# Real Time Systems

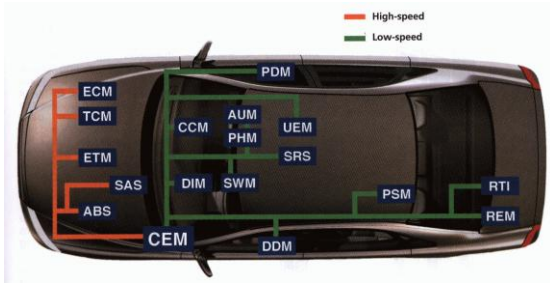
Wang Yi, "google Wang Yi"

What are "Real-Time Systems"?

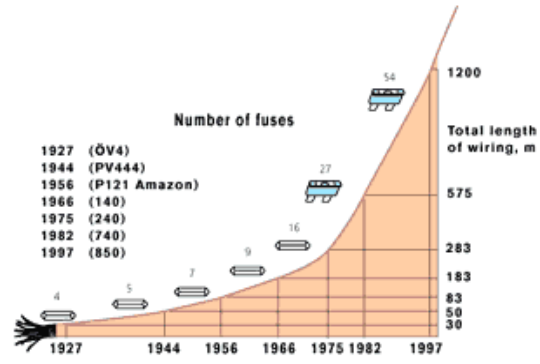
Course Information

## An Example Real-Time System

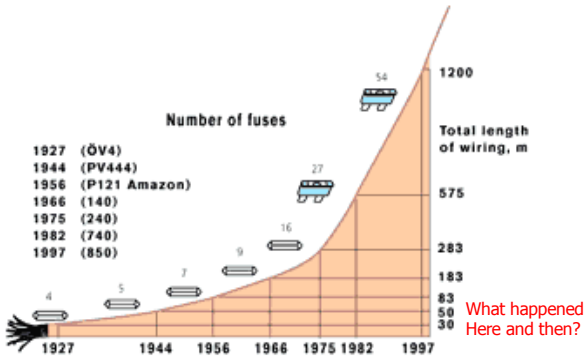
65-70 ECU's/micro-processors in the newest S80
Upto 30-40% of the total cost of a car: control system




The Evolution of Automotive Electronics

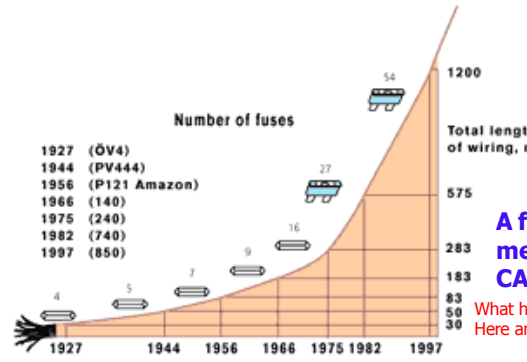



The Evolution of Automotive Electronics




What happened
Here and then?

The Evolution of Automotive Electronics

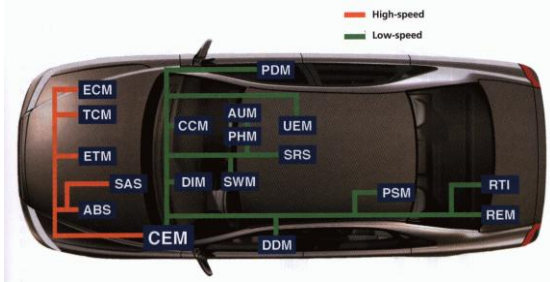



**A few meters CAN-bus!**

What happened
Here and then?

## A Real-Time System

65-70 ECU's/micro-processors in the newest S80
Upto 30-40% of the total cost of a car = control system




## Embedded Real-Time Systems

Consume 99% of all CPUs in the world!

**Computers that do not look like a computer**

- **Safety-critical: Reliability = must!**
- **Mass production: a bug = milions of dollars**

## Towards a "standard" definition of RTS

- A real-time system is any information processing system which has to respond to externally generated input stimuli within a finite and specified period
  - the correctness depends not only on the logical result but also the time it was delivered
  - Failure to respond in time is as bad as the wrong response !
- The computer is a component in a larger engineering system => EMBEDDED COMPUTER SYSTEM

## Remember:

In RT systems, the correctness of computation depends not only on the results but also on the times when outputs are produced.

- Real Time =/= Fast
- Real Time =/= Time Sharing
- Real Time = just in Time (predictable)

## Main Goal of this course

Study Techniques for constructing
Real-Time Software with
predictable response times

## Further details …

- To understand the basic requirements of real-time systems, and how to program such systems so that the requirements are met.
- To understand how these requirements have influenced the design of real-time programming languages and real-time operating systems.
- To understand the implementation and analysis techniques which enable the requirements to be realized.

## Prerequisites

- Basic understanding of C
- Basic understanding of Computer Architectures.
- Basic understanding of Operating Systems

## Course Form

- Lectures
- Programming assignments (Ada, C, OS kernel)
- Playing with Legos!

- Examination
  - 4 assignments and
  - final written exam (week 43: 5 hours)

## Software and Lab assignments

- Real Time Programming (Ada)
- Programming with OS Kernel (LegOS)
- Response Time Analysis (FpsCal)
- Modeling and Analysis (UPPAAL/TIMES)

## Literature

- Real Time Systems, J.W. Liu 2000
- On-line materials (appear in real-time ☺)

- Further readings:
  - Real-Time Systems and Programming Languages, Alan Burns and Andy Wellings, Addison Wesley, 2001.
  - Hard Real Time Computing Systems - Predictable Scheduling Algorithms and Applications, Giorgio Buttazzo, Springer, 2005.

## People to help you!

- Lecturer:
  - Wang Yi, office: 1235, yi@it.uu.se

- Assistants:
  - Pavel Krcal, office 1217, pavelk@it.uu.se
  - Simon Tschirner, office 1218, simon.tschirner@it.uu.se

## Course Outline (lectures)

- Introduction
  - Characteristics of RTS
- Real Time Operating Systems (RTOS)
  - OS support: scheduling, resource handling
- Real Time Programming Languages
  - Language support, e.g. Ada tasking
- Scheduling and Timing Analysis of RT Software
  - Worst-case execution and response time analysis
- Design and Validation
  - Modeling, Verification and Testing
- Reliability and Fault-Tolerance
  - Fault tolerant, failure recovery, exception handling
- Distributed real time systems
  - Real Time Communication: CAN Bus