TU Delft

Delft University of Technology

The Delft UoT Master Curriculum on Embedded Systems

Hans-Gerhard Gross and Arjan van Gemund

Faculty of Electrical Engineering,
Mathematics & Computer Science
Delft University of Technology

Mekelweg 4, 2628 CD Delft, Netherlands
h.g.gross, a.j.c.vangemund @ tudelft.nl

Outline

- Motivation for developing ES MS program
- Objectives and learning outcome
- Context of the program
- TU Delft ES program
- 2 Example courses
  - Real-time systems course
  - Embedded real-time systems course
- Didactics

Motivation

- Embedded Systems (ES)
  - Combinations of hard- and software
  - For monitoring/controlling the natural environment
- ES are becoming more
  - Complex
  - Intelligent
  - Distributed
  - Embedded
- ES development involves more different skills

Motivation

- ES design is a multidisciplinary effort, involving disciplines such as
  - Electrical Engineering
  - Computer Science
  - Mathematics
  - Physics
  - Mechanical Engineering
- And experience in
  - Real-time computing
  - Systems Architecture
  - Control Theory
  - Safety and Security
  - Computer Networking, etc.

Motivation

- Graduates with different backgrounds find it difficult to communicate and cooperate in multidisciplinary projects
  - Different languages
  - Different concepts
  - Different abstractions
  - Different notations
  - Different models
- Systems cannot longer be developed with two different threads
  - Hardware
  - Software

Objectives and Learning Outcome

- According to the “Dublin Descriptors” of the European Joint Quality Initiative (www.jointquality.org)
- Plus
  - Awareness of the role of ES in society
  - Capability of determining ES requirements
  - Familiarity with ES realization platforms
  - Testing and optimization of ES

Objectives and Learning Outcome

Student can

- View a system development holistically
- Master complex systems
- Describe structure behavior of systems
- Master contemporary development techniques
- Perform requirements engineering, design, implementation and testing
- Devise new theories, techniques, and tools
- Present and communicate ideas and visions

Objectives and Learning Outcome

Student can

- Work in a team
- Reflect
- Contribute to research
- Deal with compromises
- Make decisions based on risk
- Implement life-long learning

Context of the Program

- MS Embedded Systems is issued by the 3TU, federation of the 3 Technical Universities in NL (www.3tu.nl)
  - TU Eindhoven
  - TU Twente
  - TU Delft
- "Virtual" University located within the 3TUs
- Students can take courses from the 3 "real" Universities

TU Delft
Delft University of Technology

Context of the Program

- 3TU starting from April 1st, 2007
- Combines ca 70 existing Prof. chairs + 30 new ones acquired over next 5 years
- 5 centers of excellence
  - High-tech Systems and Materials
  - Technologies for Sustainable Energy
  - ICT Services and Applications (MS ES)
  - Fluid and Solid Mechanics
  - Applications of Nano-Technology

3TU ES Program

- 2 year 120 EC program
  - 1 EC = 28 hrs nominal study load
- Intake is mainly
  - Computer Science bachelors
  - Electrical Engineering bachelors
  - Other related bachelors may be admitted (Entry level determined by Admissions Board)
- Some 80% of the curriculum is identical in the 3 Technical Universities, however Delft has particular focus on
  - Embedded software (strong software engineering group)
  - Bridging the gap between the discrete and continuous domain
  - System level engineering (rather than component-level)

TU Delft ES Program

Homologation (10 EC)

For EE Students
- Operating Systems
- Systems Programming
- Software Engineering

For CS Students
- Modeling & Control
- Digital Signal Processing
- Digital Systems

Common Mandatory Part (30 EC)
- Embedded architectures (hardware)
- Real-Time Systems (software)
- Embedded Real-Time Systems (hardware/software)
- Performance Analysis (hardware)
- System Validation (hardware/software)
- Software Testing (software)

Electives (40 EC)
- Including traineeship (12 EC)

Thesis (40 EC)
- Including introductory assignment (10 EC)

TU Delft
Delft University of Technology

TU Delft ES Program: Electives

- Transmission Systems Engineering
- Methods and Algorithms for System Design
- Advanced Image Processing
- Information Theory
- Electronic Design Automation
- Reliability Engineering
- Silicon Sensors
- Displays and Actuators
- Micro-System Integration
- Multimedia Compression
- Cryptography
- Mobile and Wireless Networking
- Systems on Chip
- Computer Arithmetic
- Real-time Artificial Intelligence and Automated Speech Recognition
- Compiler Construction
- Parallel Algorithms and Parallel Computers
- Advanced Software Engineering
- Cooperative Agent Based Systems
- High Performance Computing
- Software Architecture Recovery and Modeling
- Computational Logic and Satisfiability
- Pattern Recognition
- Systems Specification Models
- Distributed Algorithms
- Knowledge based control systems
- Optimization in systems and control
- Modern robotics
- Modeling and control of hybrid systems

Example Courses

- Mandatory Program
  - Real-Time Systems
    - Taught in the 2nd quarter
      - 7 weeks / 4 hrs lecture + 4 hrs lab per week
    - Typical RT problems within a computer
  - Embedded Real-Time Systems
    - Taught in the 3rd quarter
      - 7 weeks / 4 hrs lecture + 4 hrs lab per week
    - Extension of the problems into a “physical” Environment

Example Course: Real-Time Systems

- Mainly hands-on real-time system experiments with supporting lectures
- Students are given tasks to perform own research
  - Comparison of standard Linux timing operations and the RTAI timing operations, accuracy of time measurements,
  - Development of a high-resolution timer based on the processor clock and comparison of that timer with the standard Linux and RTAI timers,
  - Development of a code framework under RTAI for worst-case execution time (WCET) analysis experiments and scheduling experiments

Example Course: Real-Time Systems

- Dynamic worst-case execution time analysis for standard algorithms, e.g., sorting, searching, computer graphics, etc., optimization of algorithms toward higher analyzability,
- Evaluation of dynamic timing analysis through code coverage analysis,
- Design of constant execution time algorithms (WCET-oriented programming),
- Search-based execution time analysis; comparison between manual testing, random testing, and application of a genetic algorithm as test case generator

Example Course: Real-Time Systems

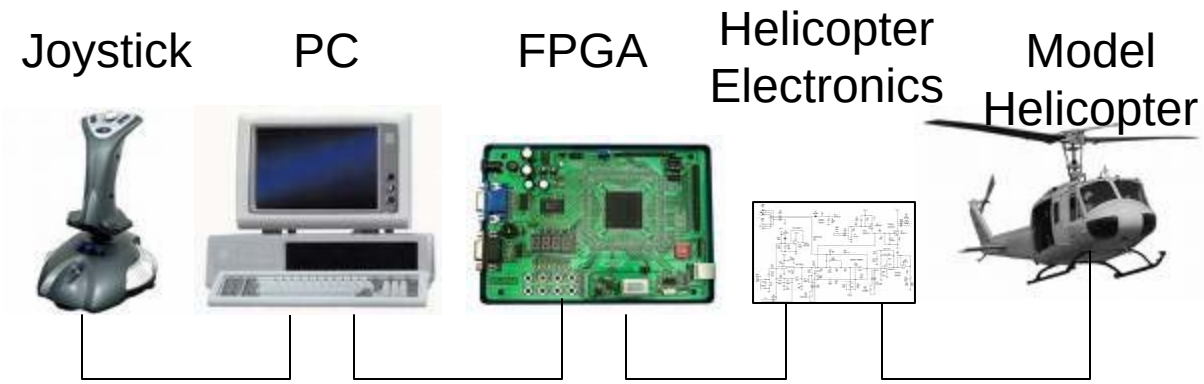
- Students are supposed to make their own research and draw their own conclusions
- Documentation of their findings through a research paper (6—8 pages, individual assignment)
- Lectures provide
  - Terminology
  - Examples
  - Additional Exercises
  - Mini-tests for self-assessment

Example Course: Real-Time Systems

- Evaluation
  - Too much basic programming
    - Provision of code templates
  - Biggest problem: writing a research paper
    - Provision of the best papers from the past
  - Most students like the idea of a research paper

Example Course: Embedded RT Systems

- Mainly embedded system project
  - Development of control software for a model helicopter
  - Control of the helicopter through a “joystick”
  - Problem: hardware/software integration



Example Course: Embedded RT Systems

- Challenges
  - Concurrent real-time programming and debugging (PC, FPGA, controllers, signal filters)
  - Hardware/software co-design: which components are implemented in hardware/software?
  - Teamwork, including project management
  - Mastery of the different disciplines involved

Example Course: Embedded RT Systems

- Supporting Lectures on
  - Helicopter mechanics
  - Modeling and simulation
  - Control theory
  - Digital filter theory
  - VHDL programming
  - X32 architecture and programming
  - Basic electronics
- Assessment based on a report (group assignment)

Example Course: Embedded RT Systems

- Evaluation
  - Difficulties with VHDL programming
    - Provision of existing VHDL components
  - Problems with signal filtering
  - Limited Lab resources (8 working places and only 1 model helicopter)
  - Steep learning curve with subjects outside their curriculum
  - High student satisfaction with the course

TU Delft
Delft University of Technology

Didactics

- New quality program BKO
  - Basiskwalificatie Onderwijs
  - All staff (age < 45) have to participate
- Active and Collaborative Learning
  - Student centered activities vs. teacher centered activities
  - Student/student interaction vs. teacher/student interaction
  - Higher-order (deep) learning approaches vs. shallow learning approaches

Didactics

- Real own problem solving rather than presenting the problems and their solutions
- Team activities rather individual assignments
- Group assignments and discussions
- Peer-reviews
- Exercises in class
- Self-assessment
- Academic working (reflection, research paper)

TU Delft
Delft University of Technology

Thank you.

- Questions
- Discussion