Modern Embedded Computing: Designing Connected, Pervasive, Media-Rich Systems

Peter Barry, Intel
Patrick Crowley

Intel Summit 2/23/2013
St. Louis

- “Gateway to the West”
- Population: ~3 million
- Diverse economy: IT, engineering, plant & life sciences, manufacturing, aerospace, healthcare
Washington University (www.wustl.edu)

- Private research & teaching university
- 10,000 students (4,000/6,000 undergrads/grads)
- 10,000 faculty and staff
- Research focus: >$1B per year, 23 Nobel prize winners
- Washington U. Med School, among the best in the world
- Very strong research activity in CS & Networking
Key Points

• Peter Barry (Intel) and I have written a textbook on embedded systems
  – Lab-oriented, Atom-based
  – Covers all aspects of modern embedded systems

• We are looking for
  – Feedback
  – Adopters!
Why a New Textbook?

• Embedded systems have changed dramatically in recent years, and existing textbooks on the subject do not reflect this.

• Most existing text books focus on one or more of
  – Microcontrollers, 8- or 16-bit systems
  – Real-time systems
  – Applications in deeply embedded control

• Many modern embedded systems are
  – Connected
  – Media-rich
  – Highly integrated

• This core insight drives the development of the book, and distinguishes it from existing texts.
Connected

• Nearly all embedded systems include:
  – IP networking stacks
  – Link connectivity via a combination of wired and wireless network interfaces.

• Core feature sets often rely upon connectivity
Media-Rich

• Many embedded systems include graphical user interfaces with
  – high-resolution 2D and 3D graphics
  – Video and audio encode and decode
  – Inputs and outputs supporting standard and high-definition video and audio

• Modern systems provide these features through a combination of sw & hw
  – Strong implication for embedded software design
Aggressive Platform Integration

• For reasons of power efficiency, performance, and size, chips and chipsets for embedded systems are highly integrated

• On-die or on-package implementations of
  – memory and I/O controllers
  – accelerators for computationally intensive tasks such as encryption, media and compression
  – multiple programmable processor cores

• Modern engineers need a practical understanding of these sw & hw interactions
Book Overview

• Undergraduate embedded design engineering course
  – Lab-based, with exciting hands-on projects
  – For undergraduates in Computer Science, Computer Engineering, Electrical Engineering
  – 2nd year or later

• Covers all aspects of modern embedded systems
  – Example embedded platforms
  – Processors and chipsets
  – Operating systems, open-source firmware
  – Application frameworks
  – Power management
  – Networking, Multimedia

• Examples based on Intel Atom
  – Platform used to make principles and examples concrete, and for exercises
  – x86-based reference eases integration with other courses & lab resources

• Highlights and Features
  – Linux-based system software
  – Real-time OS and app concepts
  – Multimedia encoding & decoding
  – ACPI system and power management
  – Programming in C & Python
  – Platform and data security
  – App development with Android and Qt
  – Course material includes
    • Lecture slides
    • Homework & exam question bank
    • Projects
Book Organization: Part 1

• Principles of Modern Embedded Systems
  – Chapter 1 - Embedded System Landscape
  – Chapter 2 – Attributes of Embedded Systems
  – Chapter 3 - The Future of Embedded Systems
CHAPTER 3  The Future of Embedded Systems

Technology Trends ........................................................................................................... 23
Computation ....................................................................................................................... 24
Connectivity ....................................................................................................................... 25
Storage ............................................................................................................................... 29
Sensing .............................................................................................................................. 30
Issues, Applications, and Initiatives .................................................................................. 30
Energy ............................................................................................................................... 30
Security ............................................................................................................................. 32
Health ............................................................................................................................... 33
Challenges and Uncertainties ........................................................................................... 34
Open Systems, Internet Access, and Neutrality ................................................................. 34
Privacy ............................................................................................................................... 35
Successful Commercialization ......................................................................................... 36
Summary ........................................................................................................................... 36
If the available supply of energy is in question, the only substantive recourse is to lower demand. And, since demand has historically tracked economic development, efficiency must be the goal. (One way to dramatically reduce energy consumption would be to dramatically cripple civilization.) The largest and fastest growing economies must accelerate the efficiency of their energy consumption; they must do more with less.

In this direction, embedded systems will surely play a substantive role. From smart grid technologies that improve the efficiency of energy distribution to smart appliances and building climate control systems that adapt operations to both user-driven need and the cost of energy to the interactive interfaces that close the energy consumption loop for users to help guide informed consumption, the rise of embedded systems will be a primary enabler of future energy efficiency initiatives.
Book Organization: Part 2

• Embedded Systems Architecture & Operation
  – Chapter 4 - Embedded Platform Architecture
  – Chapter 5 - Embedded Processor Architecture
  – Chapter 6 - Embedded Platform Boot Sequence
  – Chapter 7 - Operating Systems Overview
  – Chapter 8 - Embedded Linux
  – Chapter 9 - Power Optimization
  – Chapter 10 - Embedded Graphics & Multimedia Acceleration
  – Chapter 11 - Digital Signal Processing
  – Chapter 12 - Network Connectivity
  – Chapter 13 - Application Frameworks
  – Chapter 14 - Platform and Content Security
  – Chapter 15 - Advanced: SMP, AMP and Virtualization
## CHAPTER 12  **Network Connectivity** ................................................................. 347

- Networking Basics ......................................................................................... 349
- Layering and Network Software ................................................................. 349
- Node Operation and Network Hardware ..................................................... 350
- Sockets and a Simple Example .................................................................. 351
- TCP/IP Networking .................................................................................... 353
- Governance, the IETF, and RFCs ............................................................... 354
- Addresses, Packets, and Routes ................................................................. 355
- Port Numbers, Byte Ordering, and OS Tools ............................................ 359
- Supporting Protocols and Services ............................................................ 360
- Ethernet ......................................................................................................... 361
- History ......................................................................................................... 361
- Protocol Description .................................................................................. 362
- Ethernet MAC Addresses .......................................................................... 363
- Ethernet Packet Format ............................................................................. 363
- A Gigabit Ethernet Controller and Its Features ........................................ 364
- Wi-Fi and IEEE 802.11 ............................................................................. 365
- History ......................................................................................................... 366
- Protocol Description .................................................................................. 366
- Frame Format .............................................................................................. 367
- A Wi-Fi Adapter and Its Features ............................................................... 368
- Bluetooth ..................................................................................................... 369
- History ......................................................................................................... 369
Chapter 12, excerpt p. 352

```python
1 import socket, sys
2 if len(sys.argv) > 1:
3     dst = sys.argv[1]
4 else:
5     dst = 'localhost'
6 # Create a socket
7 s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
8 # Open a connection to the local machine at port 7777
9 s.connect((dst, 7777))
10 # Send some bytes through the connection.
11 s.send("Mr. Watson--come here--I want to see you."")
12 s.close()
13 print "Message sent."

Listing 12.1
Sending a UDP Packet: udp_sender.py.
```

In 13 lines, including comments, we have a complete sockets program. This program sends text data via UDP, and we can see precisely how by considering each section in turn.

Lines 1–5: The `socket` and `sys` libraries are imported. The program takes a single, optional command line destination argument; `sys.argv` is an array of strings with the program name in the first location, and each subsequent location contains the next command line parameter. If no argument is given, 'localhost' is the default.
Book Organization: Part 3

• Developing an Embedded System
  – Chapter 16 - Example Designs
  – Chapter 17 - Platform Debug
  – Chapter 18 - Performance Tuning
# Sample: Chapter 17

## Platform Debug

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debugging New Platforms</td>
<td>465</td>
</tr>
<tr>
<td>A Process for Debugging a New Platform</td>
<td>466</td>
</tr>
<tr>
<td>Debug Tools and Chipset Features</td>
<td>467</td>
</tr>
<tr>
<td>Oscilloscopes</td>
<td>468</td>
</tr>
<tr>
<td>Logic Analyzers</td>
<td>468</td>
</tr>
<tr>
<td>Bus Analyzers</td>
<td>468</td>
</tr>
<tr>
<td>Power-On Self-Test (POST) Cards</td>
<td>468</td>
</tr>
<tr>
<td>JTAG Adapters</td>
<td>469</td>
</tr>
<tr>
<td>Debug Process Details</td>
<td>469</td>
</tr>
<tr>
<td>Visual Examination</td>
<td>469</td>
</tr>
<tr>
<td>Hardware Evaluation</td>
<td>470</td>
</tr>
<tr>
<td>Software Evaluation</td>
<td>474</td>
</tr>
<tr>
<td>Additional Resources</td>
<td>474</td>
</tr>
<tr>
<td>Summary</td>
<td>475</td>
</tr>
</tbody>
</table>
FIGURE 17.1
Platform Debug Process.
# Sample Semester Syllabus

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Topic</th>
<th>Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro to Embedded Systems</td>
<td>Ch. 1-3</td>
</tr>
<tr>
<td>2-3</td>
<td>Platforms, Chipsets &amp; Peripherals</td>
<td>Ch. 4, 5, 16</td>
</tr>
<tr>
<td>4</td>
<td>Bringing up a System</td>
<td>Ch. 6</td>
</tr>
<tr>
<td>5</td>
<td>Traditional and Real-Time OSes</td>
<td>Ch. 7</td>
</tr>
<tr>
<td>6</td>
<td>Linux</td>
<td>Ch. 8</td>
</tr>
<tr>
<td>7-8</td>
<td>Connectivity &amp; Network Programming</td>
<td>Ch. 12</td>
</tr>
<tr>
<td>9</td>
<td>Media Processing</td>
<td>Ch. 10, 11</td>
</tr>
<tr>
<td>10</td>
<td>App Dev with Android &amp; Qt</td>
<td>Ch. 13</td>
</tr>
<tr>
<td>11</td>
<td>Security</td>
<td>Ch. 14</td>
</tr>
<tr>
<td>12</td>
<td>Multithreading, Multiprocessing &amp; Virtualization</td>
<td>Ch. 15</td>
</tr>
<tr>
<td>13</td>
<td>Platform and App Power Optimization</td>
<td>Ch. 9</td>
</tr>
<tr>
<td>14-15</td>
<td>Platform Debug &amp; Tuning</td>
<td>Ch. 17, 18</td>
</tr>
</tbody>
</table>

- Bi-weekly assignments weeks 1-10, Final project weeks 11-14
- Mid-term and final exam
Conclusion

• Book is available now

• Online course materials should be available soon
  – Lectures slides, problem sets, projects

• We are actively seeking feedback and adopters
  – pcrowley@wustl.edu
  – peter.barry@intel.com