Android and RTOS for Medical Applications

Colin Walls
colin_walls@mentor.com

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.
Agenda

Medical electronics overview
Using an RTOS
Introduction to Android
Using Android/Linux
Another design approach
Conclusions
Agenda

Medical electronics overview
Using an RTOS
Introduction to Android
Using Android/Linux
Another design approach
Conclusions
Key Health Care Trends and Issues

- Improved Diagnoses
- Preventative Care
- Better Technology
- Cost Containment
- Quality of Care
- Privacy
- Aging Population

- Capitation
- Insurance Approval
- Outpatient Care
- HIPAA
- Electronic Medical Records
- Increased Medical Needs
- Frequent Monitoring
- Self Diagnosis

Mentor Graphics Confidential Information
Key Health Care Trends and Issues

- Improved Diagnoses
- Preventative Care
- Better Technology

Cost Containment

- Capitation
- Insurance Approval
- Outpatient Care

Privacy

- HIPAA
- Electronic Medical Records

Aging Population

- Increased Medical Needs
- Frequent Monitoring
- Self Diagnosis

Quality of Care

- Mentor Graphics Confidential Information
Key Health Care Trends and Issues

- Improved Diagnoses
- Preventative Care
- Better Technology
- Capitation
- Insurance Approval
- Outpatient Care
- Cost Containment
- Quality of Care
- Privacy
- Aging Population
- HIPAA
- Electronic Medical Records
- Increased Medical Needs
- Frequent Monitoring
- Self Diagnosis

Mentor Graphics Confidential Information
Key Health Care Trends and Issues

- Improved Diagnoses
- Preventative Care
- Better Technology
- Capitation
- Insurance Approval
- Outpatient Care

Cost Containment
- HIPAA
- Electronic Medical Records

Quality of Care

Privacy

Aging Population
- Increased Medical Needs
- Frequent Monitoring
- Self Diagnosis
Connecting Points of Care

- Mobile Care
- Outpatient Care
- Long Term Care Facility
- Emergency Room
- Hospital Room

Mentor Graphics Confidential Information

Comprehensive Solutions for
Android™  •  Nucleus®  •  Linux®
Mobile & Beyond  •  2D/3D User Interfaces  •  Multi-OS  •  Networking
Connecting Points of Care

- Outpatient Care
- Mobile Care
- Emergency Room
- Hospital Room
- Long Term Care Facility
Connecting Points of Care

- Mobile Care
- Outpatient Care
- Emergency Room
- Hospital Room
- Long Term Care Facility

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

Mentor Graphics Confidential Information
Connecting Points of Care

- Mobile Care
- Outpatient Care
- Emergency Room
- Hospital Room
- Long Term Care Facility
Key Industry Trends

- Many points of care are becoming connected to share/consolidate information.
- Growing volume of patient data increasing IT burden
  - Security and accessibility are imperatives.
- Increasing need for cost-effective medical care and to provide care outside hospitals.
Key Industry Trends (con’t)

- Strong growth of software usage within medical devices driven by more advanced graphics, user interface, and connectivity features

- UI is more than aesthetics
  - reduces operator error
  - reassures patients
Key Industry Trends (con’t)

- More portable devices imposing demands on efficient, power-conservative designs
- Designs must be evolutionary, allowing expansion for other interfaces and features without product redesign and recertification
Emerging Trends in Medical Devices

- **Wireless Connectivity**
  - 802.11 a/b/n, ZigBee, Bluetooth, cellular
  - Security protocols and additional protected setup capabilities to comply with emerging IT requirements

- **More care moving into the clinic and home**
  - More patients needing more care

- **Electronic Records**
  - Patients will carry their own critical medical information
    - From ID bracelets to electronic devices storing medical information
Medical Device Selection Criteria

- **Device Size**
- **Accuracy**
- **Data Acquisition Connectivity**
- **Special Features:**
  - automatic timing, error codes, spoken instructions or results
- **Ability to Store Results in Memory or Externally**
- **Testing Speed**
- **Battery Life / Power Consumption**
- **Display / Interface**
- **Ease of Use**
- **Technical Support**
- **Testing Speed**
- **Connectivity**
- **Reports & Analytics**
# Unique Requirements for Medical Software

## Connectivity Capability for Portable Medical Devices
- Wireless 802.1x, TCP/IP/UDP, Certified USB 2.0, Bluetooth

## Increasing BOM and Time To Market pressures
- Reduce Costs of deployment
- Scalable RTOS
- Low Memory utilization
- Tools to assist with development and debug
- Certification

## Security and Data Integrity
- File System with data integrity in case of system failure

## User Interface and Power Management
- Power Management API for battery-operated devices
- Tools for building graphics and UI for touch screen displays

---

**Mentor Graphics Confidential Information**
Agenda

Medical electronics overview

Using an RTOS

Introduction to Android

Using Android/Linux

Another design approach

Conclusions
The Case for an RTOS

- Real time performance
  - fast
  - deterministic
- Cost
  - modest direct costs
  - small memory
  - makes best use of CPU power
- Certification
  - small code reduces costs
- Power consumption
Power and the OS

![Graph showing power distribution with milliamps on the y-axis and time (in milliseconds) on the x-axis. The graph is labeled as Sinewave 71db and the line is marked as Nucleus.]
Power and the OS

Sinewave 71db

Mentor Graphics Confidential Information
Example: Anesthesiology

- Brain-State Monitoring Systems
- 2-D Graphics, Networking, USB, File System, Nucleus OS
Example: Blood Glucose Monitoring

- User Interface, USB, Nucleus OS and 802.11 wireless
Agenda

Medical electronics overview
Using an RTOS

**Introduction to Android**

Using Android/Linux

Another design approach

Conclusions
What is Android?

- An operating system for smart phones?
  - yes and no

- An integrated software platform for building smart phones?
  - almost

- An integrated software platform for building connected devices
  - application framework on top of Linux
  - open source – not GPL
What is Android?
What is Android?

Application
- Open Source or Proprietary Middleware
  - glibc or uCllibc
- Linux

Application
- Android
- Linux
Android Architecture

- **Linux**
  - 2.6.3x
    - 115 patches required
  - provides:
    - process management
    - memory management
    - security
    - networking
  - drivers:
    - display, keypad, camera, WiFi, flash, audio, IPC, power
Android Architecture

- Libraries
  - e.g.:
    - libc - bionic
    - media libraries
    - graphics
    - SQLite
Android Architecture

- **Android runtime**
  - Dalvik VM
  - Not a Java VM
    - register based
  - one instance per application
  - memory optimized
  - uses Linux to manage memory and multi-threading
Android Architecture

- Application framework
  - services and systems:
    - views
    - content providers
    - resource manager
    - notification manager
    - activity manager
  - all Java classes
  - any application can publish its capabilities

Diagram:
- Application framework
- Libraries
- Android runtime
- Linux
Android Architecture

Applications

- programs provided:
  - email
  - SMS
  - calendar
  - Web browser
  - contacts

- all applications have the same status

- Java programs
  - Uses standard Java tools
  - Converted to Dalvik bytecode
Enable Customers to build product specific applications, UI
Produce product specific SDK and emulator

Porting/optimizing for instruction sets
Integrating with hardware acceleration
Extending beyond handset assumptions
Enable product and industry specific middleware

Extending beyond handset assumptions
Enable developing product and industry specific classes

Porting and optimizing Android Dalvik VM for CPU cores and SoCs

Providing Android-ready Linux kernel
Integrating and testing board support and industry-specific device drivers
Agenda

Medical electronics overview
Using an RTOS
Introduction to Android

Using Android/Linux

Another design approach
Conclusions
The Case in Favor of Linux/Android

- Excellent middleware availability
  - connectivity
  - file storage
- Expertise available
- Post-deployment applications
- User interface
The Case Against Linux/Android

- Real time behavior
  - not intrinsically real time

- Power consumption
  - large memory footprint
  - less efficient CPU utilization

- Certification
  - large source code volume
Agenda

Medical electronics overview
Using an RTOS
Introduction to Android
Using Android/Linux

Another design approach
Conclusions
Another Design Approach

Core 1
- Android
- Linux

Core 2
- Nucleus

MCAPI
Inter Processor Communication

Shared Memory Region
System Memory
Agenda

Medical electronics overview
Using an RTOS
Introduction to Android
Using Android/Linux
Another design approach

Conclusions
Conclusions

- Medical electronics is an expanding field, with specific demands on embedded software
- Traditional RTOS addresses many needs
- Linux/Android may also be attractive
- With multicore becoming more common, multi-OS may be a good solution
Thank you

Colin Walls
colin_walls@mentor.com
http://blogs.mentor.com/colinwalls

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.