How to Rapidly Prototype Using the DaVinci™ DVSDK

A Video Security over IP (VSIP) IP-Netcam Example

Loc Truong
Catalog DSP Applications
Agenda

• Scope
• VSIP Backgrounder
• IP-Netcam System Aspects
• Matching with DaVinci™ DVSDK Content
• Summary
• Examples of IP-Netcam using DaVinci™ DVEVM
Scope

• Session illustrates how to use DVSDK as a rapid prototype vehicle for DM644x-based applications, in this case VSIP, to create an IP network camera (IP-netcam) using DVSDK.

• Rapid Prototyping is “the process of quickly putting together a working model (a prototype) in order to test various aspects of a design, illustrate ideas or features and gather early user feedback” - http://en.wikipedia.org
Rapid Prototyping with DVSDK

Scope
VSIP Backgrounder
IP-Netcam System Aspects
Matching with DaVinci™ DVSDK Content
Summary
Examples of IP-Netcam using DVEVM
VSIP Backgrounder

• Rapidly growing security market
• Market currently mixed with analog and digital solutions
• Pros and cons of analog (CCTV) solution:
  - Reliable (low-tech)
  - Infrastructure available
  - Limited scalability in performance to meet today’s demands
  - Not so secured (remember spy movies!)
  - Limited mobility
  - Has limit on distance of remote operators
  - DVR solution not very robust
VSIP Backgrounder

• Pros and Cons of Digital Solution
  ↓ New technology (digital has its challenges)
  ↓ Developing infrastructure
  ✓ More reliable (more redundancy can be built-in)
  ✓ Scalable (both in performance and deployment standpoints)
  ✓ More intelligent (smart cameras) via Video Content Analysis
  ✓ Data and transport can be secured
  ✓ Can be operated very remotely
  $ More cost-effective

→ Digital solution’s advantages far outweigh analog counterpart, creating strong demand for VSIP solutions
Video Content Analysis Examples

- People Counting
- Parking Enforcement
- License Reading
- Perimeter Protection
- Traffic Pattern

Digital technology with VCA enables VSIP to take on large tasks or remove the drudgery out of the surveillance task.
Differentiating Features of IP-Netcam

<table>
<thead>
<tr>
<th>Good Reliable Hardware:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera lens &amp; CCD sensor</td>
</tr>
<tr>
<td>Pan/ Tilt/ Zoom, PoE</td>
</tr>
<tr>
<td>Local storage, Size</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intelligence (VCA):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Detection</td>
</tr>
<tr>
<td>Object Recognition/ Tracking</td>
</tr>
<tr>
<td>Face Recognition</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compression and Networking:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG4 SP, H.264 BP, G.711, etc.</td>
</tr>
<tr>
<td>TCP/IP, HTTP, RTP, RTSP, SMNP, DNS, DHCP, etc.</td>
</tr>
</tbody>
</table>

To meet challenges of IP networks, VSIP needs a **programmable processor** to constantly adapt to changing conditions.
VSIP Backgrounder

• Types of IP-Net cameras:
  – Differences in camera sensor & lens type, level of intelligence, and robustness of features (PTZ, local storage, video quality and resolution, security, etc.) determine the product mix.
  – Intelligence include different types of Video Content Analysis (VCA) techniques such as motion detection (e.g. used in Intrusion Detection), object recognition (e.g. counting applications), object tracking, face recognition (e.g. security screening).
  – An http web server is usually included for remote control and administration and/or video streaming. Depending on the configuration, the camera has names such as web-cam, internet camera, network camera, etc.
  – Video streaming can range from simple MJPEG to MPEG4 or H.264 video compression over various IP protocols such as TCP/IP, FTP, HTTP, RTP and RTSP at different resolutions and number of streams.

• To meet challenges of IP networks, VSIP needs a programmable processor to constantly adapt to changing conditions.
IP-Netcam Examples

D-Link™ DCS-6620 Network Camera:
- User selectable MJPEG or MPEG4 SP up to D1
- PTZ, 2-way audio
- CCD image sensor with auto-focus, auto-iris, AGC

CoVi™ CVQ-21xx MultiStream HD Camera:
- Supports 1280 x 720 (3fps), D1/ CIF/ QCF (30fps)
- MPEG4 SP encode
- ePTZ
- proprietary PoE
- optional 160GB to 320GB HDD
Prototype IP-Netcam Specs

Our prototype needs the following features:

1. Capture D1-res YUV video from analog camera
2. Video compression (encoder selected later)
3. HTTP server
4. TCP/IP stack
5. Local display
6. Non-volatile storage for program and settings
7. Head-room for VCA (feature selected later)
Rapid Prototyping with DVSDK

Scope
VSIP Backgrounder
IP-Netcam System Aspects
Matching with DaVinci™ DVSDK Content
Summary
Examples of IP-Netcam using DVEVM
Typical Smart Surveillance System *

Content Analysis
- Rules Database
- Markers
- Alerts

Capture → Encode → Encryption → TCP/IP → Remote UI / Storage

Water Marking

Local Storage

Local Display (optional)

(*) Courtesy eInfoChips

Minds in Motion

Technology for Innovators™
Typical IP-Netcam System Block Diagram

GPP SoC
- CPU + I$ + D$ + MMU
  - Multitasking OS
  - Control threads
  - Network stacks

DMA Controller:
- Video in to mem
- Audio in
- Video out
- Audio out
- DSP data moves
- Other I/O

DSP:
- Video encoding
- Image encoding
- Audio encoding
- Pre-processing
- Post-processing
- VCA/Encryption

Network

Remote UI / Storage

Local Display (optional)

Video Encoder

Video In Controller

Serial Port

Video Out Controller

Audio Decoder

Flash

High-speed Local memory

DSP Local Memory

Phy

EMAC

PTZ

GPIO

Processor Interface

Memory Controller

ATA

INTC

Mem C

Processor Interface

Technology for Innovators™
How About Non-SoC Solutions?

<table>
<thead>
<tr>
<th>GPP+ Discrete Accelerators</th>
<th>Discrete GPP+DSP</th>
<th>DSP only</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Single prog model</td>
<td>✓ Flexible feature set</td>
<td>✓ Simpler design</td>
</tr>
<tr>
<td>✓ Need high MIPs</td>
<td>✓ Easy upgrade</td>
<td>✓ Single prog model</td>
</tr>
<tr>
<td>↓ Fixed feature set</td>
<td>↓ High total MIPs</td>
<td>✓ Flexible feature set</td>
</tr>
<tr>
<td>↓ Long ASIC design cycle</td>
<td>↓ Complex design</td>
<td>↓ Need high MIPs</td>
</tr>
<tr>
<td>↓ High FPGA cost</td>
<td>↓ Dual prog model</td>
<td></td>
</tr>
</tbody>
</table>

There are alternative approaches to high MIPs requirements for VSIP Net-cams

DM644x processors combine the best from these solutions in a small package using GPP + DSP + Accelerators
DM6446-based IP-Netcam System BD

DM6446 SoC

- VPSS: VENC, OSD, CCDC, Resizer, H3A, Preview
- AMR926EJS: Multitasking OS, Control threads, Network stacks
- C64x+ DSP: Video encoding, Image encoding, Audio encoding, Pre-processing, Post-processing, VCA/Encryption

Crossbars (SCRs)

EDMA:
- Audio in
- DSP

Built-in DMA masters:
- ARM/DSP, VPBE/VPFE/EMAC/ATA

- Flash
- High-speed Shared Memory

Remote UI / Storage

PTZ

Local Display (optional)

Video Decoder

Audio Decoder

ASP

GPIO

INTC

Memory Controller

ATA

EMAC

PHY

Network

Minds in Motion

Technology for Innovators

Texas Instruments
Is DM644x a fit for IP-NetCam?

Key Applicable Features:
• 300MHz ARM926EJS for control threads and networking stacks
• 600MHz c64x+ DSP core for compression and analytics
• H/W VICP accelerator
• 64-Channel EDMA
• Video Processing Sub-system with built-in NTSC/PAL encoder and On-Screen Display (OSD)
• Built-in I/Os
• High-speed internal busses
• DDR2 controller
• NOR/ NAND memory I/F
• Small BGA 16x16mm package
Rapid Prototyping with DVSDK

Scope
VSIP Backgrounder
IP-Netcam System Aspects
Matching with DaVinci™ DVSDK Content
Summary
Examples of IP-Netcam using DVEVM
DM644x DVEVM & DVSDK

**DVEVM Kit:**
- Evaluation Module
- Camera, LCD, mic, remote
- TI-demo MV Linux Support Package
- VISA Demos
- SoC Visualizer
- DV Test Bench (DVTB)
- XDC Configurations Tool Kit

**DVSDK Packages:**
- Base:
  - TI Linux-host c6x code generation tools
  - DSP-side DSP/BIOS content

  "-L" package:
  - MVL subscription (one-year access to support and updates at [www.mvista.com](http://www.mvista.com))
  - DevRocket 2.0

  "-3L" package:
  - DVSDK–L package + Spectrum Digital XDS-560 USB Emulator
Installed DVSDK at a-glance

- Demo source
- DV Test Bench
- DVEVM/SDK links to docs
- DVEVM/SDK GSG
- Copy of MVL ARM kernel & file system
DVSDK Software Framework

Application Layer (APL)
- Demo Examples w/ source:
  - A/V Decode (MPEG2/4, H.264, G.711)
  - A/V Encode (MPEG4, H.264, G.711)
  - Video Encode/Decode (MPEG4, H.264)
- Linux Packages:
  - Over 300 supplied with MVL
  - E.g. HTTP

Signal Processing Layer (SPL)
- CE Servers - combinations of:
  - MPEG2 MP DEC
  - H.264 BP DEC
  - MP3 DEC
  - H.264 BP ENC
  - MPEG4 SP DEC
  - G.711 DEC
  - MPEG4 SP DEC
  - G.711 ENC

User Space
- Linux API w/EPSI
- Available Drivers:
  - V4L2
  - FBdev
  - OSS
  - DSPLink
  - ATA
  - NOR
  - NAND
  - MMC/SD
  - I2C
  - UART
  - EMAC
  - CMEM
  - Linux Kernel 2.6

Kernel Space
- DSP/BIOS Kernel 5.xx
- Timer Driver
- DSPLink Driver
- Continuous Memory I/O Buffers (Shared Memory)

Linux API w/EPSI
- CODEC engine
- VISA API
- CODEC engine
## Software Needed in Prototype (blue)

### Application Layer (APL)
- **Demo Examples w/ source:**
  - A/V Decode (MPEG2/4, H.264, G.711)
  - A/V Encode (MPEG4, H.264, G.711)
  - Video Encode/Decode (MPEG4, H.264)

### Linux Packages:
- Over 300 supplied with MVL
- E.g. HTTP Web server

### Kernel Space
- Linux API w/EPSI

#### Available Drivers:
- V4L2
- FBdev
- OSS
- DSPLink
- ATA
- NOR
- NAND
- MMC/SD
- I2C
- UART
- EMAC
- CMEM
- Linux Kernel 2.6

### Signal Processing Layer (SPL)

#### CE Servers - combinations of:

<table>
<thead>
<tr>
<th>Driver</th>
<th>xDM</th>
<th>xDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG2 MP DEC</td>
<td>xDM</td>
<td>MP3 DEC</td>
</tr>
<tr>
<td>MPEG4 SP DEC</td>
<td>xDM</td>
<td>MPEG4 SP ENC</td>
</tr>
<tr>
<td>H.264 BP DEC</td>
<td>xDM</td>
<td>H.264 BP ENC</td>
</tr>
<tr>
<td>G.711 DEC</td>
<td>xDM</td>
<td>G.711 ENC</td>
</tr>
</tbody>
</table>

- DSKT2
- ACPIY3
- DMAN3

#### Available Drivers:
- DSPLink Driver
- Timer Driver
- DSP/BIOS Kernel 5.xx

### Continuous Memory I/O Buffers
(Shared Memory)

### A lot of SW available!

---

**Minds in Motion**

**Technology for Innovators**

---

**TI Developer Conference**
Calling Sequence – Capture

Application Layer (APL)
- IP-Netcam non-event loop:
  - suspend and wait for data
  - dequeue buffer
  - VIDENC_process
  - enqueue buffer
  - transmit buffer

Linux API w/EPSI
- /dev/video0

User Space
- Linux Kernel 2.6
- dsplink.ko
- EMAC
- cmem.ko
- davinci_vpfe.ko

Kernel Space
- Linux Kernel 2.6
- dxm.ko
- dskt2.ko
- acpy3.ko
- dman3.ko
- cmem.ko

Signal Processing Layer (SPL)

CE Server (BIOS task):
- MPEG4 encoding instance
- xDM algo
  - algActivate
  - process
- DSKT2
- ACPY3
- DMAN3
- DSPLink Driver
- Timer Driver

CODEC engine

VISA API

CODEC engine

Continuous Memory I/O Buffers
(Shared Memory)
Calling Sequence – Encode

**Application Layer (APL)**
- IP-Netcam non-event loop:
  - suspend and wait for data
  - dequeue buffer
  - VIDENC_process
  - enqueue buffer
  - transmit buffer

**Signal Processing Layer (SPL)**
- CE Server (BIOS task):
  - MPEG4 encoding instance
- xDM algo
  - algActivate
  - process
- xDM
- MPEG4 SP ENC

**User Space**
- Linux API w/EPSI
  - davinci_vpfe.ko
  - dsplink.ko
  - EMAC
  - video_buf.ko
  - cmem.ko
- Linux Kernel 2.6

**Kernel Space**
- DSP/ BIOS Kernel 5.xx
- DSPLink Driver
- Timer Driver
- DSP/ BIOS Kernel 5.xx
- Continuous Memory I/O Buffers
  - (Shared Memory)

**CODEC engine**
- VISA API
- CODEC engine
- CODEC engine

**Minds in Motion**

**Technology for Innovators**
Calling Sequence – Transmit

**Application Layer (APL)**
- IP-Netcam non-event loop:
  - suspend and wait for data
  - dequeue buffer
  - VIDENC_process
  - enqueue buffer
  - transmit buffer

**Signal Processing Layer (SPL)**
- CE Server (BIOS task):
  - MPEG4 encoding instance
- xDM algo
  - algActivate
  - process
- xDM
  - MPEG4 SP ENC

**User Space**
- Linux API w/EPSI
  - davinci_vpfe.ko
  - dsplink.ko
  - EMAC
  - video_buf.ko
  - cmem.ko
  - Linux Kernel 2.6

**Kernel Space**
- Windows API
  - dskt2
  - acpy3
  - dman3
  - dsplink driver
  - Timer Driver
  - DSP/ BIOS Kernel 5.xx

**Continuous Memory I/O Buffers**
(Shared Memory)
Putting It All Together

• The DVSDK encode demo source code is explained in detail in SPRAA96.

• Once you are satisfied with the application, it can be inserted in the file system. The boot-loader, Linux kernel and file system can be flashed to the built-in NOR or NAND flash on the DVEVM.

• How to do this for a minimum system with just an HTTP server is described in SPRAAH2 application note.

• These app notes can be found at the DM6446 folder at http://focus.ti.com/docs/prod/folders/print/tms320dm6446.html or by searching www.ti.com using the codes.
IP-Netcam Rapid Prototyping
DVSDK Usage Tips

- Save development tool set-up time using *Getting Started Guides*
- Quick access to documentation is via *dvevmindex.html* and *dvsdkindex.html* under `~/dvevm_1_10`
- Check out encode demo source to see how to use V4L2 driver to capture video under `~/dvevm_1_10/demos`
- Copy kernel source to your workspace first before make xconfig and rebuild kernel
- Use a copy of the MVL arm file system for modifying or adding content for your target file system
- Install tools under `/opt` (default)
- Register your DVEVM so you can access [www.ti.com/dvevmupdates](http://www.ti.com/dvevmupdates)
DaVinci Digital Video Evaluation Module (DVEVM)

Documentation Index

- **DVEVM Getting Started Guide** - Hardware and software overview, including how to run demos, install software, and build the demos.
- Codec Combos Data Sheets: **Encode, Decode, and Loopback (Encode/Decode)**
- **msp430lib API** Library used by the demos for communicating with the MSP430 on the DVEVM.
- **simplewudget API** Library used by the demos for drawing buttons, png and jpeg images and rendering fonts on an RGB16 surface. Depends on various open source libraries.
- Other Demo Utilities Information **fifoutil, rendezvous, and timerutil**
- **Decode Demo Information**
- **Encode Demo Information**
- **Encoded/decode Demo Information**
- **XDAIS documentation**
- **u-boot upgrade instructions**
DevRocket can simplify Platform Building process

Main Features:
- Eclipse-based IDE
- Version control (CVS)
- Uses gdb, kgdb
- Supports kernel building
- Assess system timing, usage, latency, etc.
- Target control tools

Great way to manage >300 packages in MVLinux distro
E.g. add Apache HTTP server to your target file system by selecting it from the list of packages
Rapid Prototyping with DVSDK

Scope
VSIP Backgrounder
IP-Netcam System Aspects
Matching with DaVinci™ DVSDK Content

Summary
Examples of IP-Netcam using DVEVM
Summary

• DVEVM kit contains all necessary hardware to create a IP-Netcam prototype
• Available Linux drivers, open-source Linux packages and encode Codec Engine server from DVSDK can be used to create basic IP-Netcam software framework or base platform
• This H/W & S/W prototype can help jump start the development process to leave more time to add further enhancements, optimization and testing to create a real product. That’s the *DaVinci effect*!
The Next Steps:

• Take advantage of DaVinci™ DM644x DVEVM and DVSDK rich set of features to rapid prototype your product concepts. DVEVM information is at http://focus.ti.com/docs/toolsw/folders/print/tmdxevm6446.html

• Jump start your team with TTO DaVinci™ workshop more info is at http://focus.ti.com/docs/training/traininghomepage.jhtml

• Contact your local TI office for a free evaluation of TI DaVinci™ VISA codecs or find out more at http://www.go-dsp.com/forms/TIDigitalMediaSWCM/index.htm

• In addition, DaVinci™ network of Authorized Software Providers (ASPs) and TI DSP third parties can provide customized assistance to shorten your design cycle time.
List of Currently Available DM644x Application Notes:

@ http://focus.ti.com/docs/prod/folders/print/tms320dm6446.html
Customized Support from New Authorized Software Providers

FREE 60 Day Evaluation

Credentials
- Software expertise
- Engineering services
- Application expertise
- Proven customer satisfaction

4 Hours FREE Support from Authorized Software Provider
- 4 hours during evaluation
- Up to 40 hours of support per production license

Technology for Innovators™
Rapid Prototyping with DVSDK

Scope
VSIP Backgrounder
IP-Netcam System Aspects
Matching with DaVinci™ DVSDK Content
Summary

Examples of IP-Netcam using DVEVM
“Ready-to-Go” Solution

- Web Viewer Application with GUI
- H.264, MPEG4, JPEG, G.7xx
- Object Video Content Analytics
- Codec Engine DSP/BIOS™ Link DSP/BIOS™
- MontaVista Linux
- Firmware & Drivers
- Production Ready Hardware

- Rapid market entry
- Optimized BOM
- eInfochips Support Team

TI | Developer Conference

Minds in Motion

Technology for Innovators™
Hands-on Demos

• TCP/IP Client/Server IP Netcam Demo
  – The DaVinci™ DVEVM is used in a configuration where it captures analog video, compress in MPEG4 format and stream via TCP/IP to another DVEVM which decompresses the video for display (CCTV model)

• MJPEG Demo
  – The DaVinci™ DVEVM runs an Apache HTTP server to stream JPEG images. User can view captured video live via any Web Browser
Digital CCTV via TCP/ IP IP-Netcam Demo

Capture & Send App
- Capture
- MPEG4 Encode
- TCP/IP Xmit

Receive & Display App
- TCP/IP Rcv
- MPEG4 Decode
- Display

DM6446 DVEVM
- NTSC cable to video in
- 10/100 Router
- Ethernet cables to/ from router
- NTSC cable to video out

Technology for Innovators™
MJPEG Demo

Note: this demo is available as app note SPRAAH9
How to Rapidly Prototype Using the DaVinci™ DVSDK

Q&A