

Application Note No. LAN-xxx
Version: 1.0
Autor: H. Fendrich
Datum: 05.07.2012

Historie:

Version	Änderung	Datum	Autor
1.0	Erstellung des Dokuments	02.04.2012	H. Fendrich
1.1	Erweiterung mediabus Funktionalität	05.07.2012	H. Fendrich

Inhaltsverzeichnis

1	Einleitung	2
1.1	Verwendung des ISP-Verarbeitungsblock	3
1.2	OMAP3 Camera-ISP Treiber	3
1.3	Anwendung des Kamera Treiber und des ISP Treiber	4
1.3.1	Voraussetzungen	5
1.3.2	Umsetzung des gstreamer Beispiels	5
1.3.3	Aufbau des Scripts	6
1.3.4	Zusätzliche Funktionen	7

1 Einleitung

Der OMAP-35xx Controller besitzt einen Verarbeitungsblock (ISP) um die Signale eines CMOS-Kamerasensors direkt zu grabben und zu verarbeiten. Die Kameradaten können dabei parallel oder im CSI-2 (CCP2) Format vorliegen. Dieser ISP (imaging signal processor) Verarbeitungsblock ist nachfolgend dargestellt.

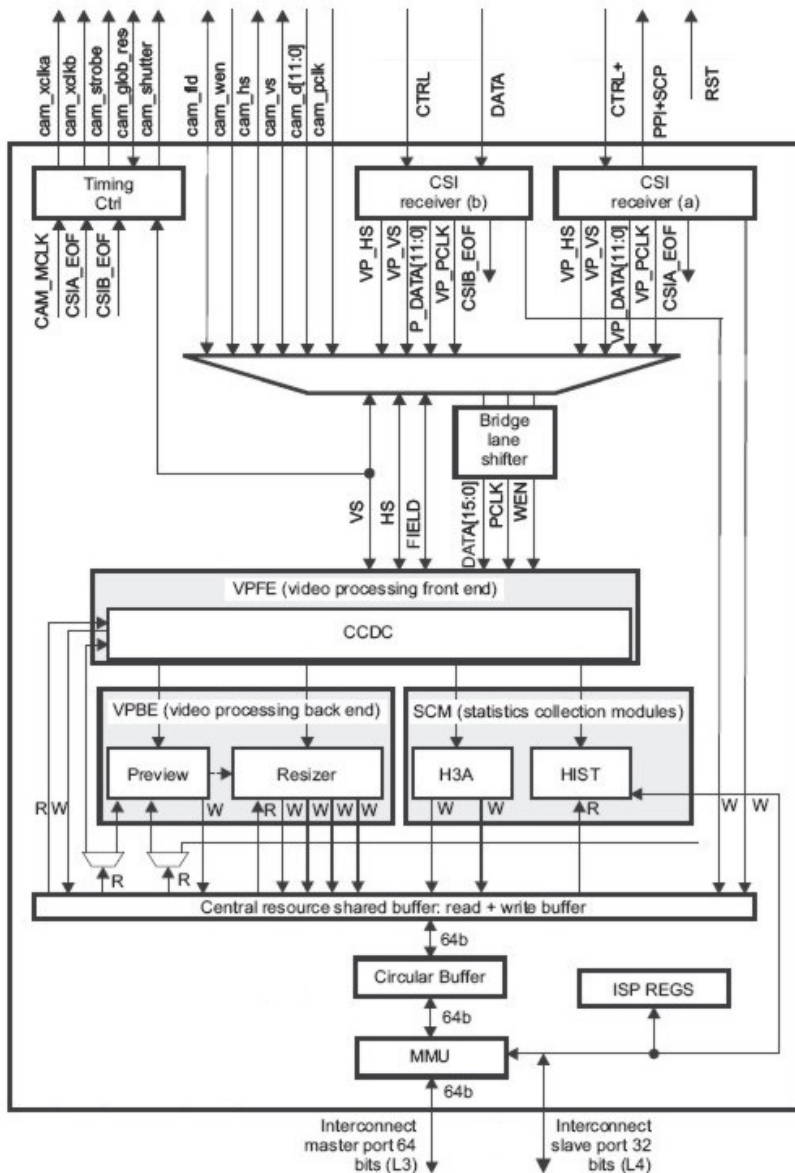


Abbildung 1 „OMAP-35xx, ISP Overview“

1.1 Verwendung des ISP-Verarbeitungsblock

Im folgenden soll die Verwendung des ISP-Verarbeitungsblock mit einer CMOS-Kamera auf Basis folgender Komponenten erläutert werden:

- PHYTEC phyCARD-L Entwicklungskit mit einem OMAP-35xx Controller
- PHYTEC Kameraboard der phyCAM-S - Serie
- PHYTEC Linux-BSP ab 3.2.xx Kernel

Die Bilddaten sollen dann über das *Video4Linux2* – Interface (V4L2) mittels des freien Multimedia *GStreamer* Frameworks abgeholt und auf dem Display dargestellt werden.

Damit die Kameras und der ISP-Block aus einer Anwendung heraus über das v4l2 Interface verwendet werden können, müssen diese in der Software abgebildet werden. Hierfür stehen in der Linux mainline ab Kernelversion 3.2.x folgende Komponenten zur Verfügung:

- Kameratreiber.** Diese müssen auf „v4l2-subdevice und mediabus“ - Funktionalität umgestellt sein. V4l2-subdevice für diese Sensoren sind schon im Mainline Kernel umgesetzt, die mediabus Anpassungen noch nicht. Hierfür müssen Ein- und Ausgänge der Hardwareblöcke als Mediabus "Pads" abstrahiert werden. Für folgende von PHYTEC verfügbaren Kameraboards sind die Kameratreibern schon teilweise bzw. vollständig umgesetzt:
 - VM-006-Serie (Aptina mt9m001)
 - VM-007-Serie (Aptina mt9v022)
 - VM-008-Serie (Techwell tw990)
 - VM-009-Serie (Aptina mt9m131)
 - VM-010-Serie (Aptina mt9v024)
 - tbd. VM-011-Serie (Aptina mt9p031)
- OMAP3 Camera-ISP Treiber.** Dies ist ein Video4Linux2 (V4L2) Treiber der ebenfalls mit v4l2-subDevice Funktionalität ausgestattet ist
- Media Controller Framework.** Dieser wird genutzt um den ISP-Treiber zu konfigurieren und die gewünschten Verknüpfungen herzustellen.
<http://lxr.free-electrons.com/source/Documentation/media-framework.txt?a=arm>

1.2 OMAP3 Camera-ISP Treiber

Detaillierte Informationen zum ISP-Treiber erhalten Sie unter:

http://www.omappedia.org/wiki/Camera-ISP_Driver

In der Treiberstruktur sind folgende Hardwareblöcke des ISP abgebildet:

- Sensor
- CSI2
- CCP2
- CCDC
- Preview
- Resizer
- SCM (H3A and Histogram)

© PHYTEC Messtechnik GmbH 2012

Europe: Support Hotline: +49 (6131) 9221-31 • <http://www.phytec.de>

North America: Support Hotline: + 1-800-278-9913 • <http://www.phytec.com>

1.3 Anwendung des Kamera Treiber und des ISP Treiber

Nachfolgen sollen Bilddaten einer VM-010-COL-LVDS (mt9v024) Kamera auf dem phyCARD-L Entwicklungskit mit Hilfe des gstreamers zur Anzeige gebracht werden.

Innerhalb des ISP sollen dabei folgende Komponenten durchlaufen werden:

- Daten der Kamera am CPI-Interface („camera parallel interface“)
- abholend der Daten mit dem **CCDC Block**
- umrechnen der RAW Bayer Daten in YCrCb im **Preview Block**
- skalieren der Daten auf Displayformat im **Resizer Block**
- Übergabe an die v4l2 Schnittstelle

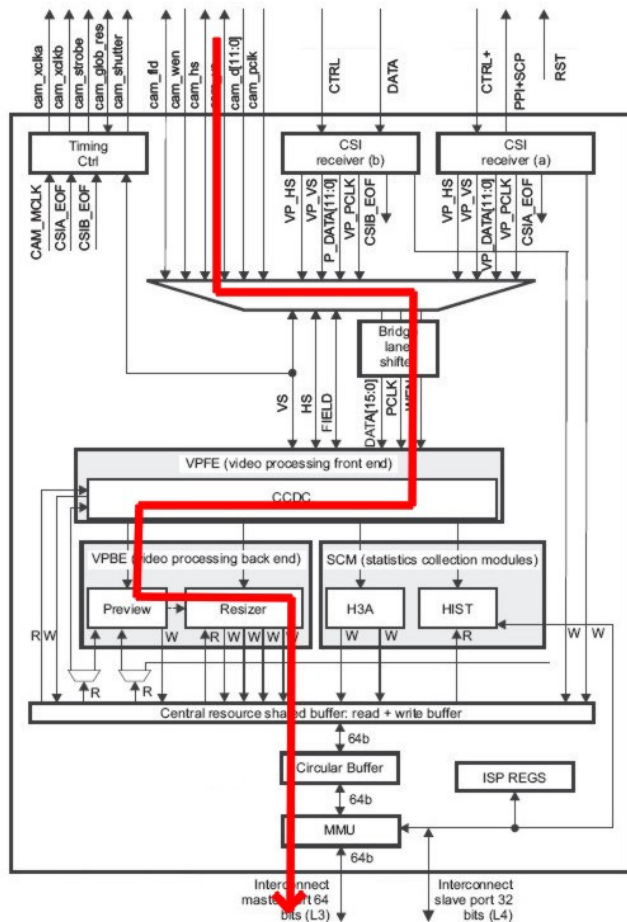


Abbildung 2 „OMAP-35xx, Beispiel der ISP Verwendung“

1.3.1 Voraussetzungen

Folgende Komponenten sind zum Umsetzen des Beispiels notwendig.

- VM-010-COL-LVDS (mt9v024) Kamera
- phyCARD-L Entwicklungskit
- Linux BSP ab 3.2.xx Kernel

Im Linuximage müssen folgende Komponenten vorhanden sein.

- Kameratreiber mt9v022 (gepatcht für mt9v024 und als v4l2 subDevice Treiber)
- OMAP3 Camera-ISP Treiber
- gstreamer Framework (<http://gstreamer.freedesktop.org/>)
- Media Controller Framework

1.3.2 Umsetzung des gstreamer Beispiels

Script zur Anzeige des Bildes der Kamera auf dem Display des phyCARD-L Kits:

```
-----
#!/bin/sh

echo "load driver"
echo "====="
modprobe omap3_isp

echo "configure ISP with media_control"
echo "====="

media-ctl -r

media-ctl -f "\mt9v022 3-0048\":0 [SGBRG8 752x480 (0,0)/752x480],
\OMAP3 ISP CCDC\":0 [SGBRG10 752x480],
\OMAP3 ISP CCDC\":2 [SGBRG10 752x479],
\OMAP3 ISP preview\":0 [SGBRG10 752x479(8,4)/738x471],
\OMAP3 ISP preview\":1 [UYVY 738x471],
\OMAP3 ISP resizer\":0 [UYVY 738x471],
\OMAP3 ISP resizer\":1 [UYVY 640x480]"

media-ctl -l '16:0 -> 5:0[1], 5:2 -> 7:0[1], 7:1 -> 10:0[1], 10:1 -> 12:0[1]'

echo "starting gstreamer ..."
echo "====="
gst-launch-0.10 \
    v4l2src device=/dev/video6 ! \
    ffmpegcolorspace ! \
    fbdevsink 2>/dev/null
-----
```

Es erfolgt die Anzeige des Farb-Livebildes mit einer Auflösung von 640x480 Bildpunkten auf dem Display des phyCARD-L Kits.

© PHYTEC Messtechnik GmbH 2012

Europe: Support Hotline: +49 (6131) 9221-31 • <http://www.phytec.de>

North America: Support Hotline: + 1-800-278-9913 • <http://www.phytec.com>

1.3.3 Aufbau des Scripts

a) Laden des Treibers.

Beim Laden des „omap3_isp“ Treibers wird bei angeschlossener Kamera der vorhandenen „mt9v022“ Treiber gleich mit geladen. Kontrolle der geladenen Treiber ist mit „lsmod“ möglich.

b) Konfiguration des OMAP3 Camera-ISP Treiber mit dem Mediacontrol-Tool

Die Konfiguration erfolgt üblicherweise in 3 Schritten:

- media-ctl -r (Reset der Einstellungen)
- media-ctl -f (Setzen der Eigenschaften der einzelnen Blöcke inkl. des Kameratreibers)
- media-ctl -l (Setzen der Verknüpfungen (Ablauf) der einzelnen Blöcke)

Auszug der Funktionen des Mediacontrol-Tools:

```
media-ctl -p (print device topology)
media-ctl -r (reset)
media-ctl -f (set format)
media-ctl -l (setup links)
media-ctl -h (help)
```

Links and formats are defined as

```
link      = pad, '->', pad, '[', flags, ']' ;
format    = pad, '[', fcc, ' ', size, [ ' ', crop ], ']' ;
pad       = entity, ':', pad number ;
entity    = entity number | ( '\"', entity name, '\"' ) ;
size      = width, 'x', height ;
crop      = left, ',', top, ',', size ;
```

where the fields are

```
entity number  Entity numeric identifier
entity name    Entity name (string)
pad number     Pad numeric identifier
flags          Link flags (0: inactive, 1: active)
fcc            Format FourCC
width          Image width in pixels
height         Image height in pixels
```

Weiterführende Informationen über den Status der OMAP3 Camera-ISP Treiber und den Einstellungsmöglichkeiten finden Sie in den Sourcen des jeweiligen BSPs.

c) Aufbau der gstreamer Kette.

Wichtig ist, daß die Quelle der Bilddaten für den gstreamer ein Ausgang aus der OMAP3-ISP-Kette ist, der als „/dev/video[0..n]“ abgebildet wird (Kontrolle mit „media-ctl -p“).

1.3.4 Zusätzliche Funktionen

PHYTEC stellt für eine weitreichende Nutzung der Kamerasensoren und der ISP-Schnittstelle noch folgende Funktionen in Ihren BSPs zur Verfügung.

- a) Anpassen der Kameraeigenschaften an eine Applikation oder für die Testphase, indem bestimmte Kameraregister vom Kunden konfiguriert werden können.

Mit dem gstreamer plug in „i2c file...“ und eine für jede Kamera passende Konfigurationsdatei liefert PHYTEC die Möglichkeit die Kameraregister nach eigenen Vorstellungen zu modifizieren. In der Gstreamer Kette wird dazu folgende Zeile erweitert:

```
v4l2src device=/dev/video6 ! \
i2c file register-settings-mt9v022.txt show=0 dev /dev/v4l-subdev[0..n] ! \
ffmpegcolorspace ! \
```

Die Anwendung der Registersettings sind im Dokument „LAN-052“ erklärt.

- b) Konfiguration des „Bridge-lane shifters“ / “Data-lane Shifters“

Der „Brigge lane shifter“ sitzt vor dem CCDC-Block des ISP und bestimmt wie die ankommenden parallelen Kameradaten zugeordnet werden. Somit kann für verschiedene Hardwarekonstellationen die Lage der parallelen Kameradaten zugeordnet werden.

Table 12-27. Data-Lane Shifter

Sensor	Connected to	Data Lane Shifter 0	Data Lane Shifter 1	Data Lane Shifter 2	Data Lane Shifter 3	Note
8 bits	[7:0]	8 bits	6 bits	4 bits	2 bits	
	[9:2]	10 bits	8 bits	6 bits	4 bits	
	[11:4]	12 bits	10 bits	8 bits	6 bits	
10 bits	[13:6]	14 bits	12 bits	10 bits	8 bits	CSI2 only
	[9:0]	10 bits	8 bits	6 bits	4 bits	
	[11:2]	12 bits	10 bits	8 bits	6 bits	
12 bits	[13:4]	14 bits	12 bits	10 bits	8 bits	CSI2 only
	[11:0]	12 bits	10 bits	8 bits	6 bits	
	[13:2]	14 bits	12 bits	10 bits	8 bits	CSI2 only
14 bits	[13:0]	14 bits	12 bits	10 bits	8 bits	CSI2 only

Abbildung 3 „OMAP-35xx, Data-Lane Shifter“

Bei der phyCARD-L liegen die Kameradaten 8-Bit Breit an cam_D[11:4]. Der Data-Lane Shifter ist im phyCARD-L BSP auf =2 voreingestellt.

Durch eine Änderung das Parameters:

```
bootargs="$bootargs board_pca102_phycam.dls=2"
```

in der „config“ Datei können auch andere Zuordnungen [0..3] getroffen werden.