Universitat Autònoma de Barcelona

2021/2022

Embedded Systems

Code: 104378
ECTS Credits: 6

Degree	Type	Year	Semester
2503758 Data Engineering	OT	4	0

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

Material is mostly in English.

Prerequisites

For a full understanding of the contents of the subject it is necessary to have a basic skill in programming and a good knowledge of how programs run on computers. To do this, you must have passed *Advanced Programming*, as well as *Physical Basics for Data Acquisition* to understand the principles of digitization of information. It is advisable to have completed *Data Structures* and *Parallel Programming*.

Objectives and Contextualisation

Embedded systems are responsible for very specific sets of functions that usually act as a high-level interface between applications and the physical world. Therefore, they deal with the processing of data captured by sensors and, also, with the control of motors and actuators of all kinds. Currently, there are systems embedded in almost any "thing" and, if things are complex, they can carry many such as those that may be in a car.

Because embedded systems are closely related to the physical environment around them, the algorithms they implement must meet many requirements, often very stringent and contradictory to each other. For example, having a high performance and consuming little energy.

Thus, for the development of embedded systems it is necessary to design robust algorithms that can be verified to operate in critical environments and have a development and execution cost within the margins delimited by the requirements of the application.

The aim of this course is for students to achieve the following objectives:

- Know the various areas of application of embedded systems.
- Get acquainted with present-day applications, including real-time applications.
- Understand the aspects of security, reliability, and robustness of systems.
- Know the methodology of the development of embedded systems.
- Understand the various models of computation of systems.

- Have practical skill with the design and manipulation of state-oriented computation models.
- Know the basic elements of the architectures of embedded systems.
- Be able to estimate implementation costs based on the system computation models.
- Know the problem of system partitioning and different strategies to solve it.
- Have the rudiments of programming for the implementation of systems.

Competences

- Conceive, design and implement the most appropriate data acquisition system for the specific problem to be solved.
- Demonstrate sensitivity towards ethical, social and environmental topics.
- Prevent and solve problems, adapt to unforeseen situations and take decisions.
- Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.
- Students must develop the necessary learning skills to undertake further training with a high degree of autonomy.

Learning Outcomes

1. Demonstrate sensitivity towards ethical, social and environmental topics.
2. Design the most efficient data acquisition system for a system to support autonomous driving.
3. Prevent and solve problems, adapt to unforeseen situations and take decisions.
4. Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.
5. Students must develop the necessary learning skills to undertake further training with a high degree of autonomy.

Content

1. Introduction to the development of embedded systems.
2. Embedded systems design methodology: Computation models, architectures, and design process.
3. Embedded systems development: Software synthesis, simulation, and embedded operating systems.

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 cyber-physical systems, and to explain cases that situate in context the knowledge and the abilities that are acquired during the course. The second part will be devoted to the discussion of 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 cyber-physical 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 cyber-physical systems.

As in all areas of engineering, the development of embedded systems involves making decisions based on often contradictory criteria. In the case studies, care will be taken to include ethical, social and environmental criteria. Similarly, the ability to adjust them to adapt to incidents in the development process and changes in specifications will be encouraged.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Problem-solving: Problem solution proposals and discussion	12	0.48	2, 3, 5, 4
Project: Course project development	12	0.48	2, 3, 5, 4
Theory: Attendance and participation in theory classes	22	0.88	1, 2, 5, 4
Type: Supervised			
Project: Course project follow-up reporting	6	0.24	2, 3, 5, 4
Tutoring: Additional problem-solving activities	6	0.24	1, 2, 3, 4
Type: Autonomous			
Assignment: Project development and report writing	12	0.48	1, 2, 3, 5, 4
Problem-solving: Reporting solutions to proposed problems	24	0.96	1, 2, 3, 5, 4
Theory: Study	26	1.04	2, 4

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, 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 disregarded, 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 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. For a total of 6 laboratory sessions there will be 5 follow-up reports, which count 10% each, and 1 final report and project defense (50%).

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 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 andplagiarism

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	1, 2, 4
Final exam	50%	2	0.08	1, 2, 4
Final project report and defense	12,5%	10	0.4	2, 3, 5
Follow-up project reports (5)	12,5%	10	0.4	2, 3, 5
Make-up exam	50%	2	0.08	1, 2, 4

Bibliography

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_syste

Explains a method to program state machines in C which resembles the one presented in the course.

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.

Edward A. Lee and Sanjit A. Seshia. (2017) *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, Second Edition, MIT Press.
A full course on embedded systems with much more theoretical background. (See also: https://ptolemy.berkeley.edu/)

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.

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.

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.

Oliver H. Bailey. (2005). *Embedded Systems Desktop Integration*. Wordware Publishing.
Complementary information about the hardware-software communication aspect of embedded systems.

Software

CoppeliaSim, EDU Version, Coppelia Robotics [https://www.coppeliarobotics.com/]

ZeroBrane Studio, ZeroBrane [https://studio.zerobrane.com/]

Draw.io, diagrams.net [https://app.diagrams.net/]