Distributed Real-Time Systems

Tony Larsson
(Course, Lectures and Seminars)

Edison Pignaton de Freitas
(Labs and Project Work Assistance)

Web adress:
http://www2.hh.se/staff/tola/drts/index.html

Course Plan

- 7 lectures
- 7 discussion/presentation seminars
- 3 lab exercises
- Read and discuss scientific papers in seminars
- In a group of X persons present one topic
- Written examination

After the course you shall:

- Understand the principles of real-time systems with focus on task:
  - scheduling,
  - allocation and
  - distribution.
- Be able to describe how real-time tasks can share a processor over time
- Understand how tasks in a distributed system should be distributed over its computer nodes
- Have experience in discussing and presenting scientific results in the area

Examination

- Active participation and presentation in seminars (with quiz for training) (80% participation required)
- Lab/simulation exercises (pass)
- Written (graded) examination (based on text book, seminar papers and lectures)

Seminars

- All participants should read a scientific paper related to the topic of the seminar
- At each seminar one group of 3-4 students present their findings and understanding of the topic area for the seminar
- The group that present a topic must investigate other related reference material

Simulation labs

- To get some familiarity with simulation three lab/exercises will be performed
- One is devoted to introducing the use of Matlab and TORSCHE for timing analysis
- The other two are related to scheduling of tasks on one processor and/or cooperation between tasks on different processors

Course Book

Kopetz, Hermann., “*Real-Time Systems - Design Principles for Distributed Embedded Applications”***,** Kluwer Academic Publishers, Boston, Massachusetts, 1997.

Available as e-book via the library (ebrary).

This book contain 14 chapters.

Read two chapters each week.

Complementary Books

Simon D. E. , “An Embedded Software Primer”, Pearson Education, 1999. Hands on oriented primer

Lewis Daniel W., “Fundamentals of EMBEDDED SOFTWARE – Where C and Assembly Meet”, Prentice Hall, 2001. A more complete hands on oriented book

Shaw Alan C., “Real-Time Systems and Software”, Wiley, 2001. More theoretical but rather easy to read book

Murthy Siva Ram C and Manimaran G., “Resource Management in Real-Time Systems and Networks”, MIT Press, 2001. A book that also covers network aspects

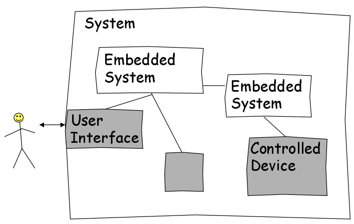
Buttazzo G.C., “HARD REAL-TIME COMPUTING SYSTEMS”, Kluwer Academic Publishers 1997. Theoretical, easy to read book that focuses on scheduling

Burns A. and Wellings A., “Real-Time Systems and Programming Languages”, Addison-Wesley, 2001. Focus on Real-Time programming

Colling Jim, “SOFTWARE ENGINEERING for REAL-TIME SYSTEMS”, Addison Wesley, 2003. A broad system design oriented approach

Embedded System

- Part of external system or intelligent product (to control it)
- Hidden or invisible to user of system
- The user must see a product not a computer



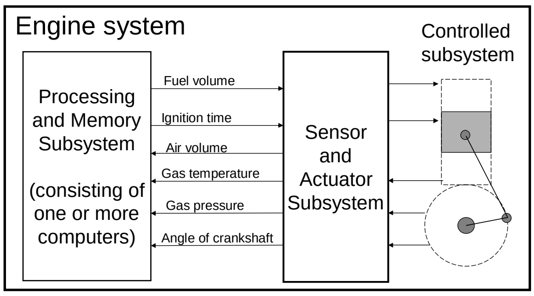
Program handled control of tasks

- Program driven control of physical system
- Computer programs implement tasks
- The control of a physical system can be divided in several concurrent tasks
- Tasks must be activated and performed at regular intervals or at specific events

More on types of tasks

- Tasks may run on different computer nodes
- Some tasks run uninterrupted and without pause until they finish
- Other tasks are able to take a break and wait for other events to happen
- Yet other tasks can be interrupted (preempted) by other more important tasks

Embedded real-time system

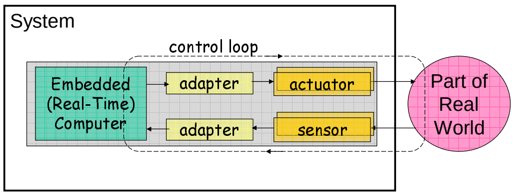


What is it about? – a few examples

- Tasks must be performed correct including within specified time limits (deadlines)
- Scheduling between tasks must be feasible
- Worst case execution times (WCET) must be estimated
- Tasks have to be prioritized
- Low priority tasks wait on higher priority tasks while they are active

Embedded Real-Time System

- Interacts with the real-world
- Provides additional value to the system



Embedded System

An embedded system is a computer system used inside a system to contribute to this other systems functions and characteristics.

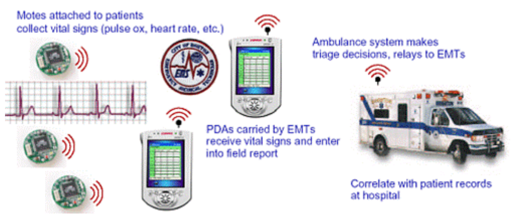
Example: The computer system used in a camera, washing machine or elevator.

What functions or characteristics does such a computer add to a camera or a washing machine?

How are cooperating ES useful

- Improved burning process in combustion engines – for reduced pollution and energy consumption
- Coordinated elevator, rail and road traffic systems – reduce delays, pollution, energy consumption and disturbing noise
- Patients can be monitored and supported in ways that enables them to stay at home

The CodeBlue Project



Wireless Sensor Networks for Medical Care Source: Harvard University

Real-Time System

A computer system that works in a real-time environment and does its tasks not only correct but also within required time limits.

Example: the computer used in an airplane, engine or traffic control system.

Comment: To be meaningful a real-time system must be dependable also in other respects (than timeliness).

Distributed System

A computer system where tasks are:

- allocated on more than one processor node
- run close to sensors or actuators
- communicate and cooperate to perform common system level tasks

Dependable System

A dependable system is:

- Reliable
- Available
- Safe
- Secure
- Maintainable

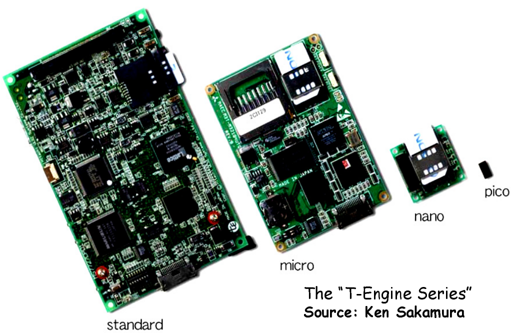
Background

- Since around the 70's cheap microprocessors & controllers, A/D and D/A converters, operational amplifiers, PLC and PID controllers and DSPs have been important application enablers
- Long before one made real-time systems by use of:
  - rope, sticks and stones,
  - water magazines, floats and locks,
  - mechanics (moment wheels, centrifugal regulators, ...),
  - pneumatics (air accumulators, pumps, motors & valves),
  - hydraulics (pumps, motors and valves)
  - electro-mechanics, e.g. relays and step motors
- We also enabled distributed systems using: sound, light, flags, semaphores, letter and Morse alphabets

Enabling technologies

- Smaller, faster and cheaper processors
- Small, cheap and roomy memories
- Mobility by use of wireless networking:
  - mobile phone, WLAN, Bluetooth, ZigBee, Infrared, …
- Cheap user interfaces/terminals:
  - smart cards, mobile phones and PDAs

Hardware of many sizes



The "T-Engine Series"
Source: Ken Sakamura

Enabling technologies, cont.

New and/or better sensors for:

- geographical position and precise global time (e.g., GPS)
- vision (picture and movie)
- biometrics for e.g. smell
- motion, tension, vibration, flow, acceleration, …

Actuators and transducers for:

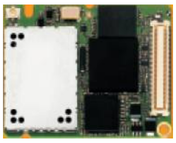
- motion, gyro stabilization, energy scavenging, …

Machine to Machine (M2M)

- M2M data links between systems, remote devices and individuals can make businesses more effective
- M2M enhance independence of time and location

Applications:

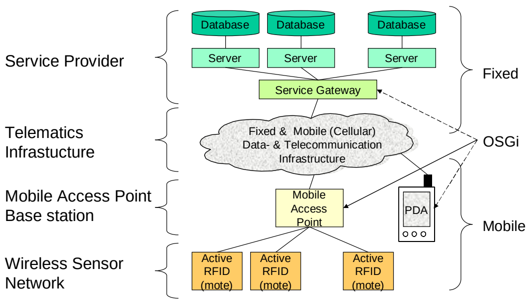
- Security
- Automatic meter reading
- Vending machines
- Elevators and escalators
- Industrial applications
- Cargo tracking
- Road traffic information
- Traffic control systems
- Telemedicine



Visions

- Pervasive, ambient, invisible, embedded and networked real-time computing systems
- Utilise information & communication everywhere – at all times and in daily life
- Support end-users ability to access and process information available in the environment or context
- Enable smarter functions by cooperation
- Wireless connections makes it viable to borrow resources/services between autonomous systems

T4 – System Architecture



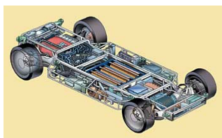
Trends

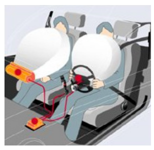
- Network coupled embedded systems
- Temporary sharing of services/resources
- Access to the environment, everywhere
- Autonomous components cooperate via open standardised interfaces
- Self-configured components reduce complexity
- From pull via push to ambient computing

Applications

- paper and power plant process control
- robotics and industrial automation
- medical monitoring and control
- mobile phones and personal digital assistants
- smart cards and rfid-tags
- base stations and telecommunication exchanges
- routers and firewalls
- stock trading and e-commerce
- defence systems
- traffic supervision & control
- home networks and burglar alarm systems
- electronic books
- cameras
- drive by wire, automatic gearshift, engine control
- airbags and safety belt systems
- anti-lock break systems, stability control, anti-slip control
- fly by wire aircraft control
- navigation systems
- handicap equipment
- etc.

Automobile applications





Economically very important

- There are **orders of magnitude more of embedded controllers** than all PCs, internet servers and main frame computers taken together
- The **value added** by use of an embedded real-time computer **can very be large,** yet difficult to measure

Consequences

- Networking and cooperation enables new services
- Invisible computers will work around the clock, everywhere for the end user
- System components must to a larger extent be autonomous and self configuring
- System components will provide resources and services to each via standardized interfaces and distribution means

Wireless control and monitoring

Typical ZigBee/IEEE 802.15.4 Applications Source: Chipcon AS

LOCATION-AWARE SERVICES

Navigation and direction-giving
Offer travelers directional guidance

Geographic messaging
Send message to anyone within a specified area or distance

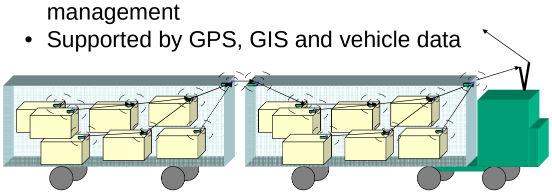
Who-is-around service
Determine who is located in a geographic area

Neighbor-and-service discovery
Find out neighboring devices and services

Source-position identification
Determine the geographic position of the sender

Real-Time Monitoring

- Complement to logistic, business and fleet management
- Supported by GPS, GIS and vehicle data



Active RFID sensors on each pallet and container accessible via mobile infrastructure (Mobitex, GPRS, UMTS, Satellite)

Distributed real-time environment

Applications operates and acts in the **real world**

For example enabling:

- Efficient and safe traffic control
- Less polluting engines
- Safety systems in cars, aircraft's, boats and trains
- Stock exchange transactions
- Multi-player games
- Tracking of goods or other mobile objects