Universitat Autònoma de Barcelona

2022/2023

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

Contact

Name: Dolores Isabel Rexachs del Rosario
Email: dolores.rexachs@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

Ana Ripoll Aracil
Betzabeth del Carmen León Otero
Vicente José Ivars Camañez

Prerequisites

Although there are no formally established prerequisites, a good knowledge of the basic operation of a computer, digital systems and programming fundamentals is highly recommended. (Fundamentals of Computer Science, Digital Systems and C language).

Objectives and Contextualisation

The basic objectives of the subject are:
• Define the operation of a general-purpose computer system and an embedded system based on microcontrollers.
• Acquire fundamental assembly language programming skills.
• Give an overview of the most common peripherals, how they work, how they are controlled, and how they connect to a computer using high-level languages, such as 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: processors and microcontrollers. Typical architectures.

- Historical evolution of processors: a generational classification with representative examples.

- Instructions Set Architecture (ISA). Assembler programming.

INPUT / OUTPUT CONCEPTS

- The Input / Output (I/O) space in computers.

- Classification of I / O devices. I / O modules.

- Programming of I / O devices.

- Synchronization with the processor: Scheduled survey, interruptions, and DMA.

- Examples of I / O devices.

MEMORY SYSTEM

- Characteristics of storage systems

- Types of memories.

- Memory organization

- Hierarchy of memory.

DESIGN OF SYSTEMS BASED ON MICROCONTROLLERS

- Microcontrollers: internal organization, external interface. Memory map.

- Tools for the design of systems based on microcontrollers: evaluation boards, simulators, and monitors.

- Debuggers: basic concepts, hardware and software techniques, advanced concepts (Background Debug Mode).

- Microcontroller programming: Assembly language and C language.

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.

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			
Laboratory classes	10	0.4	5, 4, 6, 8, 9
Seminar classes	12	0.48	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: Throughout the course, two partial tests are carried out where the student verifies that they have acquired the knowledge and skills defined in the objectives of the subject, as well as their skills worked in the classes. The percentage of the first partial test is 40% concerning the final theory mark, and the percentage of the second partial test is 60% concerning the final theory mark. The % of the final theory grade, with respect to the total grade for the subject, is 60%.

To obtain the theory mark in the continuous assessment, a mark greater than or equal to 4 must be obtained in the partial tests, and the average mark of these partial tests (taking into account the weights described), must be greater than or equal to 5.

-Laboratory practical sessions: In these practical sessions, the students put into practice, in a development environment, the knowledge acquired in the lectures and seminars. The% of the final mark of these practical sessions, with respect to the total mark of the subject, is 30%.

Attendance at all practice sessions is mandatory. An absence from a practical session for officially justified reasons is tolerated, but in any case, before being absent, the professor of the practical session must be consulted in advance. It must be taken into account that the practical sessions are non-recoverable activities, therefore, failing them with a grade lower than 5 means not being able to pass the course.

To pass the course, the final grade for these practice sessions must be greater than or equal to 5.

-Practice test: In this test, the student verifies that she has achieved the knowledge acquired in the practice sessions. The % of the mark of this practice test, concerning the total mark of the subject, is 10%.

To pass the course through continuous assessment, the mark of this practical test must be greater than or equal to 4.

To pass the subject in the continuous evaluation, each of the above items must be individually passed, that is, the final theory grade must be greater than or equal to 5, and the grade for the laboratory practice sessions and 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. For the recovery test, the student may keep grades of these continuous assessment tests that are greater than or equal to 5.
To pass the subject, through the process of recovery, the grades must be greater or equal to 4 and the average grade of these grades on partial tests must be greater or equal to 5. Likewise, the grade of the practice test obtained in the recovery test must be greater or equal to 4 and the average grade of this practice test with the total grade of the practical sessions must be greater or equal to 4.

According to the academic regulations of the UAB, the student can apply for recovery whenever it has been submitted to a set of evaluation activities representing at least, two-thirds of the total grade of the subject. Of these, those students who have, as a final grade of all evaluable activities of the subject, a grade higher than 3.5, may present themselves in the recovery.

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. No further opportunity will be made available if students do not participate in this review.

e) Qualifications

An overall grade of 5 or higher is required to pass the subject. In order to pass the course, it will be necessary to obtain a minimum grade of 5 in the activities. In case of not reaching the minimum required in any of the evaluation activities, if the calculation of the final mark is equal to or greater than 5, a 4.5 mark will be placed on the grade.

To pass the course with honors, the final grade must be 9.0 or higher. Because the number of students with this 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	10%	1	0.04	1, 2, 5, 6, 3, 7, 9, 10
Individual test 1: Processor and Instruction Set	24%	4	0.16	1, 5, 6, 3, 7, 8, 9, 10
Individual test 2: Input / Output System	24%	2	0.08	1, 2, 5, 4, 6, 3, 7, 8, 9, 10
Individual test 2: Memory System	12%	2	0.08	1, 5, 4, 6, 3, 7, 8, 10
Laboratory work	30%	2	0.08	1, 2, 5, 4, 6, 3, 7, 8, 9, 11, 10

Bibliography

- "Organización y Arquitectura de Computadores. Principios de estructura y funcionamiento" Stallings, William. Prentice Hall
- 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

- Joseph Yiu. (2011). The Definitive Guide to the ARM Cortex-M0 Elsevier. / Yiu, Joseph, Llibre en línia

Software

Code Warrior (FRDM-KL25Z)

C Compiler (gcc)

Assembler (ARM)