NATIONAL UNIVERSITY OF LIFE AND ENVIRONMENTAL SCIENCES OF UKRAINE

Machines and equipment design department

APPROVED
Faculty of Design and Engineering
"___" June 2025

COURSE SYLLABUS

Operating Systems and Programming Languages for Robots and UAVs

Field of knowledge: 13 "Mechanical Engineering"
Specialty: 133 "Industrial Engineering'
Educational program "Industrial Engineering"
Faculty of Design and Engineering
Developer: Associate Professor, Department of Machine and Equipment Design
PhD in Technical Sciences, Associate Professor Viktor KRUSHELNYTSKYI

Kyiv – 2025

Course description:

Operating Systems and Programming Languages for Robots and UAVs
(title)

The course covers the fundamentals of programming in C++, including the creation of functions, libraries, and the use of object-oriented programming. It also introduces the principles of single-board computers, operating systems, and simulators for robotic applications. The curriculum includes project development within the ROS environment, focusing on robot motion control, sensor data processing, navigation, trajectory planning, manipulator control, and data visualization. Significant attention is given to the use of conditional statements, loops, arrays, working with ADC (Analog-to-Digital Converters), and developing user interfaces for displaying information on screens.

Field of knowledge, Specialty, Education program, Degree	
Degree	*Bachelor*
Specialty	133 "Industrial Engineering"
Education program	*Industrial Engineering*
Educational and Professional Programme	*Robotics and robotic systems and complexes*
Course Description	
Type	*Elective*
General volume of hours	*150*
Number of credits ECTS	*5*
Number of modules	*2*
Course project	-
Control form	*Exam*
Course indicators for full-time and part-time forms of higher education	
Year of preparation	*4*
Semester	*8*
Lectures	*13 h.*
Practical classes	-
Lab classes	*39 h.*
Self-study	*98 h.*
Individual work	-
Number of weekly classroom hours for full-time studies	*4 h.*

1. Purpose, Competencies, and Learning Outcomes of the Course

The purpose of the course is to acquaint students with the fundamental concepts, principles, and tools used in the development and control of robots and unmanned aerial vehicles (UAVs).

Acquired Competencies:

Integral Competence (IC): The ability of a person to solve complex specialised tasks and practical problems in a certain field of professional activity or during the

learning process, which involves the application of certain theories and methods of relevant sciences and is characterized by complexity and uncertainty of conditions.

General Competencies (GC):

GC1. Ability for abstract thinking.

GC4. Ability to search, process, and analyse information from various sources.

GC5. Ability to generate new ideas (creativity).

GC6. Ability to conduct research at a certain level.

GC8. Ability to act socially responsibly and consciously.

GC10. Skills in using information and communication technologies.

GC11. Ability to work in a team.

GC13. Ability to preserve and enhance moral, cultural, and scientific values and achievements of society based on understanding the history and development patterns of the subject area, its place in the overall system of knowledge about nature and society, and in the development of society, technology, and engineering; also, ability to use various forms of physical activity for active recreation and maintaining a healthy lifestyle.

Special (Professional) Competencies (SC):

SC4. Ability to implement engineering developments in industrial mechanical engineering considering technical, organizational, legal, economic, and environmental aspects throughout the entire life cycle of a machine: from design, construction, operation, maintenance, diagnostics, to disposal.

SC6. Ability to assess the technical and economic efficiency of typical systems and their components based on the application of analytical methods, analog analysis, and available data.

SC7. Ability to make effective decisions regarding the selection of structural materials, equipment, and processes, combining theory and practice to solve engineering problems.

SC9. Ability to conduct commercial and economic activities in the field of industrial mechanical engineering.

Program Learning Outcomes (LO):

LO1. Knowledge and understanding of the principles of technological, fundamental, and engineering sciences that form the basis of industrial mechanical engineering in the relevant field.

LO2. Knowledge and understanding of mechanics and mechanical engineering and their development prospects.

LO4. Ability to perform engineering calculations to solve complex problems and practical tasks in industrial mechanical engineering.

LO7. Ability to prepare production and operate products using automated life cycle support systems.

LO10. Understanding occupational safety issues and legal aspects of engineering activities in industrial mechanical engineering, with skills in forecasting social and environmental consequences of technical tasks implementation.

LO11. Proficiency in oral and written communication with the engineering community in both the state and foreign languages.

LO13. Understanding the structures and functions of enterprises in industrial mechanical engineering.

2. Program and Structure of the Course

Titles of Content Modules and Topics	Number of hours						
	full-time						
	total	including					
		l	p	lab	ind	self-st.	weeks
Module 1. Programming language C++							
Topic 1. Writing C++ programs	41	2	-	12	-	27	1-4
Topic 2. Creating functions and libraries	22	2	-	6	-	14	4-6
Topic 3. Object-oriented programming	14	2	-	3	-	9	6-7
Total for module 1	77	6	-	21	-	50	-
Module 2. Single-board computers and operating systems							
Topic 4. Single-board computers	6	2	-	-	-	4	7-8
Topic 5. Operating systems and simulators for robotics	17	2	-	4	-	11	8-9
Topic 5. Developing projects using ROS	50	3	-	14	-	33	9-13
Total for module 2	73	7	-	18	-	48	-
Total hours	150	13	-	39	-	98	-

3. Lecture Topics

№	Topic Title	Hours
1	Writing C++ programs	2
2	Creating functions and libraries	2
3	Object-oriented programming	2
4	Single-board computers	2
5	Operating systems and simulators for robotics	2
6	Developing projects using ROS	3

4. Laboratory Topics

№	Topic Title	Hours
1	Loops and branching	2
2	Conditional operator when working with ADC	2
3	Switch-case operator	2
4	Working with arrays	2
5	Displaying information on an LCD	4
6	Creating functions	2

7	Creating libraries	2
8	Creating a library for a segment display	2
9	Object-oriented programming	3
10	Installing an operating system on a single-board computer	2
11	Simulator for robotics	2
12	Robot motion control	2
13	Processing sensor data for robot decision-making	2
14	Trajectory planning and navigation	2
15	Manipulator control	2
16	Computer vision	2
17	Data visualization in ROS	2
18	Creating ROS packages	2

5. Self-preparation works

№	Topic title	Hours
1	Analysis of programming languages for programmable logic controllers (PLCs)	12
2	Programming languages for robotics	10
3	Implementation of encryption and decryption algorithms using C++	10
4	Distributed control systems for robots	10
5	Machine learning algorithms for robots and UAVs	12
6	Development of software for the visualisation and analysis of data obtained from robots	8
7	Research and comparison of different operating systems for robots and UAVs	8
8	Analysis of robot motion planning algorithms	8
9	Process scheduling in operating systems	10
10	Operating systems in automotive engineering	10

6. Methods and tools for assessing learning outcomes: Assessment is conducted through modular tests, defense of laboratory works, and the final evaluation of learning outcomes is carried out in the form of an exam.

7. Teaching methods

The instruction of this discipline involves the use of verbal, visual, and practical teaching methods, as well as self-preparation work.

8. Assessment of learning outcomes

Students' knowledge is evaluated on a 100-point scale, which is then converted into the national grading system in accordance with the current "Regulations on Exams and Tests at NUBiP of Ukraine."

8.1. Distribution of Points by Types of Educational Activities

Type of education activity	Learning outcomes	Evaluation
Module 1. Programming language C++		
Laboratory work 1	The learning outcomes (LOs) defined by the curriculum include knowing the basics of the C++ programming language, principles of object-oriented programming, creating and using functions and libraries, including those for segment displays. Students should understand the operation of conditional statements, loops, and selection operators. They should be able to implement signal processing from ADCs, output information to liquid crystal displays, and develop software solutions for embedded systems.	**8**
Laboratory work 2		**8**
Laboratory work 3		**8**
Laboratory work 4		**8**
Laboratory work 5		**8**
Laboratory work 6		**8**
Laboratory work 7		**8**
Laboratory work 8		**8**
Laboratory work 9		**8**
Self-preparation work for module 1		**18**
Module 1 test		**10**
Total for Module 1	-	**100**
Module 2. Single-board computers and operating systems		
Laboratory work 10	The learning outcomes (LOs) defined by the curriculum include understanding the principles of single-board computers, simulators for robotics, basic concepts and tools of the ROS (Robot Operating System) environment, methods of processing data from robot sensors, control of robot and manipulator motion, trajectory planning and navigation, fundamentals of computer vision and data visualization, as well as the process of creating ROS packages.	**8**
Laboratory work 11		**8**
Laboratory work 12		**8**
Laboratory work 13		**8**
Laboratory work 14		**8**
Laboratory work 15		**8**
Laboratory work 16		**8**
Laboratory work 17		**8**
Laboratory work 18		**8**
Self-preparation work for module 2		**18**
Module 2 test		**10**
Total for Module 2	**-**	**100**
Education work		**0,7*(M1*77 + M2*73)/150 ≤ 70**
Exam		**30**
Total for the course		**(Education work+Exam) ≤ 100**

8.2. Higher Education Student Knowledge Assessment Scale

Learner's assessment, points	Grade according to the national grading system (exams/credits)
90 – 100	Excellent
74 – 89	Good
60 – 73	Satisfactorily
0 – 59	Unsatisfactorily

8.3. Evaluation policy

Deadline and re-examination policy:	Papers submitted after the deadline without good reason are given a lower grade. Modules can be retaken with the permission of the lecturer if there are good reasons (for example, illness)
Academic Integrity Policy:	Cheating during tests and exams is prohibited (including using mobile devices). All papers and essays must have correct text references to the literature used
Attendance Policy:	Class attendance is mandatory. For objective reasons (for example, illness, international internship), training can be carried out individually (online, in agreement with the dean of the faculty)

9. Educational and Methodological Support:

1. Крушельницький, В. В. *Електронний курс "Операційні системи та мови програмування роботів і БПЛА"* [Electronic resource]. Навчально-інформаційний портал НУБіП України. https://elearn.nubip.edu.ua/course/view.php?id=4228
2. Мосіюк, О. О., & Федорчук, А. Л. (2022). *Операційні системи та системне програмування: навчально-методичний посібник*. Житомир: Вид-во ЖДУ ім. Івана Франка.
3. Гаркуша, І. М. (2020). *Конспект лекцій з дисципліни "Операційні системи" для студентів галузі знань 12 "Інформаційні технології" спеціальності 126 "Інформаційні системи та технології"* (73 с.). Дніпро: НТУ «ДП».
4. Михайлов, Є. П., & Лінгур, В. М. (2019). *Навчальний посібник з дисципліни "Маніпулятори та промислові роботи" : для студентів бакалаврів*. Одеса: Одеський національний політехнічний університет.
5. Погребняк, Б. І., & Булаєнко, М. В. (2018). *Операційні системи : навч. посібник*. Харків: ХНУМГ ім. О. М. Бекетова.

10. Recommended sources of information

1. FreeRTOS. *FreeRTOS documentation*. Retrieved from https://www.freertos.org/Documentation/RTOS_book.html
2. Open Source Robotics Foundation. *ROS: Getting started*. Retrieved from https://www.ros.org/blog/getting-started/#
3. Packt Publishing. *Hands-On Embedded Programming with C++17* [GitHub repository]. Retrieved from https://github.com/PacktPublishing/Hands-On-Embedded-Programming-with-CPP-17
4. Open Source Robotics Foundation. *Ubuntu install of ROS Melodic*. Retrieved from http://wiki.ros.org/melodic/Installation/Ubuntu
5. NVIDIA Developer. *Tutorials*. Retrieved from https://developer.nvidia.com/embedded/learn/tutorials
6. Arduino. (n.d.). *Tutorials*. Retrieved from https://docs.arduino.cc/tutorials/
7. Tinkercad. *Learn how to use Tinkercad*. Retrieved from https://www.tinkercad.com/learn/circuits

8. Tinkercad. *Official guide to Tinkercad Circuits*. Retrieved from https://www.tinkercad.com/blog/official-guide-to-tinkercad-circuits
9. Arduino. *Arduino IDE*. Retrieved from https://www.arduino.cc/en/software/
10. Arduino. *Arduino language reference*. Retrieved from https://www.arduino.cc/reference/en/