



NTNU – Trondheim
Norwegian University of
Science and Technology

Introduction

TDT4258 Energy Efficient Computer Systems
(formerly Microcontroller System Design)

Stefano Nichele

Department of Computer and Information Science

2013, January 25th

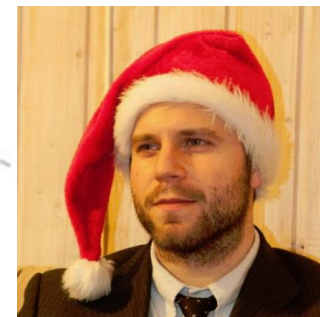
Agenda

- Lab introduction
- Organization (where, when, groups ...)
- Grading
- Equipment / Tools
- Presentation of Exercise 1



Assistants

- Vit. Ass. Stefano Nichele nichele@idi.ntnu.no



- Und. Ass. Rune Holmgren raane.holm@gmail.com

(Monday and Tuesday)



Lab

Room 458, IT-Building (4th floor)



NTNU – Trondheim
Norwegian University of
Science and Technology

When

- Und.ass will be available in scheduled lab-time, but he is not there to do the exercises for you!
 - Monday, 10.15 – 13.00 (**next week on Friday 10.15 - 13.00)
 - Tuesday, 15.15 – 18.00 (**next week on Friday 14.15 - 17.00)
- The lab is always available
- The exercises are challenging, you are supposed to understand many things by yourself, **START EARLY!!**

Exercises

- Before each exercise there will be an introductory lecture
- 3 exercises, approx. 3/4 weeks for each exercise
- Submission deadline in the semester schedule
- (Brief) presentation to vit.ass the week after submission (every group will present at least one exercise, you can be selected for presentation more than once). The presentations will be held in the lab.



Groups

- 3 people in each group (with exceptions: 2 could be fine depending on the number of students, 1 means a lot of work to do)
- Chart on the door of the lab
 - Choose the day you want supervision in the lab and write your names in the corresponding section
 - Max 9/10 groups per day
 - If you do not have a partner write your name in the day/time that suits you best (try to find someone else with no lab partner on the chart)
 - Those signed for particular hours have priority over the others (in case more students want to work in the lab with und.ass help)



Submissions

- Deadline fixed in the semester plan
- 10% score reduction for each day in late
- The submission should contain:
 - The code you have produced
 - Report (IN ENGLISH)
 - Clear description (pictures, diagrams, flow charts, etc.)
 - Well-structured
 - Step-by-step description
 - Should be delivered on It's Learning. If you have delivery problems send the report and the code to vit.ass by email (nichele@idi.ntnu.no)



Grading

- At least the minimum requirements described in the exercise text should be fulfilled
- Quality of the code and technical solutions (energy efficiency)
- Report quality
- Tests
- Demonstration for vit.ass
- Solutions beyond the minimum requirements. These must be documented in the report to be evaluated
- Final mark for TDT4258:
10% Exercise 1 + 20% Exercise 2 + 20% Exercise 3 + 50% Exam

Exercises: objectives

- Microcontroller programming (C and assembler)
- I/O programming
- Interrupts
- Programming on Linux Kernel
 - Create your own hardware driver
- Development using GNU tools



PC in the lab

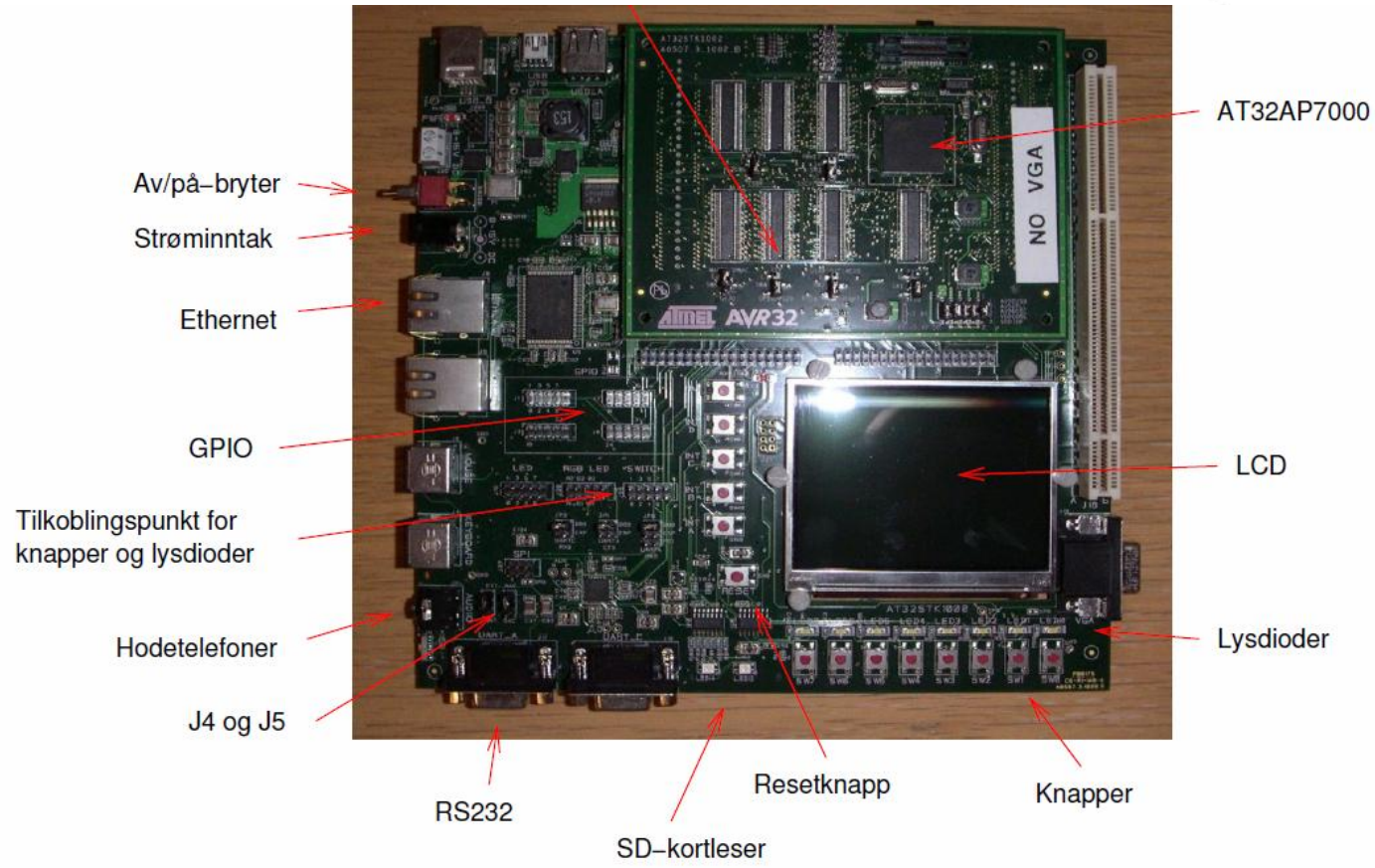
- Running Ubuntu (GNU) / Linux
- The exercises require some knowledge of Unix-based systems as well as command line
- For those not familiar with Unix-based systems:
 - Exercise booklet provides an introduction
 - Exercise booklet has links to information on the Internet
- IDI account required
 - To create one talk to Guru or Ellen Solberg (student service desk - 1st floor IT-Bygget)
- To enable IDI account and to get access to the lab send an email to nichele@idi.ntnu.no with your name, username, student card number.

Exercises

- 3 Exercises:
 - Ex. 1 Assembler
 - Ex. 2 C
 - Ex. 3 C and Linux
- Programming of an embedded system
- Exercise booklet on the Web (course webpage)
 - Read carefully
 - Contains information needed to conduct the exercises
 - Contains exercise requirements



STK1000

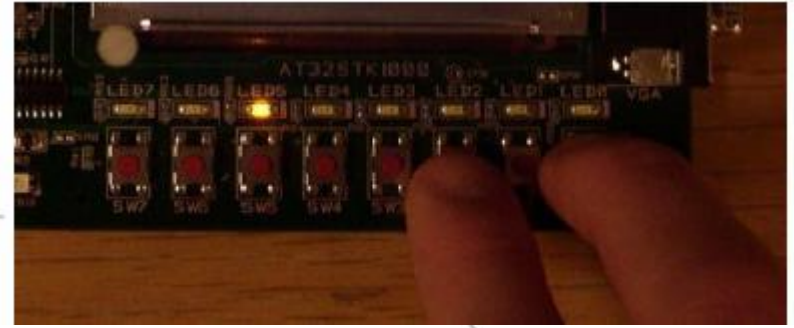


AVR32

- 32-bit RISC processor from Atmel
- AT32AP7000:
 - AVR32-based microcontroller
 - Many built-in I/O devices:
 - General I/O pins (buttons, LEDs)
 - DAC (audio)



Exercise 1



- The goal: Assembly programming of a 32-bit RISC processor
- AVR32 – AT32AP7000
- STK1000 development boards
- Create a program in AVR32-assembly to:
 - Program I/O devices: read buttons and set LEDs on and off
 - Implement interrupt handling
- GNU tools: GCC, GNU LD, GNU Make, GNU Debugger (GDB)

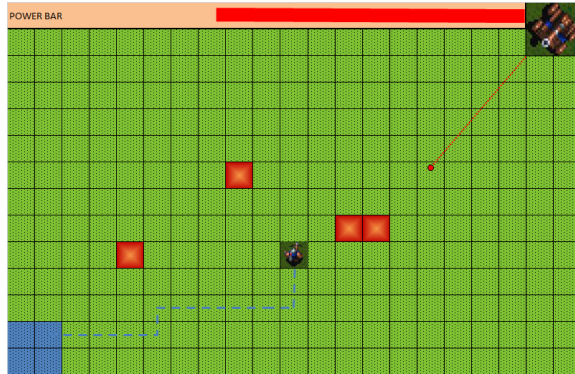
Exercise 2

- The goal: C-programming of AVR32 without operating system support
- Create a program that:
 - Makes sounds by DAC on a development board
 - Implement interrupt handling
- GNU tools: GCC, GNU LD, GNU Make, GDB



Exercise 3

- The goal: C programming with embedded Linux (Linux OS in the background)
- Create the Scorched Land Defence game:
 - Linux function calls
 - Make drivers for the buttons and LEDs (kernel drivers)
 - Programming hardware in Linux



Feedback

- Exercise booklet – what can be done better?
- Lab
 - Machines
 - Tools
- Include a section with your evaluation of the assignment in each report

Lab Schedule

Week	Monday	Tuesday	Thursday	Friday
4				Presentation Exercise 1 (12.15) Und.Ass. in Lab until 16.00
5				Exercise Gr A (10.00-13.00) Exercise Gr B (14.00-17.00)
6	Exercise Gr A	Exercise Gr B		
7	Exercise Gr A	Exercise Gr B		Deliver Exercise 1 (12.00) Presentation Exercise 2 (12.15)
8	Exercise Gr A	Exercise Gr B		
9	Exercise Gr A	Exercise Gr B		
10	Exercise Gr A	Exercise Gr B		
11	Exercise Gr A	Exercise Gr B		Deliver Exercise 2 (12.00) Presentation Exercise 3 (12.15)
14			Exercise Gr A (time TBD)	Exercise Gr B (time TBD)
15	Exercise Gr A	Exercise Gr B		
16	Exercise Gr A	Exercise Gr B		
17	Exercise Gr A	Exercise Gr B		Deliver Exercise 3 (12.00)

