

# DVA438 – Project in Embedded Systems

Ali Balador and Tiberiu Seceleanu

Mälardalen University  
2020 11 10





# The Team



Tiberiu Secoleanu, examiner



Ali Balador, supervisor



Zenepe Satka, lab technician



Jukka Mäki-Turja, customer



# Basic Information

- Course: Project in Embedded Systems (PES)
- Code: DVA438
- Level: advanced
- Subject: computer science / engineering
- Credits: 7.5 ECTS
- Duration: 10 November 2020 - 15 January 2021

The course materials are available on Canvas.



# Prerequisites (The Input)

- English language
- 90 ECTS in computer science or engineering
- passed "Embedded Systems I" or equivalent
- So you know
  - the EVK110/AVR32 platform
  - how to program AVR32 without OS support
  - how to program AVR32 with FreeRTOS support
  - theory behind RTOS
  - how to organize and run a project
  - how to analyze problems, produce deliverables
  - how to present the outcome



# The Experience (The Output)

- Organization, planning, and execution, and evaluation of a project in embedded systems.
- Analyze complex problems in embedded systems
- Specify requirements on advanced embedded systems.
- Describe the aptitudes needed to solve the problem.
- Apply advanced methods, tools and standards to solve the problem.
- Identify related problems and their solutions, and propose own solutions.
- Document and present the project results both orally and in a scientific report.
- Compare the proposed solution with the existing research solutions.



# Presence

- Attend every meeting
- You can miss only one meeting without excuse
- Any other missed meeting requires sick note from doctor
- Attendance at API and final meetings is obligatory!
- Presence at your group meetings is up to you
- However, a group might exclude inactive members (after feedback with Tiberiu and Ali)



# Communication

- Every group selects one mailperson, where communications only happen through that person.
  - Exceptions: problems within the group that require direct contact
  - Unavailability of mailperson
- Emails should include [DVA438 - N] (N = groupnumber) in the subject.
- For problems within the group, you should try to solve it yourself, if not possible contact Ali and Tiberiu
- For health issues, contact Ali and Tiberiu



# Failure

- Common causes for failing the course:
  - unfinished software
  - insufficient attendance
  - group exclusion
  - insufficient documentation
  - cheating (e.g. plagiarism)
- Failure ⇒ possibly second chance to present the system again later to a higher standard
- Single persons can fail with the rest of the group passing the course





# Requirements to Pass the Course

- To be active in your group
- All deliverables (code and report) are approved
- Attend at least seven project meetings & intro (this) & both API and final seminars
- Final presentation



# Cheating

- We follow a zero tolerance policy against cheating! including:
  - plagiarism in any form (e.g. paraphrasing),
  - abuse of controlled substances (e.g. neurostimulants like Modafinil) or
  - employing counseling services (e.g. hire someone for programming).
- Persons caught cheating will be reported to the board (i.e. probably suspended from the course immediately and thus failing the course).



# Grading

- 50% based on report
- 50% based on
  - Mutual group evaluation
  - Grading during regular supervision meetings (incl. sporadic feedback)
  - Seminars (API & final)
  - Demo
- Grade 3 (pass): >54%
- Grade 4 (good): >77%
- Grade 5 (very good): >91%



# What should be done

- Ten weeks
- Four parts
  - Requirements Analysis: understand what to do, identify work-packets
  - Design: select tools, define time-plan, assign responsibilities
  - Implement: follow the time-plan, implement
  - Finish: wrap up, document, sell
- Eventually, sell system to customer



# Preparation for the course

- If you haven't taken "Embedded Systems I" course, we strongly recommend that you go through the slides of that course and make sure you understand the theory behind RTOS!



# Until the first meeting on November 13

- Each group has team meeting
- Each group selects one mailperson
- Each group selects tools (versioning, reporting, . . .)
- Each group identifies roles and rotation (who is leader when?)
- Go to Canvas, find the project files. In there: find the order and read it
- Each group presents a tentative time plan



# Upcoming weeks

- 13 November: supervision: present requirement analysis document
- 20 November: supervision: updated requirement analysis
- **24 November: API meeting**
- 27 November: supervision: present design document
- 4 December: supervision: updated system design
- 11 December: supervision: updated system design
- 18 December: supervision: updated system design
- **19 December - 6 January: vacation**
- 8 January: supervision: final checking
- **15 January: final seminar**



# Deliverables 1

1. Requirement analysis: What is provided? What is required?  
What are potential problems?
2. Time plan: Milestones, person hours, deadlines, responsibilities, . .
3. APIs:
  - provide a detailed suggestion for the API incl. motivation
  - send suggestion until **22 November end of day to Ali**
  - every report will be provided to all groups the next day
4. Design (after finishing 1-3)
5. Test cases (**14 December end of day**)
6. Implementation (after finishing 4,5)





# Deliverables 2

## 1. Final report

- written in a good academic/scientific way (no use, make, do); write as if intended for to be read
- including one page individual report\*
- **deadline 12 January end of day**

## 2. Presentation at the final seminar

- show the benefits of your product
- show the test cases
- show the limitations

## 3. Tech demo with Zenepe

- each group shows their product
- Zenepe tests each group member on all project-parts

\* A report has an introduction and a conclusion. A report is not a bullet-list! The one page refers to 10pt IEEE two-column style without figures (The report might include figures, though).



# Final Report

- The final report contains one section with each individual report
  - each student compares ROSA with another suitable OS (not Windows, iOS, ...) taking a reference to the guest lecture
- Gradual evolution from req. analysis, via design document, to become final report
- LTEX(overleaf) is strongly advised



# API Meeting

- API split into four parts
  - Scheduler
  - Tasks
  - Clock
  - Semaphores
- Each part discussed in cut-set group
- Each group sends one member to each cut-set group
- Each cut-set group agrees on an API and presents that in the end of the seminar



# Test cases

- Each group specifies up to three test cases
- Test cases must comply with API
- Deadline: **14 December end of day**
- Each group receives up to three test cases (randomly)
- The final report must contain a section on these
- State if your system survives your own and the given test cases
- If not, state how you fixed it





**Any  
Questions?**