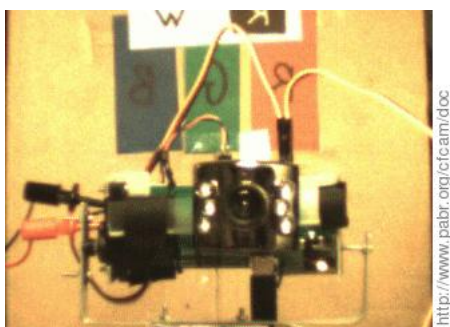

CFcam - Interfaçage d'une caméra CMOS avec une carte Gumstix Connex

Copyright © 2006, 2007, 2008 pabr@pabr.org
Tous droits réservés. (All rights reserved.)

Ce document explique comment connecter un capteur vidéo CMOS numérique directement au port CompactFlash d'une carte Gumstix Connex (ou plate-forme similaire à base de PXA255).



READ THE HYPERTEXT VERSION HERE:
<http://www.pabr.org/cfcam/doc/cfcam.fr.html>

Historique des versions		
1.2	2008-09-07	Avertissement relatif aux conflits de bus
1.1	2006-12-07	Décodage RGB. Autres capteurs CMOS. Performances. Driver dans pxaRC.
1.0	2006-09-12	Première publication

Table des matières

1. Introduction	3
2. Composants	3
3. Construction	3
4. Logiciel	4
5. Fonctionnement	4
6. Résultats	4
6.1. Images	4
6.2. Performances	6
6.3. Améliorations envisageables	6
Bibliographie	7

1. Introduction

Contrairement au PXA270, le processeur PXA255 n'a pas d'interface d'acquisition vidéo. Pour traiter des flux vidéo avec une latence minimale, il faut en principe utiliser un capteur CMOS cadencé par un oscillateur dédié, contrôlé par le bus I2C, remplissant à débit constant une FIFO dont la sortie serait connectée au bus mémoire du processeur. Un exemple est décrit de façon détaillée dans [FS2006].

Ce document propose une solution plus simple et plus économique (mais moins fiable dans certains cas) reposant sur l'observation qu'un capteur CMOS fonctionne raisonnablement bien avec une horloge non périodique.

2. Composants

- gumstix connex [http://docwiki.gumstix.org/Basix_and_connex]
- cfstix [http://docwiki.gumstix.org/Expansions#92-pin_expansion_boards]. Nous n'utiliserons pas les spécificités de CompactFlash. Seuls nous intéressent le connecteur Hirose 92 points, les buffers sur les bus d'adresses et de données, et le connecteur au pas de 1,27 mm.

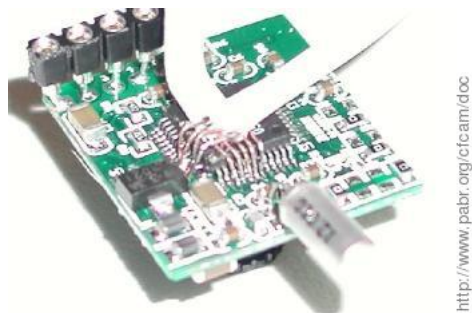
Avertissement

Ne connectez pas un capteur CMOS au port CF d'une carte netcf ou wifistix-cf, car les sorties ne sont pas trois-états et seraient donc en conflit avec celles du circuit LAN91C111 ou du module Wi-Fi.

- Un connecteur CF femelle. Sources possibles :
 - connecteur CF standard (difficile à acheter à l'unité) ;
 - à partir d'une carte CF défectueuse ;
 - connecteur femelle générique 2x25x1,27 mm (non conforme à la norme CF, mais suffisamment compatible).
- Un capteur CMOS monté sur un circuit imprimé. Sources possibles :
 - démonter une webcam USB (modèles testés : SN9C102 [<http://www.mnementh.co.uk/sonix/>]+OV7630 et SN9C120+HV7131R) ;
 - C3088/C3188 avec horloge externe (non testés) ;

3. Construction

Figure 1. SN9C102 instrumenté



- Connecter GND au USB_GND de la caméra.
- Connecter V_BATT (+5 V) au USB_VBUS de la caméra.

- Connecter nPIOR à SEN_CLK (également dénommé MCLK).
- Connecter S_IMG[0..8] à XMD[0..8].

4. Logiciel

- Désactiver /etc/init.d/S30pcmcia.
- Activer SKTSEL : pxa_gpio_mode(GPIO54_pSKTSEL_MD).
- Activer le décodage de la plage d'adresses PCMCIA : MECR=MECR_CIT.
- Programmer les timings I/O : MCIO(0)=0 fonctionne; MCIO(0)=(10<<14)|(5<<7)|10 donnerait une marge de sécurité supplémentaire.
- Transférer par DMA quelques mégaoctets à partir de 0x2000000 (slot 0), en mode WIDTH2|BURST4.
- Decoder les données brutes. En pratique il s'agit de supprimer les synchronisations horizontales et verticales et de reconstruire les couleurs à partir d'une image Bayer.

La documentation des contrôleurs mémoire et DMA est dans [PXA255_DEVEL].

[PXARC] contient une implémentation complète de cette approche.

5. Fonctionnement

Le système fonctionne de la façon suivante :

- Tant que le SN9C1xx ne détecte pas d'hôte USB, il reste en veille avec SEN_CLK en haute impédance.
- Dans sa configuration par défaut, le capteur est capable de produire des images. Sinon, nous devrions le configurer à l'aide d'I2C.
- Chaque lecture à l'adresse 0x20000000 génère une impulsion sur SEN_CLK. Bien que les accès DMA se déroulent en rafales, ceci suffit à cadencer le capteur.

6. Résultats

6.1. Images

Figure 2. Caméra connectée au cfstix et pointée vers un miroir

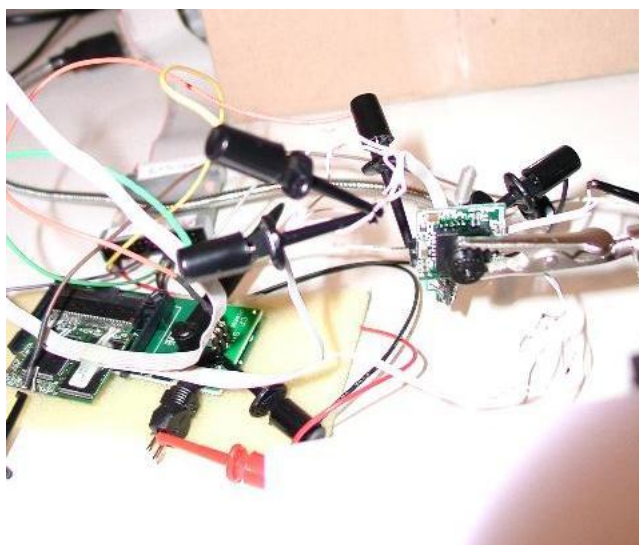


Figure 3. Données brutes d'une capture DMA (un mot de 16 bit sur deux, décalés d'un bit vers la droite)

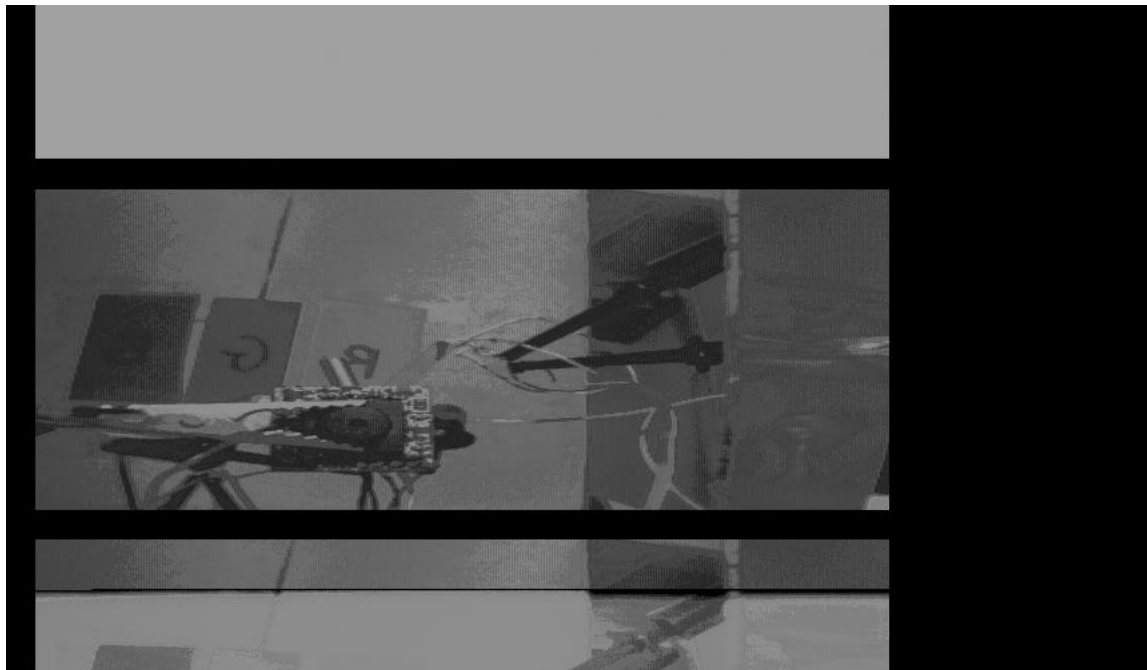


Figure 4. Signaux (SEN_CLK, S_PCK, S_IMG7)

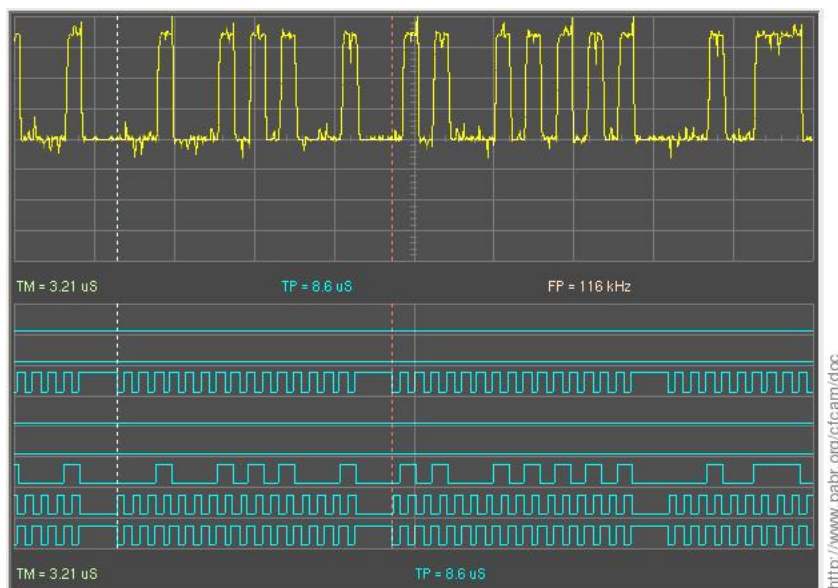
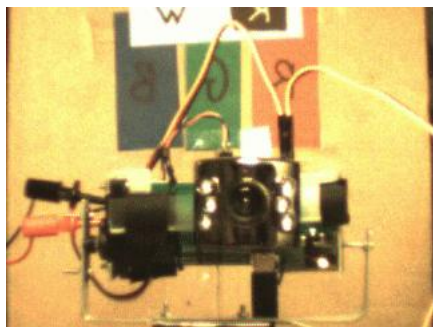
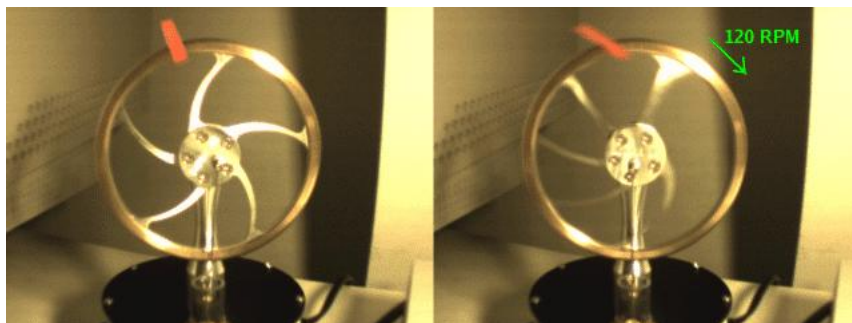


Figure 5. Une autre webcam modifiée (SN9C120+HV7131R)**Figure 6. Caméra pointée vers un miroir ; décodage Bayer**

6.2. Performances

Figure 7. Artefacts d'un capteur à *rolling shutter*

Les performances sont limitées par la complexité du cycle de lecture CF/PCMCIA. Un prototype (Figure 5, « Une autre webcam modifiée (SN9C120+HV7131R) ») atteint 320x240x4im/s et 640x480x4im/s. Avec un capteur CMOS générant un pixel par cycle d'horloge, on devrait obtenir 320x240x28im/s et 640x480x8im/s (soit 3,8 M pixels/s, intervalles de synchro compris). À titre de comparaison, l'interface de capture vidéo du PXA270 peut cadencer des capteurs jusqu'à 52 MHz.

La figure ci-dessous montre un effet indésirable de ce faible débit lorsque le capteur est de type *rolling shutter* : la déformation des objets mobiles est très prononcée.

6.3. Améliorations envisageables

- Lire le bus avec des cycles pour mémoire statique au lieu de cycles CF/PCMCIA, afin d'augmenter le débit. Pour cela, il faudrait récupérer un signal nCS sur le connecteur à 92 broches.

Bibliographie

[PXA255_DEVEL] *Intel PXA255 Processor. Developer's manual. 27869302.pdf.*

[PXARC] *pxaRC - Logiciels pour le modélisme et la robotique sur Linux/PXA255/PXA270 .* <http://www.pabr.org/pxarc/doc/pxarc.fr.html> .

[FS2006] *Linux Robot with Omnidirectional Vision.* George Francis et Libor Spacek. <http://cswww.essex.ac.uk/mv/omnipapers/gwfran06with-header.pdf>.