

# Embedded Systems

Martin Schoeberl

`mschoebe@mail.tuwien.ac.at`

Very Small Information Systems, 2006

# Outline

- 1 Introduction
  - Definition and Characteristics
  - Interface to the World
  - Resource Constraints
  - Real-Time Systems
  
- 2 Our Example System

# What Are Embedded Systems?

## Definition

An **embedded system** is a computer systems that is part of a larger system.

## Example

- Washing machine
- Car engine control
- Mobile phone

# What Are Embedded Systems?

## Definition

An **embedded system** is a computer systems that is part of a larger system.

## Example

- Washing machine
- Car engine control
- Mobile phone

# ES Characteristics

- Often mass products
  - 98% of the processors are in ES
  - Sometimes very specialized systems
- No or minimal user interface
- Resource constraints
- Must usually fulfill strict timing
- Usually runs forever (no reboot)

# ES Characteristics

- Often mass products
  - 98% of the processors are in ES
  - Sometimes very specialized systems
- No or minimal user interface
- Resource constraints
- Must usually fulfill strict timing
- Usually runs forever (no reboot)

# ES Characteristics

- Often mass products
  - 98% of the processors are in ES
  - Sometimes very specialized systems
- No or minimal user interface
- Resource constraints
- Must usually fulfill strict timing
- Usually runs forever (no reboot)

# ES Characteristics

- Often mass products
  - 98% of the processors are in ES
  - Sometimes very specialized systems
- No or minimal user interface
- Resource constraints
- Must usually fulfill strict timing
- Usually runs forever (no reboot)



# ES Characteristics

- Often mass products
  - 98% of the processors are in ES
  - Sometimes very specialized systems
- No or minimal user interface
- Resource constraints
- Must usually fulfill strict timing
- Usually runs forever (no reboot)

# Interface to the World

- Minimal user interface
  - Buttons, lamps (e.g. elevator)
  - Small display
- Sensors
  - Switch (0/1)
  - Temperature
  - Camera
- Actuators
  - Relay (On/Off)
  - Servo motor
- Communication

# Interface to the World

- Minimal user interface
  - Buttons, lamps (e.g. elevator)
  - Small display
- Sensors
  - Switch (0/1)
  - Temperature
  - Camera
- Actuators
  - Relay (On/Off)
  - Servo motor
- Communication

# Interface to the World

- Minimal user interface
  - Buttons, lamps (e.g. elevator)
  - Small display
- Sensors
  - Switch (0/1)
  - Temperature
  - Camera
- Actuators
  - Relay (On/Off)
  - Servo motor
- Communication

# Interface to the World

- Minimal user interface
  - Buttons, lamps (e.g. elevator)
  - Small display
- Sensors
  - Switch (0/1)
  - Temperature
  - Camera
- Actuators
  - Relay (On/Off)
  - Servo motor
- Communication

# Resource Constraints

- Systems have to be *cheap*
- Memory
  - few 100 Bytes to few MB RAM
  - few KB to MB ROM
- Speed
  - few MHz up to a few 100MHz
  - Energy consumption
- Communication
  - Serial line
  - Special networks (Fieldbus)
- A lot of legacy systems

# Resource Constraints

- Systems have to be *cheap*
- Memory
  - few 100 Bytes to few MB RAM
  - few KB to MB ROM
- Speed
  - few MHz up to a few 100MHz
  - Energy consumption
- Communication
  - Serial line
  - Special networks (Fieldbus)
- A lot of legacy systems

# Resource Constraints

- Systems have to be *cheap*
- Memory
  - few 100 Bytes to few MB RAM
  - few KB to MB ROM
- Speed
  - few MHz up to a few 100MHz
  - Energy consumption
- Communication
  - Serial line
  - Special networks (Fieldbus)
- A lot of legacy systems



# Resource Constraints

- Systems have to be *cheap*
- Memory
  - few 100 Bytes to few MB RAM
  - few KB to MB ROM
- Speed
  - few MHz up to a few 100MHz
  - Energy consumption
- Communication
  - Serial line
  - Special networks (Fieldbus)
- A lot of legacy systems

# Resource Constraints

- Systems have to be *cheap*
- Memory
  - few 100 Bytes to few MB RAM
  - few KB to MB ROM
- Speed
  - few MHz up to a few 100MHz
  - Energy consumption
- Communication
  - Serial line
  - Special networks (Fieldbus)
- A lot of legacy systems

# Real-Time Systems

- Imagine a car accident
  - What happens when the airbag is fired too late?
  - Even one ms too late is too late!
- Timing is an important property
- *Conservative* programming styles

# Real-Time Systems

- Imagine a car accident
  - What happens when the airbag is fired too late?
  - Even one ms too late is too late!
- Timing is an important property
- *Conservative* programming styles

# Real-Time Systems

- Imagine a car accident
  - What happens when the airbag is fired too late?
  - Even one ms too late is too late!
- Timing is an important property
- *Conservative* programming styles

# Our Example System

- A Java processor board
- FPGA based
  - Processor is *software*
  - Can be configured
  - HW accelerator
- Interfaces
  - Digital IO, Analog input
  - Serial line
  - Ethernet
- Resources
  - up to 100MHz CPU
  - 1MB memory
  - 512KB + 32MB Flash

# Our Example System

- A Java processor board
- FPGA based
  - Processor is *software*
  - Can be configured
  - HW accelerator
- Interfaces
  - Digital IO, Analog input
  - Serial line
  - Ethernet
- Resources
  - up to 100MHz CPU
  - 1MB memory
  - 512KB + 32MB Flash

# Our Example System

- A Java processor board
- FPGA based
  - Processor is *software*
  - Can be configured
  - HW accelerator
- Interfaces
  - Digital IO, Analog input
  - Serial line
  - Ethernet
- Resources
  - up to 100MHz CPU
  - 1MB memory
  - 512KB + 32MB Flash



# Our Example System

- A Java processor board
- FPGA based
  - Processor is *software*
  - Can be configured
  - HW accelerator
- Interfaces
  - Digital IO, Analog input
  - Serial line
  - Ethernet
- Resources
  - up to 100MHz CPU
  - 1MB memory
  - 512KB + 32MB Flash

# Summary

- Embedded systems are **part** of a bigger system.
- ES systems are **small**.
- ES programming is programming with **resource constraints**.

# For Further Reading



**Wikipedia.**

http:

[//en.wikipedia.org/wiki/Embedded\\_system](http://en.wikipedia.org/wiki/Embedded_system)



**M. Schoeberl.**

*JOP: A Java Optimized Processor for Embedded Real-Time Systems.*

PhD thesis, Vienna University of Technology, 2005.

<http://www.jopdesign.com/thesis/thesis.pdf>