Embedded Systems
Lecture 1: Introduction

Björn Franke
University of Edinburgh
Overview

• Definitions, Motivation

• Examples of Embedded Systems

• Characteristics of Embedded Systems

• Course Overview

• Coursework
Lecturers & Teaching Assistant

- Björn Franke
  bfranke@inf.ed.ac.uk

- Michael O’Boyle
  mob@inf.ed.ac.uk

- Harry Wagstaff
  s0803652@inf.ed.ac.uk
Definition of an Embedded System

- “Embedded Systems are information processing systems embedded into a larger product”
  (Peter Marwedel, TU Dortmund)

- “Embedded software is software integrated with physical processes. The technical problem is managing time and concurrency in computational systems.”
  (Edward Lee, Berkeley)

- “Cyber-Physical (cy-phy) Systems (CPS) are integrations of computation with physical processes”
  (Edward Lee, Berkeley)

- Cyber-physical system (CPS) = Embedded System (ES) + physical environment
Example of an Embedded System

- Automotive electronics
  - ABS: Anti-lock braking systems
  - ESP: Electronic stability control
  - Airbags
  - Efficient automatic gearboxes
  - Theft prevention with smart keys
  - Blind-angle alert systems
  - In-car entertainment systems
  - ... etc ...

- Multiple networks
  - Multiple networked processors
Another Example

- Avionics
  - Flight control systems,
  - anti-collision systems,
  - pilot information systems,
  - power supply system,
  - flap control system,
  - entertainment system,
  - …

Dependability is of utmost importance.
Motivation for Studying Embedded Systems

• Trend in Information Processing Systems towards
  - Ubiquitous computing, Pervasive computing, Ambient intelligence,
  - Post-PC era

• Requires holistic approach involving embedded software, embedded hardware and physical environment

• Additional constraints and challenges:
  Power/Energy, Cost, Dependability, Real-Time Processing, ...

• Underrepresented in teaching
Importance of Embedded Systems

• $6bn embedded processors market in 2012, 12-15% growth in the next two years

• 49.7% of Americans own smartphones [www.itfacts.biz, March 31, 2012]

• Average car has about 15 microprocessors in it. S-class has 63 microprocessors; a 1999 BMW 7-series has 65 [Microprocessor Report 2009]

• Average middle-class household has about 40 to 50 microprocessors in it [Microprocessor Report 2009]

• …, the market for remote home health monitoring is expected to generate $225 mln revenue in 2011, up from less than $70 mln in 2006, according to Parks Associates. [www.itfacts.biz, Sep. 4th, 2007]
Embedded Hardware

- Domain/application-specific: Optimised for one fixed application/domain
- Energy-efficiency often more important than raw performance, especially for battery operated devices
- Power constraints: Cooling, power supply, ...
- Cost: Low cost for large volume device vs Non-recurring engineering cost
- Programmability: ASIC (no flexibility), ASIP, CPU, FPGA (lots of flexibility)
- Complexity: Composed of individual building blocks (IP blocks)
TI OMAP5430 SoC

Dynamic memory manager
L2 cache

ARM Cortex-M4
ARM Cortex-A15
3D graphics

PowerVR SGX544-Mp
CS4x DSP
IVA-HD video accelerator

Image signal processor

L3 Network-on-chip interconnect

Timers, Int Controller, Mailboxes, System DMA

M-Shield system security technology: SHA-1/SHA-2/MD5, DES/3DES, RNG, AES, PKA, secure WDT, keys, crypto DMA

L4 peripherals
Multi-pipe display sub-system (DSS)

Debug & trace c-ITAS/STP/PTM

Emulator pod, Trace analyzer

Fast IRDA

Example
Another Example
Embedded Software

- Real-time: Timing constraints set by physical environment
- Reactive: Response to physical environment
- Concurrency: Physical environment is not sequential
- Dependability: Impact on physical environment, safety-critical
- Reliability: Fixing bugs in the field may be costly/impossible
- Efficiency: Manual optimisation required
- (Lack of) Abstraction: Exposure of underlying hardware to the programmer
Course Overview

1. Introduction
2. Interfacing with the Environment
3. Coursework
4. Models of Computation 1 & 2
5. Imperative Programming Languages
6. Embedded Hardware
7. Power/Energy/Faults
8. Scheduling Theory
9. Real-Time Operating Systems
10. Guest Lecture
11. Worst-Case Execution Time
12. Mapping & Scheduling for Multi-Core 1
13. Mapping & Scheduling for Multi-Core 2
14. Mapping & Scheduling for Multi-Core 3
15. HW & SW Optimisations 1
16. HW & SW Optimisations 2
17. Dynamic Voltage Scaling/Dynamic Frequency Scaling
18. Revision
Coursework Overview

• Two parts of individual coursework

• Coursework 25% of total course mark

• 50/50 split of marks

• Alarm clock specification & implementation

• Networked audio player

• Freescale Kinetis K70 Tower Module (ARM Cortex M4)
Textbook and Course Website

• Recommended textbook:
  Peter Marwedel
  “Embedded System Design”
  ISBN 13 978 94 007 0256 1

• Other textbooks:
  Alan Shaw, Real-Time Systems and Software, John Wiley & Sons

• Course website:
  www.inf.ed.ac.uk/teaching/courses/es
Summary

• Examples of Embedded Systems

• Embedded Hardware and Software

• Course Overview
Preview

- Next Lecture: Interfacing with the Environment
- Input (Sensors), Output (Actors)
- Analog-Digital Conversion, Digital-Analog Conversion