SYLLABUS

Computational models for embedded systems

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babes-Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Master
1.6. Study programme/Qualification	Distributed systems in Internet
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	**Computational models for embedded systems**					Discipline code	**MME8026**
2.2. Course coordinator					Assoc. Prof. dr. Vescan Andreea		
2.3. Seminar coordinator					Assoc. Prof. dr. Vescan Andreea		
2.4. Year of study	2	2.5. Semester	1	2.6. Type of evaluation	E	2.7. Discipline regime	Compulsory

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	**4**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**2**
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	**28**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					84
Additional documentation (in libraries, on electronic platforms, field documentation)					14
Preparation for seminars/labs, homework, papers, portfolios and essays					14
Tutorship					3
Evaluations					4
Other activities:					0
3.7. Total individual study hours	**119**				
3.8. Total hours per semester	**175**				
3.9. Number of ECTS credits	**7**				

4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	

5. Conditions (if necessary)

5.1. for the course	Video projector, Internet access
5.2. for the seminar /lab activities	Laboratory with computers; model checking tools, FSM/PN tools

6.1. Specific competencies acquired [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	• advanced programming skills in high-level programming languages • use of theoretical foundations of computer science as well as of formal models
Transversal competencies	• efficient development of organized activities in an interdisciplinary group and the development of empathetic abilities for interpersonal communications, to relate to and cooperate with various groups • use of efficient methods and techniques to learn, inform, research and develop the abilities to bring value to knowledge, to adapt at the requirements of a dynamical society and to communicate efficiently in Romanian language and in an international language

6.2. Learning outcomes

Knowledge	The student knows: • The graduate is able to present and explain methods, algorithms, paradigms and techniques used in various branches of computer science. • The graduate is able to define/identify/understand research problems in computer science.
Skills	The student is able to: • The graduate is able to write a scientific/technical report. • The graduate has the knowledge to apply model-based software development techniques.
Responsibility and autonomy:	The student has the ability to work independently to obtain: • The graduate is familiar with the concepts related to software modelling and is able to implement functional and non-functional requirements described in specific documents for the analysis and design of software systems. • The graduate has the necessary skills to use research support tools.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• know and understand fundamental concepts of embedded computation; • to develop skills in modeling embedded systems with various computational models; • to describe and verify safety and liveness properties of the system being modeled.
7.2 Specific objective of the discipline	• will acquire theoretical aspects regarding specification, designing and verification of an embedded system; • will acquire theoretical aspects regarding various computational models for embedded systems; • will know how to model a system and specify restrictions on functionalities

8. Content

8.1 Course	Teaching methods	Remarks
(1) Introduction. Model: Why? What? How? Types of systems. Requirements and Safety Requirements.	Interactive exposure Explanation, Conversation Didactical demonstration	
(2) Model checking	Interactive exposure Explanation, Conversation Didactical demonstration	
(3) Model checking	Interactive exposure Explanation, Conversation Didactical demonstration	
(4) Synchronous models Asynchronous models.	Interactive exposure Explanation, Conversation Didactical demonstration	
(5) Automotive Invited Lecture	Interactive exposure Explanation, Conversation Didactical demonstration	
(6) IoT + Real time:	Interactive exposure Explanation, Conversation Didactical demonstration	
(7) Finite State Machines (1)+(2)	Interactive exposure Explanation, Conversation Didactical demonstration	
(8) Finite State Machines (1)+(2)	Interactive exposure Explanation, Conversation Didactical demonstration	
(9) Petri nets	Interactive exposure Explanation, Conversation Didactical demonstration	
(10) Timed models	Interactive exposure Explanation, Conversation Didactical demonstration	
(11) Hybrid systems	Interactive exposure Explanation, Conversation Didactical demonstration	
(12) Dynamical systems	Interactive exposure Explanation, Conversation Didactical demonstration	
(13) Research report presentation by students	Interactive exposure Explanation, Conversation Didactical demonstration	
(14) Research report presentation by students	Interactive exposure Explanation, Conversation Didactical demonstration	
Bibliography Bibliography **Books**		

[Kat08] C. Baier, J.-P. Katoen, Principles of Model Checking, ISBN 978-0-262-02649-9, 2008
[Ari08] M. Ben-Ari, Principles of the Spin Model Checker, ISBN 978-1-84628-769-5, 2008
[Noe05] T. Noergaard, Embedded systems architecture: a comprehensive guide to engineers and programmers, Elsevier, 2005
[Hoa04] Hoare, CAR (2004) (1985), Communicating Sequential Processes, Prentice Hall International
[Pon02] M. Pont, Embedded C, Addison-Wesley, 2002

[Boo67] Taylor Booth (1967) Sequential Machines and Automata Theory, John Wiley and Sons, New York. Library of Congress Catalog Card Number: 67-25924.

Articles

[Har87] D. Harel, "Statecharts: A Visual Formalism for Complex Systems", Sci. Comput. Programming 8 (1987), 231-274
[Pet66] Petri, CA (1966) Communication with automata. DTIC Research Report AD0630125

Tutorials

During lectures/seminars/laboratories tutorials will be given for each assignment.

8.2 Seminar / laboratory	Teaching methods	Remarks
Seminar 1, 2, 3 Model Cheking Specifying safety and liveness requirements.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Seminar 4, 5, 6 Finite State Machines Project Activity Using Finite State Machines or/and PetriNets to model an embedded system	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Seminar 7 Delivery of projects (not delivered in Seminar 3 or Seminar 6)	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	

Remark:

- Students will search and use model cheking tools suitable for their Model Checking Project Activity. http://spinroot.com/spin/whatispin.html
- Students will search and use FSM/PN tools suitable for their FSM/PN Project Activity.

Bibliography

See from Courses content.

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- This course follows the IEEE and ACM Curriculla Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;
  http://www.seas.upenn.edu/~cis540/
  https://inst.eecs.berkeley.edu/~ee249/fa07/
  http://www.ict.kth.se/courses/IL2202/
  http://users.abo.fi/lmorel/MoCs/
  http://bears.ece.ucsb.edu/class/ece253/

Course content is considered very important by the software companies for improving advance embedded systems modeling and verifying skills.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	The correctness and completeness of the accumulated knowledge of computational models for embedded systems.	Written exam (in the regular session) InClass - Quiz TakeHome - Evaluation of the research report documentation+presentation	50%
10.5 Seminar/laboratory	Problem definition	Evaluation of investigation of the Problem definition	10%
	Problem definition and specification in JSpin, Show that it is possible to reach the desired end state	Evaluation of the project (modeling, verification properties)	20%
	Use Finite State Machine to model the embedded system.	Evaluation of the project (modeling, I/O, computational model used)	20%

Remark evaluation: Research Paper on a topic related to Embedded systems as extra credit for evaluation.

Remark .

- Seminar/Laboratory assignments/Projects laboratory work may not be redone in the retake session.
- Written exams can be taken during the retake session.
- Students from Previous Years to current academic year
    - All the above rules apply to students from previous years.
    - Seminar/Laboratory assignments and practical laboratory activity must be redone during didactic activity time (in the 14 weeks before normal session).
- The final grade computed with the given formula must be at least 5 in order to pass the exam.

Final grade = 50%WrittenExan (10%InClassExam+40% ReportSLR)+10%ProblemStatement+20%ProjectJSpin+20%ProjectFSM

10.6 Minimum standard of performance

Each student has to prove that:

- (s)he acquired an acceptable level of knowledge and understanding of the computational models for embedded systems;
- (s)he has the ability to establish certain connections and to use the knowledge in solving different problems.

Successful passing of the exam is conditioned by the final grade that has to be at least 5.

11. Labels ODD (Sustainable Development Goals)[2]

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable*.”.

Not applicable.

Date: ...	Signature of course coordinator Assoc. Prof. Vescan Andreea	Signature of seminar coordinator Assoc. Prof. Vescan Andreea

Date of approval: ...	Signature of the head of department Assoc.prof.phd. Adrian STERCA