Universitat Autònoma de Barcelona

2020/2021

Embedded Systems

Code: 102791
ECTS Credits: 6

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

The proposed teaching and assessment methodology that appear in the guide may be subject to changes as a result of the restrictions to face-to-face class attendance imposed by the health authorities.

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, mainly, in English.

Teachers

Joaquín Saiz Alcaine

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 structures around the following activities:

Theory classes: Each consists of a presentation of a series of "knowledge pills" that either disseminate the necessary knowledge for the analysis and the design of embedded systems or put in context the knowledge and the abilities that are acquired during the course (for example, the explanation of how the electronic part of a car works) or state the problems that will be dealt with in the corresponding seminars.

On the class time, the corresponding pills will be published. Note that they can have different formats and will be available since. There will be a section for each class in the classroom discussion forum.

Problem-solving seminars: Discussion of small case studies (for example, control of a microwave oven) to consolidate theoretical knowledge regarding the analysis and design of embedded systems.

Practical sessions: Follow up of project development, like of a mobile robot controller stack. Each session will deal with a specific aspect regarding the implementation of embedded systems.

On problem-solving and practical sessions' time, there will be immediate assistance by the teachers through the means available at that moment or, at least, via the classroom communication forum in the Campus Virtual.

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: The course project is a teamwork and the final report and defense must detail what each member has done.

T02.01. Work autonomously: Each person must solve problems and do its corresponding part of the project on its own.

T02.03. Manage time and available resources. As for the course project, teams must work in an organized way and each member has to commit to his or her 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 project 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			
Assignment: Project development 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:

$$n = \max(x \cdot 50\% + c \cdot 25\% + p \cdot 25\%, x \cdot 75\% + p \cdot 25\%)$$

where x, c, and p are the grades of the exam, continuous assessment, and project parts, respectively.

The final grade will be, at most, a 4.5 if x or p <5. In other words, exam and project must be passed separately.

Note that, in case the continuous assessment does not improve the final grade, it is not taken into account, thus the final grade is the maximum of the grades with or without the continuous assessment.

The exam grade (x) is the grade obtained from the final exam, which is to be held on-site and which can be retaken.

Continuous assessment grade (*c*) is calculated from a weighted average of continuous assessment tests along the course. Typically, there will be three of these.

The grade awarded for the project (*p*) will be the result of a weighted average of the grades of the follow-up reports and the final report and project defense, which count up to 50%. Two follow-ups are planned, which weight 25% each. In case of a face-to-face activity, the number and procedure for the follow-ups can change.

b) Assessment activities schedule

The dates of the continuous assessment tests and 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.

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 not be 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:

- Project, 25% of the final grade

The continuous assessment can be made up by the final examination.

There is a make-up exam for the final examination, too.

d) Assessment review procedure

Assessment activities can be reviewed any time after corresponding grades are published and before the deadline for the revision of the final exam.

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, but for the make-up exam.

e) Grading

A "non-assessable" grade is assigned to students that have not participated in any assessment activity. In any other case, not participating in an assessment activity scores 0 in the weighted average computation.

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 (exam, continuous assessment or project) of 0 in the computation of the final score and 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
Continuous assessment tests (3)	25	6	0.24	2, 3, 4, 8
Final exam	50	2	0.08	4, 5
Make-up exam	50	2	0.08	4, 5
Project defense	112,5	12	0.48	2, 1, 3, 6, 8, 7
Project follow-up reports (2)	12,5	8	0.32	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.