Design Considerations for Low Power Embedded Systems

Andrew Patterson
Business Development Director

Comprehensive Solutions for
Android™ • Nucleus® • Linux®
Mobile & Beyond • 2D/3D User Interfaces • Multi-OS • Networking
## Market Change

<table>
<thead>
<tr>
<th>World Population</th>
<th>6.3 Billion</th>
<th>6.8 Billion</th>
<th>7.2 Billion</th>
<th>7.6 Billion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected Devices</td>
<td>500 Million</td>
<td>12.5 Billion</td>
<td>25 Billion</td>
<td>50 Billion</td>
</tr>
</tbody>
</table>

Nucleus in >2.3 Billion devices

<table>
<thead>
<tr>
<th>Connected Devices Per Person</th>
<th>2003</th>
<th>2010</th>
<th>2015</th>
<th>2020</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.08</td>
<td>1.84</td>
<td>3.47</td>
<td>6.58</td>
</tr>
</tbody>
</table>

More connected devices than people

Connected data source: Cisco IBSG, April 2011

Nucleus data source: Vision Mobile, 100 million club, Oct 2010
Challenges

- Super Low Cost + High Production Volume
- Limited Size + Memory Constraints
- Low Power
- Wireless Security
- IPv6
- ZigBee
- WiFi
- Bluetooth
Nucleus 3rd Generation

New Configuration Technology
include only what you really need

Connectivity
Wi-Fi, Bluetooth, Zigbee, USB3, ...

Power Management
built in as kernel services

Footprint Optimizations
across all of OS and middleware

Kernel Improvements
Average: 27.5% faster  Min ROM: 16% smaller  Min RAM: 18% smaller

Hardware and Peripheral Support
highlights HW specifics e.g. TI DSPLink, double framebuffer, ...
Power Management

Battery Life

– End users prize longer battery life of devices
– Allows manufacturers to ship with smaller batteries both reducing cost and physical device size & weight

Simplified Mechanical Design

– Less power dissipation means less heat to remove from the device, elements such as cooling fans may be eliminated
– Potential cost and size savings

Easier Standards Compliance

– Meet “Energy Star” type power consumption requirements
– Meet FCC radiation/RF noise requirements
The need to Optimize Battery Power

SmartPhone Battery Life

St. Jude Medical (NYSE:STJ) advised doctors and patients last week that it's recalling some of its Eon and Eon Mini pain management implants after hundreds of the neurostimulation devices had to be taken out when their batteries failed too soon or because they overheated while recharging (Aug 2012)
Key Principle of Power Conservation

Turn Off everything you are not using…

Power use categories
- Static, devices that must be always on
- Dynamic, enabled when needed
What is Nucleus?

Integrated Tools Solution

Sourcery TOOLS

IDE
Compiler
Debugger
Probe
Analyzer

NUCLEUS

Application

Graphics UI Engine
Connectivity SW Stacks
Storage, File, Net
Nucleus Operating System
Device Drivers

Proven, Small Footprint,
Power Management,
Integrated MW Stack
Example: power optimized networking application

System Lines of Code (SLOC)

- Task scheduling
- Hardware abstraction (driver interface)
- Empty idle task
- BSP / drivers

- Power aware BSP/drivers
- Optimized idle task
- Automatic tick suppression
- DVFS aware OS and BSP/drivers
- System state and DFVS operating point management abstraction

Bare Metal

RTOS

Power Aware RTOS
Ready to Start developing in under 15 minutes…

- Self-Contained, Completely Integrated Development Platform
- Unified Installer – up and debugging in under 15 minutes
  - Subsequent development cycles (build -> download -> debug) even faster!
Power Management Principles

CPU needs control over as many peripherals as possible
Power Constraint Pyramid
Hardware considerations for low power design
Choosing Hardware

- Functionality and Performance
- Power Consumption Ratings
- Peripheral power and clock gating
- CPU Idle and other low power modes
- Dynamic Voltage and Frequency Scaling (DVFS)
- Low-power modes for memory sub-systems
  - SDRAM self-refresh
- Hibernate blocks
  - Minimum RAM
  - Oscillator, RTC, Wake-up/Alarm sustain system while in hibernate
Hardware Design Considerations

- Use power efficient components such as DC-DC converters
- Enable software controlled power & clock gating of connected peripherals
  - Driver controlled
- Ensure connected peripherals do not limit the design due to their minimum voltage and/or frequency requirements
- Price
### Freescale i.MX Power Management Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>i.MX23</th>
<th>i.MX25</th>
<th>i.MX28</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal Power Supply</strong></td>
<td>DC-DC switched converter Four linear regulators 3.3 V, 1.2 V, 1.8 V, 2.5 V</td>
<td>DC-DC switched converter Four linear regulators 3.3 V, 1.2 V, 1.8 V, 2.5 V</td>
<td></td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>Adaptive Voltage Control (AVC) CLK_H auto-slow Clock gating Silicon speed sensor Multiple peripheral clocks (domains)</td>
<td>Dynamic Voltage and Frequency Scaling (DVFS) Clock gating Active Well Bias (AWB)</td>
<td>Dynamic Voltage and Frequency Scaling (DVFS) Clock gating Active Well Bias (AWB)</td>
</tr>
<tr>
<td><strong>Low Power Mode</strong></td>
<td>Standby Deep-sleep</td>
<td>Wait Doze Stop Sleep</td>
<td>Standby Deep-sleep</td>
</tr>
<tr>
<td><strong>SDRAM</strong></td>
<td>2.5 V DDR1 1.8 V Mobile DDR (mDDR)</td>
<td>3.3 V SDRAM 1.8 V DDR2 1.8 V mDDR</td>
<td>1.8 V DDR2 1.8 V mDDR 1.5 V Low Power DDR2 (LP-DDR2)</td>
</tr>
</tbody>
</table>
### Freescale Kinetis Power Management Features

#### Power Modes

<table>
<thead>
<tr>
<th>Chip Mode</th>
<th>Chip Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Run</td>
<td>LLS (Low Leakage Stop)</td>
</tr>
<tr>
<td>Normal Wait- via WFI</td>
<td>VLLS3 (Very Low Leakage Stop3)</td>
</tr>
<tr>
<td>Normal Stop- via WFI</td>
<td>VLLS2 (Very Low Leakage Stop2)</td>
</tr>
<tr>
<td>VLPR (Very Low Power Run)</td>
<td>VLLS1 (Very Low Leakage Stop1)</td>
</tr>
<tr>
<td>VLPR (Very Low Power Wait) – via WFI</td>
<td>BAT (backup battery only)</td>
</tr>
</tbody>
</table>
Use Cases
On Use Cases

- System’s usage model
  - Determines the maximum possible power savings achievable using OS/BSP/Application
- Peripherals used in one “use case” may be turned off in another
- Idle/Standby
Hypothetical Medical Monitoring Device

- Device takes a complete measurement
- Device uploads a set of measured data
- User checks vital life measurements using a built in display
- Device is idle awaiting the next measurement
# Planning Use Cases

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Average Current (mA)</th>
<th>Duration (sec)</th>
<th>Frequency per day</th>
<th>Total time (sec/day)</th>
<th>Energy Used (mAh/day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vitals Measurement</td>
<td>158</td>
<td>1</td>
<td>288</td>
<td>288</td>
<td>13</td>
</tr>
<tr>
<td>Data Upload</td>
<td>250</td>
<td>3</td>
<td>288</td>
<td>864</td>
<td>60</td>
</tr>
<tr>
<td>User Vitals Check</td>
<td>320</td>
<td>30</td>
<td>15</td>
<td>450</td>
<td>40</td>
</tr>
<tr>
<td>Ideal</td>
<td>1</td>
<td></td>
<td></td>
<td>84798</td>
<td>24</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>136</td>
</tr>
</tbody>
</table>
Comprehensive Solutions for

Android™ - Nucleus® - Linux®

Mobile & Beyond • 2D/3D User Interfaces • Multi-OS • Networking

Android is a trademark of Google Inc. Use of this trademark is subject to Google Permissions. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
OS and BSP/Driver considerations for power management
OS Choice Considerations

- Ability to take advantage of hardware’s low power features
- Ability to abstract low power features from the application
- Common power framework for driver development and ability to set independent power requirements for each driver
Driver States

Up to 256 Driver State power modes may be defined.

<table>
<thead>
<tr>
<th>System State</th>
<th>Audio Amplifier</th>
<th>Display</th>
<th>Backlight</th>
<th>SD/MMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>STANDBY</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>PLAYBACK</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>USER_IDLE</td>
<td>ON</td>
<td>ON</td>
<td>DIM</td>
<td>ON</td>
</tr>
<tr>
<td>USER_ACTIVE</td>
<td>ON</td>
<td>ON</td>
<td>BRIGHT</td>
<td>ON</td>
</tr>
</tbody>
</table>

Figure 1: Example System Power States definition.
BSP (drivers)

- Set clear power requirements for each device driver
  - Set power budget per driver state (ON, OFF, Sleep, etc)
  - Specify minimum Operating Points at which the driver has to operate
  - Require full integration with an OS power framework (participating in DVFS, etc)
- Each driver should aggressively manage power beyond the system imposed limitations
- Start verifying power for BSP from the beginning
- Treat power requirements the same as functional requirements
Nucleus Power Management Services

- Power-aware kernel
- Idle CPU power management and tick suppression
- Operating point control
- Dynamic Frequency and Voltage Scaling
- System state control
- Fine grain control of peripheral states
- Hibernate and Standby modes
- Low-power sleep modes like hibernate and standby
Looking at our pyramid from above

- = Hardware power management facilities
- = Nucleus Power Management Services
- = Application Software
Nucleus Power Management with Hibernate

Nucleus Power Management APIs

- Idle Scheduler
- DVFS Service
- Peripheral State Service
- System State Service

Device Manager

- Hibernate & Standby OPS

CPU Idle & CPU Wakeup Functions
- CPU State Driver
- Peripheral Driver with Hibernate
- Peripheral Driver with Hibernate
- Peripheral Driver with Hibernate

Nucleus Kernel
- Hardware Agnostic

Nucleus BSP
- Hardware Specific
Tick Suppression Disabled

Measured Power Data

TICK SUPPRESSION DISABLED

All of these spikes between the busy times are ticks (tick suppression was disabled for this part of the demo test pattern).

BUSY (Tasks Running)
Tick Suppression

Unsuppressed Ticks (3261 hardware timer max value suppresses 17 ticks, in this test we set it to 15)

BUSY (Tasks Running)
CPU Idle Determination

GO LOWER ONLY IF $\Rightarrow$
A power-aware RTOS should abstract these details and enable the user to make application level DVFS decisions.

<table>
<thead>
<tr>
<th>1.5V</th>
<th>mHZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP #3</td>
<td>454</td>
</tr>
<tr>
<td>OP #2</td>
<td>297</td>
</tr>
<tr>
<td>OP #1</td>
<td>63</td>
</tr>
<tr>
<td>OP #0</td>
<td>1</td>
</tr>
<tr>
<td>Standby</td>
<td></td>
</tr>
<tr>
<td>Hibernate</td>
<td></td>
</tr>
</tbody>
</table>
System State Service

Provides a concept of a global system power state. Each device driver should have a designated power state for each system state.

Implements Request / Release minimum system state logic. Applications may request a minimum system state required for their proper operation.

<table>
<thead>
<tr>
<th>System State</th>
<th>Serial</th>
<th>Touch</th>
<th>Display</th>
<th>Ethernet</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS_STANDBY2</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>SS_STANDBY1</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>SS_IDLE2</td>
<td>ON</td>
<td>ON</td>
<td>DIM</td>
<td>OFF</td>
</tr>
<tr>
<td>SS_IDLE1</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>SS_ACTIVE</td>
<td>ON</td>
<td>ON</td>
<td>BRIGHT</td>
<td>ON</td>
</tr>
</tbody>
</table>
Power Management Example

System State 1: LCD = ON; USB = ON
Operating Point: CPU Freq = 400 MHz

```
pm_status = NU_PM_Set_Current_OP(eMHZ_400);
pm_status = NU_PM_Set_System_State(eLCD_ON_USBH_ON);
```
System State 2: LCD = DIM; USB = ON

```c
pm_status = NU_PM_Set_System_State(eLCD_DIM_USBH_ON);
```
Power Management Example

**System State 3**: LCD = **OFF**; USB = **ON**

```
pm_status = NU_PM_Set_System_State(eLCD_OFF_USBH_ON);
```
System State 4: LCD = OFF; USB = OFF

```
pm_status = NU_PM_Set_System_State(eLCD_OFF_USBH_OFF);
```
System State 4: LCD = OFF; USB = OFF
Operating Point: CPU Freq = 133 MHz

\[
\text{pm\_status} = \text{NU\_PM\_Set\_Current\_OP}(\text{eMHZ\_133});
\]
Power Consumption at Various OPs

Operating Point voltage (1.5V)

- **Hibernate**
  - OP#0 1 MHz: 470 mA
  - OP#1 63 MHz: 380 mA
  - OP#2 297 MHz: 230 mA
  - OP#3 454 MHz: 380 mA

- **Standby**
  - OP#0 1 MHz: 200 mA
  - OP#1 63 MHz: 230 mA
  - OP#2 297 MHz: 370 mA
  - OP#3 454 MHz: 470 mA

SOC Current Consumption (mA)
i.MX28 Board

Nucleus Hibernate Announcement, FTF June 20, 2012
Impact on Battery Life ..

<table>
<thead>
<tr>
<th></th>
<th>mAh (Board)</th>
<th>Percentage Usage per hr</th>
<th>mAh</th>
<th>Battery (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP #3</td>
<td>470</td>
<td>10%</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>OP #2</td>
<td>370</td>
<td>5%</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>OP #1</td>
<td>230</td>
<td>10%</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>OP #0</td>
<td>200</td>
<td>15%</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Standby</td>
<td>38</td>
<td>20%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Hibernate</td>
<td>0</td>
<td>40%</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>126</td>
<td>19</td>
</tr>
</tbody>
</table>

Nucleus Power Management Framework

<table>
<thead>
<tr>
<th></th>
<th>mAh (Board)</th>
<th>Battery (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP #3</td>
<td>470</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

No Power Management
Real-world customer .. Tier 1 Medical

<table>
<thead>
<tr>
<th>Operating Point</th>
<th>Win CE</th>
<th>Nucleus</th>
</tr>
</thead>
<tbody>
<tr>
<td>454 Mhz</td>
<td>148mA</td>
<td>78mA</td>
</tr>
<tr>
<td>300 Mhz</td>
<td>102mA</td>
<td>58mA</td>
</tr>
</tbody>
</table>

Customer measured on i.MX28

Device shown is not the real device

Requirement:
2 x AA batteries required to run 24 hrs
Inflexion UI

Run-time Engine

- Performs all the graphics rendering on the target
- Supports 2D / 3D rendering and XML driven menus

UI Designer

- Host based tool to design the complete UI
- Decouples UI design from application implementation
Benefits

Nucleus RTOS

- Proven RTOS, Power, footprint, memory usage, reliability

Sourcery

- Best in class GNU based toolchain

Nucleus RTOS + Sourcery

- Highest level of OS integration
- RTL optimizations

Nucleus ReadyStart

- Complete solution: runtime and tools fully integrated
- Key technologies: broad connectivity support, power management built into the kernel, UI
- Business model: RTOS, Tools, & UI from a Single Provider
Try it

http://www.mentor.com/embedded-software/downloads/

Mentor Embedded Linux Lite for Freescale QorIQ and PowerQUICC

Mentor Embedded Linux Lite for LSI ACP2148v2

Mentor Embedded Linux Lite for Broadcom XLR/XLS
Demonstrations