**CS-473**  
**Embedded systems**  
Beuchat René

<table>
<thead>
<tr>
<th>Cursus</th>
<th>Sem.</th>
<th>Type</th>
<th>Credits</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer science</td>
<td>MA1, MA3</td>
<td>Opt.</td>
<td>4</td>
<td>English</td>
</tr>
<tr>
<td>Cybersecurity</td>
<td>MA1, MA3</td>
<td>Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical and Electronical Engineering</td>
<td>MA1, MA3</td>
<td>Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microtechnics</td>
<td>MA1, MA3</td>
<td>Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robotics, Control and Intelligent Systems</td>
<td>MA1, MA3</td>
<td>Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robotics</td>
<td>MA1, MA3</td>
<td>Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC master EPFL</td>
<td>MA1, MA3</td>
<td>Opt.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Summary**

Last year with R.Beuchat as teacher. The main topics of this course are understanding and designing embedded system on a programmable circuit (FPGA). Students will be able to design a camera or a LCD controller on an FPGA in VHDL and will use their controller through a softcore processor.

**Content**

- Microcontrollers and their associated programmable interfaces (GPIO, Timer, SPI, A/D, PWM, interrupts)
- Hardcore/softcore processors (ie. NIOS II, ARM)
- Memory organizations, little/big endian
- Synchronous busses, dynamic bus sizing (ie. Avalon Bus in Memory Mapped mode)
- Processor busses, busses realized in a FPGA
- Serial busses (ie. UART, SPI, i2c, ...)
- How an LCD graphical screen and a CMOS camera work
- FPGA-based conception of Embedded Systems
- Embedded systems with processors on FPGAs

Laboratories provide knowledge & practice to develop an embedded system based on an FPGA device.

**Keywords**

microprocessors, microcontroller, FPGA, embedded systems, SoC, programmable interface

**Learning Prerequisites**

**Required courses**

Introduction to computing systems, Logic systems, Computer architecture

**Recommended courses**

Electronic, Programming (C/C++), Project System On Chip

**Important concepts to start the course**

- Computer architecture (processor, memory, programmable interfaces)
- Processor Architecture (PC, registers, ALU, instruction decoding, instruction execution)
- Knowledge of C programming language
• Knowledge of VHDL

Learning Outcomes
By the end of the course, the student must be able to:
• Design an embedded system on an FPGA
• Analyze a specific problem to be solved and propose an FPGA-based system to solve it
• Implement a solution to the given problem
• Realize and simulate the design
• Test the developed solution on an FPGA
• Use complex development tools and hardware debugging tools such as a logic analyzer and an oscilloscope

Transversal skills
• Use a work methodology appropriate to the task.
• Negotiate effectively within the group.
• Set objectives and design an action plan to reach those objectives.
• Continue to work through difficulties or initial failure to find optimal solutions.
• Use both general and domain specific IT resources and tools
• Make an oral presentation.

Teaching methods
Ex cathedra and exercises, laboratories by specific sub-topics, final mini-project

Expected student activities
• Reading and deepening of course concepts
• Preparation of exercises performed in the laboratory
• Writing reports on different labs
• Realization of a final mini-project by group with oral presentation, report and demonstration

Assessment methods
With continuous control.
all labs 30%, mini-projet 20%, oral exam 50%

Supervision
Others Course on Moodle with forum

Resources
Bibliography
Teaching notes and suggested reading material on moodle
Specialized datasheets (micro-controllers, FPGA) and standards (i.e., SPI, I2C, Amba, Avalon, etc.)
Book downloadable freely: https://www.arm.com/resources/ebook/fundamentals-of-soc
Fundamentals of System-on-Chip Design on Arm Cortex-M Microcontrollers

Notes/Handbook
Documents and slides provided on moodle

**Moodle Link**
- [https://go.epfl.ch/CS-473](https://go.epfl.ch/CS-473)

**Prerequisite for**

CS-476 Real-time embedded systems