UNIVERSITÄT DORTMUND

# Embedded Systems

Peter Marwedel
Informatik 12
Univ. Dortmund
Germany

# Future of IT?

According to forecasts charac-terized by the terms such as

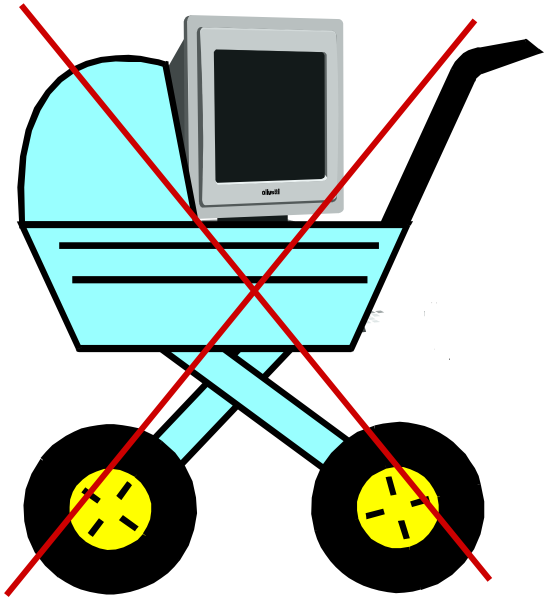
- *Post-PC era*
- *Disappearing computer*
- *Ubiquitous computing*
- *Pervasive computing*
- *Ambient intelligence*
- *Embedded systems*

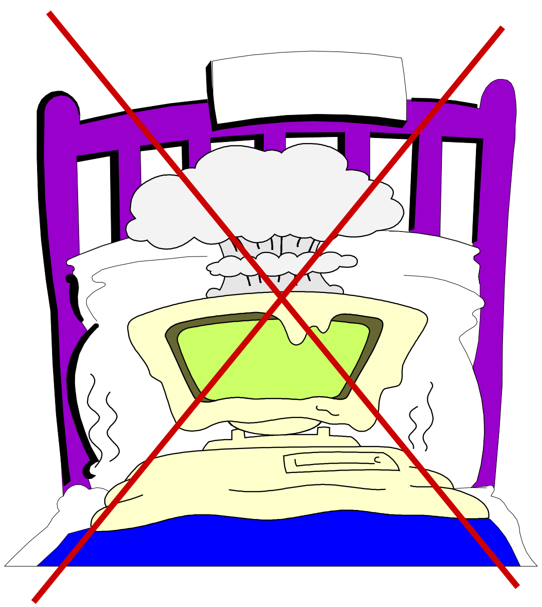






# What is an embedded system?





# Embedded Systems

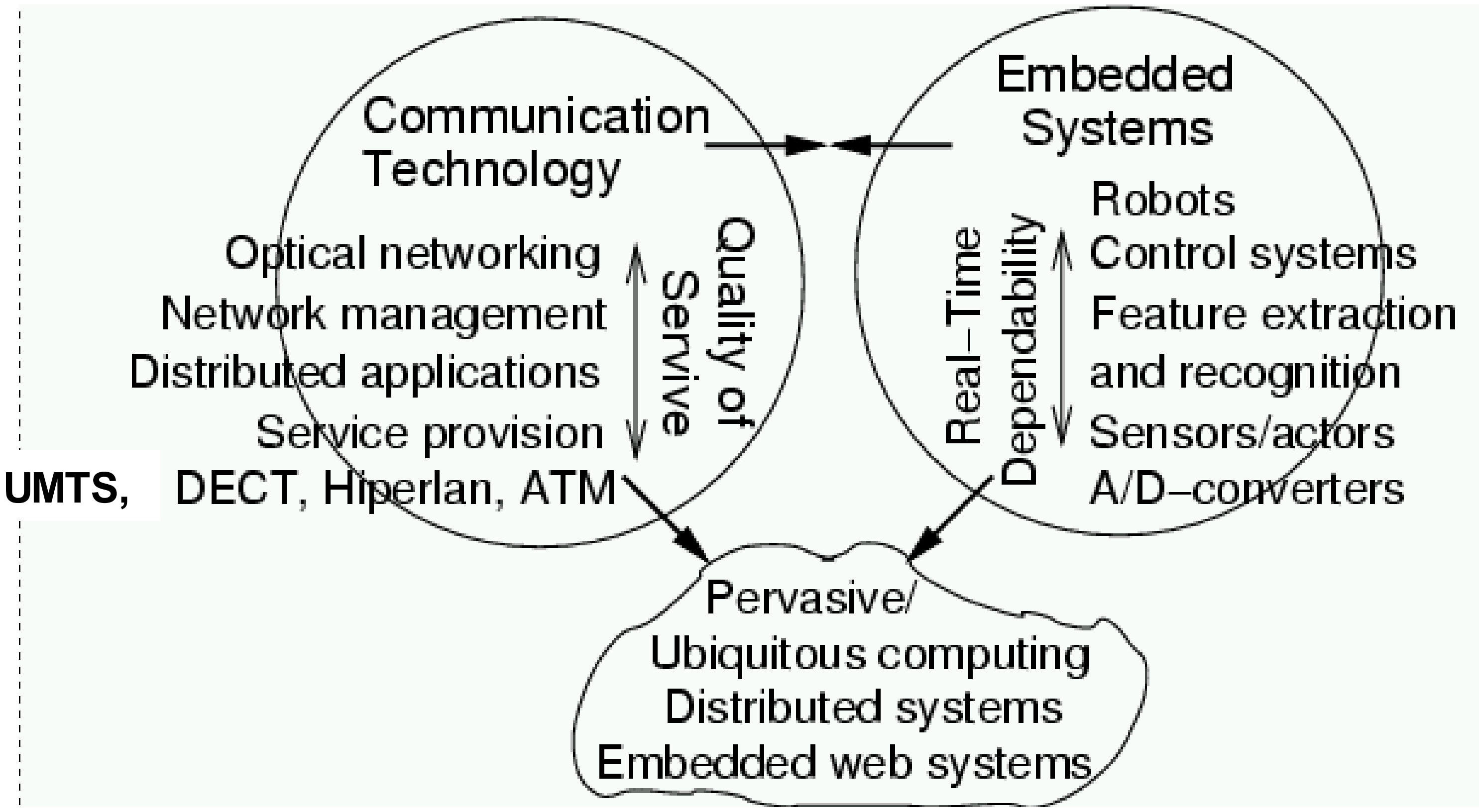
Embedded systems (ES) = **information processing systems embedded into a larger product**

!

Main reason for buying is **not** information processing

# Embedded systems and ubiquitous computing

Ubiquitous computing: Information anytime, anywhere.
Embedded systems provide fundamental technology.



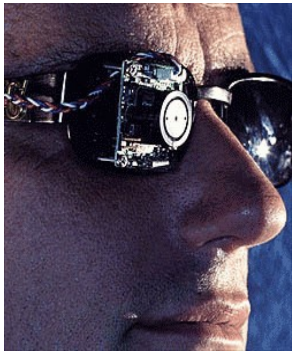
# Application areas (1)

- Automotive electronics
- Aircraft electronics
- Trains
- Telecommunication


# Artificial Eye

Several approaches

– Camera attached to glasses; computer worn at belt; output directly connected to the brain, “pioneering work by William Dobelle”. Previously at [www.dobelle.com]



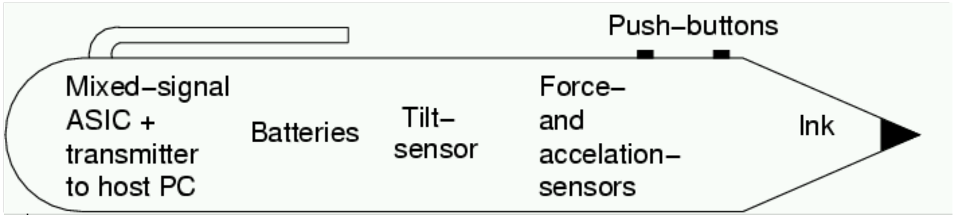
– Translation into sound; claiming much better resolution.
[http://www.seeingwithsound.com/etumble.htm]

# Application areas (3)

- Military applications

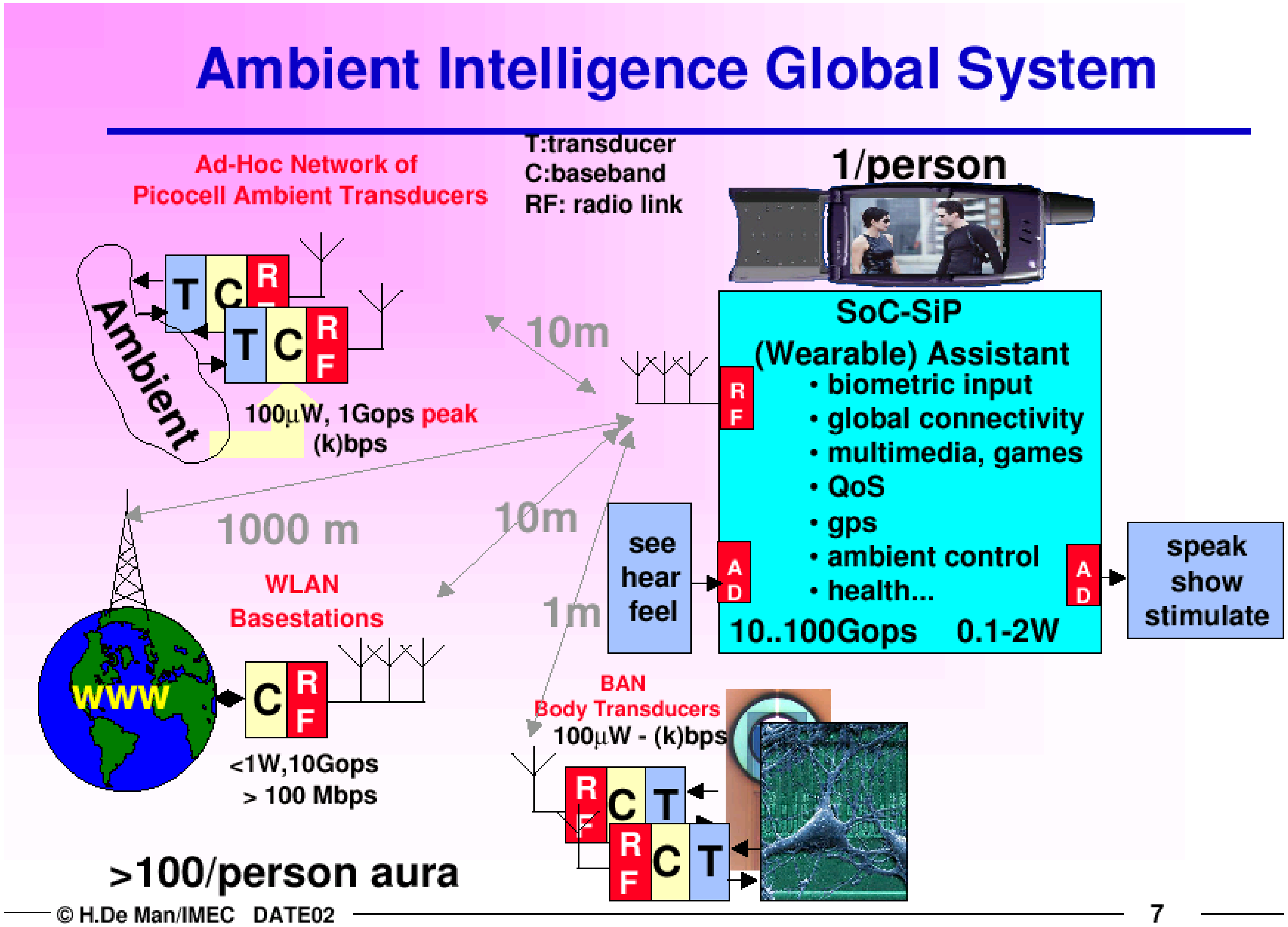
http://www.submarine.co.mp/wallpaper/submarine_640.jpg

- Authentication



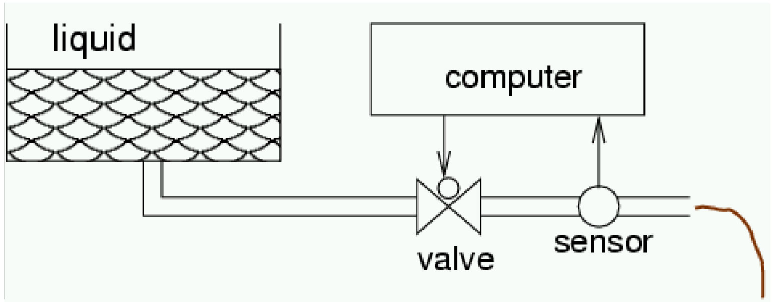
# Application areas (4)

- Consumer electronics



# Application areas (5)

- Fabrication equipment




- Smart buildings

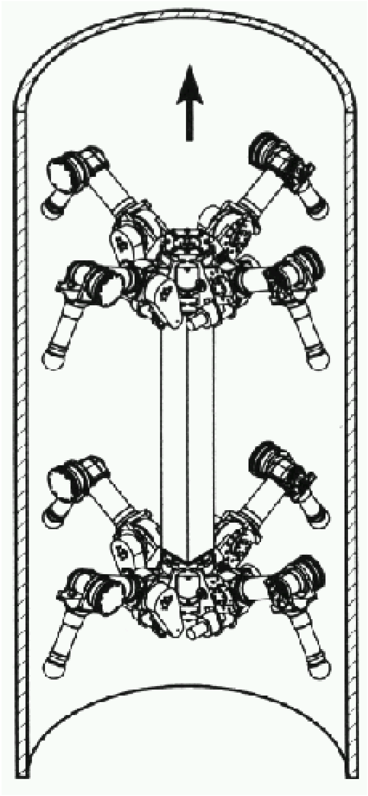


Show movie
http://www.date-conference.com/con 2003/keynotes/index.htm

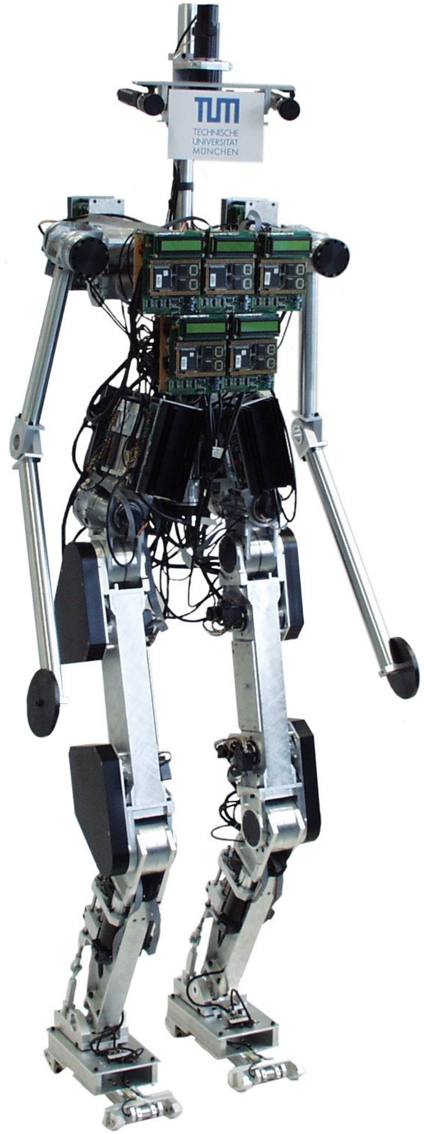
# Application areas (6)

- Robotics

„Pipe-climber“



Robot „Johnnie“ (Courtesy and ©: H.Ulbrich, F. Pfeiffer, TU München)



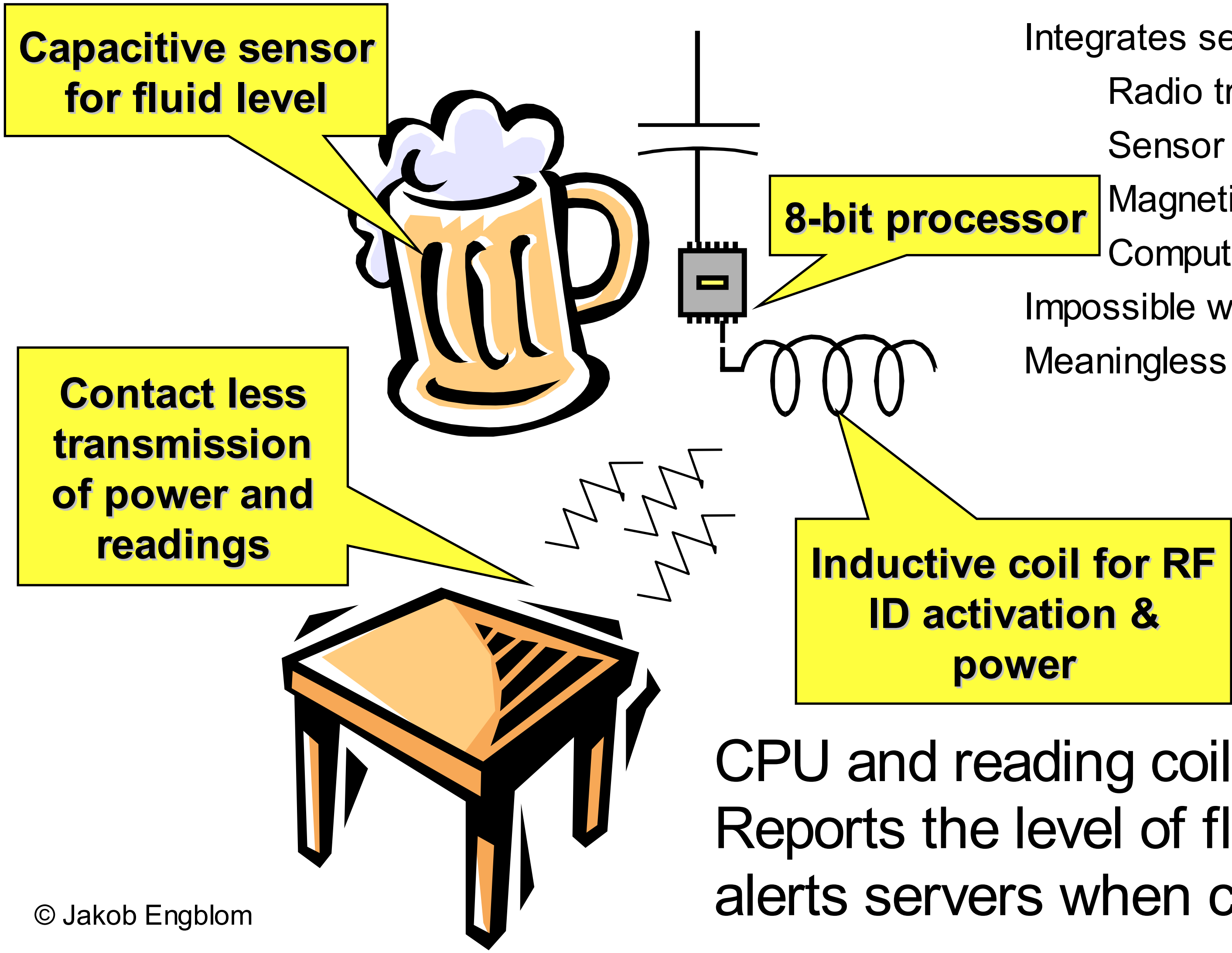


Show movie of 2-legged robot(s)

# Examples

Some embedded systems from real life

# Smart Beer Glass




Integrates several technologies:
- Radio transmissions
- Sensor technology
- Magnetic inductance for power
- Computer used for calibration

Impossible without the computer

Meaningless without the electronics

CPU and reading coil in the table. Reports the level of fluid in the glass, alerts servers when close to empty

UPPSALA UNIVERSITET

Information Technology

# Smart Beer Glass

- Typical embedded solution
- Integrates several technologies:
  - Radio transmissions
  - Sensor technology
  - Magnetic inductance for power
  - Computer used for calibration
- Impossible without the computer
- Meaningless without the electronics

UPPSALA UNIVERSITET

Information Technology

# Pedometer

- Obvious computer work:
  - Count steps
  - Keep time
  - Averages
  - etc.
- Hard computer work:
  - Actually identify when a step is taken
  - Sensor feels motion of device, not of user feet

# Mobile phones


- Multiprocessor
  - 8-bit/32-bit for UI
  - DSP for signals
  - 32-bit in IR port
  - 32-bit in Bluetooth
- 8-100 MB of memory
- All custom chips
- Power consumption & battery life depends on software

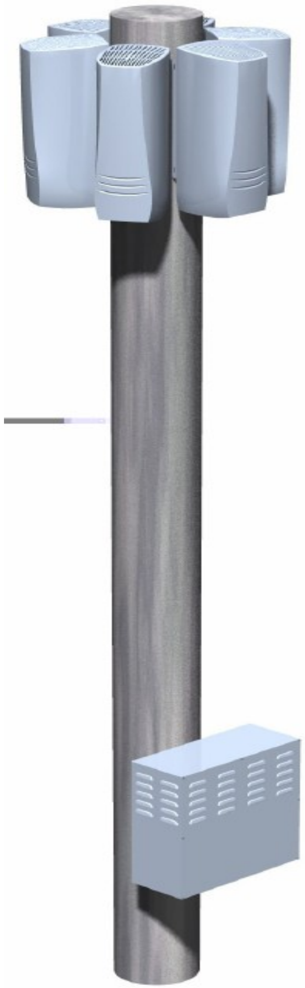
 

UPPSALA UNIVERSITET

Information Technology

# Mobile base station

- Massive signal processing
  - Several processing tasks per connected mobile phone
- Based on DSPs
  - Standard or custom
  - 100s of processors



# Telecom Switch

- Rack-based
  - Control cards
  - IO cards
  - DSP cards
  - ...
- Optical & copper connections
- Digital & analog signals

# Smart Welding Machine

- Electronics control voltage & speed of wire feed
- Adjusts to operator
  - kHz sample rate
  - 1000s of decisions/second
- Perfect weld even for quite clumsy operators
- Easier-to-use product, but no obvious computer

UPPSALA UNIVERSITET

Information Technology

# Sewing Machine

- User interface
  - Embroidery patterns
  - Touch-screen control
- ”Smart”
  - Sets pressure of foot depending on task
  - Raise foot when stopped
- New functions added by upgrading the software

UPPSALA
UNIVERSITET

Information Technology

# Forestry Machines

UPPSALA UNIVERSITET

Information Technology

# Forestry Machines

- Networked computer system
  - Controlling arms & tools
  - Navigating the forest
  - Recording the trees harvested
  - Crucial to efficient work
- Processors
  - 16-bit processors in a network

# Operator Panel


- Embedded PC
  - Graphical display
  - Touch panel
  - Joystick
  - Buttons
  - Keyboard
- But tough enough to be “out in the woods”

UPPSALA UNIVERSITET

Information Technology

# Cars

- Multiple processors
  - Up to 100
  - Networked together
- Multiple networks
  - Body, engine, telematics, media, safety

# Cars

- Functions by embedded processing:
  - ABS: Anti-lock braking systems
  - ESP: Electronic stability control
  - Airbags
  - Efficient automatic gearboxes
  - Theft prevention with smart keys
  - Blind-angle alert systems
  - ... etc ...

UPPSALA UNIVERSITET

Information Technology

# Cars

- Large diversity in processor types:
  - 8-bit – door locks, lights, etc.
  - 16-bit – most functions
  - 32-bit – engine control, airbags
- Form follows function
  - Processing where the action is
  - Sensors and actuators distributed all over the vehicle

# Extremely Large

- Functions requiring computers:
  - Radar
  - Weapons
  - Damage control
  - Navigation
  - basically everything
- Computers:
  - Large servers
  - 1000s of processors

UPPSALA UNIVERSITET

Information Technology

# Inside your PC

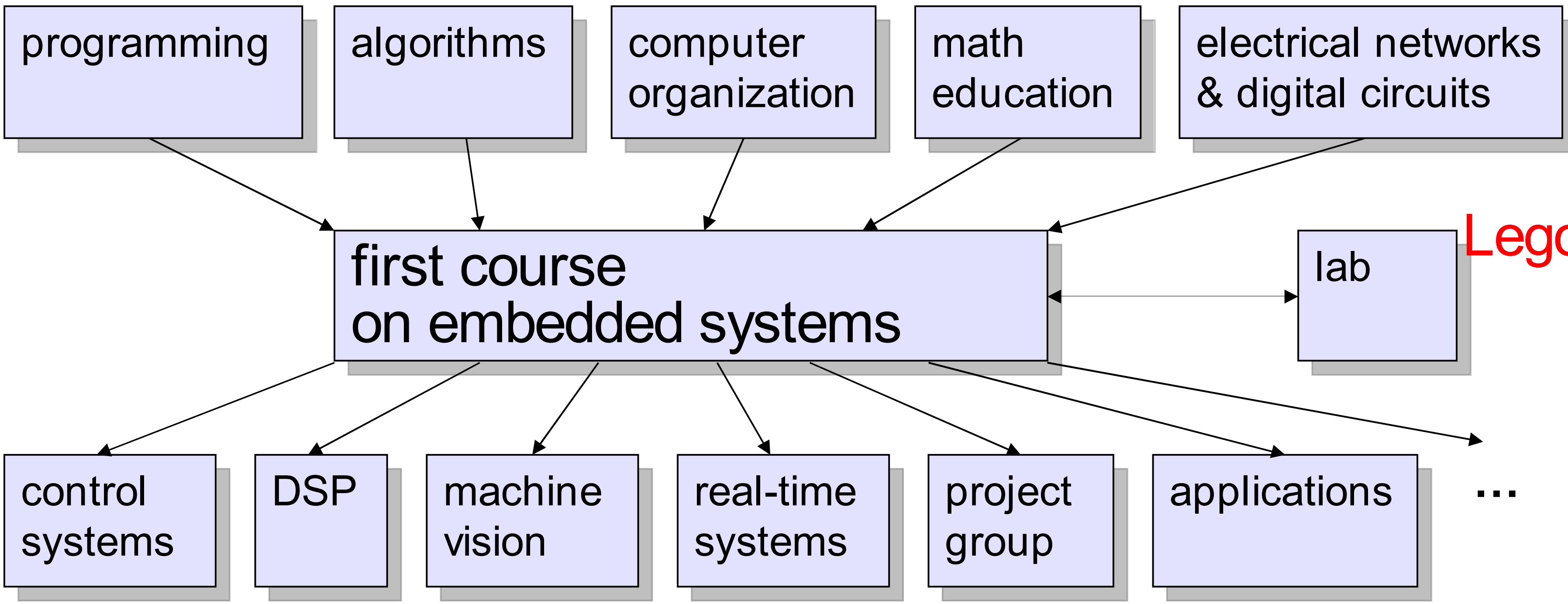
- Custom processors
  - Graphics, sound
- 32-bit processors
  - IR, Bluetooth
  - Network, WLAN
  - Harddisk
  - RAID controllers
- 8-bit processors
  - USB
  - Keyboard, mouse

# If you want to play

- Lego mindstorms robotics kit
  - Standard controller
    - 8-bit processor
    - 64 kB of memory
  - Electronics to interface to motors and sensors
- Good way to learn embedded systems

# Concept of ES education at Dortmund



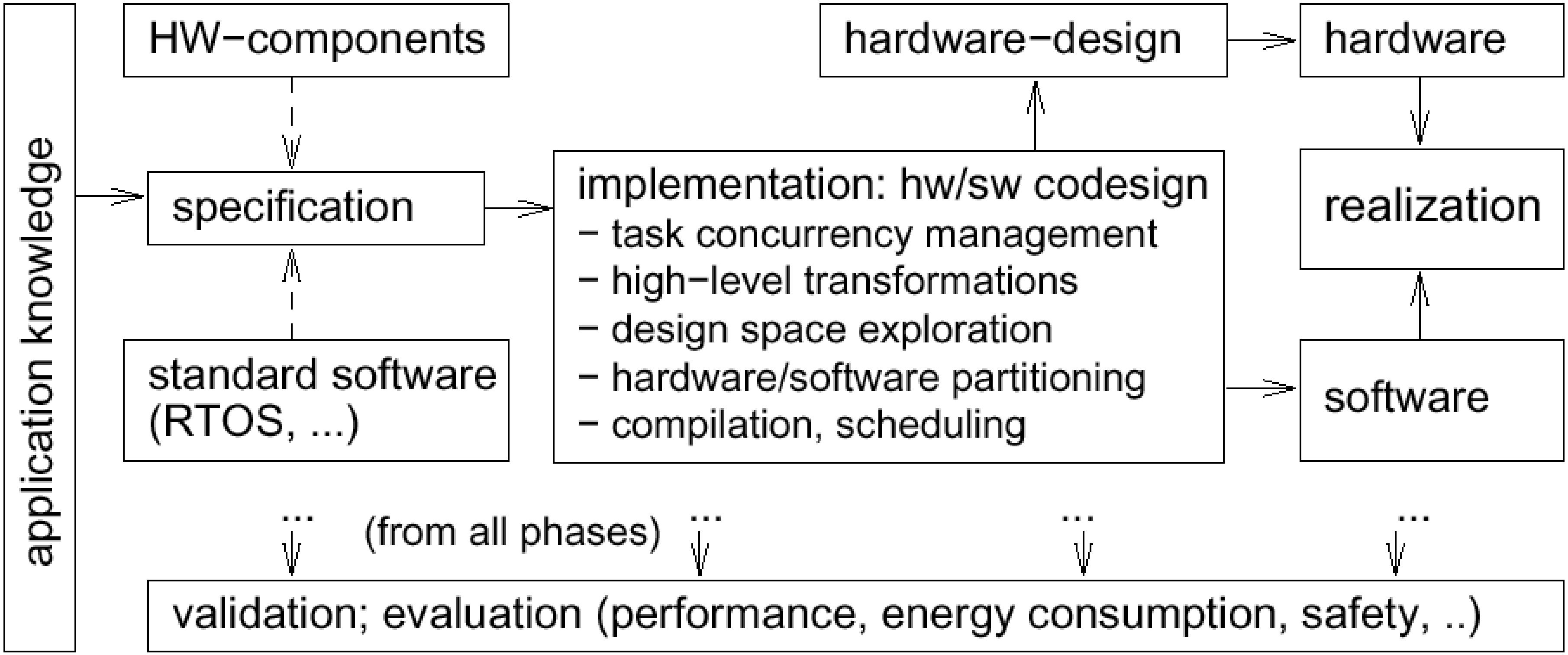

- Can typically be taught in 4th or 5th term
- Provides motivation and context of other work in the area
- Mix of students and courses from CS and EE departments

# Structure of the CS curriculum at Dortmund - current 4.5 year diploma program -

| Term | | | | |
|---|---|---|---|---|
| 1 | Computer organization | | Programming & semantics | Math education |
| 2 | Circuits & communication | OS | Algorithms | |
| 3 | HW lab | Networks | SW lab | |
| 4 | | Databases | … | |
| 5 | Embedded systems fundamentals | Software engineering | … | |
| 6 | Advanced topic in ES | … | … | |
| 7 | Project group | … | | |
| 8 | | … | … | |
| 9 | Thesis | | | |

All dependences met

# Structure of this course



# Broad scope avoids problems with narrow perspectives reported in ARTIST curriculum guidelines

"The lack of maturity of the domain results in a large variety of industrial practices, often due to cultural habits"

"curricula … concentrate on one technique and do not present a sufficiently wide perspective."

"As a result, industry has difficulty finding adequately trained engineers, fully aware of design choices."

Source: ARTIST network of excellence:
Guidelines for a Graduate Curriculum on Embedded Software and Systems, http://www.artist-embedded.org /Education/Education.pdf, 2003

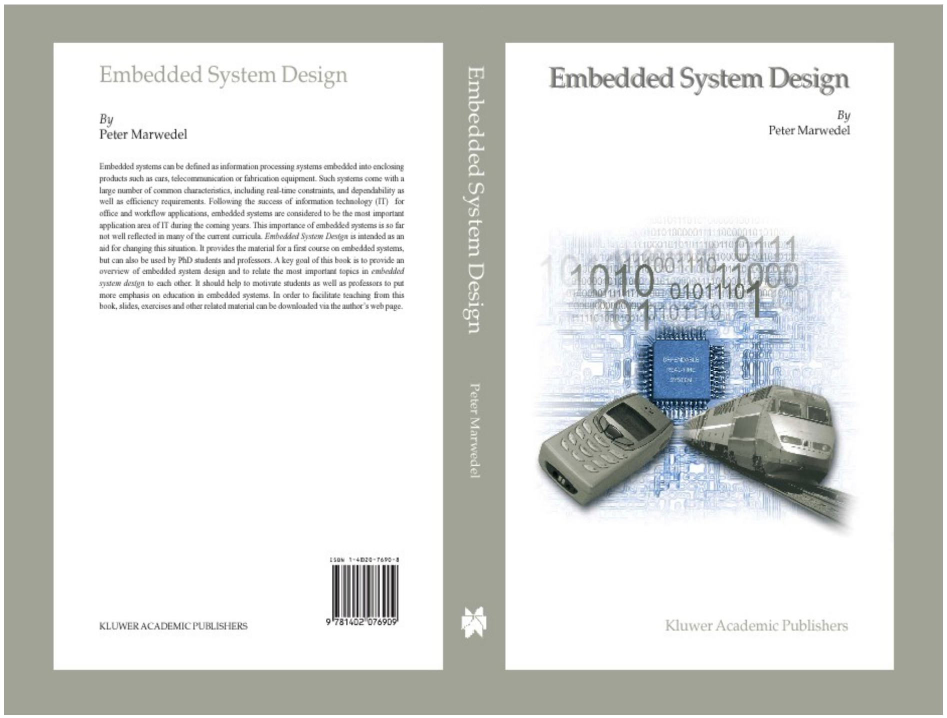
# Scope consistent with ARTIST guidelines

"The development of ES cannot ignore the underlying HW characteristics. Timing, memory usage, power consumption, and physical failures are important."

$$\int P \, dt$$

"It seems that fundamental bases are really difficult to acquire during continuous training if they haven't been initially learned, and we must focus on them."

# Textbook




3 Styles:

- Original hardcover version, Kluwer, 2003, >100 $/€
- Reprint, lighter cover borders, thicker paper, same price/ content; Corrections available on web site (see slides)
- 2nd edition, Springer, "2005", scheduled for end of October 2005, soft cover, with corrections, 37-39€
- German edition, scheduled for 2007, volunteers for proof-reading of first chapters sought

# Slides

- Slides are available at:
  http://ls12-www.cs.uni-dortmund.de/~marwedel/kluwer-es-book
  Master format: Powerpoint;
  Derived formats: OpenOffice, PDF
- Changes for term of winter 2006/7:
  - Additional "more in-depth sections".
  - Selected updates.

# Motivation for Course

"Information technology (IT) is on the verge of another revolution. Driven by the increasing capabilities and ever declining costs of computing and communications devices, IT is being embedded into a growing range of physical devices linked together through networks and will become ever more pervasive as the component technologies become smaller, faster, and cheaper... These networked systems of embedded computers ... have the potential to change radically the way people interact with their environment by linking together a range of devices and sensors that will allow information to be collected, shared, and processed in unprecedented ways. ... The use of [these embedded computers] throughout society could well dwarf previous milestones in the information revolution."

*National Research Council Report (US) Embedded Everywhere*

Source. Ed Lee, UC Berkeley, ARTEMIS Embedded Systems Conference, Graz, 5/2006]

# Growing importance of embedded systems (1)

- Growing economical importance of embedded systems THE growing market according to forecasts, e.g.:
  - *Worldwide mobile phone sales surpassed 156.4 mln units in Q2 2004, a 35% increase from Q2 2003, according to Gartner* [www.itfacts.biz]
  - *The worldwide portable flash player market exploded in 2003 and is expected to grow from 12.5 mln units in 2003 to over 50 mln units in 2008* [www.itfacts.biz]
  - *Global 3G subscribers will grow from an estimated 45 mln at the end of 2004 to 85 mln in 2005, according to Wireless World Forum.* [www.itfacts.biz]

# Growing importance of embedded systems (2)

- *The number of broadband lines worldwide increased by almost 55% to over 123 mln in the 12 months to the end of June 2004, according to Point-Topic.* [www.itfacts.biz]
- *Today's DVR (digital video recorders) users - 5% of households - will grow to 41% within five years, according to Forrester.* [www.itfacts.biz]
- *The automotive sector … ensures the employment of more than 4 million people in Europe. Altogether, some 8 million jobs in total depend on the fortunes of the transport industry and related sectors - representing around 7 per cent of the European Union's Gross National Product (GNP)* [OMI bulletin]

# Growing importance of embedded systems (3)

- .. *but embedded chips form the backbone of the electronics driven world in which we live ... they are part of almost everything that runs on electricity* [Mary Ryan, EEDesign, 1995]
- 79% of all high-end processors are used in embedded systems

The future is embedded, Embedded is the future!

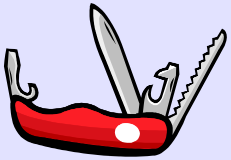
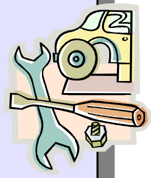
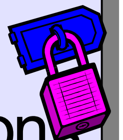
- Foundation for the „post PC era"
- ES hardly discussed in other CS courses
- ES important for Technical University
- ES important for Europe
- Scope: sets context for specialized courses

Importance of education

# Embedded Systems

Peter Marwedel
Informatik 12
Univ. Dortmund
Germany

# Characteristics of Embedded Systems (1)

- Must be **dependable**,
  - **Reliability *R(t)* =** probability of system working correctly provided that is was working at *t*=0
  
  - **Maintainability *M(d)* =** probability of system working correctly *d* time units after error occurred.
  
  - **Availability *A*(*t*)**: probability of system working at time *t*
  - **Safety**: no harm to be caused
  
  - **Security**: confidential and authentic communication
  

  Even perfectly designed systems can fail if the assumptions about the workload and possible errors turn out to be wrong.

  Making the system dependable must not be an after-thought, it must be considered from the very beginning

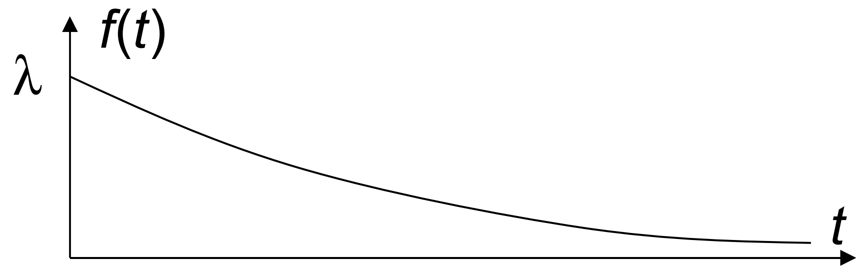
More in-depth:

# Reliability: f(t), F(t)

- Let $T$: time until first failure, $T$ is a random variable
- Let $f(t)$ be the density function of $T$

Example: Exponential distribution

$f(t)=\lambda e^{-\lambda t}$

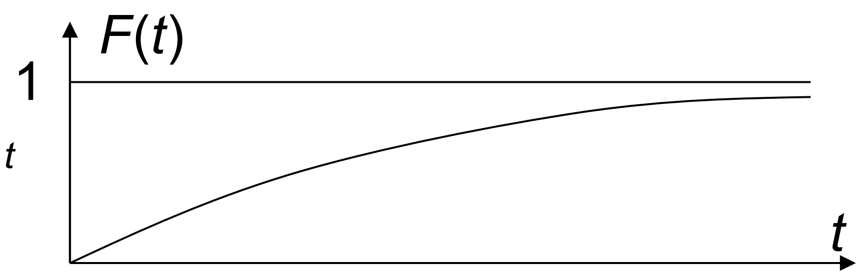



- $F(t)$ = probability of the system being faulty at time $t$:

$F(t) = \Pr(T \leq t)$ $\qquad$ $$F(t) = \int_0^t f(x)\,dx$$

Example: Exponential distribution

$$F(t) = \int_0^t \lambda e^{-\lambda x} dx = -[e^{-\lambda x}]_0^t = 1 - e^{-\lambda t}$$



# Reliability: *R*(*t*)

- **Reliability** $R(t)$ = probability that the time until the first failure is larger than some time $t$:

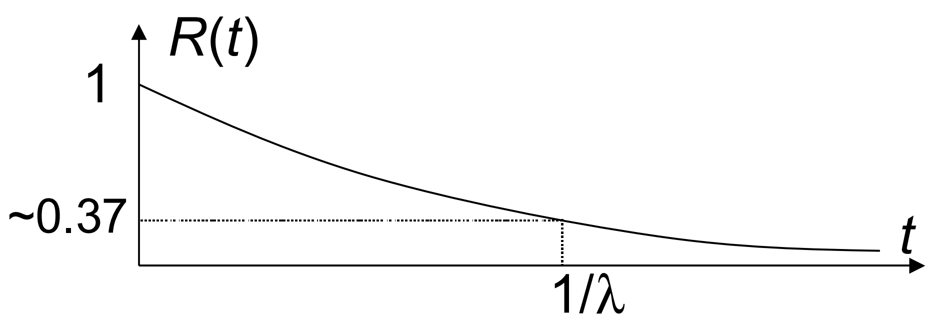
$R(t)=\Pr(T>t),\ t\geq 0$ $\qquad R(t)=\int_t^\infty f(x)dx$

$$F(t)+R(t)=\int_0^t f(x)dx+\int_t^\infty f(x)dx=1$$

$R(t)=1-F(t)$ $\qquad f(t)=-\frac{dR(t)}{dt}$

Example: Exponential distribution

$R(t)=e^{-\lambda t};$



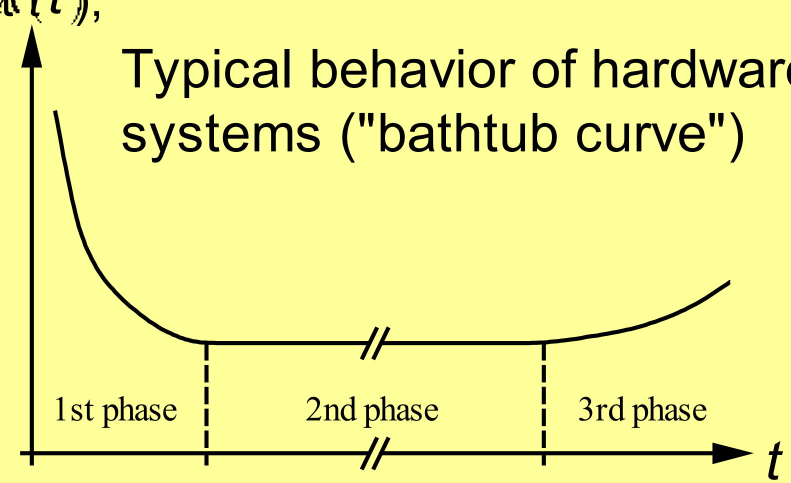
# Failure rate

The failure rate at time *t* is the probability of the system failing between time *t* and time *t+Δt:*

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\Pr(t < T \le t + \Delta t \mid T > t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{F(t+\Delta t) - F(t)}{\Delta t R(t)} = \frac{f(t)}{R(t)}$$

Conditional probability ("provided that the system works at t");

P(A|B)=P(AB)/P(B)




Typical behavior of hardware systems ("bathtub curve")

For exponential distribution:

$$\frac{f(t)}{R(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda$$

FIT = expected number of failures in $10^9$ hrs.

# MTTF = *E*{*T* }, the *statistical mean* value of *T*

$$\text{MTTF} = E\{T\} = \int_0^\infty t \cdot f(t)\, dt$$

According to the definition of the statistical mean value

Example: Exponential distribution

$$\text{MTTF}_{\text{exp}} = \int_0^\infty t \cdot \lambda\, e^{-\lambda t} dt = \cancel{-\left[t \cdot e^{-\lambda t}\right]_0^\infty} + \int_0^\infty e^{-\lambda t} dt$$

$$\int u \cdot v' = u \cdot v - \int u' \cdot v$$

$$\text{MTTF}_{\text{exp}} = -\frac{1}{\lambda}\left[e^{-\lambda t}\right]_0^\infty = -\frac{1}{\lambda}[0-1] = \frac{1}{\lambda}$$

MTTF is the reciprocal value of failure rate.

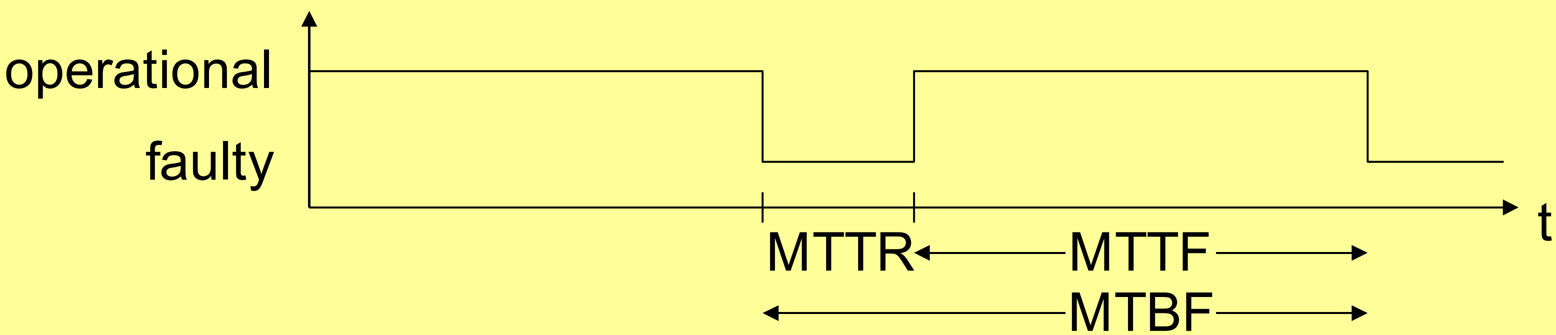
# MTTF, MTTR and MTBF

*end*

MTTR = mean time to repair
(average over repair times using distribution $M(d)$)
MTBF* = mean time between failures = MTTF + MTTR

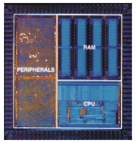
$$\text{Availability } A = \lim_{t \to \infty} A(t) = \frac{\text{MTTF}}{\text{MTBF}}$$

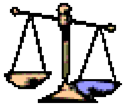
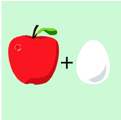
Ignoring the statistical nature of faults …




* Mixed up with MTTF, if starting in operational state is implicitly assumed

# Characteristics of Embedded Systems (2)

- Must be **efficient**
  - Energy efficient 
  - Code-size efficient (especially for systems on a chip) 

  - Run-time efficient 
  - Weight efficient 
  - Cost efficient 

- **Dedicated** towards a certain **application**
  Knowledge about behavior at design time can be used to minimize resources and to maximize robustness
- **Dedicated user interface**
  (no mouse, keyboard and screen) 

# Characteristics of Embedded Systems (3)

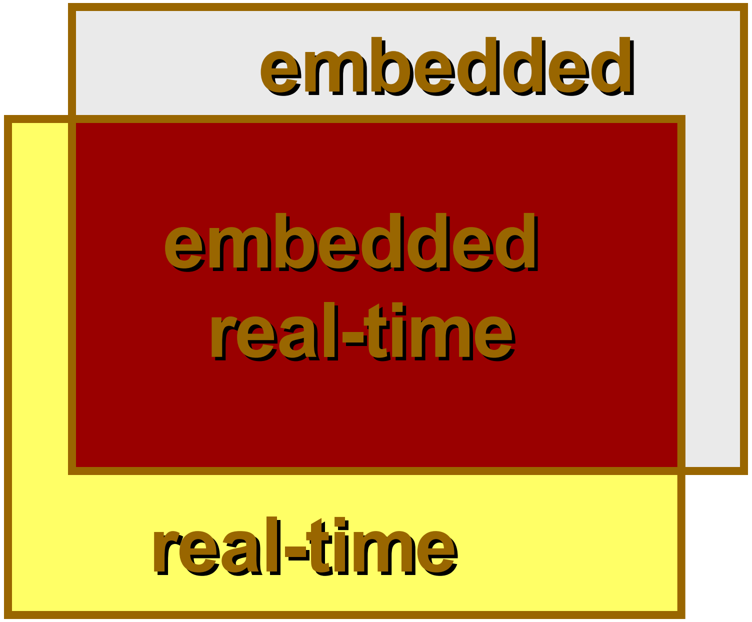
- Many ES must meet **real-time constraints**
  - A real-time system must react to stimuli from the controlled object (or the operator) within the time interval **dictated** by the environment.
  - For real-time systems, right answers arriving too late are wrong.
  - **„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe“** [Kopetz, 1997].
  - All other time-constraints are called **soft**.
  - A guaranteed system response has to be explained without statistical arguments

UPPSALA UNIVERSITET

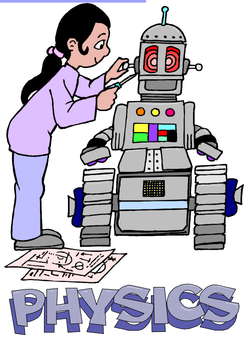
Information Technology

# Real-Time Systems

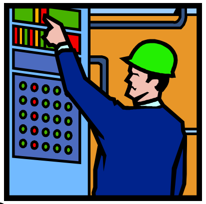
- Embedded and Real-Time
  - Synonymous?
- Most embedded systems are real-time
- Most real-time systems are embedded



# Characteristics of Embedded Systems (4)




- Frequently **connected to physical environment** through sensors and actuators,
- **Hybrid systems** (analog + digital parts).



- Typically, ES are **reactive systems**:
  **„A reactive system is one which is in continual interaction with is environment and executes at a pace determined by that environment“** [Bergé, 1995]
  Behavior depends on input **and current state.**
  ☞ automata model appropriate,
  model of computable functions inappropriate.

# Characteristics of Embedded Systems (5)

- ES are **underrepresented in teaching** and public discussions:
  *„Embedded chips aren't hyped in TV and magazine ads ...* [Mary Ryan, EEDesign, 1995]

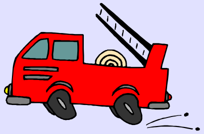
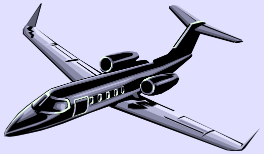
Not every ES has all of the above characteristics.

**Def.: Information processing systems having most of the above characteristics are called embedded systems.**

Course on embedded systems makes sense because of the number of common characteristics.

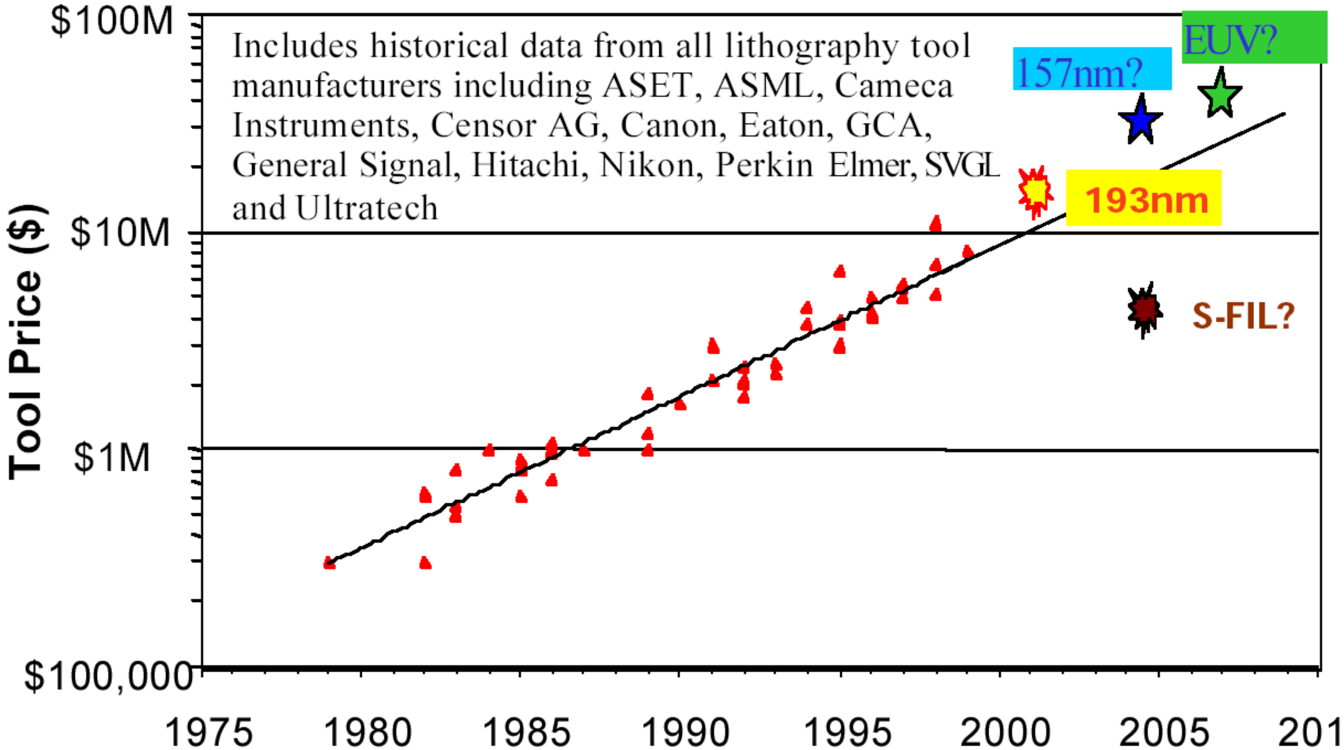
# Quite a number of challenges, e.g. dependability

Dependability? 

- Non-real time protocols used for real-time applications (e.g. Berlin fire department) 
- Over-simplification of models (e.g. aircraft anti-collision system) 
- Using unsafe systems for safety-critical missions (e.g. voice control system in Los Angeles; ~ 800 planes without voice connection to tower for > 3 hrs 

# Challenges for implementation in hardware

- Lack of flexibility (changing standards).
- Mask cost for specialized HW becomes very expensive



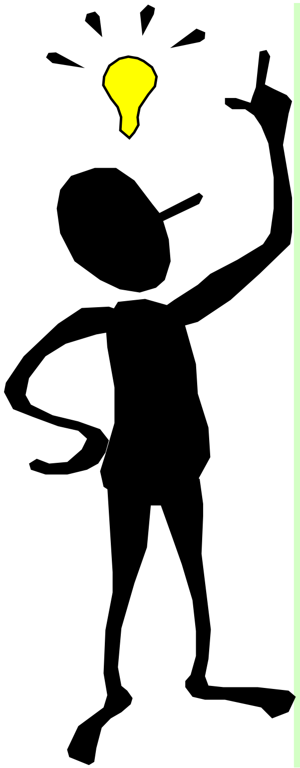

☞Trend towards implementation in Software

[http://www.molecularimprints.com/Technology/tech_articles/MII_COO_NIST_2001.PDF9]

# Importance of Embedded Software and Embedded Processors

"... the New York Times has estimated that the average American comes into contact with about 60 micro-processors every day...."
[Camposano, 1996]

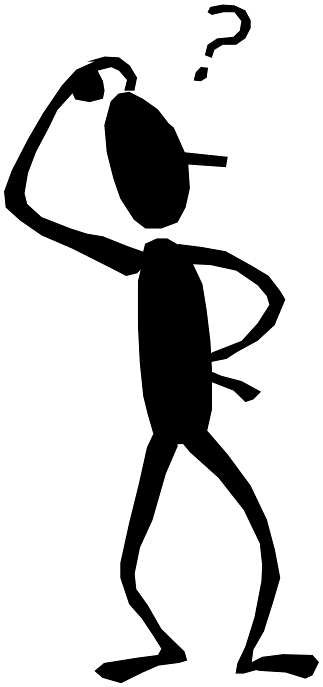
Latest top-level BMWs contain over 100 micro-processors
[Personal communication]



Most of the functionality will be implemented in software

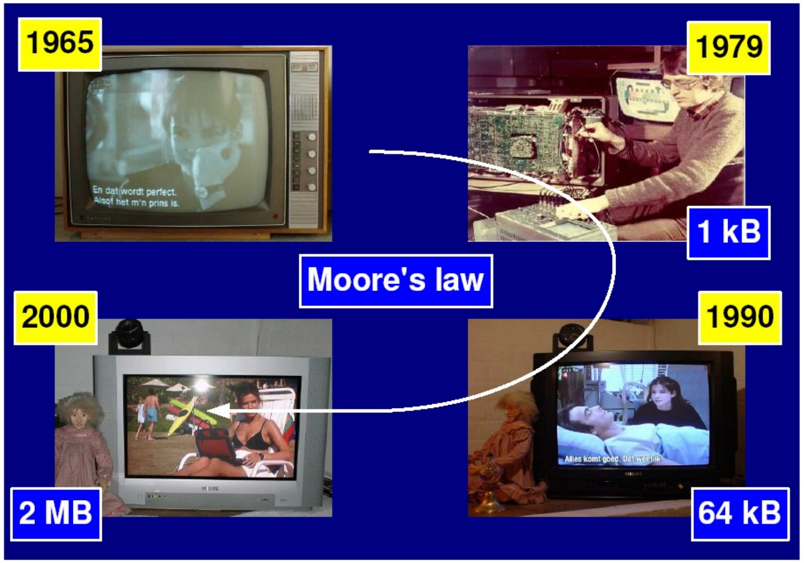
# Challenges for implementation in software

If embedded systems will be implemented mostly in software, then why don't we just use what software engineers have come up with?

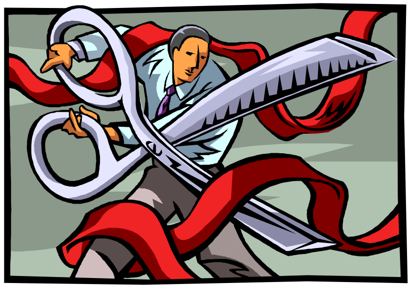


# Software complexity is a challenge

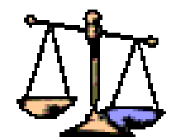
- Exponential increase in software complexity
- In some areas code size is doubling every 9 months [ST Microelectronics, Medea Workshop, Fall 2003]
- *... > 70% of the development cost for complex systems such as automotive electronics and communication systems are due to software development*
[A. Sangiovanni-Vincentelli, 1999]




Rob van Ommering, COPA Tutorial, as cited by: Gerrit Müller: Opportunities and challenges in embedded systems, Eindhoven Embedded Systems Institute, 2004



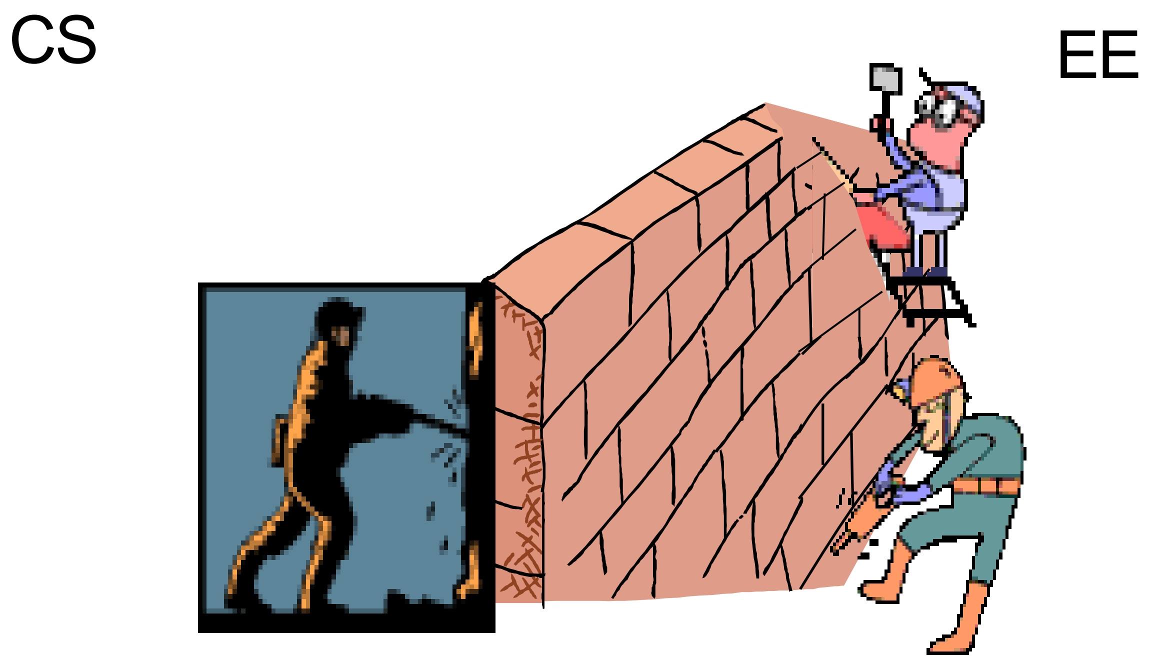
# Challenges for Embedded Software



- Dynamic environments
- Capture the required behaviour!
- Validate specifications
- Efficient translation of specifications into implementations!
- How can we check that we meet real-time constraints?
- How do we validate embedded real-time software? (large volumes of data, testing may be safety-critical)

# It is not sufficient to consider ES just as a special case of software engineering

**EE knowledge must be available,**
**Walls between EE and CS must be torn down**



# Summary

- Definition of embedded systems
- Application areas
- Examples
- Curriculum
- Growing importance of embedded systems
- Characteristics
  - Reliability
- Challenges in embedded system design