Embedded Networks

Jörg Kaiser

Institut for Distributed Systems (IVS)
Embedded systems and Operating Systems (EOS)

Summer Term 2012

Organization

Lecture: Prof. Dr. Jörg Kaiser
Institut für Verteilte Systeme (IVS)
Arbeitsgruppe Eingebettete Systeme und Betriebssysteme
kaiser@ivs.cs.uni-magdeburg.de

Secretary: Petra Duckstein / Dagmar Dörge
Bu29 / Room 405
duckstein@ivs.cs.uni-magdeburg.de

Exercises: Christoph Steup
Institut for Distributed Systems (IVS)
Department Embedded Systems and Operating Systems
steup@ivs.cs.uni-magdeburg.de

Organization

Lectures:	Tuesday	9:00 - 11:00	G29-E037
Exercises:	Wednesday	11.00 - 13:00	G29-334
	Wednesday	13:00 - 15:00	G29-334

Requirements: Need: Vordiplom, Bachelor
Nice: VL Betriebssysteme 1,
VL Technische Informatik II,
VL Embedded Systems.

Creditpoints: 6 ECTS

Successful participation: Exercises, Exam

Course Category: Informatik II and III

Organization

- Exercises: Infos on the web.
- Slides on the web

  http://ivs.cs.uni-magdeburg.de/eos/lehre/SS2011/vl_en/
- infos also available via UNIVIS

Participants must register on the web-page :

http://eos.cs.uni-magdeburg.de/register/

Literature:

Paulo Veríssimo, Luís Rodrigues:
Distributed Systems for System Architects
Kluwer Academic Publishers, Boston, January 2001

Hermann Kopetz:
Distributed Real-Time Systems
Kluwer Academic Publishers, 1997

Konrad Etschberger:
CAN - Controller Area Network, Grundlagen, Protokolle, Bausteine, Anwendungen
Carl Hanser Verlag, München, Wien, 1994

Sape Mullender (Hrsg.):
Distributed Systems
ACM Press, 1989

Further literature will be provided during the course.

On-line Documentation:

CAN:	http://www.can-cia.de
Profibus:	http://profibus.com/downloads.html
FIP:	http://worldfip.org/downloads
LON:	http://echelon.com

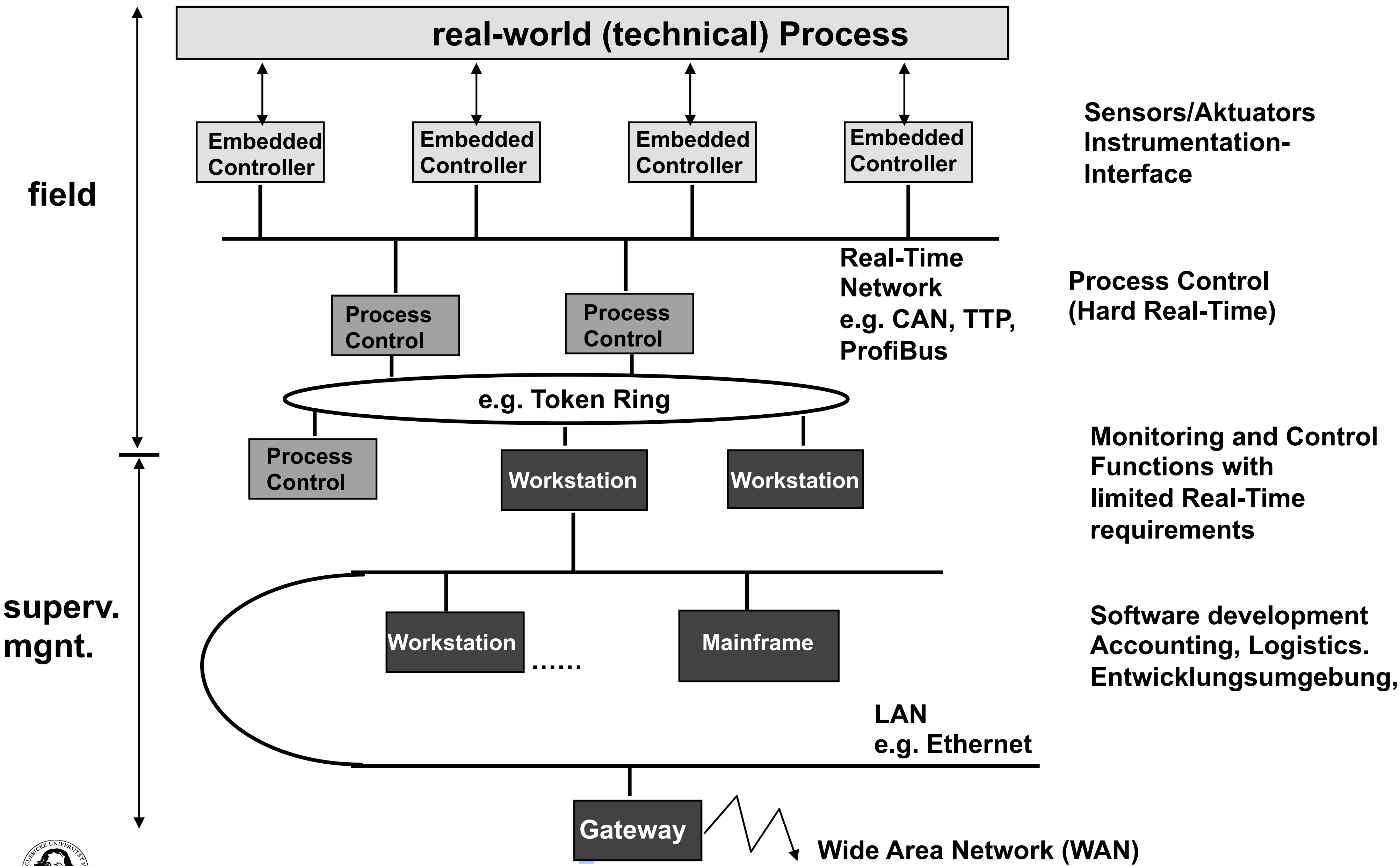
Embedded Networks
or
Communication networks to monitor and control the physical environment

Application Areas for Embedded Networks

- **Industrial Automation**
- **Automotive**
- **Buildings**
- **Mechanical Engineering**

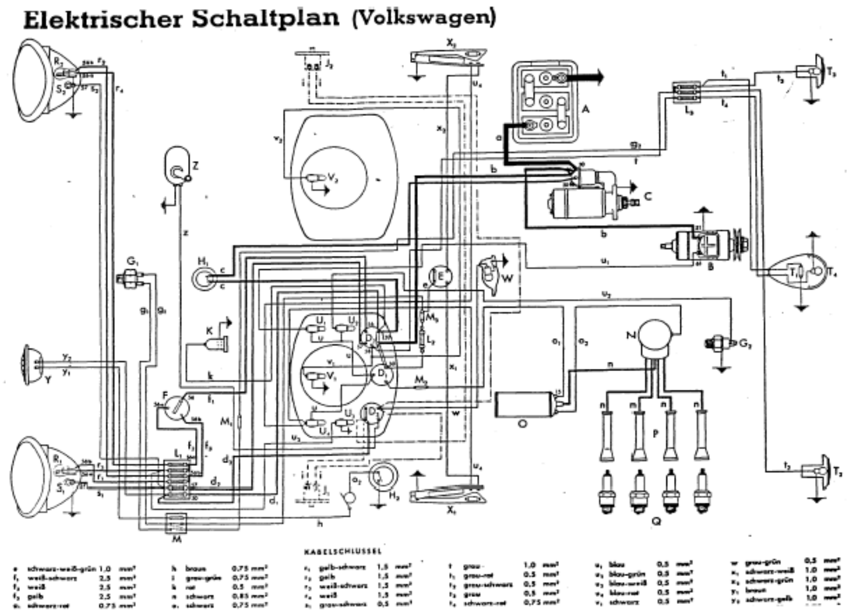
⇨ **The Network is the Computer !**

Embedded Networks in a CIM environment



Controlling a Car

Yesterday

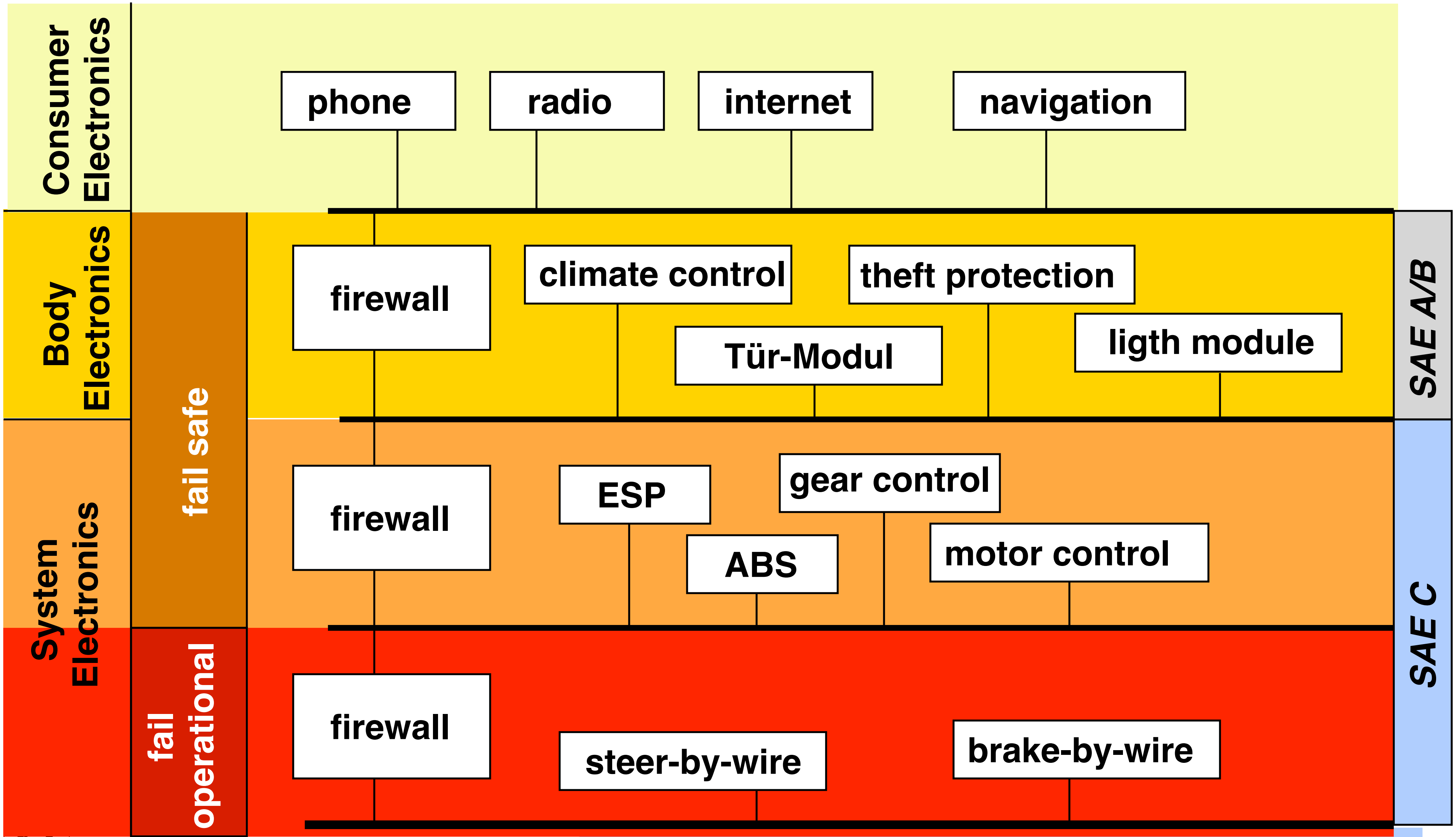


Today

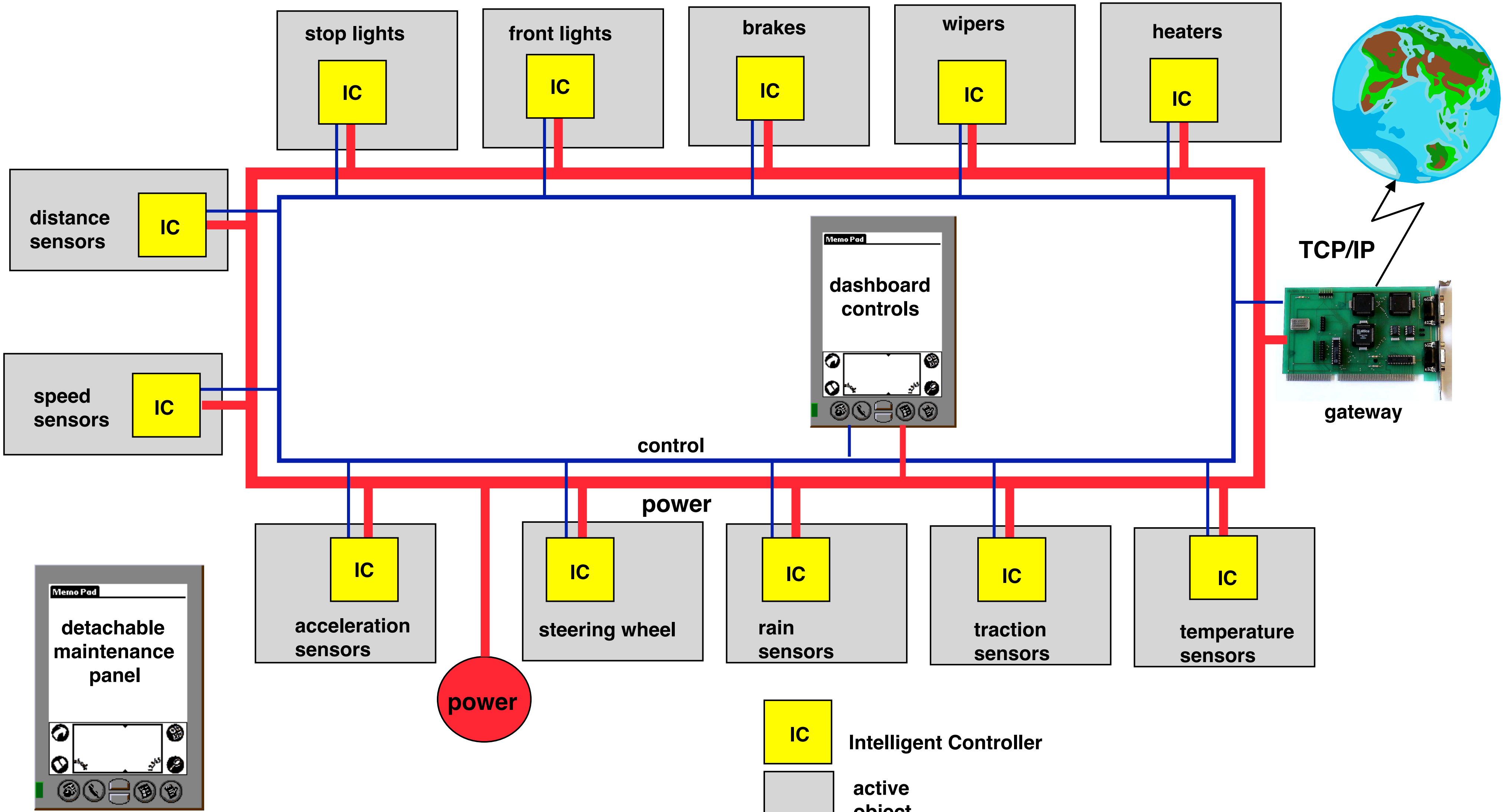
- **11.136 electrical parts**
- **61 ECUs**
- **Optical bus for information and entertainment**
- **Sub networks based on proprietary serial bus**
- **35 ECUs connected to 3 CAN-Busses**
- **2500 signals in 250 CAN messges**

Levels of Communication in a CAR

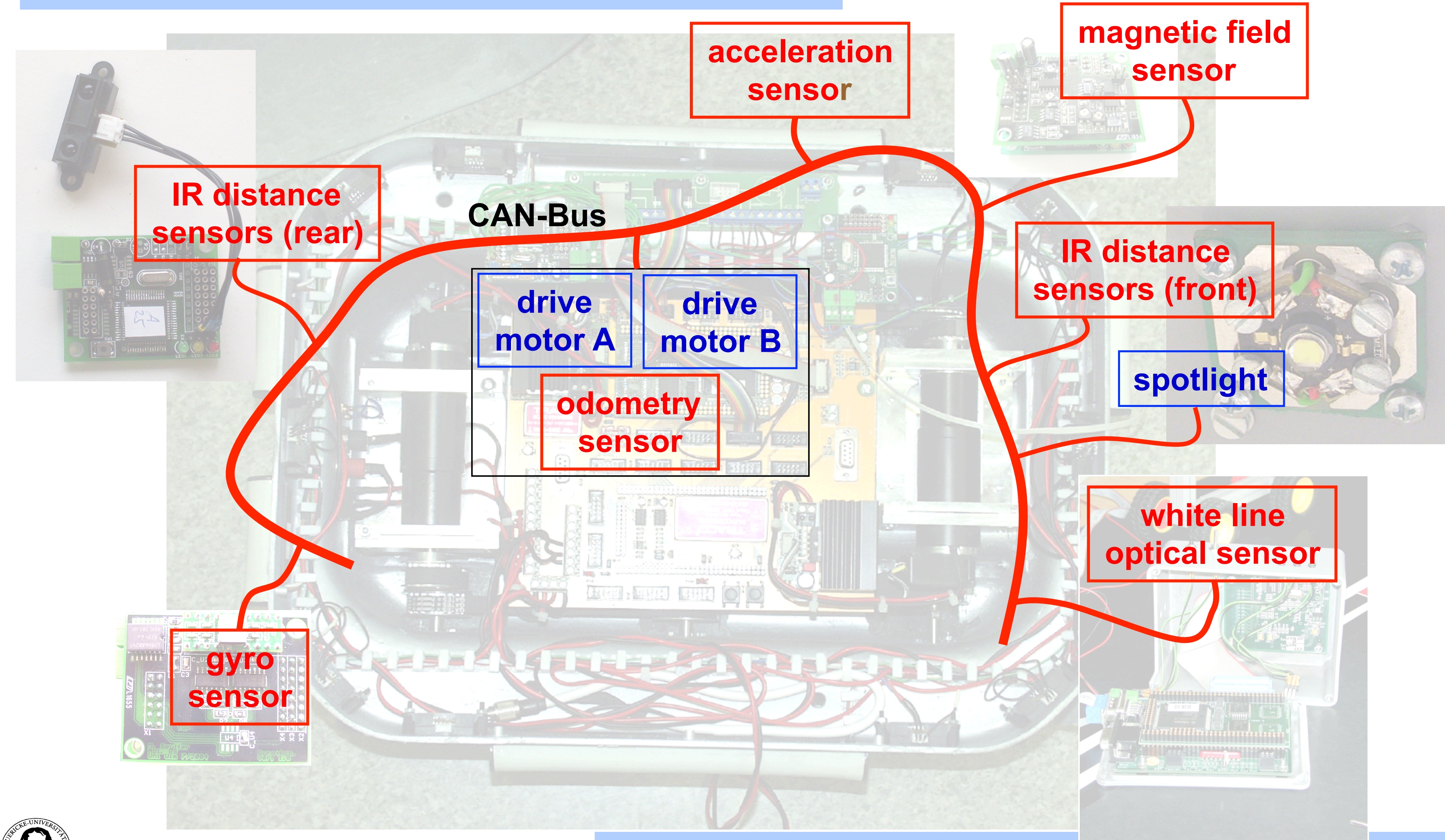
T. Führer, B. Müller, W. Dieterle, F. Hartwich, R. Hugel, M. Walther:
„Time Triggered Communication on CAN“



Future: Distributed Cooperative Control



Distributed Control with Co-operating Smart Components



Requirement:
Predictability of Communication !

Sources of Unpredictability ?

Sources of Unpredictability

Network is a shared medium

→ Arbitration, Collisions

Sender and Receiver must run in Sync

→ bounded buffers, lost messages

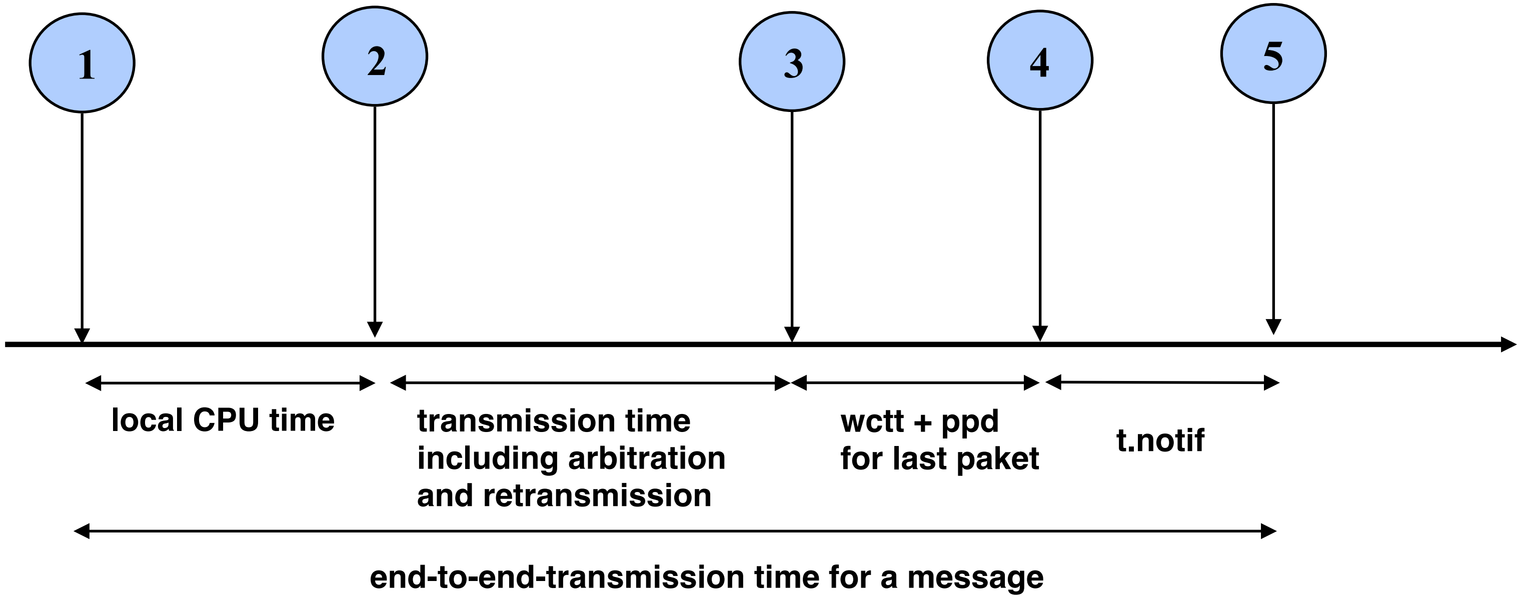
Failures, faults, errors

→ re -configuration, -covery, -send

Requirements for a predictable communication system

- **bounded, predictable transmission times**
- **execution time for protocol stack is bounded and small**
- **variations of the execution time (Delay Jitter) is small**
- **error detection in sender and receiver**
- **error detection with minimal latency**
- **no thrashing under high load conditions (constant throughput)**
- **support for multicast communication**
- **support for many-to-many communication**
- **Composability**

End-to-End communication costs



1. Send-task becomes ready

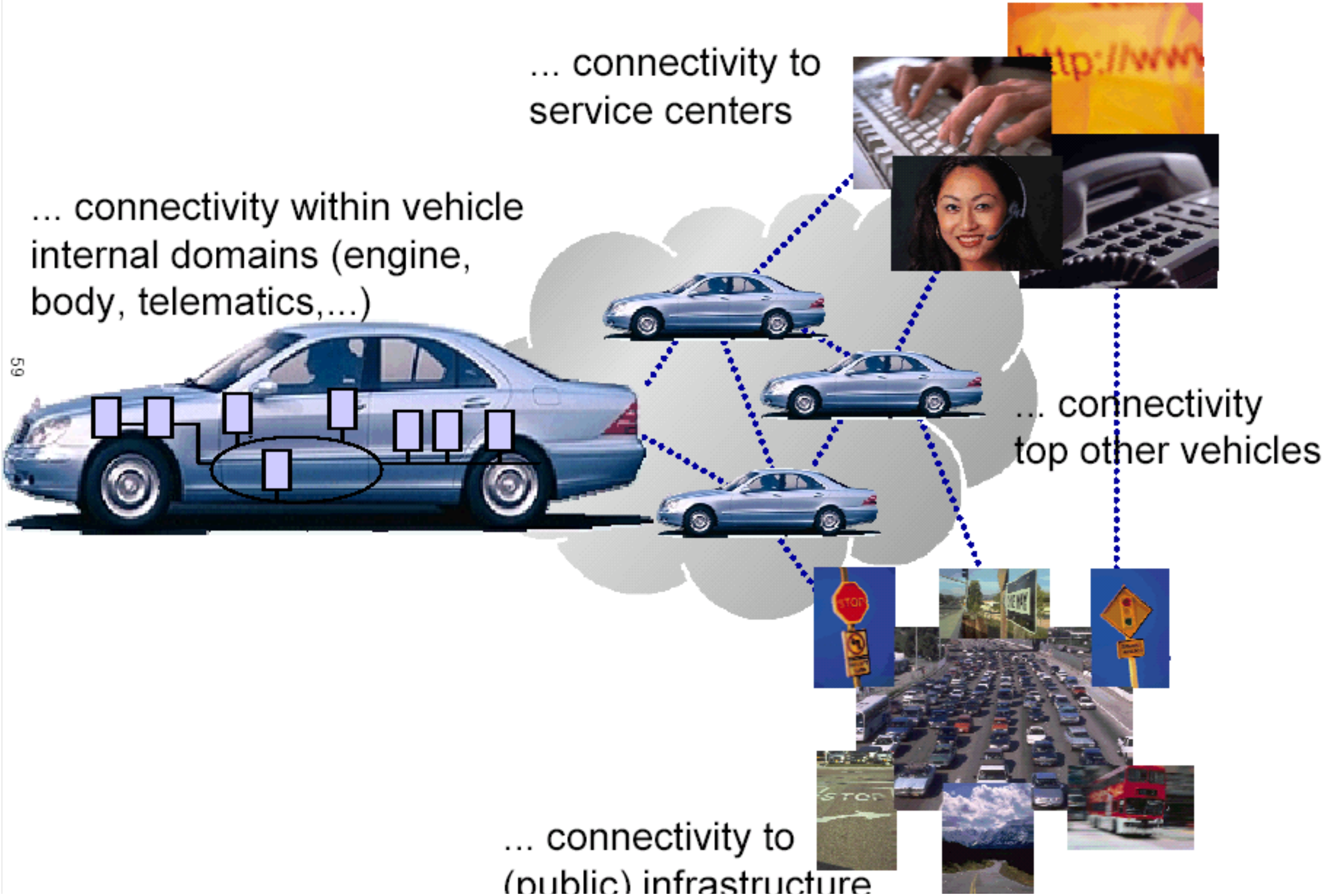
2. Latest point in time when the message is in the ordered transmission queue (OQ).

3. All pakets of message m in OQ are put to the network medium.
   Transmission of last paket starts.
   wctt: worst case transmit time
   ppd: physical propagation delay

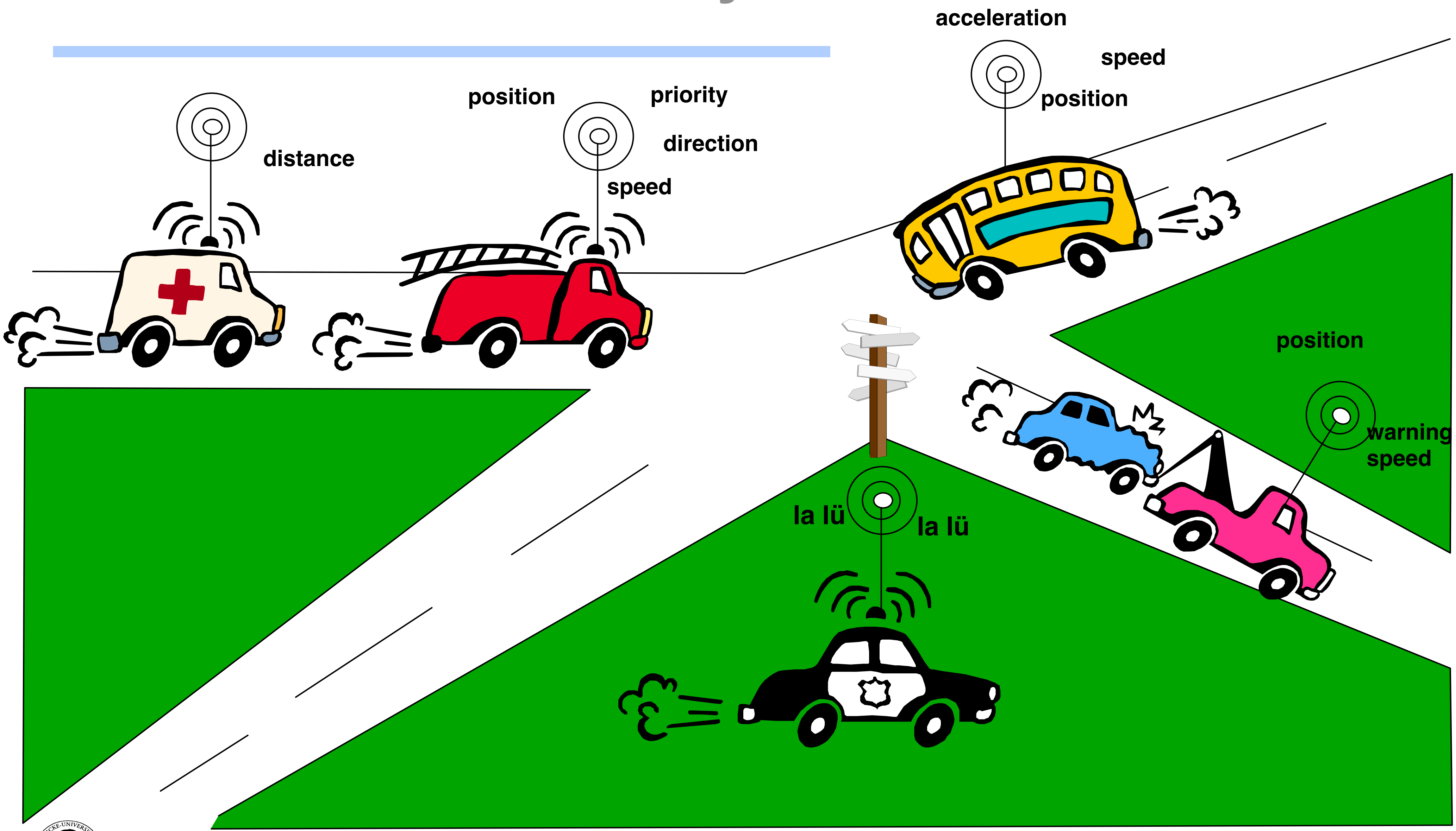
4. Last paket of m reaches the Communication Controller of receipient.

5. "Paket received" interrupt is triggered.
   t.notif: worst case delay between successful reception of the paket (in the CC) and notification of the task.
   Receive task will become ready at this time instant..

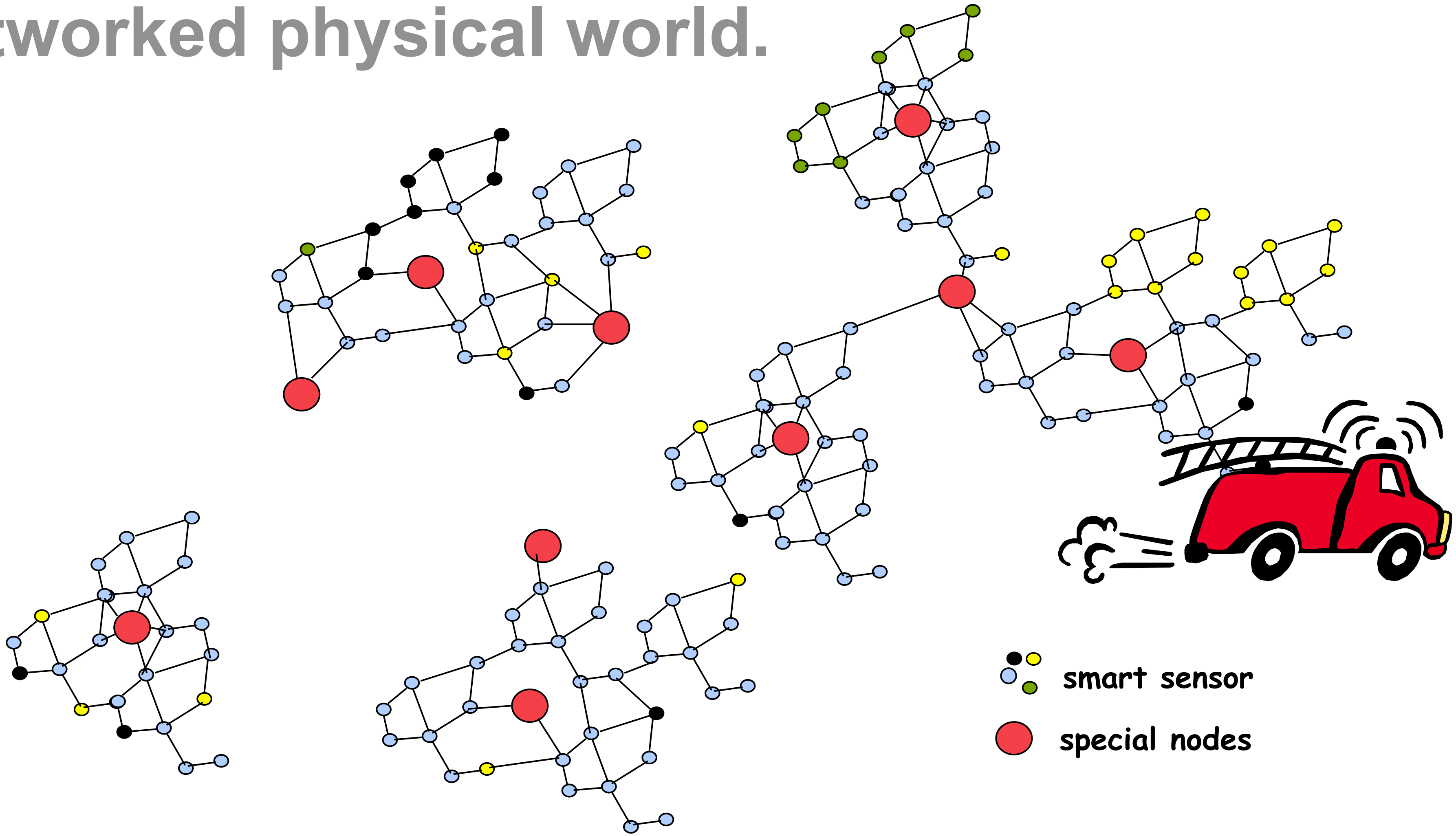
Vehicles become a means of communication with ...



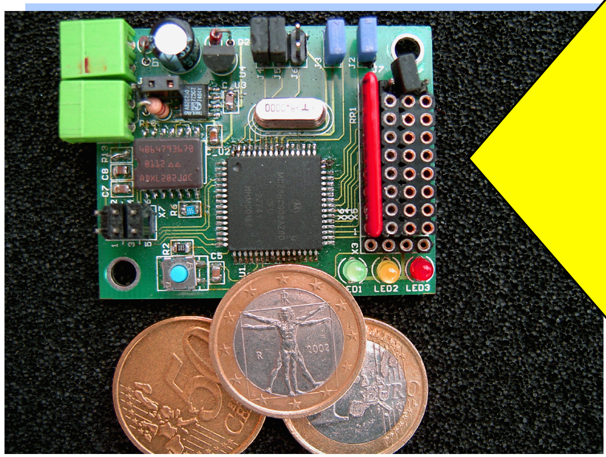
Autonomous sentient systems



"Embedded Everywhere": A networked physical world.

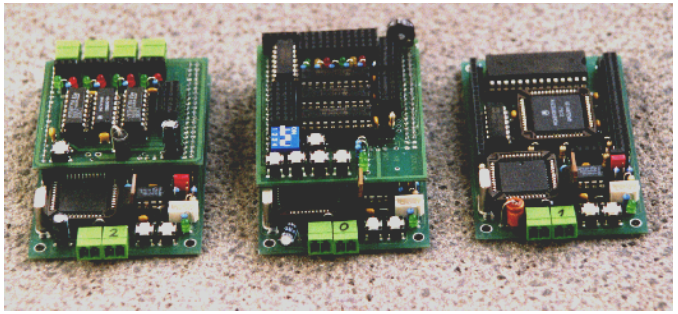


Hardware for Sensornets "Smart Dust"

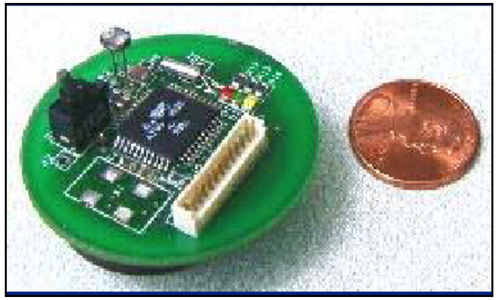


Developed Sensors at CORE
- **infrared motion detector**
- **infrared distance sensor**
- **acceleration sensor**
- **embedded gyro**
- **weather station**
- **magnetic field detector**
- **in-house location system**

tiny-board, CORE, Ulm

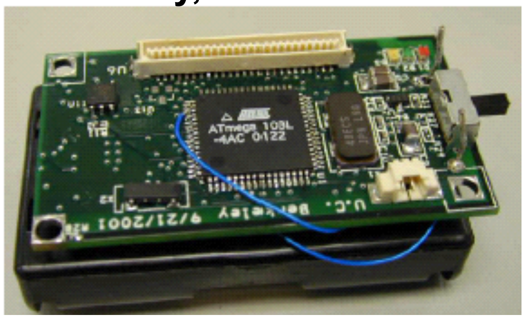


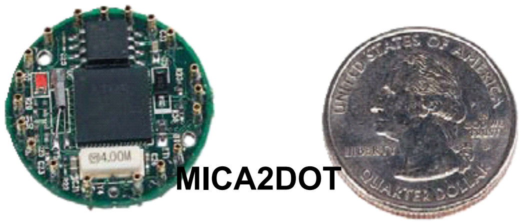
68HC11 CAN-Sensor Boards, CORE, Ulm



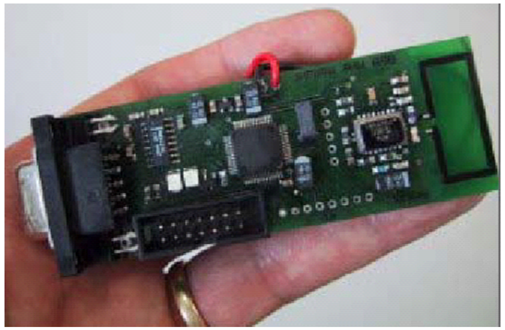
WeC „Smart Rock“ UCB

a mica mote,
Berkeley, Crossbow



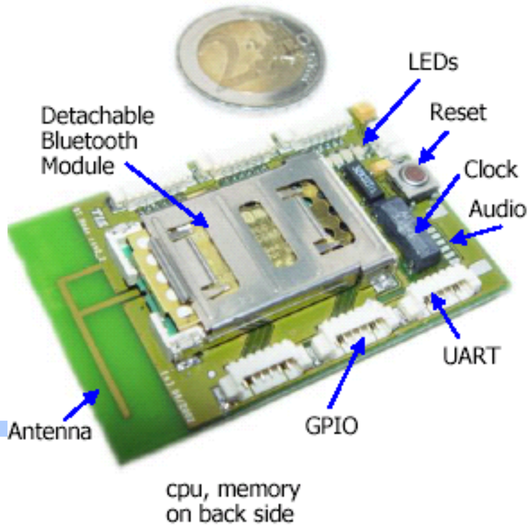


MICA2DOT

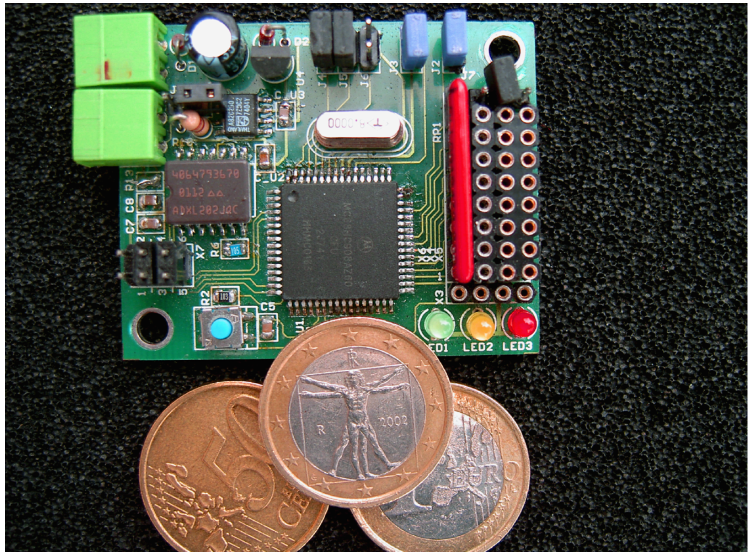


The EYES prototype

Smart-its: ETH Zurich,



Tiny Properties



Designed for experimentation:
Basic Board + Piggyback extension

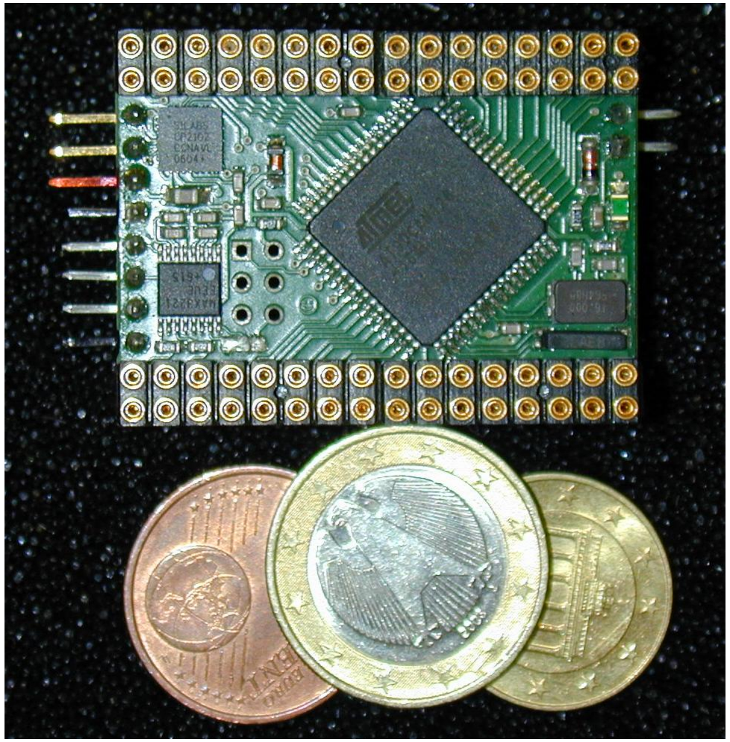
Basic board:

- Processor 68HC908AZ60 (60k Flash, 2k RAM)
- Power regulator (linear or switched) 6-14 V
- LEDs for checks, configuration jumpers
- CAN-Bus Network Interface
- Sockets for AD, C&C, digital I/O
- Sockets for asynch.and synch serial comms.

Power consumption:

Processor ~ 250 mW @ 16MHz
Radio link (Easy Radio, 19kbit/sec): ~150mW(transmit), ~75mW(idle)
9V Block (565 mAh): ~ 8h@continuous operation, ~30 days@10ms/sec

AVR Properties



Designed for experimentation:
Basic Board + Piggyback extension

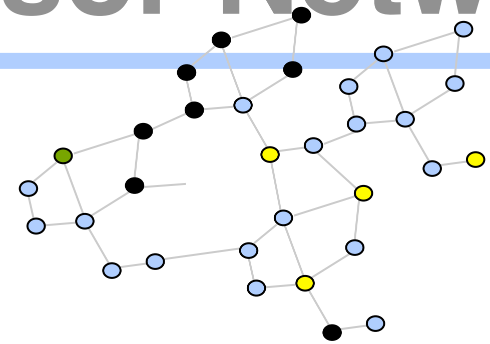
Basic board:

- Processor Atmel AVR AT90CAN128
  (128k Flash, 4k RAM)
- Power regulator (linear or switched) 2.7-24 V
- LEDs for checks, configuration jumpers
- CAN-Bus Network Interface
- Sockets for AD, C&C, digital I/O
- Sockets for asynch. and synch serial comms.

Power consumption:

- Processor ~ 160 mW @ 16MHz and 5V
- Radio link (802.15.4, 250Kbps)
- 9V Block (565 mAh): ~17,5h@continuous operation, ~70 days@10ms/sec

Sensor Networks



Components:

- heterogeneous Sensors
- stationary and mobile entities
- very large number of components
- through away product (in the true sens of the word)
- life time = battery life time
- constraints in performance and memory

Behaviour:

- sponteneous behaviour
- not always active
- division of labour required

Network:

- bandwidth constraints
- Multi-hop
- Aging of information
- Quality of dissemination

Sensor Networks

- wireless communication with low bandwidth
- only a few standards for dedicated applications
- alternation of sleep and active times is a challenge for MAC protocols
- inherently multi-hop
- address- , contents- und location-based routing

Embedded networks: Fieldbusses vs. sensornets

common properties:

- communicate information to perceive and control the physical environment,
- transfered information is subject to aging,
- meeting indivudual timing constraints is more important than throughput,
- considers trade-offs concerning energy consumption, bandwidth, reliability and priority of message traffic.

major differences:

	fieldbusses	sensornets
number of nodes	low to moderate	very large (in theory)
safety	very high to moderate	low
predictability	very high	low to moderate
number of hops	1 to few	many
indiv. failure probability	very high to moderate	very low

Embedded Networks

- o Introduction
- o Dependability and fault-tolerance
  - * Attributes and measures of Dependability
  - * Basic techniques of Fault-Tolerance
- o Time, Order and Clock synchronization
- o The physical network layer
- o Protocols for timely and reliable communication
  - * Introduction, problem analysis and categories
  - * Industrial Automation & Automotive Networks
    - * Industrial Ethernet, Interbus-S, ProfiBus, WorldFip,
    - * Controller Area Network (CAN-Bus)
    - * Time Triggered Protokolls (TTP/C, FlexRay)
    - * Real-Time CSMA-Networks (Byteflight, VTCSMA)
    - * Timed Token protocol, Braided Ring
- o Sensornets
  - * Requirements for sensor nets
  - * Protokols for wireless communication
  - * Energy-efficient MAC-protocols

Research in EOS

System Software for Distributed Sensor Actuator Systems

- **Protokolls for wired and wireless embedded networks**
- **Component-oriented middleware for resource-constraint systems**
- **Advanced conceps for dependable, distributed sensor-actor systems**
- **Dynamic sensor fusion**
- **Mobile co-operating robots**
- **Model-based software development**
- **Programming embedded systems**