An Undergraduate Embedded Software Laboratory for the Masses

André Stollenwerk, Chate Jongdee, Stefan Kowalewski
stollenwerk@embedded.rwth-aachen.de

Workshop on Embedded Systems Education (WESE’09)
October 15th 2009
Outline

- Motivation
- Short Facts on the new Lab Course
- Organizational Aspects
- Evaluation Board
- Topics
- Experiences
- Summary and Outlook
Bologna Process → Impact to Curricula

- Graduation Scheme was Adapted to Bachelor and Master of Science
- Courses have to be Rearranged
- New Courses were Established
- Analysis of Existing Curriculum was Performed
Particular Requirements to new Courses

- Experiences with Hardware Related Issues
  - CPU and Hardware Registers
  - Interrupts
  - Memory-Models

- Getting in Practical Touch with System Software
  - Bootloader
  - Scheduling
  - Memory Management
  - Programming Paradigms (Especially Hardware Programming)
Outline of the new Lab Course

- Mandatory 3rd Semester BSc (with no Requirements)
- Implementing an Operating System on an ATmega in C
- Practically Familiarizing with Operating Systems Algorithms
- Working with restricted hardware resources
- Testing and Documenting Code
- Teamwork
- Presenting ones Results (5 Min Presentations)
Facts and Figures on the Lab Course

- Used Microcontroller: ATmega 644
- 2\textsuperscript{nd} Year Students
- 4 ECTS Credits
  - 18 Hours of Presence Exercises
  - ~ 100 Hours of Self Studies (Homework)
- ~ 200 Students (in Teams of 2 Students)
- Experiments Spread over approx. 3.5 Months
- No Grades (just Passed or Failed)
- Established in Winter Term 2007 (3\textsuperscript{rd} Time this Semester)
Organizational Aspects

- 6 Experiments with 3 Hours of Presence Exercise each
- Guided by 1 Research Assistant and 3 Student Assistants
- Group Sizes of approx. 30 Students (15 Teams)
- Test Programs for each Experiment
- Different Ways of Supporting the Students
  - Guiding Documentations
  - Consultation-Hours
  - Moderated Bulletin Board
  - Reliable and Easy to Handle Evaluation Board
  - Testing on Real Hardware
Our Evaluation Board

- 2 x 16 Character LCD
- Serial Interface
- ISP
- JTAG
- Oscillator
- Microcontroller
- Debugging “Helpers”
- 4 Buttons
- 8 LEDs
- RS232

EMBEDDED SOFTWARE LABORATORY

RWTH AACHEN UNIVERSITY
Topics

- **Bootloader**
  - Receiving new Programs through Serial Interface (RS232)
  - Configuring the Microcontroller (Fuses and LockBits)
  - Byte Order (Intel HEX vs. Big-Endian)

- **Scheduler**
  - Handling the Processes' Stacks and CPUs Registers
  - Preemptive and Cooperative Multitasking
  - Problems with Certain Work Flows due to Preemptivity
  - Different Scheduling Strategies

Command Flow during Scheduling

```
function a()
{
    do_something();
    something_else();
}
```

```
function b()
{
    some_code;
    more_code;
    ...
}
```

Scheduler

- save old task context
- find the next task
- restore new tasks context
- rearrange stackpointer

Different Scheduling Strategies
Topics

- Memory Management
  - Dynamically Allocated Private and Common Memory
  - Read-Write-Dependencies
  - Access Restrictions
  - Different Allocation Strategies
  - Accessing Internal and External Memory
    - External Memory also includes a Latch

8 Bit Latch \(\rightarrow\) 8 k Byte RAM \(\rightarrow\)

Accessing Internal Memory
Topics

- Task Manager
  - User Interface to Previous Components
  - Start / Stop / Pause Different User Applications
  - Change Scheduling Priorities / Strategies
  - Change Memory Allocation Strategies

- Application Tasks (Last Experiment – Choice 1 out of 4)
  - Communication Between Two Microcontroller Boards
  - PS/2 Keyboard Driver + Simple Text-Based Game
  - RFID Reader + Access Control to Task Manager
  - Digital Picture Frame
Application Tasks

- Communication
  - Serial Communication via UART
    - Different Format Settings
    - Handling Frame Format and Parity Errors
  - Serial Communication via Optical Interface
    - LED + Photo Resistor
    - Reading from ADC
    - Taking Care of Electrical Issues like Noise

- PS/2 Keyboard Driver
  - Using USART for Basic Electrical Communication
  - Handling different Keyboard Maps
  - Timing during Communication
Application Tasks

- **RIFD Reader**
  - 125 kHz Read-Only Tags
  - External Receiver Board based on EM4095
  - Challenges due to Interrupt / Timer Issues
  - Little bit of Signal Coding (Manchester Code)

- **Digital Picture Frame**
  - SD-Card + Color LCD on an extra PCB (both SPI)
  - FAT File System
  - Different Image Formats (None With Transformations… ;-) )
Experiences

- Easy and Reliable Hardware is a very Basic Requirement
- Testing Environments are Needed for Preparation at Home
- Auxiliary Documentation → Everybody on the same Level
- Experiments’ Documentation as Detailed as Possible
- Not to Restrict Students Creativity
- Biggest Challenge – Finding the Right Trade-Off

→ Resulting in Better Skilled and Motivated Students
Conclusions and Outlook

- Covering Huge Amounts of Students
- Introducing Students to Embedded Hardware Programming
- Familiarizing Students with System Software Concepts

- Create more Testing Possibilities
- Evaluate the new Application Tasks
Tanks for your attention

Questions?
Details on the Evaluation Board

- Exchangeable Microcontroller (ATmega 16…ATmega 644)
- Exchangeable Crystal Oscillator (currently 20 MHz)
- All Ports (32 PINs) Lead Through to the Outside
- 2x16 Character LCD
- JTAG
- ISP
- RS232 Interface
- 4 Buttons
- 8 LEDs
Extract of the BSc Curriculum in Computer Science at RWTH Aachen University

- **1st Semester**
  - Lecture “Computer Engineering”
  - Lecture “Programming”

- **2nd Semester**
  - Lab Course “Electrical Engineering“
  - Lecture “Data Structures and Algorithms”

- **3rd Semester**
  - Lecture “Operating Systems and System Software“
  - Lab Course “Hardware Programming“
Memory Arrangement

Stack Task n

Global Vars

Heap Map

Payload

Stacks

Main & ISR Stack