Embedded systems engineering
Distributed real-time systems

David Kendall

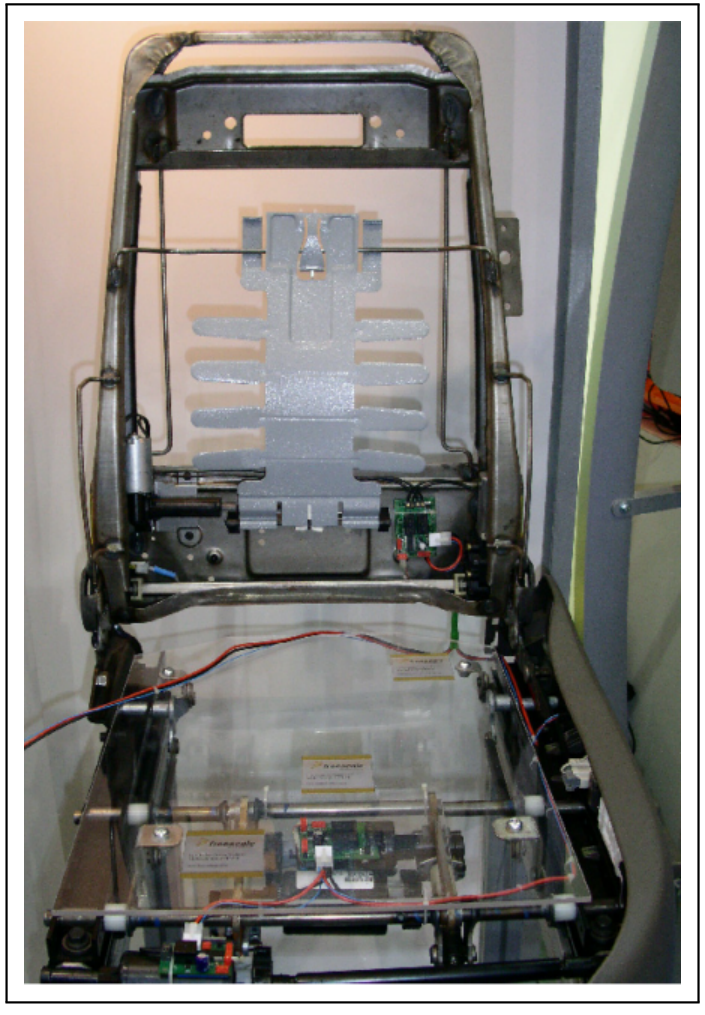
Introduction

- . . . to embedded systems
  - which are *distributed* and *real-time*
- . . . to engineering
- . . . to embedded systems engineering
- . . . to the module

Embedded systems are everywhere

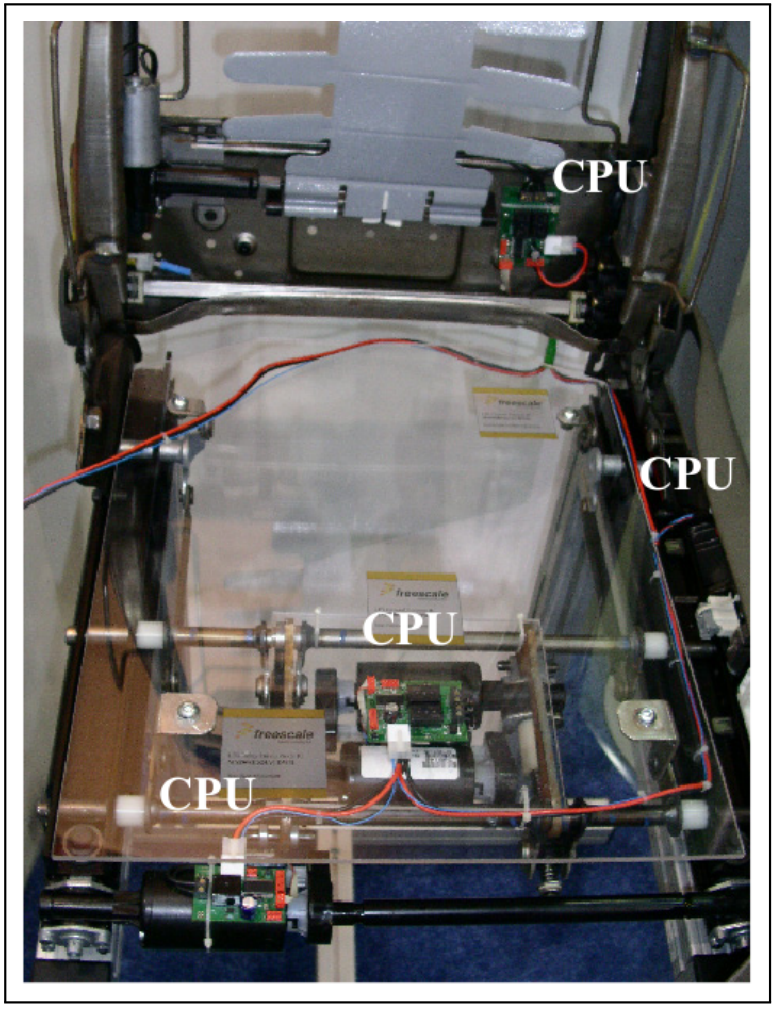
How many CPUs in a car seat?

- Photo from Convergence 2004
  - automotive electronics show



How many CPUs in a car seat?

- Low speed LIN network to connect seat motion control nodes
- This is a distributed embedded system
- CPUs
  - Front-back motion
  - Seat tilt motion
  - Lumbar support
  - Control button interface



How many CPUs in a car?

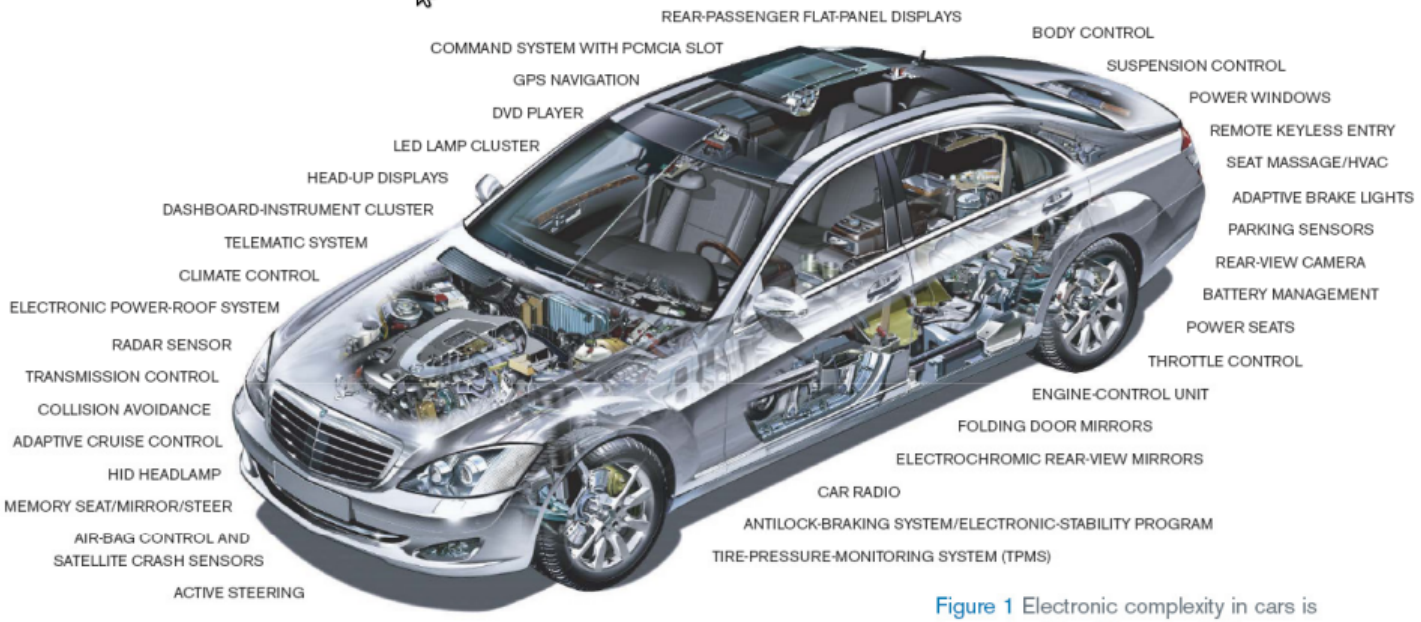
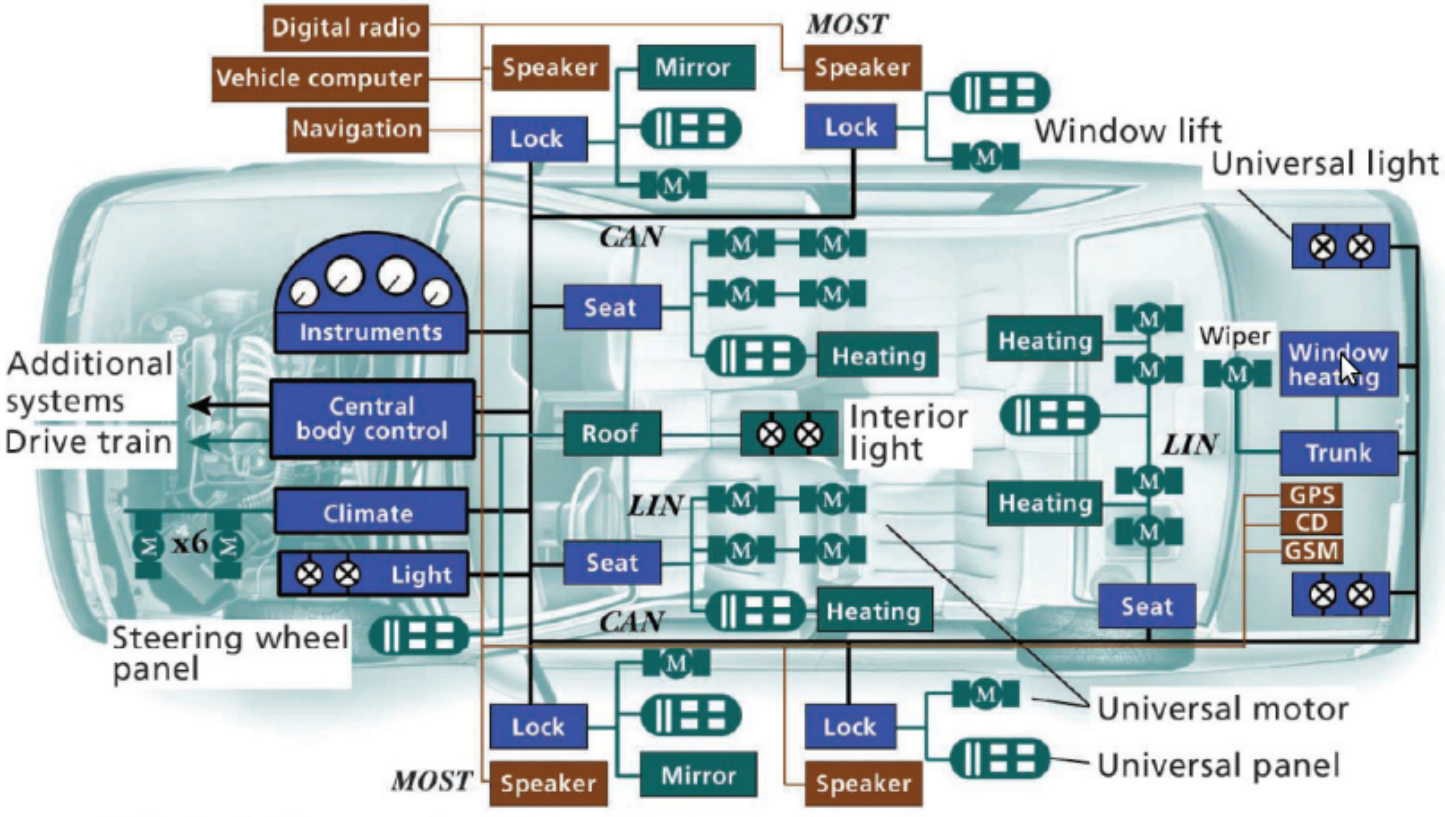


Figure 1 Electronic complexity in cars is increasing. New Mercedes S-Class cars employ at least 70 networked ECUs (electronic control units); 10 years ago, most cars had three ECUs (photo courtesy of DaimlerChrylser; source: Gartner Research, November 2005).

Car as a distributed embedded system

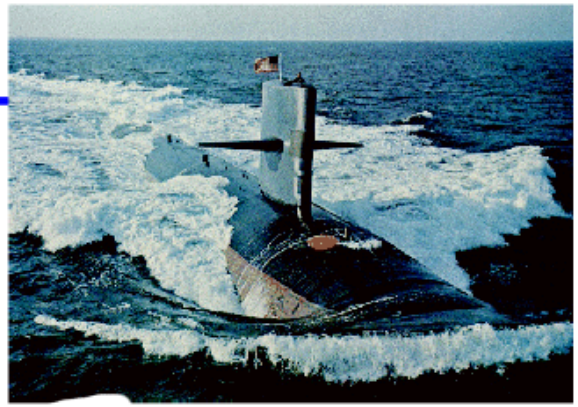


CAN Controller area network
GPS Global Positioning System
GSM Global System for Mobile Communications
LIN Local interconnect network
MOST Media-oriented systems transport

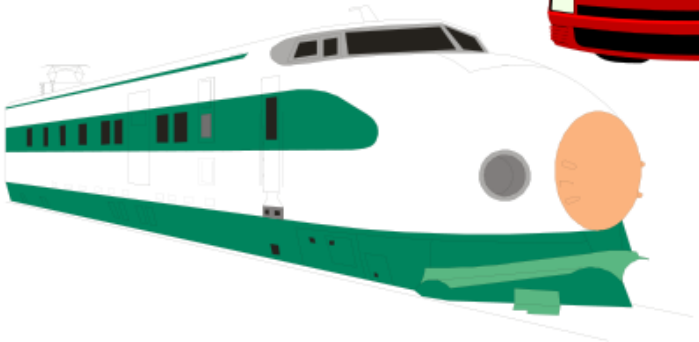
[Leen02]

Embedded systems






Embedded Systems - Computers Monitoring/Controlling their environment


Embedded systems characterstics

- concurrent – composed of multi-tasking and/or distributed processes
- communicating – specialized processes communicate in order to achieve some overall system function
- real-time – timing requirements are established by the environment
- resource-constrained – limited resources: processing, memory, peripherals, power, . . .

Engineering

The American Engineers' Council for Professional Development defines "engineering" as:

> *[T]he creative application of scientific principles to design or develop structures, machines, apparatus, or manufacturing processes, or works utilizing them singly or in combination; or to construct or operate the same with full cognizance of their design; or to forecast their behavior under specific operating conditions; all as respects an intended function, economics of operation and safety to life and property.*

Embedded systems engineering

Application of scientific principles to

- the design of embedded systems
- the construction of embedded systems with “full cognizance” of their design
- the forecasting of the behaviour of embedded systems under specific operating conditions

“all as respects an intended function, economics of operation and safety to life and property”

Embedded systems engineering

Application of scientific principles to design, construction and analysis of systems that are

- concurrent,
- communicating,
- real-time and
- resource-constrained

Concurrency

- arises when multiple processes (tasks) share a single processor
- allocation of the processor to a different task can be
  - event-triggered
    - response to any of many possible interrupts: periodic timer overflow, the arrival of a message on a CAN bus, the pressing of a switch, the completion of an A/D conversion, ...
  - time-triggered
    - response to one source of interrupt only: usually a periodic timer
- managing and reasoning about concurrency is a major challenge for the embedded systems engineer

Communication

- Computing nodes that communicate with each other form a distributed system
- Economic and safe allocation of resources in a distributed system requires a predictable communication network
- Predictable communication networks include
  - CAN, TTCAN
  - TTEthernet

Real-time

- A real-time system is a system where the total correctness of an operation depends not just on the logical correctness of the result but also on its temporal correctness i.e. the time at which it is produced.
- A deadline specifies the time by which an operation must complete and deliver its result.
- A hard real-time system is a system that is considered useless if an operation misses a single deadline.
- A soft real-time system tries to meet its deadlines but can tolerate an occasional missed deadline, perhaps giving reduced service quality.

Introduction to the module

- View the module home page

Summary of the structure

- Uni-processor solutions
  - Time-triggered
  - Event-triggered
- Distributed solutions
  - Networks for embedded systems
    - Focus on Controller Area Network (CAN)
  - Analysing networked embedded systems
    - Distributed Response Time Analysis
- Other topics
  - C programming for embedded systems
  - Methods, tools, standards

Acknowledgements

- Philip Koopman
  http://www.ece.cmu.edu/ ece649/lectures/01_intro.pdf