The Video Clip Player: Philips Nexperia™ PNX0106 and Linux based platform

Armin Gerritsen & Marcin Klecha
Advanced Systems Laboratory

08-04-2006
Advanced Systems Laboratory

• Vision
  – A world where everyone can always connect to information, entertainment and services.

• Mission
  – To prototype advanced applications, one or two “Moore’s Law” generations ahead, using Nexperia™ platforms.
  – To gain insight on future requirements, identifying disruptive market and technology trends.

• Strategy
  – Show PS’ strengths in “Connected Consumer” through prototype demonstrators built on Nexperia™.
  – Develop market understanding through strategic customer engagement together with Business Clusters and Marketing and Sales.
  – Link program to business direction through Business Relevance and Technology Program Boards.
ASL: Where?
ASL: Where?
Outline

• Video Clip Player
  – Introduction
  – System Overview and Use Cases

• Linux & Software architecture
  – Video: Framebuffer Implementation
  – Audio: EPICS DSP
  – UPnP and DTCP-IP protection
  – Linphone: VoIP application

• Linux and prototyping
  – History
  – Measurements: powermanagement
  – Next steps
Video Clip Player: Introduction

The Video Clip Player (VCP) shows Linux-based applications and is based around the Nexperia™ PNX0106 System-on-Chip solution.

The VCP is hard-drive based reference design featuring a colour QCIF+ LCD and provides features such as FM radio, audio playback, photo viewing and video playback on LCD or TV.

The VCP has multiple connectivity options like USB 2.0, USB OTG, UART, SPI, BlueTooth, NFC and Ethernet.

The VCP is using the latest technologies like UPnP, VoIP and DTCP-IP.
Video Clip Player: Introduction

And what about Linux?

Linux played an essential role in the development and usage of the VCP in three ways

- The VCP runs Linux. Linux enabled Philips Semiconductors to quickly have a demonstrator platform to promote its Nexperia™ PNX0106 IC.

- The VCP can be easily used as a test-platform for Philips departments, to test the latest ideas and technologies.

- The VCP was prototyped using Linux, allowing IP to be validated, actual use-cases to be tested and performance measurements to be done, all before the IC itself was available.
Video Clip Player: Nexperia™ PNX0106

- ARM926 EJ-S
  - I-cache
  - D-cache

- Memory Controller
- YUV OUT
- LCD
  (6800/8080)
- SPI x2
- ADC 10bits
- GPIO
- Audio DAC
  - Headphone
- Audio ADC
- SPDIF
- I²S In/Out x2
- I²C x2
- SDRAM
- SRAM
- EEPROM
- FLASH
- ATA/IDE
- MCI
- USB2.0 HS OTG
- IEEE1394
- UART(2x)

Advanced Systems Laboratory Philips Semiconductors - Armin Gerritsen/Marcin Klecha - 08-04-2006
Video Clip Player: System overview

- TV OUT
- SAA7121
- MIC
- FM Radio
- TEA5761
- USB 2.0 Device
- USB OTG
- NFC
- BlueTooth
- Obex
- PAN
- 802.11
- Ethernet
- FM Radio
- TEA5761
- USB 2.0 Device
- USB OTG
- NFC
- BlueTooth
- Obex
- PAN
- 802.11
- Ethernet
Video Clip Player: Use Cases

• Audio/Video Player (showing MP3 and MPEG4)
• Audio Recording
• FM Radio
• Pictures Slideshow (LCD and TV)
• USB OTG (Pictures Download from a camera)
• USB 2.0 Device – PC Connection
• VoIP (Voice-over-IP)
• Send/Receive pictures via BlueTooth
• iRadio – audio streaming over TCP/IP
• Games
• UPnP Server with DTCP-IP link protection
• NFC mifare card reader
Video Clip Player: Audio/Video/Pictures

Main window of the GUI

- Video Player
- Audio Player
- Audio Applications
- Pictures
- Games
- Settings

JPEG slideshow to a TV or LCD

Audio/Video playback to LCD/TV
- *.3gp, *.mp4
- MPEG-4 SP + AMR Audio

Audio Recording and playback of the recording
From a MIC on the board

Audio MP3 playback from a CF/HDD via a headphone or speakers

FM Tuner - Radio
Video Clip Player: Connectivity and Networking
Video Clip Player: Connectivity and Networking

SEND or RECEIVE a picture from VCP-1 to a mobile phone via Bluetooth.
Video Clip Player: Connectivity and Networking

SEND or RECEIVE a picture from VCP-I to a mobile phone via BlueTooth

Ethernet, BlueTooth or WiFi
Video Clip Player: Connectivity and Networking

- SEND or RECEIVE a picture from VCP-I to a mobile phone via BlueTooth
- Audio/Video/Picture streaming from VCP-I
- Image printing from VCP-I
- Control AV streaming and image printing
- Browse the VCP's content using UPnP technology.
Video Clip Player: Connectivity and Networking

SEND or RECEIVE a picture from VCP-1 to a mobile phone via BlueTooth

Audio/Video/Picture streaming from VCP-1

Browse the VCP’s content Using UPnP technology,

Control AV streaming and image printing

DTCP-IP protected

iRadio - Audio streaming from Internet to VCP-1

Image printing from VCP-1

Ethernet, BlueTooth or WiFi

Internet
Video Clip Player: Connectivity and Networking

SEND or RECEIVE a picture from VCP-1 to a mobile phone via BlueTooth

Audio/Video/Picture streaming from VCP-1

Browse the VCP's content Using UPnP technology,

Control AV streaming and image printing

Ethernet, BlueTooth or WiFi

DTCP-IP protected

NFC card reader e.g. easy WiFi set-up

iRadio - Audio streaming from Internet to VCP-1

Image printing from VCP-1

Internet

Advanced Systems Laboratory Philips Semiconductors - Armin Gerritsen/Marcin Klecha - 08-04-2006
Video Clip Player: VoIP (SIP) telephony

Linphone
Telephony on Linux

Linphone
Home network or Hotspot

SIP Proxy: Siproxd

Linux PC

Linphone

VCP-1

Linphone

Home network or Hotspot

Internet

VCP-1

Linphone

Home network
Outline

• Video Clip Player
  – Introduction
  – System Overview and Use Cases

• Linux & Software architecture
  – Video: Framebuffer Implementation
  – Audio: EPICS DSP
  – UPnP and DTCP-IP protection
  – Linphone: VoIP application

• Linux and prototyping
  – History
  – Measurements: powermanagement
  – Next steps
Linux & Software architecture: Framebuffer

• VCP is using the Linux framebuffer architecture
  – Enables the usage of many of the Linux applications and graphic libraries
    (e.g. Qt(e)/Qtopia and DirectFB)

• ‘Out of the box’ the Linux framebuffer architecture doesn’t support typical LCD’s for CE devices
  – The absents of address lines prevents applications from directly accessing the memory of the LCD.
  – Therefore a virtual memory buffer is required, which needs to be synchronised with the real LCD memory.
Linux & Software architecture: Framebuffer

1. mmap
2. - Find address
   - Program MMU
   - Convert to virtual
3. Write via virtual address directly into video-RAM

Application

Video-RAM

driver
Linux & Software architecture: Framebuffer

1. Application
2. Driver
3. Video-RAM

- Find address
- Program MMU
- Convert to virtual

Write via virtual address directly into video-RAM
Linux & Software architecture: Framebuffer

• How to use the framebuffer architecture, if you cannot mmap its memory?
How to use the framebuffer architecture, if you cannot mmap its memory?

Solution: Use an intermediate buffer

=> Very small application modification required
Linux & Software architecture: Framebuffer

• The VCP implementation provides the following solutions
  – Manual or a timer based synchronisation
  – Using DMA to do synchronisation in the background

• *Making a virtue of necessity:* During synchronisation use hardware features in the LCD and/or PNX0106 to do accelerated video post processing.
  E.g.
  – flipping the screen
  – endianess conversion
  – colour space conversion
  – windowing
  – … *and more …*
Linux & Software architecture: Framebuffer

- The VCP implementation provides the following solutions
  - Manual or a timer based synchronisation
  - Using DMA to do synchronisation in the background

- Making a virtue of necessity: During synchronisation use hardware features in the LCD and/or PNX0106 to do accelerated video post processing.
  E.g.
  - flipping the screen
  - endianess conversion
  - colour space conversion
  - windowing
  - … and more …

Extensions to the architecture, which don’t break compatibility!
Linux & Software architecture: Audio

EPICS DSP

• The PNX0106 has a build in EPICS DSP
  – Firmware loadable via hotplug or udev
  – Statistics and control via sysfs and ioctl
  – /dev/epics/[wma/mp3/…]

write / memcpy

control / mmap

circular (input)

EPICS module

FIFO (output)

read / memcpy
Linux & Software architecture: Audio

**EPICS DSP**

- Example of loading using `udev`

```bash
mkdir /dev/epics
mknod /dev/epics/mp3 c 254 0
mknod /dev/epics/wma c 254 1
insmod e7b.ko &
sleep 1
echo 1 > /sys/class/firmware/epics0/loading
cat /usr/lib/firmware/epics_firmware > /sys/class/firmware/epics0/data
echo 0 > /sys/class/firmware/epics0/loading
```
Outline

• Video Clip Player
  – Introduction
  – System Overview and Use Cases

• Linux & Software architecture
  – Video: Framebuffer Implementation
  – Audio: EPICS DSP
  – UPnP and DTCP-IP protection
  – Linphone: VoIP application

• Linux and prototyping
  – History
  – Measurements: powermanagement
  – Next steps
Linux & Software architecture: UPnP

- Universal Plug and Play (UPnP) is an architecture for peer-to-peer network connectivity of intelligent devices or appliances, particularly within the home.
  - UPnP uses internet standards and technologies, such as TCP/IP, HTTP, and XML
  - Allows devices to automatically connect and work together.

- UPnP facilitates enabling simple and reliable connectivity between stand-alone devices, and easy configuration into a home network.
  - Without any need for user intervention, a new UPnP-enabled device is automatically registered and configured into the network.
  - This is including an announcement of the services and content the device has available.
Linux & Software architecture: UPnP
Linux & Software architecture: UPnP

Browse and play with mobile phone the content of VCP-using Bluetooth or WiFi.
Linux & Software architecture: UPnP

*Browse and play* with mobile phone the content of VCP-using Bluetooth or WiFi.
Browse and play with mobile phone the content of VCP using BlueTooth or WiFi.

Browse the VCP's content using UPnP technology.
Control AV streaming and image printing.
Linux & Software architecture: UPnP

Browse and play with mobile phone the content of VCP using Bluetooth or WiFi.

Audio/Video/Picture streaming from VCP-1

Browse the VCP’s content using UPnP technology.
Control AV streaming and image printing

Ethernet, Bluetooth or WiFi
Linux & Software architecture: UPnP

Browse and play with mobile phone the content of VCP-using BlueTooth or WiFi.

Audio/Video/Picture streaming from VCP-1

Browse the VCP’s content
Using UPnP technology,

Image printing from VCP-1

Control AV streaming and image printing

Ethernet, BlueTooth or WiFi
• The DTCP-IP protection is optional by DLNA

Digital Living Network Alliance (DLNA) aligns industry leaders in the CE, mobile, and PC industries. It provides a shared vision of a wired and wireless interoperable network devices in the home enabling a seamless environment for sharing and growing new digital media and content services.

**Linux & Software architecture: DTCP-IP**

- DTCP AKE daemon
- dtcp.cgi
- Web server (thttpd)
- File system
- DTCP source device (Melody)
- AKE
- CGI
- AKE

- AKE decrypt
- URL download (wget)
- DTCP-aware application
- DTCP sink device
- HTTP

**DTCP-IP**: Digital Transmission Content Protection over Internet Protocol

**CGI**: Common Gateway Interface

**AKE**: Authentication and Key Exchange
Linux & Software architecture: DTCP-IP

VCP load by DTCP-IP encryption

- Total system
- DTCP-IP encryption

Authentication
Encryption
Download
• GUI integration in Qtopia

• At default the VoIP-application waits for calls when started
• ‘Waiting’ is stopped when application exits
Linux & Software architecture: Linphone (VoIP)

- **Application Processor** - Qtopia
- **Linphone application** – command line user interface
- **oSIP** – open SIP library
- **oRTP** – open Real Time Protocol library
- **Voice codecs** – run on application processor (Media Streamer)

**Diagram details:**
- TCP/UDP/IP
- /dev/eth0
- 802.3 MAC
- 802.11 MAC
- /dev/dsp
- EPICS
- Audio AD/DA
- Optional DSP (for voice codecs)
- OSS/ALSA compliant audio interface
- Audio in/out

---

Advanced Systems Laboratory Philips Semiconductors - Armin Gerritsen/Marcin Klecha - 08-04-2006
Linux & Software architecture: Linphone (VoIP)

Implementation details

• Protocols
  – TCP/UDP/IP
  – SIP, RTP protocols provided with the application (oSIP, oRTP)

• Codecs running on the application processor or EPICS DSP
  – A/µ-law PCM
  – GSM and Speex

• Interfaces
  – Ethernet and/or WiFi (/dev/ethX)
  – OSS compliant audio interface
Outline

• Video Clip Player
  – Introduction
  – System Overview and Use Cases

• Linux & Software architecture
  – Video: Framebuffer Implementation
  – Audio: EPICS DSP
  – UPnP and DTCP-IP protection
  – Linphone: VoIP application

• Linux and prototyping
  – History
  – Measurements: powermanagement
  – Next steps
Linux and prototyping: History

Linux enabled a smooth transition from a prototyping system to a very advanced and compact design with extended networking and connectivity capabilities.
Linux and prototyping: History

• Why Prototyping?
  – Proof of concept – verification of new features and architecture validation
  – IP (subsystems) verification in the whole system
  – A prior verification of a reference design (components connectivity)
  – Development of prototyping SW for demo purposes and re-use for reference design evaluation
  – Reduce time and risk for a reference design delivery

• Why use Linux?
  – Prototype platforms often have it provided
  – CE-OS’es often require expensive licences
  – Linux clean separation between drivers, generic kernel and user-space allows easy transfers from prototype to validation to demonstration
  – Therefore *quick* results!
The Clock Generation Unit (CGU) of the PNX0106 allows many different clock settings:

- base clocks
- dividers
- independent spread stages

All of them are controllable during runtime!
Linux and prototyping: Dynamic Power Measurements

A board under test

Actual real-world use-cases could be measured!

e.g. MP3 playback, effects on power usage in various scenario’s
Linux and prototyping: ABISS
Low Power HDD Scheduler

• **Active Block I/O Scheduling System (ABISS)**
  - The algorithm was developed by Philips Research.
  - It is a scheduling framework in kernel space.
  - Targeted at low latency and low power streaming.
  - Allows multiple streams; both reading and writing.
  - Simple and non-intrusive API
  - Obsoletes buffering in streaming applications and adds low power properties.

Result:

~200mW HDD down from ~1.7W for a 2Mbit/s stream
Linux and prototyping: ABISS

VCP architecture

Resource daemon

Application

Power daemon

User space

Kernel space

POSIX API (VFS)

ABISS scheduler

File system

Page cache

ABISS elevator

HDD
Linux and prototyping: Next steps

• Extending the Audio/Video capabilities of the PNX0106 with an external video co-processor
  – e.g. PNX0106 + PNX4103
  – Based on ARM926 and Philips TriMedia™ architecture

• Linux plays again an important role!

• More details see CELF 2007 …
Linux and prototyping: Next steps

_PNX0106 + PNX4103_

**PNX4103**
Emulation Platform

**ARM Versatile Platform**

**LCP1500 (PCI)**

**TV-OUT**
Linux and prototyping: Next steps

PNX0106 + PNX4103

PNX4103 Emulation Platform

ARM Versatile Platform

LCP1500 (PCI)

TV-OUT

Linux

Linux

Emulation Platform

Advanced Systems Laboratory Philips Semiconductors - Armin Gerritsen/Marcin Klecha - 08-04-2006