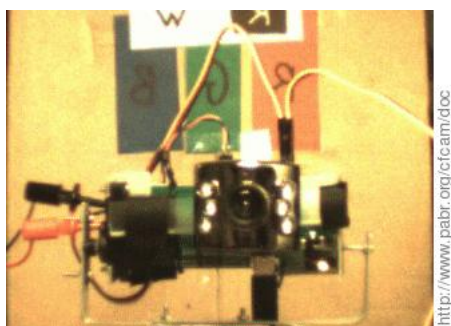

CFcam - Connecting a CMOS camera to a Gumstix Connex motherboard

Copyright © 2006, 2007, 2008 pabr@pabr.org
All rights reserved.

This document explains how to connect a digital CMOS image sensor directly to the CompactFlash port of a Gumstix Connex computer (or similar PXA255-based platform).



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

Revision History		
1.2	2008-09-07	Warnings about bus conflicts
1.1	2006-12-07	RGB decoding. More CMOS sensors. Performance results. Driver in pxaRC.
1.0	2006-09-12	Initial release

Table of Contents

1. Introduction	3
2. Parts required	3
3. Hardware	3
4. Software	4
5. Operation	4
6. Results	4
6.1. Pictures	4
6.2. Performance	6
6.3. Possible improvements	6
Bibliography	7

1. Introduction

Unlike the more recent PXA270, the PXA255 processor does not have hardware support for video capture. The proper way to add real-time, low-latency imaging capability to a PXA255 is with a CMOS sensor clocked by a dedicated crystal, controlled with I2C, feeding data at constant rate into a FIFO whose other end is connected to the memory bus of the processor. See [FS2006] for a detailed example.

This document suggests a simpler, lower-cost (but possibly less reliable) approach based on the observation that CMOS sensors typically work reasonably well with a non-periodic clock.

2. Parts required

- gumstix connex [http://docwiki.gumstix.org/Basix_and_connex]
- cfstix [http://docwiki.gumstix.org/Expansions#92-pin_expansion_boards]. We don't really care about the CompactFlash functionality, but for \$25 we get a 92 pin Hirose connector, address/data bus transceiver, and 1.27 mm headers.

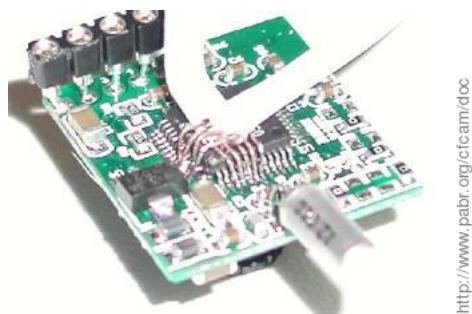
Warning

Do not connect a CMOS sensor to the CF port of a netcf or wifistix-cf board, as the outputs are not three-state and would conflict with those of the LAN91C111 or Wi-Fi chip.

- Female CF connector. Sources include:
 - Factory CF connector (hard to find in small quantities)
 - From a dead CF card
 - Generic 2x25x1.27 mm female connector (not CF-compliant, but reasonably compatible)
- CMOS sensor and break-out board. Sources include:
 - USB webcam (tested: SN9C102 [<http://www.mnementh.co.uk/sonix/>]+OV7630 and SN9C120+HV7131R)
 - C3088/C3188 with external clocking (not tested)

3. Hardware

Figure 1. SN9C102 with wires attached



- Connect GND to USB_GND of the camera.
- Connect V_BATT (+5 V) to USB_VBUS of the camera.
- Connect nPIOR to SEN_CLK (a.k.a. MCLK).

- Connect S_IMG[0..8] to XMD[0..8].

4. Software

- Disable /etc/init.d/S30pcmcia.
- Enable SKTSEL: pxa_gpio_mode(GPIO54_pSKTSEL_MD).
- Enable PCMCIA address decoding: MECR=MECR_CIT.
- Program I/O bus timings: MCIO(0)=0 works; MCIO(0)=(10<<14)|(5<<7)|10 would add a security margin.
- Transfer a few megabytes from 0x2000000 (slot 0) with DMA, using WIDTH2|BURST4.
- Decode the raw data. This typically involves removing the horizontal and vertical blanking intervals, and reconstructing RGB pixels from a Bayer image.

The memory controller and DMAC are documented in [PXA255_DEVEL].

[PXARC] contains a driver which implements all this.

5. Operation

Here is how it works (results with other sensors may vary)

- As long as the SN9C1xx does not see a USB host, it remains idle with SEN_CLK floating.
- The power-on configuration of the sensor is good enough to produce images. Otherwise, we would have to configure it with I2C.
- Each read at 0x2000000 will send a pulse on SEN_CLK. Although the DMA reads occur in bursts, this will clock the sensor.

6. Results

6.1. Pictures

Figure 2. Camera connected to cftix and pointed at a mirror

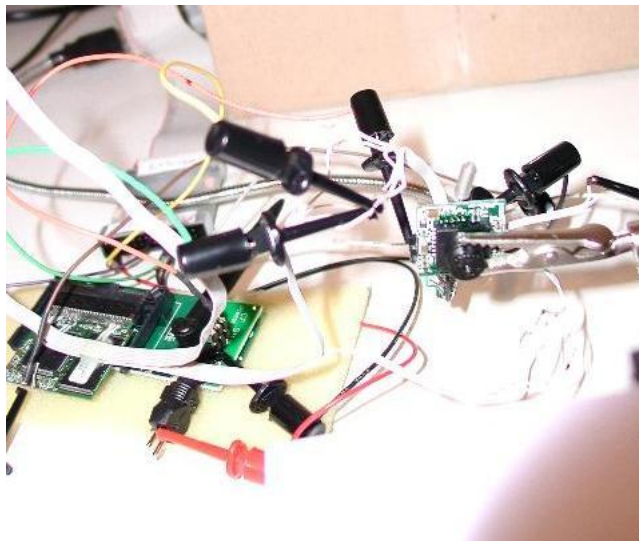


Figure 3. Raw data from DMA capture (every odd 16-bit word, right-shifted by one bit)



Figure 4. Signals (SEN_CLK, S_PCK, S_IMG7)

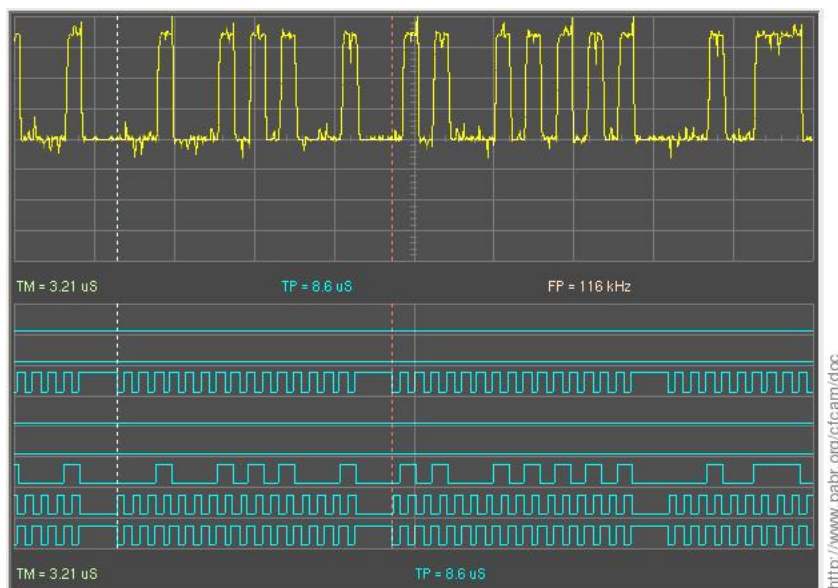
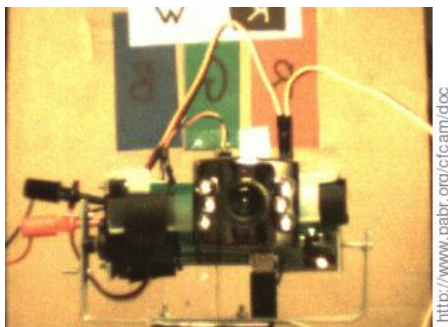


Figure 5. Another modified webcam (SN9C120+HV7131R)

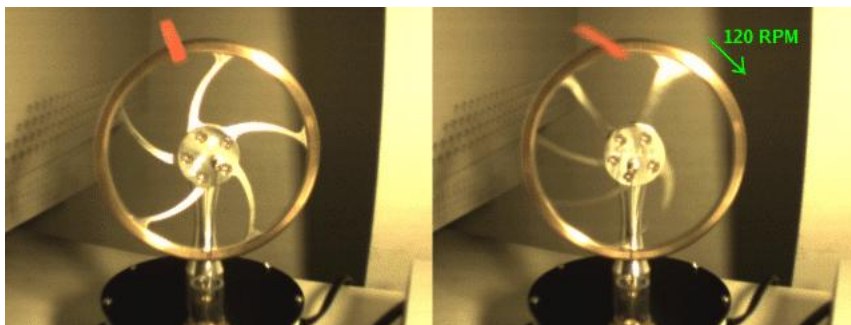


Figure 6. Camera looking at itself in a mirror; Bayer decoding



6.2. Performance

Figure 7. Rolling shutter artifacts



Performance is limited by the complexity of the asynchronous CF/PCMCIA read cycle. A prototype (Figure 5, "Another modified webcam (SN9C120+HV7131R)") has been tested at 320x240x14fps and 640x480x4fps. A CMOS sensor that outputs one pixel per clock cycle should achieve 320x240x28fps and 640x480x8fps (or 3.8 M pixels/s, including sync intervals). For comparison, the hardware capture interface of the PXA270 can clock sensors up to 52 MHz.

Figure 7, "Rolling shutter artifacts" shows an unwanted side effect of this low pixel rate when the sensor is a "rolling shutter" type: moving objects are significantly distorted.

6.3. Possible improvements

- Read sensor data with static memory bus cycles instead of CF/PCMCIA bus cycles. This may increase the pixel rate, but would require picking a nCS signal from the 92-pin connector.

Bibliography

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

[PXARC] *pxaRC - R/C and robotics software for Linux/PXA255/PXA270 .* <http://www.pabr.org/pxarc/doc/pxarc.en.html> .

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