

Application Note No.: LAN-062d_1
 Version: 1.0
 Autor: H. Fendrich
 Date: 20.10.2014

Historie:

Version	Änderungen	Datum	Autor
1.0	Erstellung des Dokuments	20.10.2014	H. Fendrich

phyCAM VM-012 - Remapping

Inhaltsverzeichnis

1	Einleitung	2
2	Pixelreihenfolge VM-012	3
3	Remapper Software.....	5
3.1	GStreamer-Plug-in " <i>gstvita-1300_remapper</i> "	5
3.2	C-Library " <i>vita-1300_remapper</i> "	6

Konventionen

Folgende Konventionen gelten in dieser Application Note:

- Text in *blauer Kursivschrift* steht für einen Hyperlink innerhalb oder außerhalb des Dokumentes. Klicken Sie auf diesen Link, um schnell zu der entsprechenden URL, Kapitel, Tabelle oder Bild zu springen.
- Text in **fett und kursiv** beschreibt eine Interaktion durch den Benutzer, die auf dem Bildschirm definiert ist.
- Text in Consolas zeigt einen durch den Benutzer frei einzugebenden Text oder Programmcode (auszugsweise, oder vollständig).
- Text in *kursiv* zeigt Eigennamen von Entwicklungswerkzeugen und Steuerelementen (Fenster, Registerkarten, Befehle etc.), die innerhalb des Entwicklungswerkzeuges verwendet werden, es findet keine Interaktion statt.
- **Weißer Text auf schwarzem Hintergrund** zeigt das Ergebnis der Benutzerinteraktion (Befehl, Programmausführung, etc.).

1 Einleitung

In den phyCAM-Kameramodulen VM-012 werden die folgenden beiden Kamerasensoren VITA 1300 der Firma OnSemiconductor eingesetzt:

- phyCAM VM-012-BW Serie: Sensor NOIV2SN1300A
- phyCAM VM-012-COL Serie: Sensor NOIV2SE1300A

Eine Besonderheit der VITA 1300 – Sensoren ist, daß die Pixel einer Bildzeile nicht in der Reihenfolge ihrer Position auf dem Sensorchip ausgegeben werden. Vielmehr sind die ausgelesenen Bilddaten in Kernen (Blöcken) organisiert, innerhalb derer die Ausgabeposition unterschiedlich gruppiert ist. Die Reihenfolge ist zudem abhängig vom eingestellten Subsampling-Modus. Die resultierende Pixelreihenfolge ist in [Abschnitt 2](#) detailliert beschrieben.

Zur korrekten Anzeige oder zur Auswertung der Bilder müssen die Pixel daher umsortiert werden. Dazu stellt PHYTEC für die VM-012 Kameramodule eine Remapper-Software unter *Embedded Linux* zur Verfügung.

Folgende Softwarekomponenten finden Sie im Lieferumfang der Kameramodule:

- ein GStreamer-Plug-in *gstvita-1300_remapper*
- eine C-Library *vita-1300_remapper*

Die Beschreibung der Softwarekomponenten finden Sie in [Abschnitt 3](#).

2 Pixelreihenfolge VM-012

Bedingt durch den internen Aufbau des Kamerasensors VITA 1300 werden Pixel nicht konsekutiv sequenziell entsprechend der physikalischen Anordnung auf dem Sensor entsprechend ausgegeben. Die Zeilen des Pixel-Arrays sind vielmehr in so genannte Kernels unterteilt, die aus Gruppen von jeweils 8 Pixeln gebildet werden (8 Pixel in x-Richtung je Zeile):

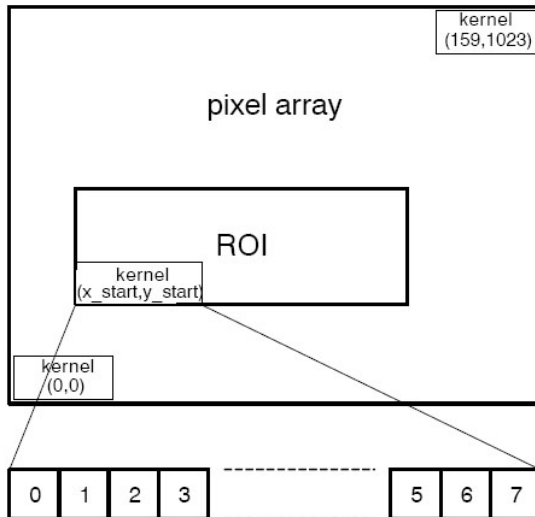


Bild 1: Anordnung der Pixel-Kernel auf dem Sensor-Array

Die Reihenfolge, in der die Pixeldaten ausgegeben werden, ist an den Kernels orientiert und abhängig vom Subsampling-Modus.

1. Ohne Subsampling (VM-012-BW /2 VM-012-COL)

Die Pixelreihenfolge ist in geraden und ungeraden Kernels unterschiedlich entsprechend der folgenden Grafik:

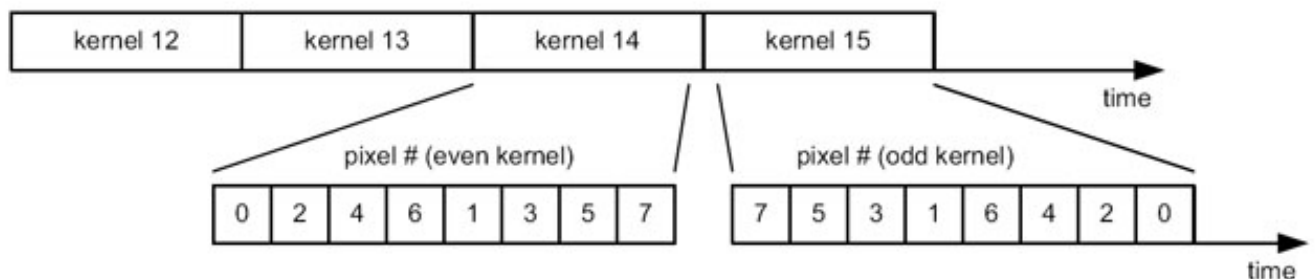


Bild 2: Pixelreihenfolge ohne Subsampling

2. Mit Subsampling (monochromer Sensor, VM-012-BW)

Im Subsampling-Modus werden zwei benachbarte Kernel zu einem Kernel zusammengefaßt (*Bild 3*). Dadurch wird die Anzahl der ausgegebenen Pixel halbiert. Es werden immer die geraden Pixel entnommen.

Im Gegensatz zur Betriebsart ohne Subsampling gibt es keinen Unterschied in der Pixelreihenfolge von geraden und ungeraden Kerneln.

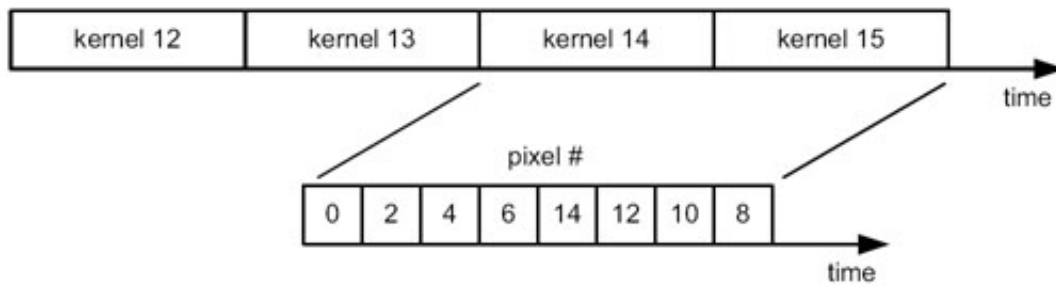


Bild 3: Pixelreihenfolge mit Subsampling im Monochrome-Modus

3. Mit Subsampling (color Sensor, VM-012-COL)

Im Subsampling-Modus werden zwei benachbarte Kernel zu einem Kernel zusammengefaßt (*Bild 4*). Dadurch wird die Anzahl der ausgegebenen Pixel halbiert. Es werden immer 2 benachbarte Pixel entnommen und dann 2 Pixel übersprungen.

Im Gegensatz zur Betriebsart ohne Subsampling gibt es keinen Unterschied in der Pixelreihenfolge von geraden und ungeraden Kerneln.

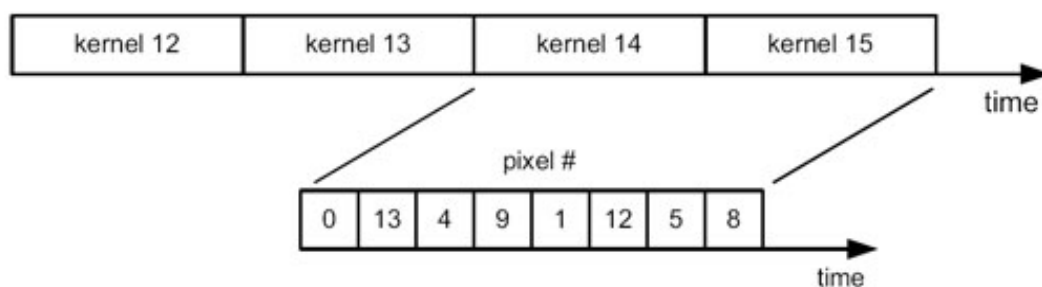


Bild 4: Pixelreihenfolge mit Subsampling im Color-Modus

3 Remapper Software

Für die Sortierung der Bilddaten der VM-012 Kameraboards werden von PHYTEC folgende Softwarekomponenten unter *Linux* bereitgestellt:

- ein GStreamer-Plug-in *gstvita-1300_remapper*
- eine C-Library *vita-1300_remapper*

Diesem liegt jeweils ein in *C* geschriebener Algorithmus zugrunde, der die vorhandenen Bilddaten aus einem Array ausliest, umsortiert und in ein Array zurückschreibt. Er verwendet also die bereits im Speicher befindlichen Bilddaten.

Der Algorithmus kann auf die Datenwortbreite eines Pixels im Array konfiguriert werden. Dazu wird die verwendete Breite eines Datenpixels (8 Bit oder 16 Bit) übergeben.

Hinweis:

Für 10 oder 12 Bit Auflösung wird die Einstellung 16 Bit verwendet, die restlichen Bits bleiben frei.

Die Routine erwartet einen Zeiger auf ein Quell-Array mit einem Image (Einzelbild), wobei die Bildbreite (x-Richtung) durch 8 teilbar sein muss. Das sortierte Bild wird im Ziel-Array abgelegt.

Dieser Algorithmus ist in der C-Library direkt umgesetzt und im GStreamer Plug-in in einen entsprechenden „gst-plug-in“ Container eingebunden.

3.1 GStreamer-Plug-in "*gstvita-1300_remapper*"

Um schnell Kamerafunktionen zu testen und ein Kamerabild zur Anzeige zu bringen ist es von Vorteil, mit einem einfachen Tool zu arbeiten. PHYTEC stellt zu diesem Zweck Skripte für das verbreitete *GStreamer* – Framework zur Verfügung. Der *GStreamer* kann ohne Programmieraufwand durch einfache Skript-Aufrufe konfiguriert werden. Die von PHYTEC mitgelieferten Skripte verwenden den *GStreamer* zum Darstellen und Übertragen von Bild-datenströmen. Weiterhin können Einzelbilder oder Bildfolgen in verschiedenen Formaten gespeichert werden.

Informationen zum freien Multimedia Framework *GStreamer* und dessen Funktion finden Sie unter www.gstreamer.net.

Damit die Bilddaten der VM-012 in den *GStreamer* Beispielen in der richtigen Datenreihenfolge verwendet werden können, hat PHYTEC den Remap-Algorithmus in einem *gst-plug-in* gekapselt.

Das *gstvita-1300_remapper* Plug-in kann direkt nach dem Abholen des Bilddatenstroms von der V4L2-Schnittstelle und der weiteren Verarbeitung im *GStreamer* eingefügt werden.

Informationen zum Plug-in erhalten sie durch die Eingabe von:

```
> gst-inspect vita1300_remapper
```

Da die Datenformate und Bitbreiten vom *GStreamer* selbst ausgehandelt werden, sind bei der Verwendung des Plug-ins lediglich folgende Parameter zu setzen:

```
vita1300_remapper [mode= ] [passthrough= ]
```

mode: 0 = kein Subsampling bei VM-012-BW/VM-012-COL (default)
 1 = mit Subsampling bei VM-012-BW (monochrom)
 2 = mit Subsampling bei VM-012-COL (color)

passthrough: false = remapper mode verwenden (default)
 true = remapper mode nicht verwenden, Daten werden ohne Sortierung weitergeleitet

Anwendungsbeispiel:

Livebildanzeige der VM-012-BW mit dem *GStreamer*

Verwendung des Remapper-mode 0

```
gst-launch-0.10 \  
  v4l2src device= /dev/video0 ! \  
  video/x-raw-gray,width=1280,height=1024 ! \  
  vita1300_remapper mode=0 passthrough=false ! \  
  mfw_isink axis-top=0 axis-left=80 disp-width=640 disp-height=480
```

3.2 C-Library "*vita-1300_remapper*"

Werden die Bilddaten der VM-012 über das V4L2-Interface von einem C/C++ - Programm (z.B. einer eigenen Applikation) abgeholt, so liegen die Bilddaten in ungeordneter Reihenfolge vor. Zum Ordnen der Bilddaten („remappen“) kann die von PHYTEC bereitgestellte Remapper-Library *vita-1300_remapper* verwendet werden. Zum Sortieren wird die darin enthaltene Funktion *vita1300_convert* verwendet.

```
enum vita1300_result vita1300_convert (  
    void *dst,  
    void const *src,  
    enum vita1300_subsampling_mode mode,  
    unsigned int w, unsigned int h,  
    unsigned int bpp);
```

Übergabewerte:

- *dst: Zeiger auf das Ergebnisarray
- *src: Zeiger auf das Quellarray
- mode: 0 = ohne Subsampling bei VM-012-BW/VM-012-COL
 1 = mit Subsampling bei VM-012-BW (monochrom)
 2 = mit Subsampling bei VM-012-COL (color)
- w: Bildbreite in Pixel (muß durch 8 teilbar sein)
- h: Bildhöhe in Pixel
- bpp: Bit pro Pixel (8 oder 16)

Rückgabewert:

```
enum vita1300_result { VITA1300_RES_OK, VITA1300_BAD_X,
                      VITA1300_BAD_MODE, VITA1300_BAD_BPP,
                      VITA1300_BAD_OVERLAP};
```

- 0: OK
- 1: Fehler, Bildbreite nicht durch 8 teilbar
- 2: Fehler, falscher Mode – Parameter
- 3: Fehler, falscher bpp – Parameter
- 4: Fehler, Quell- und Ergebnisarray überlappen sich

Damit die Funktion *vita1300_convert* im Programm verwendet werden kann, müssen einige Deklarationen bekannt gegeben werden. Dies erfolgt durch das Einbinden der VITA1300-Remapper Header-Datei *vita1300-remapper.h* im Applikationsprogramm mit:

```
include vita1300-remapper.h
```

Bei Fragen, oder wenn Sie weitere Informationen benötigen, wenden Sie sich bitte an unsere Support Mitarbeiter.

Europa (außer Frankreich):	Frankreich:
▪ +49 (6131) 9221-31	▪ +33 (0) 2 43 29 22 33
▪ support@phytec.de	▪ support@phytec.fr
Nord Amerika:	Indien:
▪ +1 (206) 780-9047, or +1 (800) 278-9913	▪ +91 (80) 4086 7046/48
▪ support@phytec.com	▪ support@phytec.in
China:	
▪ +86 (755) 3395 5875	
▪ support@phytec.cn	