Universitat Autònoma de Barcelona

**2020/2021**

# Computer Architecture and Peripherals

Code: 102685
ECTS Credits: 7.5

| Degree | Type | Year | Semester |
|---|---|---|---|
| 2500895 Electronic Engineering for Telecommunication | OB | 2 | 2 |
| 2500898 Telecommunication Systems Engineering | OB | 2 | 2 |

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: Ana Ripoll Aracil
Email: Ana.Ripoll@uab.cat

## Use of Languages

Principal working language: spanish (spa)
Some groups entirely in English: No
Some groups entirely in Catalan: No
Some groups entirely in Spanish: No

## Teachers

Dolores Isabel Rexachs del Rosario
Vicente José Ivars Camáñez

## Prerequisites

There are no prerequisites. However, students should be familiar with the most basic concepts of a computer, such as functional units and digital systems (Fundamentals of Computer Science, Digital Systems and Hardware Description Languages)

## Objectives and Contextualisation

The basic objectives of the course are:
-To define the basic operating mode of an embedded system based on a microcontroller.
-The acquisition of fundamental programming skills in assembly language.
-Provide an overview of the most common peripherals, how they work, how to control them, and how to connect them to a computer-based on a microcontroller (as well as its programming in C and assembler.

## Competences

Electronic Engineering for Telecommunication

- Develop personal work habits.
- Develop thinking habits.
- Learn new methods and technologies, building on basic technological knowledge, to be able to adapt to new situations.
- Work in a team.

Telecommunication Systems Engineering

- Develop personal work habits.
- Develop thinking habits.
- Learn new methods and technologies, building on basic technological knowledge, to be able to adapt to new situations.
- Work in a team.

## Learning Outcomes

1. Apply the basics of hardware device description languages.
2. Develop applications in real time.
3. Develop critical thinking and reasoning.
4. Develop independent learning strategies.
5. Develop systemic thinking.
6. Develop the capacity for analysis and synthesis.
7. Manage available time and resources.
8. Manage available time and resources. Work in an organised manner.
9. Perform real time, concurrent, distributed and event-based programming, and design person-computer interfaces.
10. Use the basics of software design, verification and validation in the description of hardware systems based on high level hardware description languages.
11. Work cooperatively.

## Content

INTRODUCTION. COMPUTER ARCHITECTURE

- Basic concepts: microprocessors and microcontrollers. Typical architectures.
- Evolution of microprocessors: representative examples.
- Instruction Set. Assembling Programming

MICROCONTROLLERS PROGRAMMING

- Microcontrollers: internal organization, external interface. Memory map.
- Instruction Set
- Addressing Modes
- Assembly language

INPUT / OUTPUT CONCEPTS

- Input/Output Systems
- Memory-Mapped Versus Explicit Device Interfaces.
- I/O programming
- I/O processing: Programmed I/O with polling; Interrupts; Direct Memory Access

MEMORY

- Types of memories
- Organization of memory
- Memory Hierarchy

DESIGN AND DEVELOPMENT OF A MICROCONTROLLER BASED SYSTEM

- Supporting Circuits of Microprocessor
- Development Tools

## Methodology

Lectures: Theoretical content will be taught through lectures, although students will be encouraged to actively participate in the resolution of examples.

Seminars: During problem sessions, a list of exercises will be resolved. Students are encouraged to solve the problems on their own in advance. Students will also be encouraged to present their own solutions in class. During seminar sessions, topics related to the laboratory classes will be presented. The seminars will be the natural forum in which to discuss in common the development of the practical work. The mission of the

seminars is to act as a bridge between the master classes and the practical work, which will promote the capacity for analysis and synthesis, critical reasoning, and will train the student in problem-solving.

Practicum: The student will receive a dossier with the practical work to be developed during the course. This practical work is based on the design and programming of programs in assembler and in C. That allow to understand the operation of a computer and to learn the mechanisms of the Input/Output subsystem. The practices will be developed in groups of students. The practicum includes sessions of 2 hours in the laboratory, where the implementation and debugging of the programs will be carried out.

## Activities

| Title | Hours | ECTS | Learning Outcomes |
|---|---|---|---|
| Type: Directed | | | |
| Laboratory classes | 10 | 0.4 | 5, 4, 6, 8, 9 |
| Seminar classes | 10 | 0.4 | 5, 4, 6, 3, 8, 9, 11 |
| Theory classes | 29 | 1.16 | 2, 6, 3, 9 |
| Type: Autonomous | | | |
| Study | 125.5 | 5.02 | 5, 4, 6, 3, 8 |

## Assessment

(a) Evaluation process and activities
- Partial tests: As part of continuous assessment, partial tests are carried out (between 2 or 3) where the student verifies that he has acquired the knowledge and skills defined in the objectives of the subject, as well as his competences. The % of the average grade of these tests, with respect to the total of the subject, is 60%. It must be considered that, individually, the grade of each partial test does not exceed 50% of the final grade of the course.
In order to pass the subject, by means of continuous assessment, a minimum grade of 4 must be obtained in each of these partial tests and, in addition, the average grade of the marks of these partial tests must be greater than or equal to 5.
- Laboratory practice sessions: The % of the final mark of these practice sessions, with respect to the total mark of the subject, is 30%.
In order to pass the subject through continuous assessment, the final grade of these practice sessions must be greater than or equal to 5. Attendance at all practice sessions is compulsory. An absence from a practice session for officially justified reasons is tolerated, but in any case, before leaving, the teacher of the practice session must be previously consulted. Keep in mind that, the practice sessions are non-recoverable activities, therefore, suspending them with a grade lower than 5, means not being able to pass the subject.

- Practice test: In this test, the student verifies that he has achieved the knowledge acquired in the practice sessions. In order to pass the subject through continuous evaluation, the grade of this practice test must be greater than or equal to 4 and the average grade of the total grade of the practical sessions plus that of the practice test must be greater than or equal to 5.

b)Programming of evaluation activities
The schedule of evaluation activities will be given on the first day of the course. It will be made public through the Virtual Campus and on the website of the School of Engineering, in the exam section.

c)Recovery process
Any student, who has suspended the subject by the continuous assessment method, may be submitted to a recovery test. In this test, the student must examine the partial tests of the lectures and/or the practice test,

which he has done through the continuous assessment and from which he has opted for a grade lower than 5. The student may keep, for the recovery test, the grades of these continuous assessment tests that are greater than or equal to 5.
In order to pass the subject, through the process of recovery, the grades must be greater or equal than 5 and the average grade of these grades of the partial tests must be greater or equal than 5. Likewise, the grade of the practice test obtained in the recovery test must be greater or equal than 5 and the average grade of this practice test with the total grade of the practical sessions must be greater or equal than 5.

According to the academic regulations of the UAB, the student can apply for the recovery whenever it has been submitted to a set of evaluation activities that represent, at least, two-thirds of the total grade of the subject. Of these, may be presented in the recovery, those students who have as a final grade of all the evaluable activities of the subject, a grade higher than 3.5.
d) Grade review procedure
For all other assessment activities, a place, date, and time of review will be indicated allowing students to review the activity with the lecturer. In this context, students may discuss the activity grade awarded by the lecturers responsible for the subject. If students do not take part in this review, no further opportunity will be made available.

e) Qualifications

In order to pass the course with honors, the final grade must be 9.0 or higher. Because the number of students withthis distinction cannot exceed 5% of the number of students enrolled in the course, this distinction will be awarded to whoever has the highest final grade.

## Assessment Activities

| Title | Weighting | Hours | ECTS | Learning Outcomes |
|---|---|---|---|---|
| Individual practice test | 0.2 | 1 | 0.04 | 1, 2, 5, 6, 3, 7, 9, 10 |
| Individual test: Input / Output System | 0.24 | 2 | 0.08 | 1, 2, 5, 4, 6, 3, 7, 8, 9, 10 |
| Individual test: Memory System | 0.12 | 2 | 0.08 | 1, 5, 4, 6, 3, 7, 8, 10 |
| Individual test: Processor and Instruction Set | 0.24 | 2 | 0.08 | 1, 5, 6, 3, 7, 8, 9, 10 |
| Laboratory report | 0.1 | 2 | 0.08 | 1, 5, 4, 6, 3, 7, 8, 11, 10 |
| Previous work and laboratory work | 0.1 | 4 | 0.16 | 1, 5, 4, 6, 3, 7, 8, 9, 11, 10 |

## Bibliography

- "Organización y Arquitectura de Computadores. Principios de estructura y funcionamiento" William Stallings. Ed. .Megabyte. Noriega Editores
- Felix García Carballeira "Problemas resueltos de estructura de computadores". Paraninfo.
- "Computer Organization & Design. The hardware/software interface" David Patterson/John L. Hennessy. Ed. Morgan Kaufmann Publishers.
- "Computer Systems Design and Architecture" Vicent P. Heuring / Harry F. Jordan. Ed. Addison-Wesley

- Designing Embedded Hardware, 2nd Edition By John Catsoulis Publisher: O'Reilly Media Released: May 2005
-The Art of Readable Code Simple and Practical Techniques for Writing Better Code By Dustin Boswell, Trevor Foucher Publisher: O'Reilly Media Released: November 2011
-Designing Mobile Interfaces Patterns for Interaction Design By Steven Hoober, Eric Berkman Publisher: O'Reilly Media Released: November 2011
-Making Embedded Systems Design Patterns for Great Software By Elecia White Publisher: O'Reilly Medi Released: October 2011