Universitat Autònoma de Barcelona

2019/2020

Embedded Systems

Code: 102791
ECTS Credits: 6

Degree	Type	Year	Semester
2502441 Computer Engineering	OB	3	1
2502441 Computer Engineering	OT	4	1

Contact

Name: Lluís Ribas Xirgo
Email: Lluis.Ribas@uab.cat

Use of Languages

Principal working language: catalan (cat)
Some groups entirely in English: No
Some groups entirely in Catalan: No
Some groups entirely in Spanish: No

Other comments on languages

Reference material is in English.

Teachers

Joaquín Saiz Alcaine
Daniel Rivas Alonso

Prerequisites

For the full understanding of the contents of the course, it is convenient to have a basic ability in programming and a good knowledge of how programs execute in computers. For this, you should have completed Programming laboratory and Software engineering courses (programming), as well as Computer structure, Operating systems and Computer architecture (program execution model).

Objectives and Contextualisation

This course is the first within the subject of Design of application-oriented computing systems, which deals with the development of systems that run the algorithms of specific applications under a set of requirements. For example, it is not enough for a mobile device to be capable of displaying a video, it must be done at 25 images per second, synchronizing it with audio and consuming as little energy as possible. Therefore, the main goal of the subject is to know how to design algorithms and the implications that each design has in the cost of the application depending on the very algorithm and how it is implemented, that is, in accordance to the chosen execution platform.

In this context, the Embedded systems' course objective is that students acquire the following competences:

- Understand the various fields of application of embedded systems.
- Have notions of the usual requirements of each application domain, including those in real time.
- Understand the aspects of security, reliability and robustness of the systems.
- Understand the methodology of the development of embedded systems.
- Understand the various systems' computational models.
- Have practical skills with the design and manipulation of state-based computational models.
- Understand the basic elements of embedded system architectures.
- Have acquired the rudiments of platform-based design.
- Know how to estimate implementation costs based on the computational models of the systems.
- Understand the problem of partitioning systems and be aware of the different approaches to do it.
- Have the rudiments of programming and hardware description languages for the implementation of the systems.

Competences

Computer Engineering

- Acquire personal work habits.
- Capacity to design, develop, evaluate and ensure the accessibility, ergonomics, usability and security of computer systems, services and applications, as well as of the information that they manage.
- Communication.
- Have the capacity to define, evaluate and select hardware and software platforms for the development and execution of computer systems, services and applications.
- Have the capacity to develop specific processors and embedded systems and to develop and optimise the software of said systems.
- Work in teams.

Learning Outcomes

1. Accept and respect the role of the various team members, and its different levels of dependence.
2. Analyse the requirements of computer applications.
3. Compare and evaluate the possible platforms that can fulfil the requirements of applications.
4. Identify the needs of the specific application that must be resolved.
5. Identify the security needs that embedded systems have to fulfil.
6. Manage time and resources available. Work in an organized manner .
7. Use English as the language of communication and professional relations .
8. Work independently.

Content

1. Introduction

1.1. Application domains of embedded systems

1.2. Study cases

1.3. Systems engineering

2. Design methodology

2.1. Design flow

2.2. Functional and non-functional requirements' analysis

2.3. State-based computational models

2.4. State machine nets

2.5. Hardware architectures

2.6. Real-time operating systems

2.7. Software architectures

2.8. Architectural platforms

3. Implementation methodology

3.1. Refinement

3.2. Reusability

3.3. Partitioning and scheduling

3.4. Software and hardware syntheses

Methodology

Teaching is structured in the following face-to-face activities:

- Theory classes: Presentations of course contents, with a first part that is devoted to the dissemination of the necessary knowledge for the analysis and the design of embedded systems, and to explain cases that situate in context the knowledge and the abilities that are acquired during the course like, for example, the explanation of how the electronic part of a car works. The second part will be devoted to discuss the problems that will be dealt with in the corresponding seminars.
- Problem-solving seminars: Discussion of small case studies (for example, control of a microwave oven) that serve to consolidate theoretical knowledge regarding the analysis and design of embedded systems.
- Laboratory practices: Team work at laboratory, following a walk-through guide under the supervisions of a teacher. Each session will deal with a specific aspect regarding the implementation of embedded systems.

There is a very important part of team work outside the classroom and the laboratory. In this sense, each member of each team will have to assume different roles for each assignment. This also means having to work in an organized way and know how to work autonomously when appropriate.

TRANSVERSAL COMPETENCES

In this course, students are expected to acquire autonomy and organizational skills in their own work as well as basic competence in team work, and in English.

In this sense, there will be a part of the assessment of each of the corresponding learning results:

- T03.02. Assume and respect the role of each member of the team, as well as the different levels of team dependence: Each person is required to assume a specific role for each work milestone and it should be indicated at every deliverable. Additionally, during the laboratory sessions, the members of each team will respect their roles (leader, programmer and observer).
- T02.01. Work autonomously: The assessment of each laboratory session includes the observation of how each one works. Other assessments are individual, as laboratory door pitches and participation in classes.

- T02.03. Manage time and available resources. Work in an organized way: The organization of your own work is subject to the team's role. In any case, all late submissions have penalties depending on the delay, which compel students to manage the time so that their team can meet the required time requirements.
- T04.03. Use English as the language of communication and professional relationship of reference: Most of the material is in English to promote the learning of this language and it is positively valued that the reports of the laboratory work or the solutions are written in English.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Laboratory: Course project development	12	0.48	2, 1, 3, 6, 8
Problem-solving: Problem solution proposals and discussion	12	0.48	2, 1, 3, 6, 8
Theory: Attendance and participation in theory classes	22	0.88	4, 5
Type: Supervised			
Additional problem-solving activities	6	0.24	2, 3, 6, 8
Course project follow-up tutoring	6	0.24	2, 3, 6, 8
Type: Autonomous			
Laboratory work preparation and report writing	12	0.48	2, 1, 3, 6, 8, 7
Problem solving and report writing	24	0.96	2, 1, 3, 6, 8, 7
Study	26	1.04	4, 5, 8

Assessment

a) Procedure and assessment activities' plan

The assessment is continuous with specific activities (exams and assignments) throughout the course. These assessment activities generate a series of grades that determine the final grade.

The calculation of the final grade, n, follows the expression:

$$n = t \cdot 50\% + p \cdot 20\% + q \cdot 30\%$$

where t, p, and q are the grades of the theory, problem-solving, and laboratory parts, respectively.

The final grade will be, at most, a 4.5 if t, p or q <5. In other words, each part (theory, problem-solving and laboratory) must be passed separately.

The theory grade (t) is obtained from the average between the grades of the two parts. There will be two exams, one at midterm for the first part and another at the end, for the second part. A final exam lets students improve either part or both.

Problem-solving grade (p) is calculated from a weighted average of report grades. Missing reports count as 0. Worst grades are excluded, but the quantity of grades to be considered depend on the number of problem-solving sessions that are effectively done.

The grade of laboratory work (*q*) will be the result of a weighted average of the grades of the work that is done in the laboratory and the corresponding reports. Therefore, attendance at the laboratory sessions is mandatory.

b) Assessment activities schedule

The dates of the continuous assessment theory and problem-solving tests, assignment submission deadlines will be published on the Campus Virtual (CV) and may change to adapt to eventual incidents: it will always be reported previously through the CV since it is understood that it is the usual communication platform between lecturers and students outside the classroom.

c) Re-assessment procedures

Late submissions, subject to prior notice, will be accepted and penalized with a lower grade. Late submissions without prior notice or justification of force majeure will notbe accepted. A second submission period may be opened for reports that receive a negative evaluation. Unaccepted or unsubmitted assignment reports will be scored 0 and will not have the option of a second assessment.

In accordance with the coordination of the Degree and the deanship of the School of Engineering, the following activities cannot be re-assessed:

- Problem-solving, 20% of the final grade
- Laboratory, 30% of the final grade

The final exam lets students improve the grades from partial exams, independently. There is no minimum score for either part to be re-assessed in the final examination.

d) Assessment review procedure

All assessment activities can be reviewed in tutoring hours of the teaching staff can be reviewed. For the theory and problem-solving exams, a specific place, date and time will be indicated.

Should the change of a grade be agreed upon, that grade may not be modified in a later review.

No reviews will be done after the closure of the reviews of the final exam.

e) Grading

A "non-assessable" grade will be assigned to students that have not participated in any assessment activity nor have attended any laboratory sessions. In any other case, not participating in an assessment activity, including unattendances to lab, is scored with a 0 for calculating the weighted average.

Honours will be awarded to those who obtain grades greater than or equal to 9.0 in each part, up to 5% of those enrolled in descending order of final grade. They may also be granted in other cases, provided that they do not exceed 5% and the final grade is equal to or greater than 9.0.

f) Irregularities, copies and plagiarism

Copies are evidences that the work or the examination has been done in part or in full without the author's intellectual contribution. This definition also includes attempts of copying in exams and reports, and violations of the norms that ensure intellectual authorship. Plagiarisms refer to the works and texts of other authors that are passed on as their own. They are a crime against intellectual property. To avoid plagiarism, quote the sources you use when writing the corresponding work reports or examinations.

In accordance with the UAB regulations, copies or plagiarisms or any attempt to alter the assessment result, for oneself of for others, like e.g. letting other copy, imply a final grade for the corresponding part (theory, problem solving or laboratory) of 0 and, in this case, failing the course. This does not limit the right to take action against perpetrators, both in the academic field and in the criminal.

g) Assessment of repeaters

There is no differentiated treatment for repeaters but they can take advantage of their own material from the previous year provided it is informed in the corresponding reports.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Final exam (only for grade improvement)	50	2	0.08	4, 5
Half-term exams	50	4	0.16	4, 5
Laboratory reports	30	12	0.48	2, 1, 3, 6, 8, 7
Problem-solving activities' reports	20	12	0.48	2, 1, 3, 6, 8, 7

Bibliography

(The final list is shown in the Campus Virtual.)

1. M. J. Pont. (2005). *Embedded C*. Pearson Education Ltd.: Essex, England.
   It shows how embedded systems are programmed, topic also treated in the course problem-solving part and laboratory. Therefore, it's an interesting complementary material.
2. Ll. Ribas Xirgo. (2011). "Estructura bàsica d'un computador", Capítol 5 de Montse Peiron Guàrdia, Lluís Ribas i Xirgo, Fermín Sánchez Carracedo i A. Josep Velasco González: *Fonaments de computadors*. Material docent de la UOC. OpenCourseWare de la UOC.
   [http://openaccess.uoc.edu/webapps/o2/handle/10609/12901]
   It describes the state-based machine model, the algorithmic machines, and the digital systems' basic architectures that are used in the course from a different perspective, though.
3. Ll. Ribas Xirgo. (2014). *How to code finite state machines (FSMs) in C. A systematic approach.* TR01.102791 Embedded Systems. Universitat Autònoma de Barcelona.
   [https://www.researchgate.net/publication/273636602_How_to_code_finite_state_machines_FSMs_in_C_A

   Explains a method to program state machines in C which resembles the one presented in the course.
4. Brian Bailey, Grant Martin and Andrew Piziali. (2007). *ESL Design and Verification. A Prescription for Electronic System-Level Methodology*. Elsevier.
   It gives an overview of the embedded systems' synthesis process and situates the course material, thus it's a good complement.
5. Tim Wilmshurst. (2010). *Designing Embedded Systems with PIC Microcontrollers. Principles and Applications (Second Edition)*. Elsevier.
   Complementary information to that of the course on an embedded system for controlling a robot.
6. Oliver H. Bailey. (2005). *Embedded Systems Desktop Integration*. Wordware Publishing.
   Complementary information about the hardware-software communication aspect of embedded systems.