Libre Software Meeting 2009

“Easy” H.264 video streaming with Freescale's i.MX27 and Linux

Nantes 2009: Rendez-vous Libres!
10es Rencontres Mondiales du Logiciel Libre

July 8th 2009
LSM, Nantes: “Easy” H.264 video streaming with i.MX27 and Linux
Presentation plan

1) i.MX27 & H.264 basics
2) Design by example
3) Demo
Who am I?

- Julien Boibessot
- Embedded systems engineer
- Co-founder of the “Armadeus Project” association (http://www.armadeus.org)
Platform used: APF27

- ARM9 400MHz (i.MX27 Freescale)
- Up to 256 MB RAM DDR 266MHz
- Up to 512 MB Flash NAND 16 bits
- Spartan3A 50-400k gates
- Low power (<10mW sleep mode)
1) Basics (i.MX27 & H.264)
i.MX27

- Freescale i.MX Family
- ARM926EJ-S™ core up to 400MHz
- Low power Video over IP and industrial remote control applications

- [http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=i.MX27&nodeId=0162468rH31143ZrDR66AF](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=i.MX27&nodeId=0162468rH31143ZrDR66AF)
i.MX27 Integrated peripherals

- MPEG-4/H.263/H.264 HW Codec
- High-speed CMOS Sensor Interface (CSI)
- 10/100 Mbps Ethernet MAC
- 18bits LCD controller up to 800x600 (overlay)
- 3 x MMC/SD
i.MX27 Integrated peripherals (2)

- NAND 16bits controller
- DDR-RAM 32 bits controller
- USB OTG high speed, host x 2
- CSPI x 3, SSI/I2S x 2, I2C x 2, UART x 6
Video Hardware Codec

- H.263/H.264/MPEG4
- Up to 720x486 @ 30fps
- DMA to reduce CPU usage
- Encode a format while decoding another
Video Hardware Codec (2)

- Black box
- (16 bits DSP + hardware modules)
H.264 Basics (1)

- H.264 = Advanced Video Coding (AVC) = MPEG4-Part10
- Mix H.263 (IUT-T) – MPEG4 (ISO) => Joint Video Group
- Goals:
  - Same quality than others with bitrate / 2
  - Better quality with same bitrate
H.264 Basics (2)

- Better performances but more computational power (2 to 4x)
- Flexibility with several quality level (low → high bitrates) = profiles: *Baseline*, Main, Extended, High
- Video over IP oriented
- Patented → royalties for commercial use
H.264 Basics (3)

- Network Adaptation Layer Units (NALUs)
  - Parameter sets
  - Slices (encoded image or part of image)
- suitable for transmission over packet networks
H.264 Basics (4)

- Network Abstraction Layer Unit

![Diagram of NALU Type]
H.264 Packets transport: RTP

- RTP = Real Time Protocol
- Try to bring “real time” transmission to IP
- Sequence numbering & timestamp
- Upper layer to UDP (unicast or multicast)
- Unidirectional
H.264 Packets transport: RTP

- RFC-3984: RTP Payload Format for H.264
- Several ways of forming RTP packets with NALUs:
  - Single NALU per packet
  - Single time aggregation packet
  - Multiple time aggregation packet
  - Fragmentation
RFC-3984 example

- Single NALU per packet (RTP header)

```
  Sequence Nb  Source
     80 60 00 01 00 00 01 00 5f 7e 88 67
     ~Version + payload type  Timestamp (90kHz)
```
RFC-3984 example

- Single NALU: Full RTP packet

```
<table>
<thead>
<tr>
<th>Sequence Nb</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 60 00 01</td>
<td>00 00 01 00</td>
</tr>
<tr>
<td>~Version +</td>
<td>5f 7e 88 67</td>
</tr>
<tr>
<td>payload type</td>
<td></td>
</tr>
</tbody>
</table>

```

```plaintext
00 00 00 01 04 5f 7e 88 67 ...
```

```
NALU Type
```
2) Design by example
Custom design

• Portable video recorder/transmitter

• Requirements:
  - VGA capture @ 25 fps
  - Real-time video transmit over WiFi
  - Video preview on LCD
  - Recording of video on SD Card
Video capture (hardware)

- OV9653 (CMOS sensor)
- 8/10 bits data bus + sync
- 640x480 @ 30 fps / 1024x768 @ 15 fps
- Configuration with I2C
- !! Datasheets !!
Video capture (software)

- Conf only

```
Application
/dev/v4l/video0
V4L driver
CMOS sensor
```

User space
Kernel
Hardware
Video preprocessing (hardware)

From CSI or
memory
YUV4:2:2
YUV4:2:0
RGB16/32

i.MX27 Video Pre-Processor (extracted from Freescale's datasheets and simplified)

i.MX27 Video Pre-Processor (extracted from Freescale's datasheets and simplified)

From CSI or memory
YUV4:2:2
YUV4:2:0
RGB16/32

CH-1
RESIZE

Color Space Conv.

Supports:
RGB to YUV
YUV to RGB

Channel 1
(for Display)

Line Buffer
(128x48)

CH-2
RESIZE

Color Space Conv.

Supports:
RGB to YUV

Channel 2
(for Video encoder)

To Memory YUV
4:2:0 YUV 4:2:2
(YUYV) YUV
4:4:4

Line Buffer
(128x48)
LCD preview (hardware)

- TFT 480x272
- i.MX27 internal controller
LCD preview (software)

Application

Graphic library

/dev/fb0 /dev/fb1

Framebuffer driver

LCD controller + video mem

User space

Kernel

Hardware
Video Encoding (hardware)

RAM

01011100011
01011100011
01011....

DMA

from Video pre proc (channel 2)

DMA

H.264 Codec

i.MX27
Video encoding (software)

User space

LibVPU

/dev/vpu

Kernel

VPU driver

Hardware

VPU + video mem
Video storage

- SD/MMC support in Linux since 2.6.2x
- i.MX27 controller supported
- Simple as opening and writing in a file
- Header
Video transmission

- WiFi (SDIO or USB)
- Encryption:
  - WEP: iwconfig alone
  - WPA: need of a WPA supplicant
- Socket UDP, port 6666 (free one)
- RTP as top layer (cf slide 15)
3) Demo
July 8th 2009
LSM, Nantes: “Easy” H.264 video streaming with i.MX27 and Linux
Decoding on PC

• gstreamer

$ gst-launch-0.10 udpsrc port=6666 caps="application/x-rtp, media=(string)video, clock-rate=(int)90000, encoding-name=(string)H264" ! rtph264depay ! ffdec_h264 ! xvimagesink
To Be Done

- Better image synchro with double buffering
- RTCP/RTSP: (VLC compatible)
- Tactile GUI
- Sound
Any questions ?