COURSE SYLLABUS

Academic year 2024-2025

1. Programme Information

1.1. Higher education institution	**Lucian Blaga University of Sibiu**
1.2. Faculty	Faculty of Engineering
1.3. Department	Department of Computer Science and Electrical and Electronics Engineering
1.4. Field of study	Computer Science and Information Technology
1.5. Level of study[1]	Master
1.6. Programme of study/qualification	EMBEDDED SYSTEMS

2. Course Information

2.1. Name of course	Embedded Systems Design			Code	ES.301.ZO
2.2. Course coordinator	Assoc. Prof. Macarie BREAZU, PhD				
2.3. Seminar/laboratory coordinator	conf. dr. ing. Daniel MORARIU				
2.4. Year of study[2]	2	2.5. Semester[3]	3	2.6. Evaluation form[4]	E
2.7. Course type[5]	O	2.8. The formative category of the course[6]			Z

3. Estimated Total Time

3.1. Course Extension within the Curriculum – Number of Hours per Week					
3.1.a. Lecture	3.1.b. Seminar	3.1.c. Laboratory	3.1.d. Project	3.1.e. Other	Total
2	0	1	1	0	**4**
3.2. Course Extension within the Curriculum – Total Number of Hours within the Curriculum					
3.2.a. Lecture	3.2.b. Seminar	3.2.c. Laboratory	3.2.d. Project	3.2.e. Other	Total[7]
28	0	14	14	0	**56**

Time Distribution for Individual Study[8]	**Hours**
Learning by using course materials, references and personal notes	10
Additional learning by using library facilities, electronic databases and on-site information	11
Preparing seminars / laboratories, homework, portfolios and essays	56
Tutorial activities[9]	14
Exams[10]	3
3.3. Total Individual Study Hours[11] *(NOSI$_{sem}$)*	**94**
3.4. Total Hours in the Curriculum *(NOAD$_{sem}$)*	**56**
3.5. Total Hours per Semester[12] *(NOAD$_{sem}$ + NOSI$_{sem}$)*	**150**
3.6. No. of Hours / ECTS	**25**
3.7. Number of credits[13]	**6**

4. Prerequisites (if needed)

4.1. Courses that must be successfully completed first (from the curriculum)[14]	
4.2. Competencies	

5. Conditions (where applicable)

5.1. For course/lectures[15]	video projector, internet access
5.2. For practical activities (lab/sem/pr/app)[16]	computer network, internet access

6. Specific competencies acquired[17]

Number of credits assigned to the discipline[18]			6	Credits distribution by competencies[19]
6.1. Professional competencies	PC1	approve engineering design		1
	PC2	perform project management		1
	PC3	operate open source software		1
	PC5	perform scientific research		1
	PC10	design hardware		1
6.2. Transversal competencies	TC1	apply knowledge of science, technology and engineering		1

7. Course objectives (resulted from developed competencies)

7.1. Main course objective	The students should understand the embedded system design principles, the standardized analysis and design methods imposed in that field and should be proficient in using the state-of-the-art development tools and techniques in practical projects.
7.2. Specific course objectives	• knowledge and proper operation of discipline-specific advanced concepts • usage of a variety of strategies, methods, techniques for design, implementation and evaluation • developing of a positive attitude towards (the need for validation of the theoretical aspects by) a practical application

8. Content

8.1 Lectures[20]		**Teaching methods[21]**	**Hours**
Lecture 1	Introduction. Overview. Terms and scope. Application areas. Growing importance of embedded systems.	Exposition	2
Lecture 2	Basic of developing for embedded systems. Embedded systems initialization	Exposition	2
Lecture 3	Hardware-software codesign. UML. Finite state machine as design tool.	Exposition	2
Lecture 4	Finite state machines – implementation options	Exposition	2
Lecture 5	Exceptions and interrupts. Safe and structured use of interrupts in real-time and embedded software.	Exposition	2
Lecture 6	Introduction to RTOS. Scheduling - basic concepts.	Exposition	2
Lecture 7	Tasks. Semaphores. Message queues. Pipes. Other services.	Exposition	2
Lecture 8	Scheduling of independent tasks.	Exposition	2
Lecture 9	Scheduling of dependent tasks.	Exposition	2

Lecture 10	Scheduling schemes for handling overload.	Exposition	2
Lecture 11	Execution time analysis for embedded real-time systems.	Exposition	2
Lecture 12	Modularizing an application for concurrency.	Exposition	2
Lecture 13	Synchronization and communication.	Exposition	2
Lecture 14	Common design problems. Case studies. Design errors, learning from failures.	Exposition	2
		Total lecture hours:	**28**

8.2 Practical activities

8.2.b. Laboratory		**Teaching methods**[22]	**Hours**
Laboratory 1	Studying the hardware platform to be used (e.g. HCS12 or C16x family). Familiarization with the development environment (e.g. Keil, Tasking, DAvE tools).	Experiment	2
Laboratory 2	Developing a simple application that uses parallel ports.	Experiment	2
Laboratory 3	Developing a simple application that uses timers.	Experiment	2
Laboratory 4	Developing a simple application that uses serial port.	Experiment	2
Laboratory 5	Testing some RTOS facilities to achieve parallelism.	Experiment	2
Laboratory 6	Testing some RTOS scheduling facilities.	Experiment	2
Laboratory 7	Testing some RTOS synchronization tools.	Experiment	2
		Total laboratory hours:	14

8.2.c. Project		**Teaching methods**[23]	**Hours**
Project 1	Description of projects, choosing a specific project.	Case study	2
Project 2	Designing the practical application.	Case study	2
Project 3	Implementation of the practical application.	Case study	2
Project 4	Implementation of the practical application.	Case study	2
Project 5	Testing the practical application.	Case study	2
Project 6	Testing the practical application.	Case study	2
Project 7	Proof of the application's functionality.	Case study	2
		Total project hours:	14

9. Bibliography

9.1. Recommended Bibliography	1. Peter Marwedel, "*Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*" 4th edition, ISBN-13: 978-3030609092, Springer, 2021
	2. Insup Lee et all (eds), "*Handbook of Real-Time and Embedded Systems*", ISBN 1-58488-678-1, Chapman & Hall / CRC, 2008, ULBS library code: 04/H22
	3. Giorgio C. Buttazzo, "*Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*", 3rd edition, ISBN: 1461406757, Springer, 2011
	4. Qing Li, Caroline Yao, "*Real-Time Concepts for Embedded Systems*", ISBN 1578201241, CMP Books; July 2003, ULBS library code: 04/L58
	5. Rob Williams, „*Real-Time Systems Development*", ISBN-10: 0750664711, ISBN-13: 978-0750664714, Elsevier, 2006
	6. Francis Cottet, Joëlle Delacroix, Claude Kaiser, Zoubir Mammeri, „*Scheduling in Real-Time Systems*", ISBN-13: 978-0470847664, Wiley, December 2002
	7. http://webspace.ulbsibiu.ro/macarie.breazu/ESD.htm
9.2. Additional Bibliography	1. Miro Samek, "*Practical UML Statecharts in C/C++: Event-Driven Programming for Embedded Systems*", Elsevier, 2009, ULBS library code: 04/S18

10. Conjunction of the discipline's content with the expectations of the epistemic community, professional associations and significant employers of the specific study program[24]

By periodic formal and informal meetings with members of companies in the field.

11. Evaluation

Activity Type	11.1 Evaluation Criteria	11.2 Evaluation Methods		11.3 Percentage in the Final Grade	Obs.[25]
11.4a Exam / Colloquy	• Theoretical and practical knowledge acquired (quantity, correctness, accuracy)	Tests during the semester[26]:	0%	60%	CEF
		Homework:	0%		
		Other activities[27]:	0%		
		Final evaluation:	100%		
11.4c Laboratory	• Knowledge of the equipment, how to use specific tools; evaluation of tools, processing and interpretation of results	• Written questionnaire • Oral response • Laboratory notebook, experimental works, reports, etc. • Practical demonstration		20%	CPE
11.4d Project	• The quality of the project, the correctness of the project documentation, the appropriate justification of the chosen solutions	• Self-evaluation, project presentation • Critical evaluation of a project		20%	CPE
11.5 Minimum performance standard[28] 4.50 grade at each component					

The Course Syllabus will encompass components adapted to persons with special educational needs (SEN – people with disabilities and people with high potential), depending on their type and degree, at the level of all curricular elements (skills, objectives, contents, teaching methods, alternative assessment), in order to ensure fair opportunities in the academic training of all students, paying close attention to individual learning needs.

Filling Date: 13.09.2024

Department Acceptance Date: 16.09.2024

	Academic Rank, Title, First Name, Last Name	Signature
Course Teacher	Assoc. Prof. Macarie BREAZU, PhD	
Study Program Coordinator	Prof. Arpad GELLERT, PhD	
Head of Department	Assoc. Prof. Radu George CREȚULESCU, PhD	
Dean	Prof. Maria VINȚAN, PhD	

[1] *Bachelor / Master*
[2] *1-4 for bachelor, 1-2 for master*
[3] *1-8 for bachelor, 1-3 for master*
[4] *Exam, colloquium or VP A/R - from the curriculum*
[5] *Course type: R = Compulsory course; E = Elective course; O = Optional course*
[6] *Formative category: S = Specialty; F = Fundamental; C = Complementary; I = Fully assisted; P = Partially assisted; N = Unassisted*
[7] *Equal to 14 weeks x number of hours from point 3.1 (similar to 3.2.a.b.c.)*
[8] *The following lines refer to individual study; the total is completed at point 3.37.*
[9] *Between 7 and 14 hours*
[10] *Between 2 and 6 hours*
[11] *The sum of the values from the previous lines, which refer to individual study.*
[12] *The sum (3.5.) between the number of hours of direct teaching activity (NOAD) and the number of hours of individual study (NOSI) must be equal to the number of credits assigned to the discipline (point 3.7) x no. hours per credit (3.6.)*
[13] *The credit number is computed according to the following formula, being rounded to whole neighbouring values (either by subtraction or addition*

$$No.credits = \frac{NOCpSpD \times C_C + NOApSpD \times C_A}{TOCpSdP \times C_C + TOApSdP \times C_A} \times 30\ credits$$

Where:

- NOCpSpD = Number of lecture hours / week / discipline for which the credits are calculated
- NOApSpD = Number of application hours (sem./lab./pro.) / week / discipline for which the credits are calculated
- TOCpSdP = Total number of course hours / week in the Curriculum
- TOApSdP = Total number of application hours (sem./lab./pro.) / week in the Curriculum
- C_C/C_A = Course coefficients / applications calculated according to the table

Coefficients	Course	Applications (S/L/P)
Bachelor	2	1
Master	2,5	1,5
Bachelor - foreign language	2,5	1,25

[14] *The courses that should have been previously completed or equivalent will be mentioned*
[15] *Board, video projector, flipchart, specific teaching materials, online platforms, etc.*
[16] *Computing technology, software packages, experimental stands, online platforms, etc.*
[17] *Competences from the Grids related to the description of the study program, adapted to the specifics of the discipline*
[18] *From the curriculum*
[19] *The credits allocated to the course are distributed across professional and transversal competences according to the specifics of the discipline*
[20] *Chapter and paragraph titles*
[21] *Exposition, lecture, board presentation of the studied topic, use of video projector, discussions with students (for each chapter, if applicable)*
[22] *Practical demonstration, exercise, experiment*
[23] *Case study, demonstration, exercise, error analysis, etc.*
[24] *The relationship with other disciplines, the usefulness of the discipline on the labour market*
[25] *CPE – Conditions Exam Participation; nCPE – Does Not Condition Exam Participation; CEF - Conditions Final Evaluation; N/A – not applicable*
[26] *The number of tests and the weeks in which they will be taken will be specified*
[27] *Scientific circles, professional competitions, etc.*
[28] *The minimum performance standard in the competence grid of the study program is customized to the specifics of the discipline, if applicable*