NoTA DIP implementation in resource limited devices and subsystems

30.9.2009  NoTA Conference 2009
Notava Inc. / Notava Japan Inc.
Takaharu Fujii

contact_jp@notava.com
http://www.notava.com

Copyright(c) Notava Inc./Notava Japan Inc. 2009/9/30
Agenda

» DIP in NoTA Architecture
» DIP Implementation
  » Scalability
  » Options
  » Steps
  » Current achievements
» Summary
NoTA Logical Architecture
DIP in NoTA Impl. Architecture

- Imaging Subsystem
  - Imaging
  - ISP, Capturing
  - RTOS
  - 32bit CPU
  - DD
  - 32bit CPU w MMU

- 3D Graphics Subsystem
  - 3D GFX
  - 3D GFX Accelerator
  - RTOS
  - 32bit CPU w MMU
  - DD

- Mass Storage Subsystem
  - DSC
  - Game
  - Music
  - Generic OS
  - 32bit CPU w MMU

- Intelligent Energy Subsystem
  - Cellular
  - RTOS
  - 32bit CPU
  - DD
  - Cellular Subsystem
  - Audio
  - RTOS
  - 32bit CPU
  - DD

- Audio Subsystem
  - Audio
  - RTOS
  - 32bit CPU
  - DD

- Energy Source
  - Charging / protection
  - Runtime
  - DD
  - 8bit MCU

- Mass storage (e.g. memory card, HDD)

DD: Device Deriver

Copyright(c) Notava Inc./Notava Japan Inc. 2009/9/30
Scalability of DIP Implementation

To adapt to any kinds of subsystems, DIP implementation must be scalable from the rich resource subsystems to resource limited subsystems, as well as versatile platforms.

Cost wise, DIP implementation in resource limited subsystem is very important factor.

---

**HW IP approach**
- L_INdown in HW
- L_INup/down in HW
- Whole DIP in HW

**8bit MCU, 10MHz**
- No MMU
- OS for embedded system
- Embedded 16kB RAM, 512kB Flash

**32bit CPU, 1GHz**
- MMU support
- Rich OS
- 512MB RAM + mass storage

---

Copyright(c) Notava Inc./Notava Japan Inc. 2009/9/30
DIP Implementation Options

HW IP approach

- Subsystem specific HW
  - H_IN
  - L_INup
  - L_INdown
  - OS

- HW IP
  - H_IN
  - L_INup
  - L_INdown
  - OS

- CPU
  - Mem
  - HW for Subsystem func.

- Memory
  - Accel. HW
  - OS

8bit MCU, 10MHz

- Separated CPU/MCU
  - Target
    - Code size: < 5~10%
    - Power: ~ 0%
    - CPU load: < 5%

32bit CPU, 1GHz

- Resource Sharing
  - (e.g. Audio, 3D GFX)
  - Memory

Copyright(c) Notava Inc./Notava Japan Inc. 2009/9/30
DIP Implementation Steps
towards limited resource subsystems

1. Implementation options
   » Subsystem’s main resource, separated CPU or HW impl.
   » POSIX OS like Linux or others

2. DIP porting to the specific OS platform
   » h_in/platform
   » l_in/platform

3. L_INdown adaptation for the specific interconnect
   » TCP/IP, Bluetooth, USB, FIFO are available in reference code
   » For other cases, referring ld_fifo / ld_sample can be good start
   » l_in/ld_common for OS depended code

4. Optimization for code reduction
   » Optimize buffer size
   » Limit the number of sockets
   » Remove L_Manager functionality
   » OS less approach, merge layers and so on

X. HW IP approach
Current Achievements of DIP Scalability

DIP: ~ 70kB (w some code reduction efforts)

DIP: < 100kB (depending on L_INdown adapt.)

Whole DIP in HW IP (programmable logic for some specific use case)

HW IP approach, Separated CPU/MCU, Resource Sharing

Limited Computing resource rich

No OS

T-Kernel, uTRON, uCLinux, eCos

Linux, Maemo, Android, Symbian

32bit CPU ~ 8bit MCU without MMU

32bit CPU with MMU support

A sample use case: Code size: < 20%, Power: ~ 0%, CPU load: < 20%
» DIP implementation scalability is allowed by NoTA specification.

» DIP implementations in the limited resources are available for low cost solutions.
   » Small CPU with embedded OS
   » 8bit MCU without OS

» Such DIP impl. are also applicable for small devices (e.g. sensor nodes) to extend Smart Environments.
Thank You!