COURSE OUTLINE

(1) GENERAL

SCHOOL	SCHOOL OF ENGINEERING		
ACADEMIC UNIT	DEPARTMENT OF ELECTRONICS ENGINEERING		
LEVEL OF STUDIES	GRADUATE		
COURSE CODE	8002002	**SEMESTER**	2
COURSE TITLE	Embedded Systems - Real time software		

INDEPENDENT TEACHING ACTIVITIES *if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits*	**WEEKLY TEACHING HOURS**	**CREDITS (ECTS)**
Lectures	4	9
E-learning	2	
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	Scientific Area
PREREQUISITE COURSES:	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek and English
IS THE COURSE OFFERED TO ERASMUS STUDENTS	YES
COURSE WEBSITE (URL)	http://ies.teipir.gr

(2) LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B*
- *Guidelines for writing Learning Outcomes*

The objective of this course module is to provide the students with an introductory coverage of a wide scientific field, covering audio, video and broadcasting technologies.

Upon successful completion of this course, the students possess advanced knowledge, skills and competences that enable them to:

1. Design and develop the hardware and software components of an embedded system,
2. Make use of the enabling technologies for implementing embedded systems with emphasis on Microcontrollers from various vendors and the techniques for programming their integrated peripherals using IDE programming tools in high level languages as C,
3. Apply contemporary techniques for Hardware-Software co-design of embedded systems for

Real time applications using RTOS,

4. Understand the interdisciplinary nature of various application fields of Embedded Systems,
5. Design and implement an embedded system of their choice as a final project.

General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology	*Project planning and management*
Adapting to new situations	*Respect for difference and multiculturalism*
Decision-making	*Respect for the natural environment*
Working independently	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Team work	*Criticism and self-criticism*
Working in an international environment	*Production of free, creative and inductive thinking*
Working in an interdisciplinary environment	*……*
Production of new research ideas	*Others…*
	…….

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Working independently
- Team work
- Production of free, creative and inductive thinking

(3) COURSE CONTENT

Lectures:

UNIT I: Introduction – Embedded Systems

Introduction to Embedded Systems and the important metrics for design their hardware-software components. Examples for typical Embedded systems design with emphasis on Wireless Sensor Network and RFID applications

UNIT II: Implementation Technologies for Embedded Systems

Contemporary technologies for Embedded systems implementation with analysis of their comparative advantages. Modern Microcontrollers with emphasis on ARM and MSP430 MCUs. Metrics for MCU selection according to the Embedded System requirements.

UNIT III: Microcontrollers Programming techniques for Embedded systems development

Basic programming techniques for Microcontrollers based on C language. Introduction to the Integrated peripherals of Microcontrollers and the programming model followed using an Integrated Development Environment. MCU programming in practice for simple embedded systems.

UNIT IV: Real Time Operating Systems –Open Source RTOS

Real Time Operation Systems basic principles and concepts for embedded systems. Presentation of the Interrupt mechanisms and an in-depth presentation of programming interrupt handlers. Introduction to the MCU low-power modes of operation and usage with interrupt mechanisms for embedded systems implementation.

UNIT V: Embedded Systems Example 1 – Wireless Sensor Networking for Lighting Applications

Analysis of a Wireless Sensor Network node as a typical example of an Embedded System. Presentation of the various levels of abstraction used concerning the hardware and software components, as well as, the communication protocols employed from the physical to the

application level. Laboratory examples for LED lighting control based on WSNs.

UNIT VI: Embedded Systems Example 2 – Implementation of an autonomous vehicle

A project focused on the development of a small scale autonomous vehicle (UGV) or UAV is realized based on the embedded system design techniques presented.

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	• Face to face lectures in class • E-learning
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	• Use of electronic presentation with multimedia content in class, • Student support through the course webpage and the departmental e-learning platform (moodle), • Electronic communication of instructors and students, through the course webpage and by e-mail.
TEACHING METHODS *The manner and methods of teaching are described in detail.* *Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc.* *The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS*	Lectures, lab practice, homework assignments / project, study.

Activity	***Semester workload (hours)***
Lectures	52
E-learning	26
Study lecture material	52
Lab practice	26
Report on lab practice	26
Homework assignments or project and report (individual or group)	50
Study and preparation for the exams	36
Visit a company / production plant / institution	2
Course Total	**270**

STUDENT PERFORMANCE EVALUATION *Description of the evaluation procedure* *Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other* *Specifically-defined evaluation criteria are given, and if and where they are accessible to*	Final course grade = 35% x Assignments +Class/Lab participation 20%x Mid-term exam 20% x (Group) Project Report 25% x Final written exam <u>*Expected participation in learning activities:*</u> Students are expected to 1. participate in all lectures, practice sessions / laboratories and other learning activities planned for the specific

students.	semester (invited talks), 2. complete a project for the design and development of an embedded system assigned by the instructor and related to the course contents, either independently or in groups, and submit a technical report on the results by the end of the semester, 3. participate to the course final written exam. The exam covers all taught material. Students must prove mastery of the material taught and the tools used.

(5) ATTACHED BIBLIOGRAPHY

-Essential reading

1. Wayne Wolf, Computers as Components: Principles of Embedded Computing System Design
2. Steve Heath, Embedded Systems Design
3. John H. Davies, MSP430 Microcontroller Basics, NEWNES-ELSEVIER, ISBN: 978-0-7506-8276-3
4. Gadre, D. V., Programming and Customizing the AVR Microcontroller, Tziolas Publications (translated into Greek).
5. Jane Liu, Real-Time Systems
6. Bryant, O'Hallaron, Computer Systems – A Programmer's Perspective
7. Ben Ari, Principles of Concurrent and Distributed Programming

-Recommended Material

1. Silberschatz, Galvin & Gagne, Operating Systems Concepts, Chapters 4 (Processes), 6 (Scheduling), 7 (Process Synchronization) and 8 (Deadlocks)
2. Andrew Tanenbaum, Modern Operating Systems, Chapters 2 and 3
3. MSP430 Datasheets
4. ARM Datasheets
5. Datasheets for MSP430 and ARM development boards
6. Intel XScale development boards

-Relevant Journals:

1. Design Automation for Embedded Systems
2. Eurasip Journal of Embedded Systems
3. IEEE Embedded Systems Letters
4. International Journal of Embedded and Real-Time Communication Systems
5. International Journal of Embedded Systems