Real Time Systems -- 5&10hp
Wang Yi
user.it.uu.se/~yi

Course Information

People to help you!

- Lecturers:
  - Martin Stigge, office 1236, martin.stigge@it.uu.se
  - Wang Yi, office: 1235, Wang.Yi@it.uu.se

- Assistant:
  - Xiaoyue Pan, office 1246, xiaoyue.pan@it.uu.se

Webpage
http://www.it.uu.se/edu/course/homepage/realtid/ht11
Literature

- On-line materials (to appear in real-time 😊)
  - Notes (pdf) by Hanssoon and Tindell
- Further readings:

What are “Real-Time Systems”?
A Real-Time System

65-70 ECUs/micro-processors in some model of S80

Embedded Real-Time Systems

Consume 98% of all CPUs in the world!

Computers that do not look like a computer

- The main reason of buying is not for “information processing”, but “control”
- Majority of these systems must reply ... timely
- Mass production: a single bug = millions of dollars
Where are ESs?

Embedded computing systems are becoming pervasive in our society (more than $10^9$ units/year):

- Robotics
- Flight control systems
- Plant control
- Automotive
- Consumer electronics
- Multimedia systems
- Sensor/Actor Networks
People say …

Real-Time vs. Embedded

- “Real-Time” is a characteristic property of ESs
- A large class of ESs are “Hard Real-Time Systems”
- The others are “Soft Real-Time Systems”
What are "Real-Time Systems"?

Real time systems will guarantee to give a result within a specified time --- Wiki

- 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.

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 $\neq$ Fast
- Real Time $\neq$ 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 Architecture.
- Basic understanding of Operating Systems

Course Form

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

- Examination
  - 4 assignments and
  - final written exam (October 25: 5 hours)
Course Outline (lectures)

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

Lab assignments & Software (blue=5hp)

- Real Time Programming (Ada)
- Response Time Analysis (FpsCal)
- Real-Time OS Kernel (C, Lego NXT/OSEK)
- Modeling and Analysis (UPPAAL)
M.Sc Program in Embedded System: Overview

Ph.D. Program in Embedded System

5 years (4 years full time + 20% teaching duties)
- 60HP courses
- 180HP thesis work

Example Topics:
- Multi-core embedded systems
- Resource partition, application mapping and isolation
- Multiprocessor scheduling
- RT operating systems
- Avionics, Robotics, Real-time tracking
- Timing analysis of embedded software
- Modeling and verification