DESCRIPTION OF THE MODULE

TITLE OF THE MODULE	Code
Module name: Embedded OS **Course name:** Embedded OS	**4.1.9**

Teacher(s)	Department
Coordinating: Serdiuk Serhii (serdjuksn@gmail.com) Others:	Software Tools

Study cycle	Level of the module	Type of the module
BA		compulsory

Form of delivery	Duration	Language(s)
Lab session/hands on	12 weeks	english/ukrainian

Prerequisites	
Prerequisites: - Linux OS - Linux commands - Makefiles - Linux use of libraries - Advanced C - Data interfaces in embedded systems	**Co-requisites (if necessary):** - Students need to be able to use a computer

Credits of the module	Total student workload	Contact hours	Individual work hours
3,5	105	35	70

Aim of the module (course unit): competences foreseen by the study programme

- Students are able to program the Linux shell
- Students are able to perform tasks to configure and administer Linux
- Students are able to send and receive data from a spi or i2c sensor, attached to an embedded system running Linux (/dev/mem implementation)
- Students are able to write a Linux device driver, conform the available Linux sysfs functions

Learning outcomes of module (course unit)	Teaching/learning methods	Assessment methods
To know the principles of the Linux shell programming	**Classroom teaching**	exam, applied task, labs reports
To know the principles of the /dev/mem file in Linux	**Classroom teaching**	exam, applied task
To know the Linux device driver sysfs and spi interface	**Classroom teaching**	exam, case-study
Cross-compile a Linux kernel (with some changes)	**Classroom teaching**	report on student integrated project
- File and file system structure control - Process control - Distribution of user rights - Read some I2C and SPI data - Write a SPI sysfs driver	**Hands on lab session**	report on student integrated project, labs reports

Generic competences		Learning outcomes of the programme	
1.	Ability for abstract thinking, analysis and synthesis	**1.1**	Know/understand various approaches and methods of system analysis and synthesize
		1.2	Be able to implement basic methods of scientific cognition in embedded systems sphere
2.	Ability to identify, pose and resolve problems	**2.1**	Know/understand various approaches and methods to solve professional tasks
		2.2	Be able to implement various technologies to solve professional tasks
3.	Skills in the use of information and communications technologies	**3.1**	Know/understand appropriate application software and communications technologies for embedded systems design
		3.2	Be able to use technology as a tool to research, design, evaluate of embedded systems
4.	Ability to design and manage projects	**4.1**	Collaborates on corporate interiors projects, including programming client needs, conceptual design, schematic design, and design development of embedded systems
		4.2	Be able to apply component models and integration platforms in the embedded systems design
Subject specific competences		**Learning outcome of the programme**	
5.	To analyse, design, build and maintain applications in a robust, secure and efficient way, choosing the most adequate paradigm and programming languages.	**5.1**	To choose, combine and exploit different programming paradigms, at the moment of building software, taking into account criteria like ease of development, efficiency, portability and maintainability.
		5.2	To use the tools of a software development environment to create and develop applications.
		5.3	To demonstrate knowledge and capacity to apply the fundamental principles and basic techniques of parallel, concurrent, distributed and real-time programming.
6.	To develop and analyse hardware and software for embedded systems.	**6.1**	To analyse, evaluate and select the most adequate hardware and software platform to support embedded and real-time applications.
		6.2	To develop specific processors and embedded systems; to develop and optimize the software of these systems.
7.	To demonstrate knowledge and comprehension about the internal operation of a computer and about the operation of communications between components of embedded systems.	**7.1**	To demonstrate knowledge and capacity to manage and maintain computer systems, services and applications.
		7.2	To demonstrate knowledge about the characteristics, functionalities and structure of the Operating Systems allowing an adequate use, management and design, as well as the implementation of applications based on its services
8.	To analyse and evaluate computer architectures including parallel and distributed platforms, and develop and optimize software for these platforms.	**8.1**	To develop and analyse software for systems based on microprocessors and its interfaces with users and other devices.
		8.2	To design and implement system and communications software.
		8.3	To design and implement operating systems.

Themes	Contact work hours							Time and tasks for individual work	
	Lectures	Consultation	Seminars	Practical work	Laboratory work	Placements	**Total contact work**	**Individual work**	**Tasks**
1. To know the principles of the Linux shell programming	2				4		6	4	Writing Scripts
2.To know the principles of the /dev/mem file in Linux	2						2	8	Access to RAM
3. To know the Linux device driver sysfs and spi interface	4						4	8	The Linux device driver programming
4. Cross-compile a Linux kernel (with some changes)	4				3		7	8	Application of gss
5. File and file system structure control					3		3	8	Configure and administer Linux
6. Process control					2		2	8	
7. Distribution of user rights					1		1	4	
8. Read some I2C and SPI data					4		4	8	Embedded data communication
9. Write a SPI sysfs driver					6		6	14	The Linux device driver programming

Assessment strategy	Weight in %	Deadlines	Assessment criteria
Products and performance assessments	50		All labs should be passed
Written and Oral Exam	50		Grade A (excellent) - clarity of expression – excellent, confident delivery, practical tasks – full done. Grade B (good) – clarity of expression – good, thoughts and ideas clearly expressed, practical tasks - well done. Grade C (good) - clarity of expression – well-placed, delivery is fluctuate, practical tasks - well done. Grade D (passed) - clarity of expression – poor, delivery is fluctuate, practical tasks done with mistakes. Grade E (fail) - failure in theoretical or practical tasks.

Author	Year of issue	Title	No of periodical or volume	Place of printing. Printing house or internet link
Compulsory literature				
Ivan Cibrario Bertolotti,Gabriele Manduchi	2012	Real-Time Embedded Systems: Open-Source Operating Systems Perspective		CRC Press Taylor&Francis Group

Bruno Bouyssounouse, Joseph Sifacis	2005	Embedded Systems Design: The ARTIST Roadmap for Research and Development		Springer-Verlag Berlin Heidelberg
Muhammad Ali Mazidi, Rolin D. McKinlay, Danny Causey	2009	PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18		Moscow: MK-Press, Korona-Print [RU]
Tim Wilmshurst	2008	Designing Embedded Systems with PIC Microcontrollers: Principles and Applications		Moscow: MK-Press, Korona-Vek [RU]
Serdiuk S.M., Kachan O.I.	2015	Guidance for laboratory work in "Embedded systems" for students of specialty 7.05010301 "Software systems" and 7.05010302 "Software Engineering", full time study		Zaporozhye: ZNTU [UA]
Additional literature				
Bonnie Baker	2010	Real Analog Solutions for Digital Designers		Moscow: "Dodeka-XXI" [RU]
Peter Mertens	2007	Integrated data processing. Operating systems in industry		Moscow: Finansy i statistika [RU]
Christofer Negus	2004	Red Hat Linux. Bible		Moscow:Vilyams [RU]

Tempus

This project has been funded with support from the European Commission.

This publication reflects the views only of the author,

and the Commission cannot be held responsible for any use which

may be made of the information contained therein.