

Application Note

How to use phyCAM camera modules with the phyFLEX-i.MX6 SOM

Revision History

Version	Changes	Author	Date
A0	Initial Release	H. Fendrich	11.07.2013
B0	Details added Section 4 added	M. Klahr	16.07.2013
B1	Describe the new bootarg settings for cameras (in PD13.2.0).	H. Fendrich	28.08.2013
B2	New block diagrams	H. Fendrich	21.11.2013

Content

1	Overview	2
2	Camera Connectors on the Carrier Boards	4
3	Set bootarg parameters in the <i>config</i> -file for cameras (Linux PD13.2.0 and higher)	6
4	GStreamer function for improved Tearing reduction.....	8

1 Overview

The phyFLEX-i.MX6 (Quad and Dual) comes with two independent ports for phyCAM-P or phyCAM-S+ cameras. 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.

The phyFLEX-i.MX6 - module offers several ways to connect phyCAM cameras modules. In the dual and quad-core version, the i.MX-6 controller has two independent parallel camera ports: Camera_0 to CSI0/IPU#1 and Camera_1 to CSI1/IPU#2 (see figure 1).

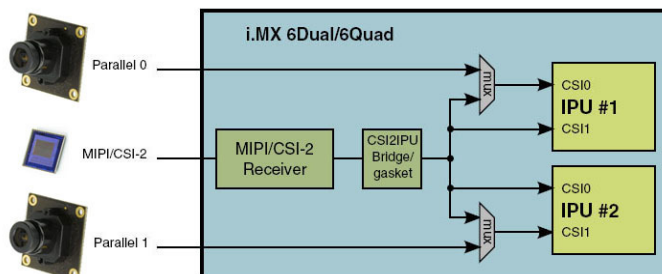


figure 1: Block Diagram Camera Interfaces of phyFLEX i.MX6 (Quad and Dual) SOM

In the solo and dual-lite - core version, the i.MX-6 controller has one independent parallel camera ports: Camera_0 to CSI0/IPU#1 (see figure 2).

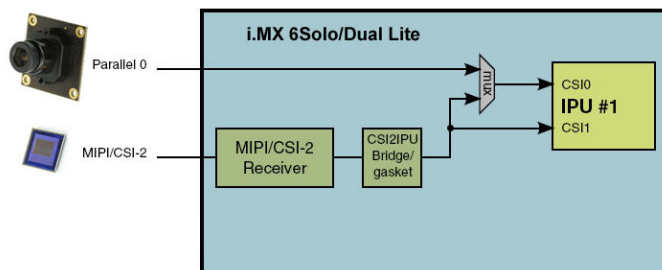


figure 2: Block Diagram Camera Interfaces of phyFLEX i.MX6 (Solo and DualLite) SOM

The two camera ports Camera_0 (also referred as CSI0) and Camera_1 (also referred as CSI1) are accessible in two ways:

- as parallel interfaces on the phyFLEX connector no. 3 (phyCAM-P or customer parallel), for connection reference see PHYTEC mapper board FLM-A-XL1

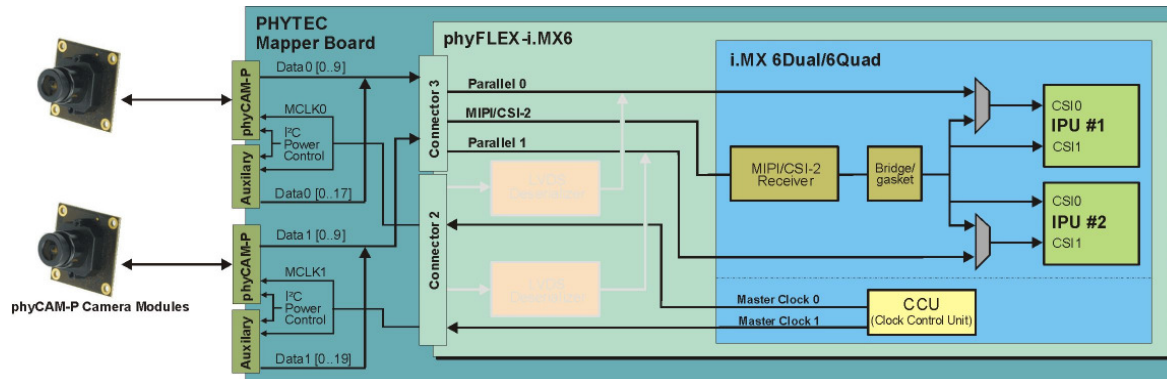


figure 3: Block Diagrams of phyCAM-P (or customer parallel) Camera Interfaces of phyFLEX i.MX6 (Quad and Dual)

- as serial LVDS interfaces on the phyFLEX connector no. 2 (phyCAM-S+), for connection reference see PHYTEC carrier board PBA-B-01

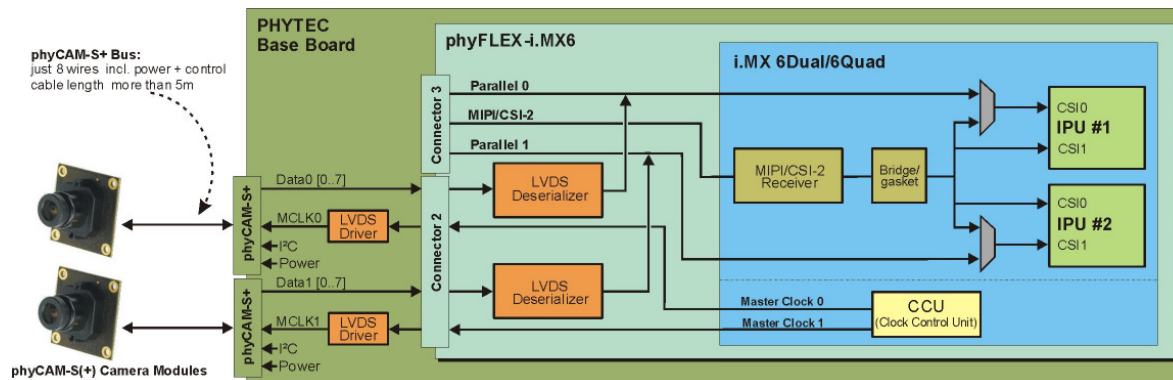


figure 4: Block Diagrams of phyCAM-S(+) Camera Interfaces of phyFLEX i.MX6 (Quad and Dual)

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 can be used, but not both at the same time.
The selection between phyCAM-P and phyCAM-S(+) 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 phyFLEX board for this channel.

The phyCAM-P or phyCAM-S(+) camera modules fit seamlessly to the phyFLEX-i.MX6. See the phyCAM-P/-S manual L-748 for more information.
Beside the phyCAM-P standard, the parallel inputs are available with 18 bits data width (CSI0) and 20 bits data width (CSI1). This can be used for customized applications that demand a high color depth.

2 Camera Connectors on the Carrier Boards

The development kits for the phyFLEX i.MX6 may contain either:

- one carrier board
- or
- one carrier board and
 - one mapper board

The carrier board connects to the connector rows 1 and 2 of the phyFLEX-i.MX6 board.
The mapper board allows in addition to this the connection of the connector row 3 of the module.

The two serial phyCAM-S+ camera inputs are available directly on the carrier board *PBA-B-01*.
See figure 5 for the location of the connectors on the carrier board.

The inputs for parallel cameras are located on the mapper board *FLM-A-XL1*, which is placed between carrier board and phyFLEX board.

If you like to connect parallel cameras and your kit does not contain a mapper board please contact Phyttec sales.
Mapper boards can be ordered separately for your development kit.

On the mapper-board *FLM-A-XL1* the two parallel interfaces are available as phyCAM-P camera-connectors (FFC connectors).

Alternatively, the two parallel ports are available as 18 bit wide (CSI0) and 20 bit wide (CSI1) interfaces on pin row connectors (2 mm pitch).

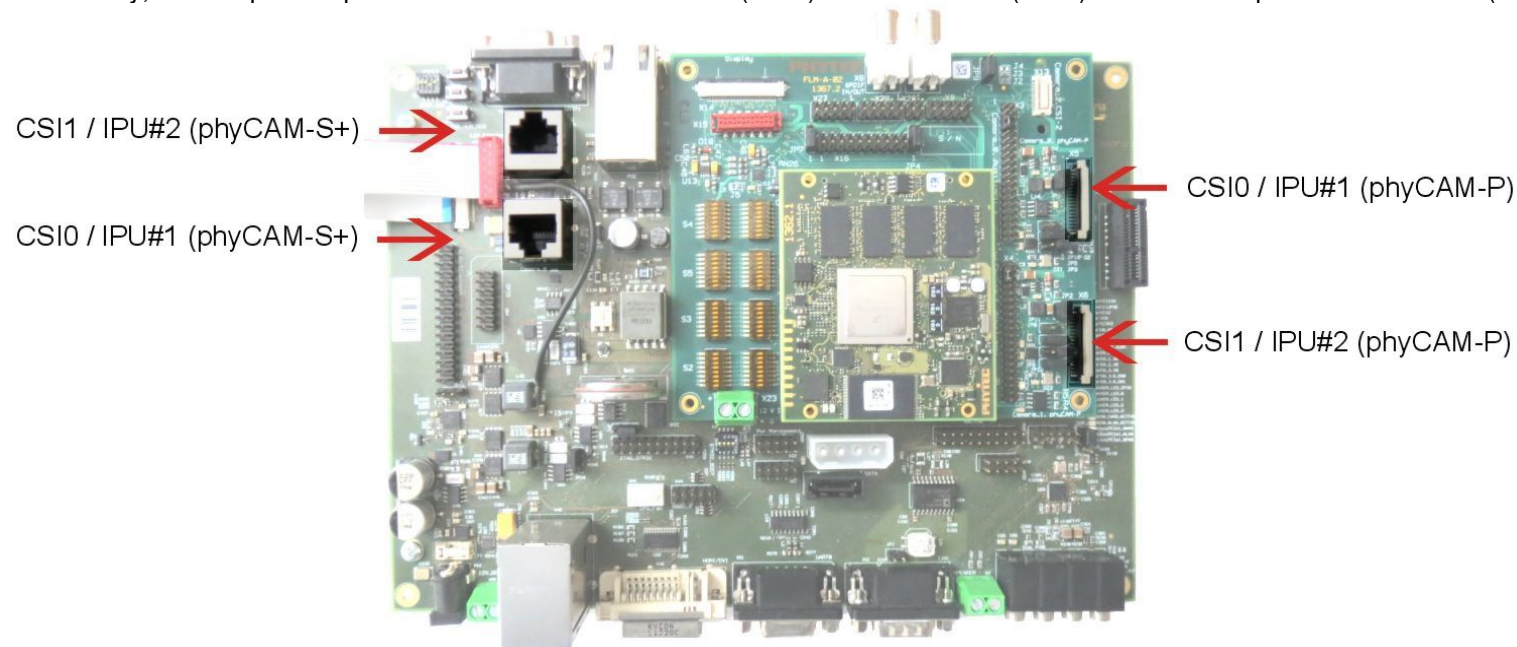


figure 5: Camera Interfaces on Carrier Board and Mapper Board for the phyFLEX i.MX6 SOM

Please note:

- For each channel CSI0/1 either the phyCAM-P or phyCAM-S+ input can be used. The selection phyCAM-P / phyCAM-S+ is made in the config file.
- Some options / configurations of the module do not allow using all input configurations for the camera interface. This especially concerns pin muxing of the i.MX6 and jumper settings.
- If more than one camera is connected: Make sure that the I²C addresses of the cameras are different. Find more about the configuration of your camera module(s) in the phyCAM - manual L-748.
- The software configuration (bootarg parameters) must match to the camera type(s) connected to the board. Read section 3 for more information about software configuration.

3 Set bootarg parameters in the *config*-file for cameras (Linux PD13.2.0 and higher)

The bootarg parameters are used to adapt the software environment to the hardware configuration and camera module(s) connected to the board.

The configuration in the config 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 bootarg parameters

To change the config-file (bootargs-cam), use the Barebox environment:

- 'cd env/bin' <enter>
- 'edit bootargs-cam' <enter>

The following settings in config-file are necessary:

- CSI0=<csi[X]_interface_type>,<csi[X]_cam_type>,<csi[X]_cam_address>
- CSI1=<csi[X]_interface_type>,<csi[X]_cam_type>,<csi[X]_cam_address>
- only for VM-010: bootargs="\$bootargs mt9v022.sensor_type=<color or mono>"
- only for VM-011: bootargs="\$bootargs mt9p031.mono=Y"

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 I ² C-address (default jumper on camera and boards)
phyCAM camera model (part number)	connected to	csi[X]_interface_type	csi[X]_cam_type	csi[X]_cam_address	
VM-006-BW (-M12 / -H)	X5 on mapper board	CSI0=phyCAM-P	VM-006	0x5D, 0x41	CSI0=phyCAM-P,VM-006,0x5D
	X6 on mapper board	CSI1=phyCAM-P	VM-006	0x5D, 0x41	CSI1=phyCAM-P,VM-006,0x5D
VM-006-BW-LVDS (-M12 / -H)	X63 on base board	CSI0=phyCAM-S+	VM-006	0x5D, 0x41	CSI0=phyCAM-S+,VM-006,0x5D
	X64 on base board	CSI1=phyCAM-S+	VM-006	0x5D, 0x41	CSI1=phyCAM-S+,VM-006,0x5D
VM-008	X5 on mapper board	CSI0=phyCAM-P	VM-008	0x44, 0x45	CSI0=phyCAM-P,VM-008,0x45
	X6 on mapper board	CSI1=phyCAM-P	VM-008	0x44, 0x45	CSI1=phyCAM-P,VM-008,0x45
	X63 on base board	CSI0=phyCAM-S+	VM-008	0x44, 0x45	CSI0=phyCAM-S+,VM-008,0x45
	X64 on base board	CSI1=phyCAM-S+	VM-008	0x44, 0x45	CSI1=phyCAM-S+,VM-008,0x45
VM-009 (-M12 / -H)	X5 on mapper board	CSI0=phyCAM-P	VM-009	0x48, 0x5D	CSI0=phyCAM-P,VM-009,0x48
	X6 on mapper board	CSI1=phyCAM-P	VM-009	0x48, 0x5D	CSI1=phyCAM-P,VM-009,0x5D
VM-009-LVDS (-M12 / -H)	X63 on base board	CSI0=phyCAM-S+	VM-009	0x48, 0x5D	CSI0=phyCAM-S+,VM-009,0x48
	X64 on base board	CSI1=phyCAM-S+	VM-009	0x48, 0x5D	CSI1=phyCAM-S+,VM-009,0x48
VM-010-BW (-M12 / -H)	X5 on mapper board	CSI0=phyCAM-P	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI0=phyCAM-P,VM-010,0x48 bootargs="\$bootargs mt9v022.sensor_type=mono"
	X6 on mapper board	CSI1=phyCAM-P	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI1=phyCAM-P,VM-010,0x4C bootargs="\$bootargs mt9v022.sensor_type=mono"
VM-010-BW-LVDS (-M12 / -H)	X63 on base board	CSI0=phyCAM-S+	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI0=phyCAM-S+,VM-010,0x48 bootargs="\$bootargs mt9v022.sensor_type=mono"
	X64 on base board	CSI1=phyCAM-S+	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI1=phyCAM-S+,VM-010,0x48 bootargs="\$bootargs mt9v022.sensor_type=mono"
VM-010-COL (-M12 / -H)	X5 on mapper board	CSI0=phyCAM-P	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI0=phyCAM-P,VM-010,0x48 bootargs="\$bootargs mt9v022.sensor_type=color"
	X6 on mapper board	CSI1=phyCAM-P	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI1=phyCAM-P,VM-010,0x4C bootargs="\$bootargs mt9v022.sensor_type= color"
VM-010-COL-LVDS (-M12 / -H)	X63 on base board	CSI0=phyCAM-S+	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI0=phyCAM-S+,VM-010,0x48 bootargs="\$bootargs mt9v022.sensor_type= color"
	X64 on base board	CSI1=phyCAM-S+	VM-010	0x48, 0x4C, 0x58, 0x5C	CSI1=phyCAM-S+,VM-010,0x48 bootargs="\$bootargs mt9v022.sensor_type= color"
VM-011-BW (-M12 / -H)	X5 on mapper board	CSI0=phyCAM-P	VM-011	0x48, 0x5D	CSI0=phyCAM-P,VM-011,0x48 bootargs="\$bootargs mt9p031.mono=Y"

	X6 on mapper board	CSI1=phyCAM-P	VM-011	0x48, 0x5D	CSI1=phyCAM-P,VM-011,0x5D bootargs="\$bootargs mt9p031.mono=Y"
VM-011-BW-LVDS (-M12 / -H)	X63 on base board	CSI0=phyCAM-S+	VM-011	0x48, 0x5D	CSI0=phyCAM-S+,VM-011,0x48 bootargs="\$bootargs mt9p031.mono=Y"
	X64 on base board	CSI1=phyCAM-S+	VM-011	0x48, 0x5D	CSI1=phyCAM-S+,VM-011,0x48 bootargs="\$bootargs mt9p031.mono=Y"
VM-011-COL (-M12 / -H)	X5 on mapper board	CSI0=phyCAM-P	VM-011	0x48, 0x5D	CSI0=phyCAM-P,VM-011,0x48 #bootargs="\$bootargs mt9p031.mono=Y"
	X6 on mapper board	CSI1=phyCAM-P	VM-011	0x48, 0x5D	CSI1=phyCAM-P,VM-011,0x5D #bootargs="\$bootargs mt9p031.mono=Y"
VM-011-COL-LVDS (-M12 / -H)	X63 on base board	CSI0=phyCAM-S+	VM-011	0x48, 0x5D	CSI0=phyCAM-S+,VM-011,0x48 #bootargs="\$bootargs mt9p031.mono=Y"
	X64 on base board	CSI1=phyCAM-S+	VM-011	0x48, 0x5D	CSI1=phyCAM-S+,VM-011,0x48 #bootargs="\$bootargs mt9p031.mono=Y"

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.
- If the CSI1 port not is used, please select "none" or the same camera as at CSI0 (with other I²C-adress).

4 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, Phytect added GStreamer examples, that use a different fbsink – function, that activates the anti-tearing mechanism.

Please use the following GStreamer-examples with Freescale “mfw_isink” for evaluation of the i.MX-6 camera:

Camera Module	Standard GStreamer Script	GStreamer Script with Freescale mfw_isink
	No tearing reduction	Framebuffer tearing reduction
VM-006	/more_mt9m001_scripts/mt9m001_bw_fbdef_640x480-sensor-roi	/more_mt9m001_scripts/mt9m001_fs_bw_fbdef_640x480
VM-010-BW	/more_mt9v024_scripts/mt9v024_bw-fbdev_752x480	/more_mt9v024_scripts/mt9v024_bw-fs_fbdev_752x480
VM-010-COL	/more_mt9v024_scripts/mt9v024_col-fbdev_752x480	/more_mt9v024_scripts/mt9v024_col-fs_fbdev_752x480
VM-011-BW	/more_mt9p031_scripts/mt9p031_bw_different_res_to_fbdev	/more_mt9p031_scripts/mt9p031_bw_fs_different_res_to_fbdev
VM-011-COL	/more_mt9p031_scripts/mt9p006_col_different_res_to_fbdev	/more_mt9p031_scripts/mt9p006_col_fs_different_res_to_fbdev

Note: After use freescale, gstreamer-sink plug-in "mfw_isink " the standard-gstreamer-sink plug-in "fbdevsink" do not working. Please restart the modul.