

Anbindung des uvcvideo Treibers für eine CAM-104H

19. September 2011

Inhaltsverzeichnis

1	Allgemeines	2
1.1	Hardware	2
1.2	Software	2
1.2.1	Gstreamer	2
2	UVCVideo Treiber	4
2.1	UVCVideo Treiber zum BSP hinzufügen	4
2.2	PTXdist Einstellungen	4
3	Gstreamer	6
3.1	Beispiele	6
3.1.1	Einzelbild-Aufnahme	6
3.1.2	Livebild auf Framebuffer(Farbe)	6
3.1.3	Livebild auf Framebuffer(Monochrome)	6

Kapitel 1

Allgemeines

In diesem Kapitel wird auf die verwendete Hardware bzw. Software eingegangen.

1.1 Hardware

Folgende Hardware wurde verwendet:

- PHYCORE-i.MX27
- PHYTEC CAM-104H

1.2 Software

Folgende Software wurde verwendet:

- OSELAS.Toolchain-2011.02.0 - *platform-arm-v5te-linux-gnueabi-gcc-4.5.2-glibc-2.13-binutils-2.21-kernel-2.6.36-sanitized*
- OSELAS.BSP-Phytec-phyCORE-i.MX27-PD11.1.0
- uvcvideo.tar.gz - *Beinhaltet uvcvideo Treiber und rules für ptxdist*

1.2.1 Gstreamer

Es wurden die folgenden gstreamer bzw. gst-plugins Versionen verwendet:

- gstreamer-0.10.25
- gst-plugins-ugly-0.10.13
- gst-plugins-good-0.10.17
- gst-plugins-base-0.10.25
- gst-plugins-bad-0.10.17

Diese sind im *OSELAS.BSP-Phytec-phyCORE-i.MX27-PD11.1.0* mit den dazugehörigen Patches enthalten. Mit dem Aufruf von *ptxdist menuconfig* im BSP Verzeichnis kann diese Software, unter folgenden Menüpunkt, ausgewählt werden.

```
PTXdist Base Configuration ->  
  Graphics & Multimedia ->  
    gstreamer ->  
      ...
```

Kapitel 2

UVCVideo Treiber

Der uvcvideo Treiber wurde so bearbeitet, dass die CAM-104H damit lauffähig ist. In diesem Kapitel wird erklärt wie man diesen Treiber zum BSP hinzufügt.

2.1 UVCVideo Treiber zum BSP hinzufügen

Das uvcvideo tar archive (uvcvideo.tar.gz) beinhaltet zwei Unterverzeichnisse *local_src* und *rules*. Diese müssen in das BSP-Verzeichnis eingefügt werden.

Dies geschieht am einfachsten mit folgenden Befehl im BSP-Verzeichnis:

```
$ tar xvf $PFAD_ZUM_ARCHIVE
```

Wobei *\$PFAD_ZUM_ARCHIVE* ein Platzhalter für den Pfad darstellt der zum Archive führt.

2.2 PTXdist Einstellungen

Nachdem wir den Treiber und Rules in ptxdist hinzugefügt haben, kann unter folgenden Menüpunkt der Treiber ausgewählt werden. Dieser wird bei dem nächsten starten von *ptxdist go* kompiliert.

```
Project Specific Configuration ->
  uvcvideo
```

Dies sollte auf "`< * >`" gestellt werden.

Zuvor wir mit den kompilieren beginnen, sollte man sicherstellen, dass v4l2 im Kernel aktiviert ist. Die schaut man mit *ptxdist kernelconfig* unter folgenden Menüpunkt nach.

```
Device Drivers ->
  Multimedia support ->
    <*> Video For Linux
    Video capture adapters ->
      <*> V4L USB devices
```

Unter dem Menüpunkt *V4L USB devices* sollte der *USB Video Class (UVC)* nicht ausgewählt werden, da dieser jetzt extern zu Verfügung steht.

Falls *ptxdist go* erfolgreich durchgelaufen ist, muss man mit *ptxdist images* fertige Images erstellen und dieses auf das Target neu flashen. Auf dem Target kann man mit dem Befehl *modprobe*

uvcvideo den Treiber laden bzw. mit dem Befehl *rmmmod uvcvideo* entladen. Um das module beim starten des Targets automatisch zu laden, verwendet man folgenden Befehl:

```
$ echo "uvcvideo" >> /etc/modules
```

Kapitel 3

Gstreamer

Es kann zu Problemen bei dem abspielen von Videodaten via gstreamer kommen. Folgende Probleme sind bekannt:

Data Flow Error Dies scheint ein Initialisierungsproblem zu sein der manchmal auftritt, gstreamer neustarten hilft.

Rauschen bei Einzelbild-Aufnahme Dies tritt nur bei dem ersten Bild auf, eine Lösung ist den num_buffers parameter von v4l2src zu erhöhen.

Schwarzes Bild Kamera neu verbinden.

Erfolgreich getestet wurde ein 2 Tage Livebild-Dauerbetrieb.

3.1 Beispiele

In diesen Abschnitt werden verschiedene Beispiele beschrieben. Das Anpassen der Auflösung ist jedoch nur zum Teil möglich. Das Video Device muss ggf. angepasst werden.

3.1.1 Einzelbild-Aufnahme

```
$ gst-launch v4l2src device=/dev/video2 num-buffers=1 !  
  video/x-raw-rgb,width=744,height=480 ! ffmpegcolorspace ! jpegenc !  
  filesink location=image.jpg
```

3.1.2 Livebild auf Framebuffer(Farbe)

```
$ gst-launch v4l2src device=/dev/video2 ! video/x-raw-rgb,width=744,height=480  
  ! ffmpegcolorspace ! fbdevsink 2
```

3.1.3 Livebild auf Framebuffer(Monochrome)

```
$ gst-launch v4l2src device=/dev/video2 !  
  video/x-raw-gray,width=744,height=480 ! ffmpegcolorspace ! fbdevsink 2
```