PHYTEC Messtechnik GmbH

Application Note

How to use phyCAM camera modules with phyCORE-i.MX6 SBC

Revision History

Version	Changes	Author	Date
A0	Initial Release	H. Fendrich	08.08.2018
A1	added VM-016 camera modul	H. Fendrich	09.07.2019
A2	added phyCAM-M cable note	H. Fendrich	17.06.2020
A3	PD21.1.0 changes added	H. Fendrich	27.05.2021

Content

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

1 Overview

The i.MX6 Microcontroller supported more then 1 camera interface (see figure 1).

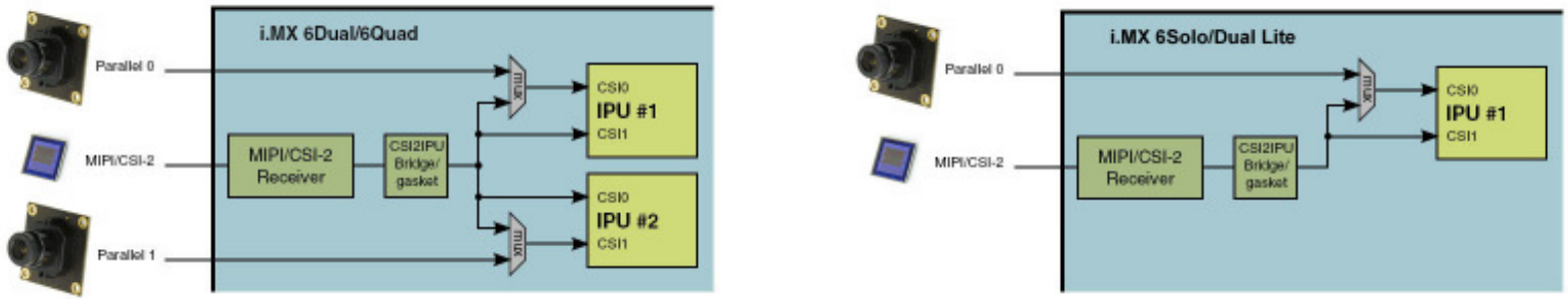


figure 1: Block Diagram Camera Interfaces of i.MX6 Controller (Quad / Dual) and i.MX6 Controller (Solo / Dual Lite)

On the phyCORE-i.MX6 the CSI0/IPU#1 camera path go out as parallel signal. (see figure 2)

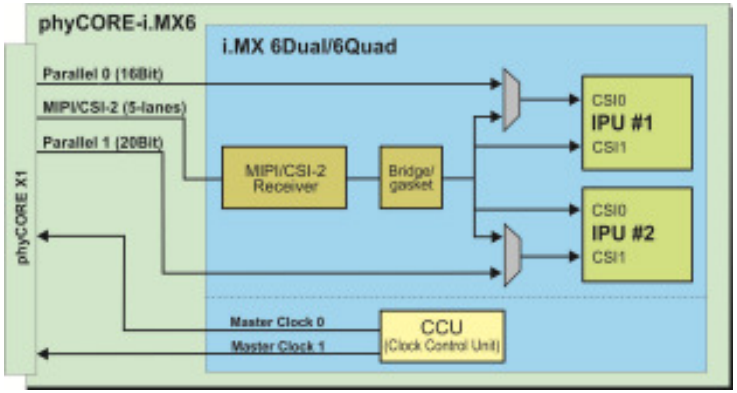


figure 2: Use of Camera_0 (CSI0 of IPU#1) and Camera_1 (CSI1 of IPU#2)

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

On the PHYTEC or customer carrier boards can the interfaces are led out as phyCAM-P (or raw-parallel) see figure 3 and/or phyCAM-S+ see figure 3 or MIPI/CSI 2 interface. For more information to phyCAM-P/S+ see manual L-748.

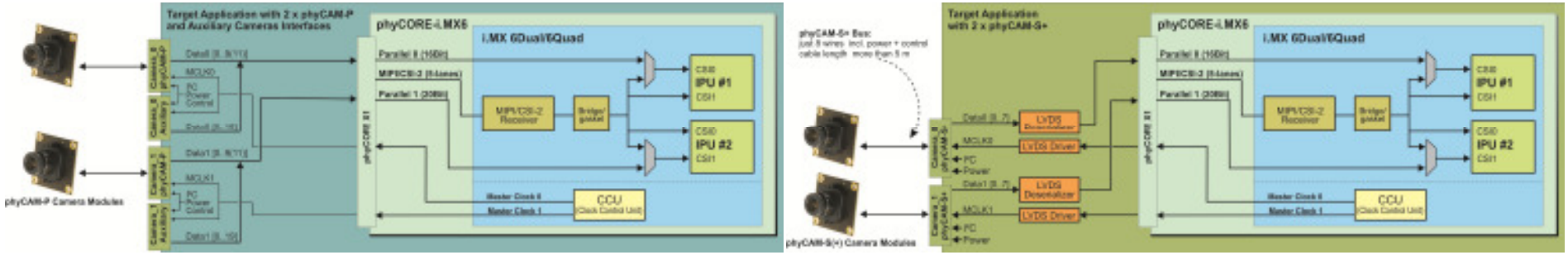


figure 3: Block Diagram of an phyCAM-P solution and an phyCAM-S (+) solution with phyCORE-i.MX6 (Quad / Dual) SBC

On the phyBOARD-Nunki baseboard are all camera interfaces led out as as phyCAM-P (or raw-parallel) as phyCAM-S+ and as MIPI/CSI-2 interfaces-I (see figure 4). Here you can connect different phyCAM-P or/and phyCAM-S(+) or phyCAM-M camera modules. See the phyCAM-P/-S manual L-748 for more information.

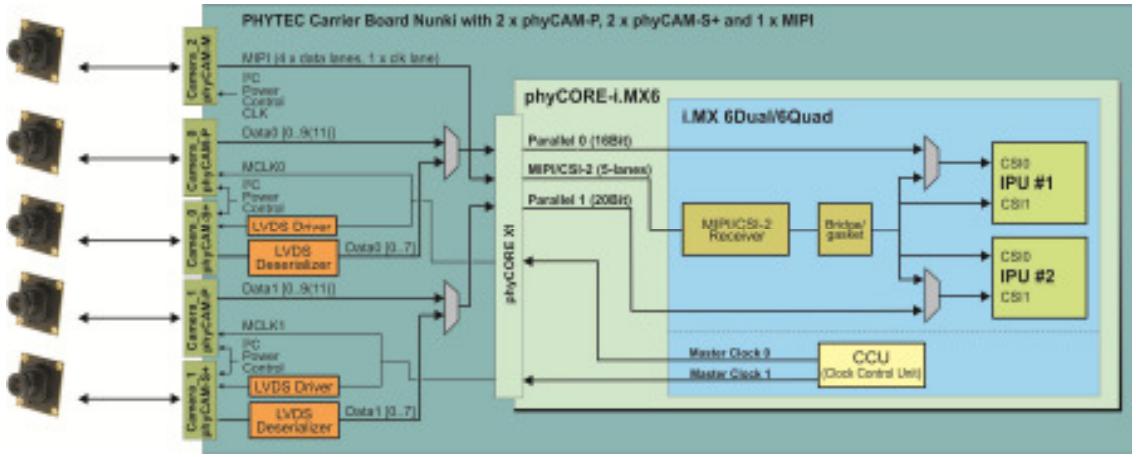


figure 4: Block Diagram of phyCAM Camera Interfaces of phyCORE-i.MX6 (Quad / Dual) and the go out on the phyBOARD-Nunki-i.MX6 - SBC

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

The BSP shipped with the Kit includes already the software drivers for the supported phyCAM 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 modul, please use pin header the phyFLEX-i.MX6 Modul.

Note:

The phyBOARD-Nunki features the CSI interface for phyCAM-P and phyCAM-S(+) camera modules and the MIPI Interface for phyCAM-M camera modules. Please find more information about the camera support in the path: …\Documentation\...

Note that the parallel and serial input of each channel is fed into the same input of the i.MX-6.
Thus, for each input either the parallel or the serial interface or the MIPI interface can be used, but not all at the same time.
The selection between phyCAM-P, phyCAM-S(+) and phyCAM-M is done by a software parameter in the config file (see below).
If a channel is configured as parallel input, this also automatically disables the LVDS deserializer on the Nunki board for this channel.

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

2 Camera Connectors on the NUNKI - Carrier Boards

The development kits for the phyBOARD-NUNKI-i.MX6 contain:
- one carrier board (NUNKI)
- one phyCORE-i.MX6 modul SOM

The carrier board connects to the phyCORE-connector, with the phyCORE-i.MX6 modul.

On the base board NUNKI (PBA-C-11-001) we convert the parallel Camera_0 and Camera_1 interface in:
- phyCAM-S+ camera interface standard (support on connector X1/X2)
- phyCAM-P camera interface standard (support on connector X6/X7)

Or led out as phyCAM-M interface standard (support on connector X32) **Note: As of PCB Version PL1489.3, the phyCAM-MI interface has been changed and is located on the back side of connector X35.**

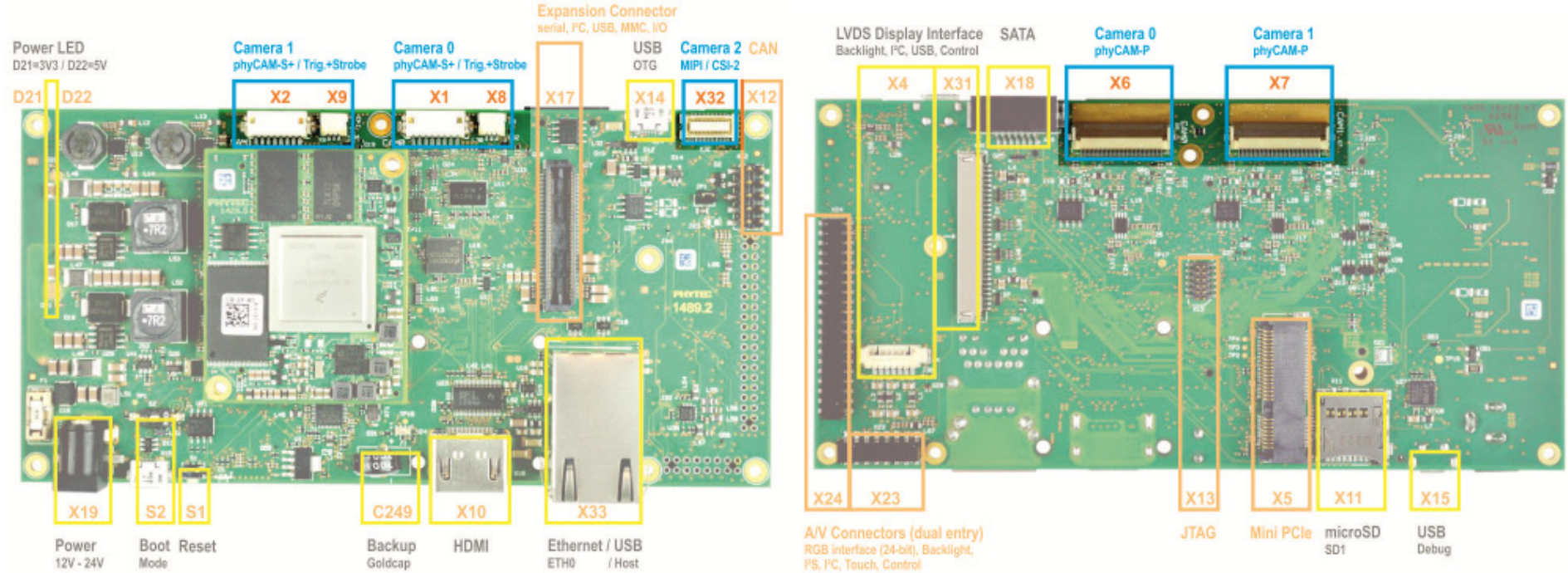


figure 5: Camera Interfaces on phyBOARD-NUNKI up PCB Version PL1489.2 for the phyCORE-i.MX6 SOM

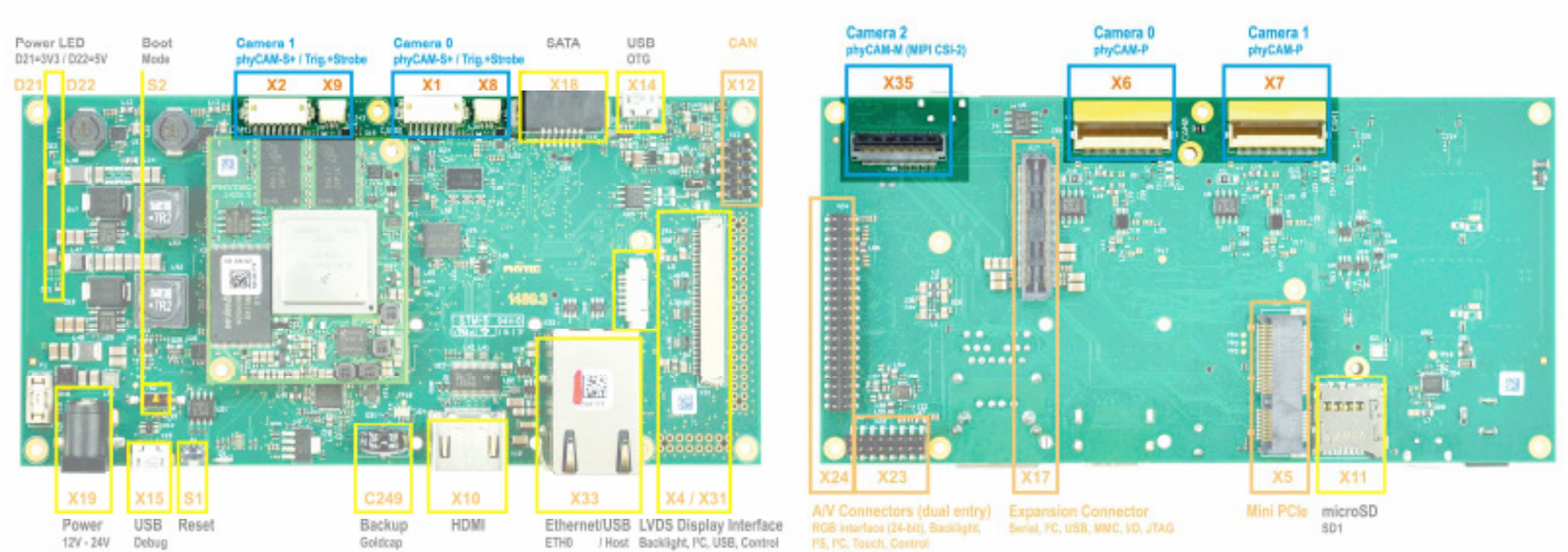


figure 6: Camera Interfaces on phyBOARD-NUNKI from PCB Version PL1489.3 for the phyCORE-i.MX6 SOM

Note: If you use the phyCAM-M interface (X35), use an FFC cable that is especially suitable for FH41 connectors (e.g. Phytec WF271). Standard FFC cables can cause a short circuit.

Connecting the phyCAM-P and phyCAM-M Camera:

- Open the lock of the FFC connector on the camera by lifting the lock upwards.
- Plug the FFC cable into the FFC socket with the contact surfaces facing downwards until you feel the stop. The reinforcement of the FFC cable (usually highlighted in color) points to the bracket of the socket.
- Lock the FFC socket by carefully pressing down on the bracket.

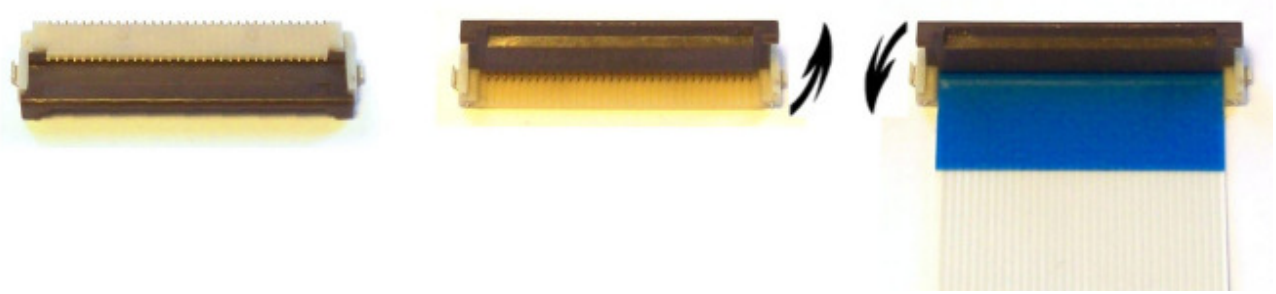


figure 7: phyCAM-P and phyCAM-M Flip Lock – FFC Socket Camera Connection

Connecting the phyCAM-S Camera:

- Push the phyCAM-S cable into the connector as far as it will go.
- The connector locks are located on the top of the connector.

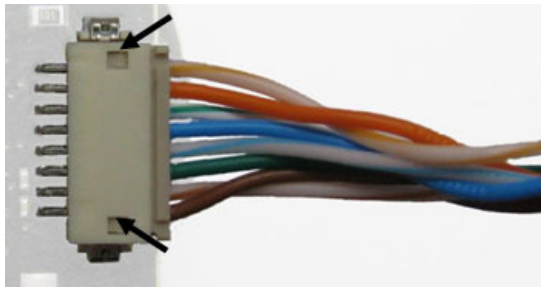


figure 8: phyCAM-S/S+ Hirose Camera Connection

3 Change the cameras or camera parameters

3.1 Change the cameras or camera parameters (YOCTO Linux PD18.1.0 and higher)

The cameras and the camera interface supported as a sub-devices. To configure is used the multimedia framework "media control".
The conversion of the formerly from PHYTEC supported camera driver, camera interface and the IPU and VPU support is not yet complete.

For more information see on FTP:
ftp://ftp.phytec.de/pub/ImageProcessing/phyBOARD-Nunki-i.MX6_linux_PD18.1.x/

The configuration in the config-expansions file must match with the camera models that are connected to the camera interfaces.
Note that model, interface type and I²C-addresses must be set correctly. Otherwise the camera(s) are not working.
Please check, if the correct camera model is set in the config file. If not, please follow the steps below to set the appropriate configuration.

Changing the setting of the config-expansions parameters

To change the config-file (config-expansions), use the Barebox environment: Use a terminal program e.g. "Putty".

- 'cd env' <enter>
- 'edit config-expansions' <enter>

The following settings in config-expansions - file are necessary:
of_camera_selection -p <csi_port> -b <csiX_cam_bus_type> -a <csiX_cam_i2c_address> <cam_type>

- csi_port = [0,1]
- csix_cam_bus_type = [phyCAM-P, phyCAM-S+] (Note: phyCAM-M not yet supported)
- csix_cam_i2c_address = [0x41-0x5D] depend of camera type/settings
- csix_cam_type = [VM-006, VM-008, VM-009, VM-010-BW, VM-010-COL, VM-011-BW, VM-011-COL]

Example: VM-010-BW-LVDS with i2C-address 0x48 on CSI0 (Camera_0) port
[of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-010-BW]

The parameters for the phyCAM – modules are shown in the table below (see next page).

After changing the settings with the editor:

- close the editor (CTRL D)
- type 'saveenv' <enter> to save
- restart PHYTEC module

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

Now you can start working with the GStreamer demo-scripts. Detailed information about the GStreamer examples can be found in the phyCAM-Manual L-748.

Hardware Configuration		Bootarg Parameters				Bootarg settings for default I²C-address (default jumper on camera and boards)
phyCAM camera model (part number)	**connected to**	**csi_port**	**csi[X]_cam_bus_type**	**csi[X]_cam_type**	**csi[X]_cam_i2c_address**	
VM-006-BW (-M12 / -H) (not yet supported)	X6 on Nunki board	0	phyCAM-P	VM-006	0x5D, 0x41	of_camera_selection -p 0 -b phyCAM-P -a 0x5D VM-006
	X7 on Nunki board	1	phyCAM-P	VM-006	0x5D, 0x41	of_camera_selection -p 1 -b phyCAM-P -a 0x5D VM-006
VM-006-BW-LVDS (-M12 / -H) (not yet supported)	X1 on Nunki board	0	phyCAM-S+	VM-006	0x5D, 0x41	of_camera_selection -p 0 -b phyCAM-S+ -a 0x5D VM-006
	X2 on Nunki board	1	phyCAM-S+	VM-006	0x5D, 0x41	of_camera_selection -p 1 -b phyCAM-S+ -a 0x5D VM-006
VM-008 (not yet supported)	X6 on Nunki board	0	phyCAM-P	VM-008	0x44, 0x45	of_camera_selection -p 0 -b phyCAM-P -a 0x45 VM-008
	X7 on Nunki board	1	phyCAM-P	VM-008	0x44, 0x45	of_camera_selection -p 1 -b phyCAM-P -a 0x45 VM-008
	X1 on Nunki board	0	phyCAM-S+	VM-008	0x44, 0x45	of_camera_selection -p 0 -b phyCAM-S+ -a 0x45 VM-008
	X2 on Nunki board	1	phyCAM-S+	VM-008	0x44, 0x45	of_camera_selection -p 1 -b phyCAM-S+ -a 0x45 VM-008
VM-009 (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-009	0x48, 0x5D	of_camera_selection -p 0 -b phyCAM-P -a 0x48 VM-009
	X7 on Nunki board	1	phyCAM-P	VM-009	0x48, 0x5D	of_camera_selection -p 1 -b phyCAM-P -a 0x5D VM-009
VM-009-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-009	0x48, 0x5D	of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-009
	X2 on Nunki board	1	phyCAM-S+	VM-009	0x48, 0x5D	of_camera_selection -p 1 -b phyCAM-S+ -a 0x48 VM-009
VM-010-BW (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-P -a 0x48 VM-010-BW
	X7 on Nunki board	1	phyCAM-P	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-P -a 0x4C VM-010-BW
VM-010-BW-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-010-BW
	X2 on Nunki board	1	phyCAM-S+	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-S+ -a 0x48 VM-010-BW
VM-010-COL (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-P -a 0x48 VM-010-COL
	X7 on Nunki board	1	phyCAM-P	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-P -a 0x4C VM-010-COL
VM-010-COL-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-010-COL
	X2 on Nunki board	1	phyCAM-S+	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-S+ -a 0x48 VM-010-COL
VM-011-BW (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-011-BW	0x48,0x5D	of_camera_selection -p 0 -b phyCAM-P -a 0x48 VM-011-BW

	X7 on Nunki board	1	phyCAM-P	VM-011-BW	0x48,0x5D	of_camera_selection -p 1 -b phyCAM-P -a 0x5D VM-011-BW
VM-011-BW-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-011-BW	0x48,0x5D	of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-011-BW
	X2 on Nunki board	1	phyCAM-S+	VM-011-BW	0x48,0x5D	of_camera_selection -p 1 -b phyCAM-S+ -a 0x48 VM-011-BW
VM-011-COL (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-011-COL	0x48,0x5D	of_camera_selection -p 0 -b phyCAM-P -a 0x48 VM-011-COL
	X7 on Nunki board	1	phyCAM-P	VM-011-COL	0x48,0x5D	of_camera_selection -p 1 -b phyCAM-P -a 0x5D VM-011-COL
VM-011-COL-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-011-COL	0x48,0x5D	of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-011-COL
	X2 on Nunki board	1	phyCAM-S+	VM-011-COL	0x48,0x5D	of_camera_selection -p 1 -b phyCAM-S+ -a 0x48 VM-011-COL
VM-012-BW (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-P -a 0x48 VM-012-BW
	X7 on Nunki board	1	phyCAM-P	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-P -a 0x4C VM-012-BW
VM-012-BW-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-012-BW
	X2 on Nunki board	1	phyCAM-S+	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-S+ -a 0x48 VM-012-BW
VM-012-COL (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-P -a 0x48 VM-012-COL
	X7 on Nunki board	1	phyCAM-P	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-P -a 0x4C VM-012-COL
VM-012-COL-LVDS (-M12/-H)	X1 on Nunki board	0	phyCAM-S+	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 0 -b phyCAM-S+ -a 0x48 VM-012-COL
	X2 on Nunki board	1	phyCAM-S+	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	of_camera_selection -p 1 -b phyCAM-S+ -a 0x48 VM-012-COL
VM-050 (-021-0 / -050-0)	X6 on Nunki board	0	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	of_camera_selection -p 0 -b phyCAM-P -a 0x4A VM-050
	X7 on Nunki board	1	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	of_camera_selection -p 1 -b phyCAM-P -a 0x4E VM-050
VM-051 (-050-0 / -105-0)	X6 on Nunki board	0	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	of_camera_selection -p 0 -b phyCAM-P -a 0x4A VM-050
	X7 on Nunki board	1	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	of_camera_selection -p 1 -b phyCAM-P -a 0x4E VM-050

Notes:

- csi[X]_cam address must be identical to the I²C-address of the corresponding camera. I²C addresses of the camera are set by hardware configuration (jumper setting on the camera and / or on the baseboard / mapper board. 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 CAM0 and CAM1.

3.2 Change the cameras or camera parameters for VM-016 camera series (YOCTO Linux PD18.1.1 and higher)

Before you can use the PD18.1.1 with a VM-016 camera , you have to be generate an new image or use the ready SD-Card image for the Nunki board. See more on ftp: ftp://ftp.phytec.de/pub/ImageProcessing/phyBOARD-Nunki-i.MX6_linux_PD18.1.x/Software/

For the VM-016 cameras with phyCAM-P and phyCAM-S+ interface are the syntax equal as in chapter 3.
For the VM-016 cameras with phyCAM-M interface is a new parameter (-A) necessary and the parameter (-p) is no longer necessary.

- dt_node = [0x10, 0x18] devicetree node, Activated the mipi-blocks to configure with media-ctl.

Hardware Configuration		Bootarg Parameters				Bootarg settings for default I²C-address (default jumper on camera and boards)
phyCAM camera model (part number)	**connected to**	**csi_port**	**csi[X]_cam_bus_type**	**csi[X]_cam_type**	**csi[X]_cam_i2c_address**	
VM-016-BW-P (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-016	0x10, 0x18	of_camera_selection -p 0 -b phyCAM-P -a 0x10 VM-016
	X7 on Nunki board	1	phyCAM-P	VM-016	0x10, 0x18	of_camera_selection -p 1 -b phyCAM-P -a 0x18 VM-016
VM-016-BW-S (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-016	0x10, 0x18	of_camera_selection -p 0 -b phyCAM-S+ -a 0x10 VM-016
	X2 on Nunki board	1	phyCAM-S+	VM-016	0x10, 0x18	of_camera_selection -p 1 -b phyCAM-S+ -a 0x10 VM-016
VM-016-COL-P (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-016	0x10, 0x18	of_camera_selection -p 0 -b phyCAM-P -a 0x10 VM-016
	X7 on Nunki board	1	phyCAM-P	VM-016	0x10, 0x18	of_camera_selection -p 1 -b phyCAM-P -a 0x18 VM-016
VM-016-COL-S (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-016	0x10, 0x18	of_camera_selection -p 0 -b phyCAM-S+ -a 0x10 VM-016
	X2 on Nunki board	1	phyCAM-S+	VM-016	0x10, 0x18	of_camera_selection -p 1 -b phyCAM-S+ -a 0x10 VM-016
phyCAM camera model (part number)	**connected to**	**dt_node**	**csi[X]_cam_bus_type**	**csi[X]_cam_type**	**csi[X]_cam_i2c_address**	**Bootarg settings for default I²C-address (default jumper on camera and boards)**
VM-016-BW-M (-M12 / -H)	X35 on Nunki board	0x10, 0x18	phyCAM-M	VM-016	0x10, 0x18	of_camera_selection -b phyCAM-M -a 0x10 VM-016 –A 0x10
VM-016-COL-M (-M12 / -H)	X35 on Nunki board	0x10, 0x18	phyCAM-M	VM-016	0x10, 0x18	of_camera_selection -b phyCAM-M -a 0x10 VM-016 –A 0x10

3.3 Change the cameras or camera parameters (YOCTO Linux PD21.1.0 and higher)

The cameras and the camera interface supported as a sub-devices. To configure is used the multimedia framework "media control".

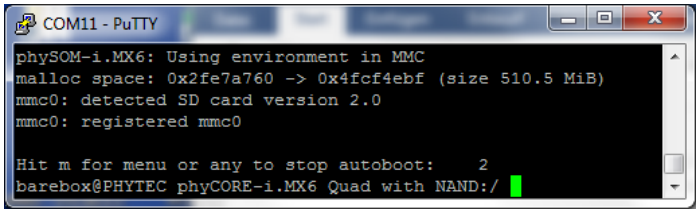
For more information see on FTP:
https://download.phytec.de/ImageProcessing/phyBOARD-Nunki-i.MX6_linux_PD21.1.x/

The old camera configuration via "config-expansions" file has been changed. The configuration is now in the barebox.
The configuration in the barebox must match with the camera models that are connected to the camera interfaces.
Note that model, interface type and I²C-addresses must be set correctly. Otherwise the camera(s) are not working.
Please check, if the correct camera model is set in the barebox. If not, please follow the steps below to set the appropriate configuration.

Changing the setting of the camera parameters

To change the camera, use the Barebox environment: Use a terminal program e.g. "Putty".

- Press space button during the barebox count. "Hit m for menu or any to stop autoboot: x"



Now you have to configure the right devicetree for your camera and camera interface.

Example: VM-010-BW-LVDS with i2C-address 0x48 on CSI0 (Camera_0) port

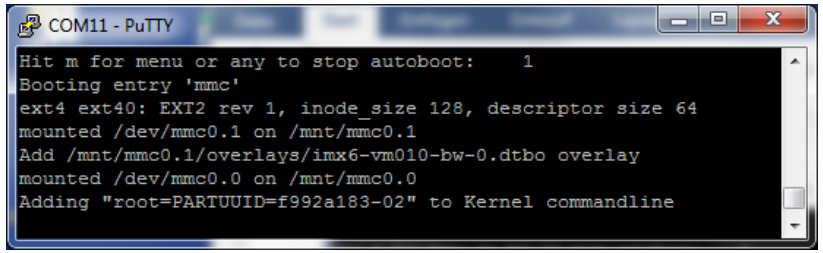
- type ‘nv overlays.select="imx6-vm010-bw-0.dtbo"’ <enter>

The parameters for the phyCAM – modules are shown in the table below.

After changing the settings with the editor:

- type 'saveenv' <enter> to save
- type 'reset' <enter> to restart

Check the set camera in the first rows of boot log:
e.g.: “Add /mnt/mmc0.1/overlays/imx6-vm010-bw-0.dtbo overlay”



Hardware Configuration		Bootarg Parameters				Device Tree settings for default I²C-address (default jumper on camera and boards)	
phyCAM camera model (part number)	connected to	CSI Port	phyCAM Type	Cam Type	CSI Port i2c Address		
VM-009 (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-009	0x48, 0x5D	nv overlays.select="imx6-vm009-0.dtbo"	0x48
	X7 on Nunki board	1	phyCAM-P	VM-009	0x48, 0x5D	nv overlays.select="imx6-vm009-1.dtbo"	0x5D
VM-009-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-009	0x48, 0x5D	nv overlays.select="imx6-vm009-0.dtbo imx6-cam-0-lvds.dtbo"	0x48
	X2 on Nunki board	1	phyCAM-S+	VM-009	0x48, 0x5D	nv overlays.select="imx6-vm009-1.dtbo imx6-cam-1-lvds.dtbo"	0x5D
VM-010-BW (-M12 / -H) VM-010-BW-LVDS (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-bw-0.dtbo"	0x48
	X7 on Nunki board	1	phyCAM-P	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-bw-1.dtbo"	0x4C
	X1 on Nunki board	0	phyCAM-S+	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-bw-0.dtbo imx6-cam-0-lvds.dtbo"	0x48
	X2 on Nunki board	1	phyCAM-S+	VM-010-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-bw-1.dtbo imx6-cam-1-lvds.dtbo"	0x4C
VM-010-COL (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-col-0.dtbo"	0x48
	X7 on Nunki board	1	phyCAM-P	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-col-1.dtbo"	0x4C
VM-010-COL-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-col-0.dtbo imx6-cam-0-lvds.dtbo"	0x48
	X2 on Nunki board	1	phyCAM-S+	VM-010-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm010-col-1.dtbo imx6-cam-1-lvds.dtbo"	0x4C
VM-011-BW (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-011-BW	0x48,0x5D	nv overlays.select="imx6-vm011-bw-0.dtbo"	0x48
	X7 on Nunki board	1	phyCAM-P	VM-011-BW	0x48,0x5D	nv overlays.select="imx6-vm011-bw-1.dtbo"	0x5D
VM-011-BW-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-011-BW	0x48,0x5D	nv overlays.select="imx6-vm011-bw-0.dtbo imx6-cam-0-lvds.dtbo"	0x48
	X2 on Nunki board	1	phyCAM-S+	VM-011-BW	0x48,0x5D	nv overlays.select="imx6-vm011-bw-1.dtbo imx6-cam-1-lvds.dtbo"	0x5D
VM-011-COL (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-011-COL	0x48,0x5D	nv overlays.select="imx6-vm011-col-0.dtbo"	0x48
	X7 on Nunki board	1	phyCAM-P	VM-011-COL	0x48,0x5D	nv overlays.select="imx6-vm011-col-1.dtbo"	0x5D

VM-011-COL-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-011-COL	0x48,0x5D	nv overlays.select="imx6-vm011-col-0.dtbo imx6-cam-0-lvds.dtbo"	0x48
	X2 on Nunki board	1	phyCAM-S+	VM-011-COL	0x48,0x5D	nv overlays.select="imx6-vm011-col-1.dtbo imx6-cam-1-lvds.dtbo"	0x5D
VM-012-BW (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-bw-0.dtbo"	0x48
	X7 on Nunki board	1	phyCAM-P	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-bw-1.dtbo"	0x4C
VM-012-BW-LVDS (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-bw-0.dtbo imx6-cam-0-lvds.dtbo"	0x48
	X2 on Nunki board	1	phyCAM-S+	VM-012-BW	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-bw-1.dtbo imx6-cam-1-lvds.dtbo"	0x4C
VM-012-COL (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-col-0.dtbo"	0x48
	X7 on Nunki board	1	phyCAM-P	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-col-1.dtbo"	0x4C
VM-012-COL-LVDS (-M12/-H)	X1 on Nunki board	0	phyCAM-S+	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-col-0.dtbo imx6-cam-0-lvds.dtbo"	0x48
	X2 on Nunki board	1	phyCAM-S+	VM-012-COL	0x48, 0x4C, 0x58, 0x5C	nv overlays.select="imx6-vm012-col-1.dtbo imx6-cam-1-lvds.dtbo"	0x4C
VM-050 (-021-0 / -050-0)	X6 on Nunki board	0	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	nv overlays.select="imx6-vm050-0.dtbo"	0x4A
	X7 on Nunki board	1	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	nv overlays.select="imx6-vm050-1.dtbo"	0x4E
VM-051 (-050-0 / -105-0)	X6 on Nunki board	0	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	nv overlays.select="imx6-vm050-0.dtbo"	0x4A
	X7 on Nunki board	1	phyCAM-P	VM-050	0x4A, 0x4E, 0x5A, 0x5E	nv overlays.select="imx6-vm050-1.dtbo"	0x4E
VM-016-BW-P (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-0.dtbo"	0x10
	X7 on Nunki board	1	phyCAM-P	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-1.dtbo"	0x18
VM-016-BW-S (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-0.dtbo imx6-cam-0-lvds.dtbo"	0x10
	X2 on Nunki board	1	phyCAM-S+	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-1.dtbo imx6-cam-1-lvds.dtbo"	0x18
VM-016-COL-P (-M12 / -H)	X6 on Nunki board	0	phyCAM-P	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-0.dtbo"	0x10
	X7 on Nunki board	1	phyCAM-P	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-1.dtbo"	0x18
VM-016-COL-S (-M12 / -H)	X1 on Nunki board	0	phyCAM-S+	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-0.dtbo imx6-cam-0-lvds.dtbo"	0x10
	X2 on Nunki board	1	phyCAM-S+	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-1.dtbo imx6-cam-1-lvds.dtbo"	0x18
VM-016-BW-M (-M12 / -H)	X35 on Nunki board	MIPI	phyCAM-M	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-mipi-0.dtbo"	0x10
VM-016-COL-M (-M12 / -H)	X35 on Nunki board	MIPI	phyCAM-M	VM-016	0x10, 0x18	nv overlays.select="imx6-vm016-mipi-0.dtbo"	0x10

Notes:

- "CSI Port i2c Address" must be identical to the I²C-address of the corresponding camera. I²C addresses of the camera are set by hardware configuration (jumper setting on the camera and / or on the baseboard / mapper board. Please refer to the phyCAM-manual and the hardware manual of the kit.
- If two cameras are used at the same time, csi[X]_cam address must be different for CAM0 and CAM1.

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

4 Demo scripts

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

There are 3 sub directories with demo scripts for the cameras:

- gstreamer-examples
- v4l2_c-examples
- opencv-examples (since PD19)

Notes: Remove the qt-demo at the first start. With the \gstreamer-examples\remove_qt_demo.sh script, else the qt-demo is alway present on the display.

4.1 GStreamer scripte

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

At the first start disable the QT-Demo. Start the script “remove_qt_demo.sh”.

Now you can start working with the GStreamer demo-scripts.
Start the scripts with den word phrase "col" or "bw" depending on the connected camera color type.

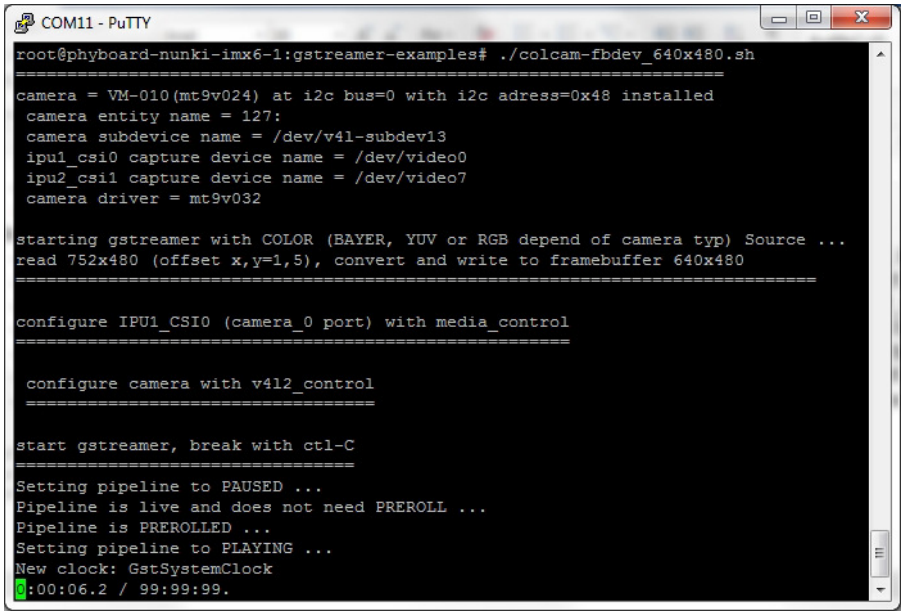
- _cam-fbdev_640x480.sh – scripts, show a livestream on display
- _cam-save_jpg_full_res.sh – scripts, save a JPG File in this directory
- _cam-save_raw_full_res.sh – scripts, save a RAW File in this directory
- func.sh – script, detect the camera typ and define the parameter for the scripts
- remove_qt_demo.sh – script, remove the qt-demo from autostart

Subdirectories:

- more_ar0144_scripts: contain more scripts for the VM-016 camera series (v4l-ctrl_ar0144.txt, list the v4l2-controls for this camera)
- more_mt9m131_scripts: contain more scripts for the VM-009 camera series (v4l-ctrl_mt9m131.txt, list the v4l2-controls for this camera)
- more_mt9p031_scripts: contain more scripts for the VM-011 camera series (v4l-ctrl_mt9p031_mt9p006.txt, list the v4l2-controls for this camera)
- more_mt9v024_scripts: contain more scripts for the VM-010 camera series (v4l-ctrl_mt4v024.txt, list the v4l2-controls for this camera)
- more_vita1300_scripts: contain more scripts for the VM-012 camera series (v4l-ctrl_vita1300.txt, list the v4l2-controls for this camera)
- phycam-m_mipi_csi-2: contain more scripts for the VM-016 phyCAM-M camera series (v4l-ctrl are listed in more_ar0144_scripts path)
- phycam_thermal_cam: contain more scripts for the VM-050/51 camera series (v4l-ctrl_vm-05x.txt, list the v4l2-controls for this camera)

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

- phytec_usb_cam: scripts for use the Phytec USB-cameras
- port_cam-1_via_ipu2-csi1: contain scripts to grab via CAM-1 Port
- tools: contain scripts to get and set the camera register direct via i2c access
- vpu_enc_dec_scripts: contain scripts how use the vpu-encoder (e.g networkstreaming or save H.264 streams)



call of “colcam-fbdev_640x480.sh” for VM-010-COL (mt9v024) series (A live image should show on the display)

All camera/video components get a separate "/dev/video[x]" or "dev/v4l-subdev[x]" device.
The v4l2 - capabilities are showed if you type: "v4l2-ctl -d [device] -L" e.g. "v4l2-ctl -d /dev/video0 -L". (see chapter 6.1)

For the first use, the camera and controller-camera interface must be configured with the tool media-ctl. See chapter 5.

4.2 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 den word phrase "col" or "bw" depending on the connected camera color type.

- ar0144_col/bw_phycam-m_full_save-raw – scripts, save a raw image from VM-016 (phyCAM-M version)in full resolution (8 and 10-Bit Formats are possible)
- ar0144_col/bw_phycam-p-s_full_save-raw – scripts, save a raw image from VM-016 (phyCAM-S+ and phyCAM-P version) in full resolution (8/10 and 12-Bit Formats are possible)
- mt9m131_col_full_save-raw – scripts, save a raw image from VM-009 in full resolution (8-Bit Formats are possible)
- mt9p006_col_full_save-raw – scripts, save a raw image from VM-011-COL in full resolution (8/10 and 12-Bit Formats are possible)
- mt9p031_bw_full_save-raw – scripts, save a raw image from VM-011-BW in full resolution (8/10 and 12-Bit Formats are possible)
- mt9v02x_col/bw_full_save-raw – scripts, save a raw image from VM-010 in full resolution (8/10 and 12-Bit Formats are possible)
- vita1300_col/bw_full_save-raw – scripts, save a raw image from VM-012 in full resolution (8/10 and 12-Bit Formats are possible)
- vm05x_full_save-raw – scripts, save a raw image from VM-050/51 in full resolution (8/10 and 12-Bit Formats are possible)

For saving 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.

A other 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*

For the first use, the camera and controller-camera interface must be configured with the tool media-ctl. See chapter 5.

4.3 OpenCV scripts

PHYTEC i.MX6 BSP PD21.1.0 includes OpenCV4.5. For use OpenCV with image output we use window manager X11. PHYTEC tested openCV with PYTHON programming language. Examples of scripts can be found in the path “opencv-examples”:

- /opencv-examples/python/python3 phycam_s_p_video_v4l2: show the live image from the camera (phyCAM-S+ and phyCAM-P version)
- /opencv-examples/python/python3 phycam_m_video_v4l2.py: show the live image from the camera (phyCAM-M version)

For the first use, the camera and controller-camera interface must be configured with the tool media-ctl. See chapter 5.
The window manager X11 must also be started. <systemctl start xserver-nodm>.

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

5 Configure camera and controller-camera-interface with “media-ctl”

Modern System-on-Chip (SoC) devices support a wide range of functionality in the way of internal hardware blocks which has resulted in a variety of ways to interconnect functions within the SoC that alter the V4L device content.

The Media Controller kernel API has been designed to expose detailed information about media devices and capture devices to userspace in a way that allows them to be interconnected in dynamic and complex ways at runtime.

Media controller devices expose entities to the media controller framework. Each entity has one or more source pads and one or more sink pads. You use the media controller kernel API (ie via *media-ctl*) to query entities, link source pads to sink pads, and set formats of pads.

We use *media-ctl* to set:

- the desired resolution and color format of the camera sensor
- the fitting resolution and color format in the CSI interface (controller-camera-interface)
- the way (and/or preprocessing) through the different hardware blocks in the controller
- the resulting camera stream is made available as “/dev/videox” device

These settings are necessary:

- once after restarting the system and using the camera for the first time
- set a new resolution, color format, or bit-depth
- set a new path through the hardware blocks

Following entities are present at the i.MX 8M Plus Image (check with “media-ctl –p”):
- entity 1: ipu1_csi0 (3 pads, 4 links)
- entity 5: ipu1_csi0 capture (1 pad, 1 link)
- entity 11: ipu1_vdic (3 pads, 3 links)
- entity 15: ipu1_ic_prp (3 pads, 5 links)
- entity 19: ipu1_ic_prpenc (2 pads, 2 links)
- entity 22: ipu1_ic_prpenc capture (1 pad, 1 link)
- entity 28: ipu1_ic_prpvf (2 pads, 2 links)

- entity 31: ipu1_ic_prpvf capture (1 pad, 1 link)
- entity 47: ipu1_csi1 (3 pads, 3 links)
- entity 51: ipu1_csi1 capture (1 pad, 1 link)
- entity 61: ipu2_csi0 (3 pads, 3 links)
- entity 65: ipu2_csi0 capture (1 pad, 1 link)
- entity 71: ipu2_vdic (3 pads, 3 links)
- entity 75: ipu2_ic_prp (3 pads, 5 links)
- entity 79: ipu2_ic_prpenc (2 pads, 2 links)
- entity 82: ipu2_ic_prpenc capture (1 pad, 1 link)
- entity 88: ipu2_ic_prpvf (2 pads, 2 links)
- entity 91: ipu2_ic_prpvf capture (1 pad, 1 link)
- entity 107: ipu2_csi1 (3 pads, 4 links)
- entity 111: ipu2_csi1 capture (1 pad, 1 link)
- entity 121: ipu1_csi0_mux (3 pads, 1 link)
- entity 127: ipu2_csi1_mux (3 pads, 1 link)

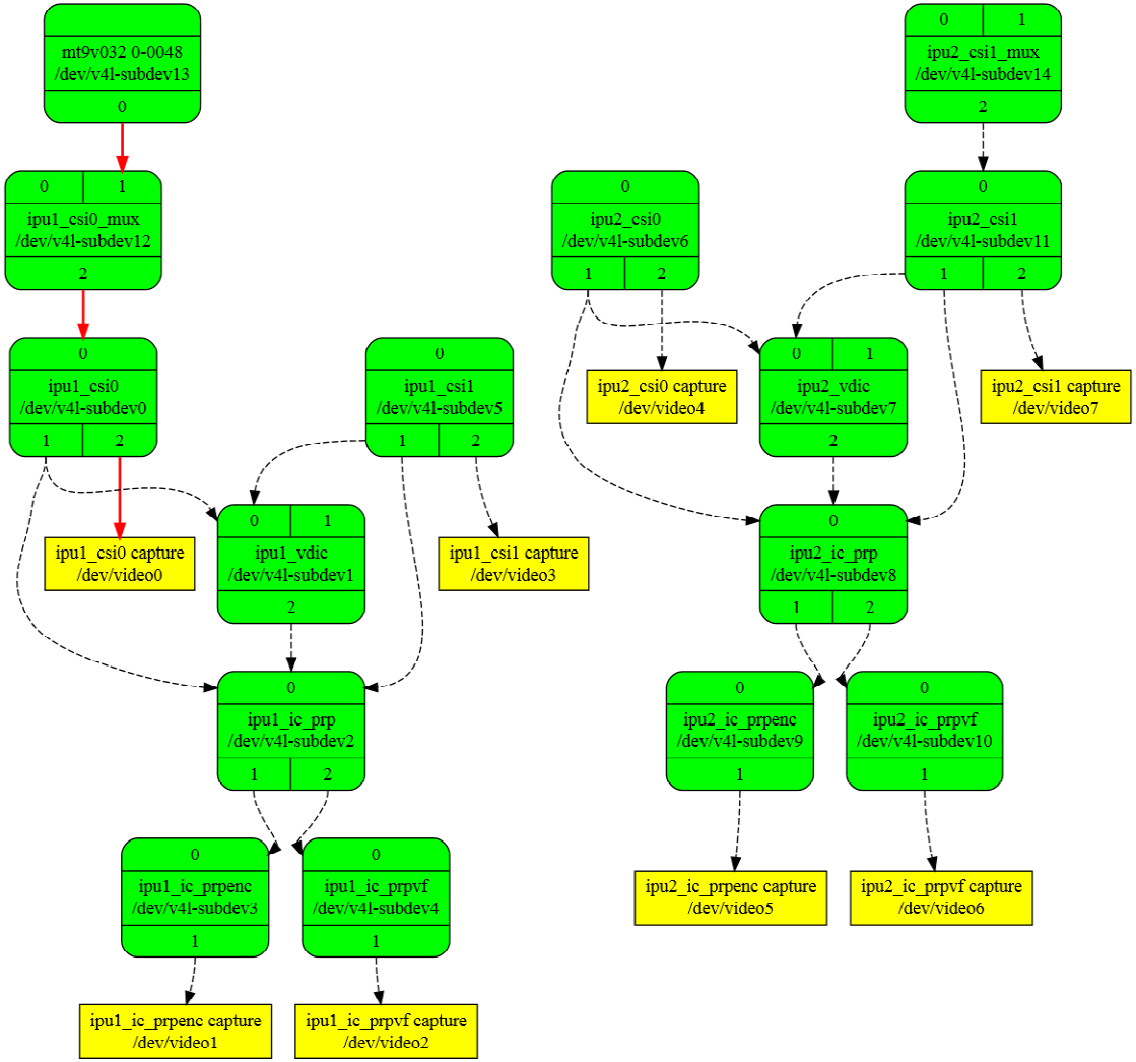
To set the *media-ctl,* we use in our demo script following points:

- *media-ctl –r:* reset all links to inactive
- *media-ctl –l:* setup the links between the hardware blocks
- *media-ctl –V:* formats the hardware blocks

For example, we will set the VM-010-COL (MT9V024) with full Resolution from IPU1_CSI0 Port to a /dev/videox device.

First we set the links:
media-ctl -r
media-ctl -l "'mt9v032 0-0048':0->'ipu1_csi0_mux':1[1]"
media-ctl -l "'ipu1_csi0_mux':2->'ipu1_csi0':0[1]"
media-ctl -l "'ipu1_csi0':2->'ipu1_csi0 capture':0[1]"
Camera -> IPU1_CSI0_mux -> IPU1-CSI0 -> IPU1-CSI0 capture (/dev/videoX)

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com



Mediagraph: VM-010 at **Camera_0** / IPU1_CSI0 Port (active path is red painted, possible paths are dashed)

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com

Then we format the hardware blocks:
media-ctl -V "'mt9v032 0-0048':0 [fmt:Y8/752x480 (1,5)/752x480]"
media-ctl -V "'ipu1_csi0_mux':2 [fmt:Y8/752x480]"
media-ctl -V "'ipu1_csi0':2 [fmt:Y8/752x480]"

Check the settings with "media-ctl –p":
Now the camera is present at "dev/video0" device. The camera is ready for access.

6 Configuration of camera features

6.1 Configuration with v4l2-ctl

To set the various camera functions (e.g. exposure, gain, ...) use please the v4l2-ctl functions. You can get an overview of the existing functions by entering the commands:

- for example: v4l2-ctl -d /dev/video0 --all (list all)
- for example: v4l2-ctl -d /dev/video0 -L (list all detail)

With this control are many features usable. For example exposure. Set one exposure value (if the automatic disable):

- v4l2-ctl -d /dev/video0 --set-ctrl=auto_exposure=1 (set AEC off)
- v4l2-ctl -d /dev/video0 --set-ctrl=exposure=480 (set the exposure time for a time, that the sensor need to generate 480 rows)

6.2 Configuration camera register direct

To set or get a camera register direct use the i2c functions in the path …/gstreamer-examples/tools/...

Note: Use this function only, if you know the register reaction. Read the register reference manual of the camera sensor manufacturer.

7 GStreamer function for improved Tearing reduction

Tearing is a visual artifact in video display where the display image shows information from two or more frames in a single screen draw.
It appears as a horizontal border or line. The image part beyond the line seems to be shifted horizontally when the image content or the camera is moved.
The tearing line(s) usually move(s) vertically across the image.
The artifact occurs when the camera framerate differs from the display frame or the camera readout cycle is not in sync with the display's refresh.
Definition and background information about the tearing effect can be found in http://en.wikipedia.org/wiki/Screen_tearing.

The Freescale i.MX-6 processor contains an anti-tearing mechanism in the IPU unit, which can reduce the tearing effect.
However, since - depending on camera settings and camera model - the frame rate might be very different from the display's refresh rate, tearing effects might still be visible even if the anti-tearing mechanism is active. For applications that are intended to display live camera images on the display, additional measures should be considered to obtain a perfect image quality. This measures can include frame rate control (trimming the camera frame rate to the display's refresh rate), multi-buffering of the camera image etc.

We recommend to activate the anti-tearing mechanism of the i.MX-6 when live camera images are shown on the display.
For evaluation purposes with the development kits, Phytec added GStreamer examples, that use a different kmssink – function, that activates the anti-tearing mechanism.
For more information to "kmssink" parameters type:
- gst-inspect-1.0 kmssink

Note: The kmssink parameter "connector=__" define the output device. If you change the output device e.g. HDMI, set the right parameter.

8 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.MX6. 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/

9 OpenCV support

PHYTEC offer an special BSP called "phytec-vision-image...". This version includes OpenCV3.3. (V4.5.2 since PD21.1.0) For use OpenCV with image output we recommend an additional windows manager. In this case, the user must add the windows manager in BSP and recompile the BSP.
PHYTEC tested openCV with pyCAM cameras and the windows manager X11. Examples of scripts can be found at the following link:
- ftp://ftp.phytec.de/pub/ImageProcessing/phyBOARD-Nunki-i.MX6_linux_PD18.1.x/Software/OpenCV/

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

Note: The openCV examples have been included in the BSP since PD19.1.0.

Support Hotline Europe:: +49 (6131) 9221-31 ♦http://www.phytec.de
Support Hotline North America: 1-800-278-9913 ♦http://www.phytec.com