A demonstration of "Visual Terminal"

May 26, 2006

Fujitsu Limited,
Fujitsu Laboratories Ltd,
Fujitsu Software Technologies Limited

Jyunji Kondo (Fujitsu Software Technologies Limited)
Agenda

• Introduction of “.u Visual” with demonstrations
  – Triple play + 1
    • Visual Communication
    • Visual AV Player
    • Visual Internet
    • 1-segment/3-segment Broadcasting Receiver

• Key Technologies
  – WideStudio/MWT
  – Reducing Startup time
.u Visual – Ubiquitous Visual Terminal

- **Visual Communication**
  - VoIP with Video / Push to talk with Video
    MPEG-4 VGA 15fps G.729a/G.711

- **Visual AV Player**
  - Audio Video Player
    MPEG-4 VGA 15fps AAC/MP3/WMA

- **Visual Internet**
  - Full Web Browser “Inspirium” by Fujitsu

+ 1-segment/3-segment Broadcasting Receiver
Hardware Configuration

- CPU: FR461 made by Fujitsu (400MHz)
- 3.7 inch VGA (640x480) TFT LCD
- CMOS sensor, 350 thousand pixels
- CF / SDIO extended slot
- Stereo Speaker, Microphone, Receiver, Vibrator
- Touch Panel
- PTT button, etc.
- Wireless LAN module (IEEE802.11b)
- And much more…
  - Extended module I/F: 1-segment / 3-segment Broadcasting Tuner
  - 3-axis acceleration sensor
## Hardware Specifications

<table>
<thead>
<tr>
<th>Items</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size / Weight</td>
<td>165 x 73 x 26[mm] / 245[gram]</td>
</tr>
<tr>
<td>CPU</td>
<td>FR461 made by Fujitsu (400MHz, 8 parallel instructions per cycle)</td>
</tr>
</tbody>
</table>
| Memory              | • SDRAM:128MB  
                     • NOR Flash ROM:64MB |
| Expression          | • 3.7inch VGA(640x480) TFT Color LCD  
                     • Touch Panel |
| Pickup device       | CMOS sensor, 350 thousand pixels |
| Wireless LAN        | Wireless LAN module (IEEE802.11b compliant) embedded |
| Extended Module     | 1-segment / 3-segment Broadcasting Tuner |
| Sensor              | 3-axis acceleration sensor |
| External I/F        | CF, SDIO, USB (when cradle is used) |
# Software Specifications

<table>
<thead>
<tr>
<th>Items</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Embedded Linux 2.6 for FR-V</td>
</tr>
</tbody>
</table>
| GUI environment  | • X-Window System  
|                  | • WideStudio/MWT                                                            |
| Middlewares      | • MPEG-4 codec  
|                  | • AAC codec  
|                  | • AVC/H.264 dec                                                             |
| Applications     | • Wireless IP video phone  
|                  | • Wireless IP video transceiver  
|                  | • AV Player  
|                  | • Full Browser (Inspirium)  
|                  | • 1-segment / 3-segment Broadcasting Receiver                              |
Visual Communication

• Video codecs
  - MPEG-4 QVGA 15FPS full-duplex
  - QVGA Portrait

• Audio codecs
  - G.729a/G.711 384kbps

• Communication Network
  - Wireless LAN(IEEE802.11b)

Only using software codecs!
Visual AV Player

- Video codecs
  - MPEG-4 SP VGA 15FPS
- Audio codecs
  - MPEG-4/2 AAC-LC

Only using software codecs!
Visual Internet

• Support Languages
  – HTML4.01 XHTML1.0, XHTML Basic
• Style Sheet
  – CSS1/2
• JavaScript1.5
• Protocols
  – HTTP1.0/1.1
• Software Keyboard
1,3-segment Broadcasting Receiver

• Video codecs
  – AVC/H.264
• Audio codecs
  – MPEG-4 AAC LC
• BML supported

Only using software codecs!
Key Technologies

• WideStudio/MWT
• Reducing Startup time
WideStudio/MWT(1)

• MWT stands for “Multi-platform Widget Toolkit”.
• Light weight GUI toolkit for embedded devices.
• High level portability using C/C++ languages.
• Main toolkit of NAB Subproject of Eclipse-DSDP
  – DSDP: Device Software Development Platform
  – NAB: Native Application Builder
WideStudio/MWT(2)

The web page of WideStudio project
WideStudio/MWT(3)

The web page of Eclipse DSDP-NAB

Mission Statement

Native Application Builder (NAB)プロジェクトはC++や他のいくつかの言語でGUIアプリケーションを作成するためのフレームワークを開発する事を目的に、ユーザが選んだGUIアプリケーションを、コードを書きながらすぐに動作するプログラムで動作させる事を目標にします。

NABのツールは現在では、SWTやJFaceと組み合わせて、C++で書かれたGUIアプリケーションを作るために使われているのが大きく変わります。

詳しくはこちら⇒
Reducing Startup time

- A case study of reducing startup time when migrating from 2.4 kernel to 2.6.

  Step1: Data Collection
  Step2: Analysis
  Step3: Refinement
  Step4: Evaluation
Step 1: Data Collection

• Utilizing Bootchart
  – See the CELF Wiki page.

http://tree.celinuxforum.org/pubwiki/moin.cgi/BootChart
Step 2: Analysis

0 sec.:
- Power On

3 sec.:
- RedBoot

4 sec.:
- Kernel boot
  - Print the boot logo
  - Print boot messages
  - Initialize kernel
  - Initialize Drivers
  - Initialize Network

15 sec.:
- Mount RootFS
  - Init

25 sec.:
- Start daemons

37 sec.:
- Start X

50 sec.:
- Inspirium launcher
Step 3: Refinement

- **Power On**
  - Remove network initialization and waiting timeout.

- **RedBoot**
  - Remove

- **Kernel boot**
  - Print the boot logo
  - Print boot messages
  - Leave it until later

- **Mount RootFS Init**
  - Initialize kernel
  - Initialize Drivers
  - Modularize

- **Start daemons**
  - Minimize and separate RootFS

- **Start X**
  - Leave them until later
  - Leave it until later

- **Inspirium launcher**
  - Switch to MWT launcher
Step 4: Evaluation (1)

- Verifying the chart.
Step 4: Evaluation (2)

0 sec.
- Power On
- RedBoot
- Kernel boot
  - Print the boot logo
- WideStudio/MWT launcher
- Start X
  - User input event
  - WideStudio/MWT applications

1.5 sec.
3.5 sec.
6 sec.
19 sec.
Reducing Startup time - Summary

• Leave things until later as much as possible.
  – Installing kernel modules.
  – Starting middlewares.

• Separate RootFS into a minimal fundamental one and the other.
  – Only mount the fundamental one in the early stage.
  – Leave mounting the other until later.

• Optimize the init procedures.
  – Operations not needed for an application launcher should be left until later.

• Use light weight GUI environment.
Topics of Bootchart

• embootchart
  – Developed by Matthew Klahn, Motorola.
  – The presentation has been made in last ELC.
    http://tree.celinuxforum.org/CelfPubWiki/ELC2006Presentations?
      action=AttachFile&do=get&target=VisualizingResUsageDuringBoot.pdf
  – Reduce overheads of Bootchart and improve accuracy of measurements.
Fujitsu

The possibilities are infinite