Syllabus for the Course «Robotic&Cyber-Phisical Systems Technology»

1. Abstract of the discipline

The course «Robotic Cyber-Physical Systems Technologies» is aimed at developing an engineering understanding of the architecture and technologies of cyber-physical systems (CPS) as the basis for modern automated and robotic production and infrastructure solutions.

Modern engineering systems are increasingly becoming cyber-physical systems, integrating physical processes, computational algorithms, sensor systems, communication networks, and actuators. Such systems are at the core of the digital transformation of industry, logistics, transportation, energy, and urban infrastructure.

Within the framework of this discipline, students will study:

- analysis of engineering problems solved by cyber-physical systems;
- formation of the concept of CPS solutions;
- methods of systems engineering for complex technical systems;
- architecture of cyber-physical systems;
- architecture of robotic systems;
- sensor and actuator components of CPS;
- robot control systems;
- software platforms for cyber-physical systems;
- application of artificial intelligence methods in robotic systems.

The course develops students' understanding of the full conceptual cycle of CPS system development — from the formulation of the engineering problem to the architectural design of the system. Particular attention is paid to the development of systems and architectural thinking in engineers capable of designing complex integrated systems.

Large Language Models (LLM) are used in the discipline as:

- a tool for engineering analysis of the subject area;
- an intelligent assistant in system design;
- a means of reflection and critical verification of engineering solutions.

2. General characteristics of the discipline

The course belongs to the professional module of the Master's degree program «AI-Augmented Digital Systems Engineering» in the field of study 09.04.02 «Information systems and technologies».

The total volume of the course is 3 ECTS credit units (108 academic hours). The discipline is implemented in the second semester.

Structure of labor intensity:

- Theoretical classes: 15 hours;
- Practical classes: 30 hours;
- Independent work of students (SDS): 63 hours.

Form of interim assessment: credit with assessment and defense of the course project.

The training is implemented in the AI-Augmented Engineering Learning format, which involves the integration of large language models into the educational process.

3. The place of the discipline in the structure of the educational program

The discipline is included in the educational track «Intelligent Cyber-Physical Systems», which forms a consistent line of training for specialists in robotic and automated systems engineering.

Within the track's structure, this discipline occupies a central methodological position, combining infrastructure technologies for the Industrial Internet of Things (IIoT) and engineering technologies for building integrated robotic systems.

The discipline builds on the learning outcomes of the course «Industrial Internet of Things (IIoT)» and is logically linked to the following discipline of the program: «Integrated robotic systems and information systems».

While the IIoT course develops an understanding of data infrastructure and device networking, this course develops an understanding of the architecture and design principles of cyber-physical systems that integrate sensors, computations, algorithms, and actuators.

The course enables the transition:
IIoT infrastructure → CPS architecture → robotic systems → integration of robotic complexes.

4. Objectives of mastering the discipline

The aim of the course is to develop students' engineering understanding of the design principles of cyber-physical robotic systems as components of modern digital and industrial infrastructures.

Key educational outcomes of the course:

- understanding the nature and architecture of cyber-physical systems;
- mastering a problem-oriented approach to the design of engineering systems;
- developing skills in analyzing engineering problems and user needs;
- mastering the methods of systems engineering of complex technical systems;
- understanding the architecture of robotic systems;
- studying the sensory and executive components of CPS;
- mastering the principles of control of robotic systems;
- understanding the software architecture of cyber-physical systems;
- developing CPS architectural design skills;

- developing the ability to critically use LLM in engineering activities.

5. Tasks within the discipline

The main objectives of the discipline are:

- studying engineering problems solved by cyber-physical systems;
- mastering methods for formulating an engineering problem;
- studying methods for analyzing system stakeholders;
- mastering methods for identifying user needs;
- studying the principles of forming system requirements;
- mastering the methods of architectural design of CPS;
- studying the structure of robotic systems;
- mastering the principles of integrating robotic components into CPS;
- studying sensor technologies;
- developing robotics actuator systems;
- studying robot control methods;
- mastering the architecture of CPS software platforms;
- studying methods for applying artificial intelligence in robotic systems.

6. Planned learning outcomes

Know:

- the nature and architecture of cyber-physical systems;
- principles of systems engineering of complex technical systems;
- methods of analyzing engineering problems;
- methods of identifying user needs;
- the structure of requirements for engineering systems;
- the architecture of cyber-physical systems;
- the structure of robotic systems;
- principles of integration of CPS and IIoT;
- types of sensor systems;
- the architecture of actuators;
- principles of control of robotic systems;
- the architecture of CPS software platforms.

Be able to:

- analyze engineering problems;
- identify the needs of system users;
- formulate requirements for the engineering system;
- design the architecture of a cyber-physical system;
- develop a structural model of a robotic system;
- design the architecture of interaction between CPS components;
- analyze technological solutions for the implementation of a CPS;

- evaluate the applicability of robotics technologies.

Have proficiency in:

- skills in architectural analysis of engineering systems;
- methods of system design for CPS;
- skills in developing the concept of a cyber-physical system;
- methods of analyzing technological solutions;
- skills in using LLM as an engineering assistant.

7. Methodological concept of the discipline

The teaching methodology is built around the idea that a cyber-physical system is the result of a sequential engineering process, starting with problem analysis and ending with system architecture.

Methodological axis of the course:
engineering problem → system users → user needs → requirements → system architecture → technological implementation.

7.1. Problem-oriented engineering

Training begins with an analysis of the engineering problem and the tasks solved by the system. The course covers:

- production and infrastructure tasks;
- limitations of existing solutions;
- economic and technological factors.

7.2. Systems engineering

Once the problem has been formulated, students study systems engineering methods. The course examines:

- the life cycle of engineering systems;
- methods of requirements formation;
- architectural design of the system;
- decomposition of complex systems into subsystems;
- integration and verification of system components;
- risk assessment and mitigation strategies in system development.

Students gain practical experience in applying systems engineering principles to real-world CPS projects, learning to balance technical, economic, and operational constraints.

7.3. Cyber-physical systems technologies

The final stage of the course examines CPS implementation technologies:

- sensor systems (types, characteristics, and selection criteria);
- actuators (electromechanical, hydraulic, pneumatic);
- control systems (centralized vs. distributed architectures);
- software platforms for CPS (real-time operating systems, middleware);
- artificial intelligence technologies (machine learning for predictive maintenance, anomaly detection);
- communication protocols and network architectures for CPS;
- safety and security considerations in CPS design.

Practical exercises involve configuring sensor networks, programming control algorithms, and integrating components into a functional prototype.

8. The role of LLM in the educational process

Large language models are used in the discipline as a tool for engineering activities.

LLM as an analytical assistant

LLMs are used for:

- analysis of the subject area and relevant literature;
- generation of alternative engineering solutions;
- formulation of architectural concepts;
- identification of potential risks and limitations;
- comparative analysis of technological options.

LLM as a design tool

LLMs support:

- development of system architecture and component diagrams;
- analysis of technical solutions and trade-offs;
- preparation of technical documentation (requirements specifications, design descriptions);
- creation of test plans and validation procedures.

LLM as an object of critical analysis

Students analyze:

- model errors and inaccuracies in generated content;
- limitations of language models in technical domains;
- risks of automatic solution generation without human oversight;
- ethical considerations in AI-assisted engineering.

The principle of mandatory verification

Any result obtained using LLM must be verified through:

- engineering analysis (feasibility, safety, cost);
- comparison of alternative solutions;
- discussion of architectural constraints and trade‑offs;
- experimental validation (simulation or prototype testing).

The course reinforces the professional principle: trust engineering analysis and system architecture rather than model output.

9. Educational technologies

The discipline uses:

- lectures in the format of engineering analysis (case studies, real‑world examples);
- seminars and design discussions (peer review, critique sessions);
- digital lab work (simulation tools, virtual prototypes);
- architectural analysis of existing CPS solutions;
- project activities (team‑based development of a CPS concept);
- use of LLM as an intelligent assistant (with critical evaluation).

10. Differentiated assessment model

The assessment is based on three levels:

Basic level

- correct description of the engineering problem;
- understanding of CPS architecture fundamentals;
- basic concept of the system (components, functions);
- ability to use LLM for information gathering.

Advanced level

- architectural analysis of the system (subsystems, interfaces);
- justification of technological solutions (performance, cost, reliability);
- argumentation for the selection of system components;
- use of LLM for design and documentation with critical review.

Research level

- analysis of alternative architectures (comparative evaluation);
- assessment of technology limitations and trade‑offs;
- critical evaluation of LLM recommendations (accuracy, relevance, safety);
- proposal of innovative solutions based on engineering analysis.

11. Final certification

The final assessment consists of a credit test with a grade and defense of a course project. The certification includes two parts:

Theoretical part

Understanding is checked in:

- architecture of cyber-physical systems (components, data flows);
- structure of robotic systems (manipulators, mobile platforms);
- principles of systems engineering (life cycle, requirements);
- CPS design methods (modelling, simulation, testing).

Practical defense

The student presents the results of work on the course project:

- description of the engineering problem (context, constraints, impact);
- analysis of system users (stakeholder map, roles);
- system requirements (functional and non-functional);
- CPS architecture (diagrams, component specifications);
- robotic components of the system (type, functions, sensors);
- technologies used (sensors, actuators, control algorithms, software);
- LLM usage in project development (tasks, verification steps).

Curriculum schedule for the course «Technologies of Cyber-Physical Robotic Systems»

Abbreviations:

- **Lect.** — lectures (hours);
- **Pract.** — practical classes (hours);
- **SDS** — Self-Directed Study (hours).

Week	**Content**	**Lectures (hours)**	**Practice (hours)**	**SDS (h)**
1	**Theoretical block:** Introduction to CPS. Role of cyber-physical systems in modern technological transformation. Connection between IIoT, robotics, and CPS. Explanation of why CPS are the foundation of Industry 4.0. Emphasis on CPS as engineering systems combining physical processes, computation, sensors, and actuators. **Classroom work:** Analysis of real-world CPS examples (robotic	1	2	4

Week	Content	Lectures (hours)	Practice (hours)	SDS (h)
	warehouses, autonomous vehicles, smart production lines). Discussion of tasks, components, and limitations. **SDS:** Analytical review of a selected CPS example (purpose, main components, engineering problem). Project area selection for the course project (industrial production, logistics, etc.). **LLM usage:** Searching and analyzing engineering examples, formulating queries, critical evaluation of answers.			
2	**Theoretical block:** Problem-driven engineering approach. Development starts with analysis of a real engineering/economic problem. Concept of problem formulation and business-technical idea. **Classroom work:** Discussion of engineering problems from industry and logistics. Group formulation of a problem solvable by CPS. **SDS:** Description of selected engineering problem (technological and economic aspects). **Course project:** Formulation of engineering problem for the selected project area. **LLM usage:** Refining problem formulations, analyzing technological approaches.	1	2	4
3	**Theoretical block:** Concept of system stakeholders and their role in designing engineering solutions. Categories of users (operators, service engineers, managers, system integrators). **Classroom work:** Construction of a stakeholder map for the selected engineering problem. Analysis of user groups and their interests. **SDS:** Description of main user groups and their roles. **Course project:** Stakeholder analysis for the project. **LLM usage:** Generating user roles and interaction scenarios, comparing model proposals with engineering analysis.	1	2	4
4	**Theoretical block:** Transition from system users to their needs (user requirements) and use cases. Formation of operational scenarios based on needs. **Classroom work:** Development of use cases for the selected system. Discussion of user actions, system functions, and constraints. **SDS:** Description of several system use cases (typical operating situations). **Course project:** List of user needs and corresponding use cases. **LLM usage:** Generating	1	2	4

Week	Content	Lectures (hours)	Practice (hours)	SDS (h)
	additional use cases, checking completeness of user needs description.			

Description of the course project for the course «Technologies of Cyber‑Physical Robotic Systems», consistent with the course logic:

Problem‑driven engineering → Systems engineering → CPS technologies

The project is carried out throughout the semester and is a cross‑cutting element of the course. In the following semester, it is used as the basis for the system implementation stage.

1. General objective of the course project

The goal of the course project is to develop a concept for a cyber‑physical robotic system designed to solve a specific engineering or production problem. The student must go through the main stages of conceptual design of a complex engineering system — from problem analysis to development of architecture and technological concept.

As a result of the project, the student must form a holistic model of the future system, including:

- description of the engineering problem;
- analysis of users and their needs;
- formation of system requirements;
- architecture of a cyber‑physical system;
- description of robotic and information components;
- justification of the technologies used.

The course project must demonstrate the student’s ability to apply systems engineering methods to develop complex technical solutions.

2. Typical project areas

The project must relate to the field of cyber‑physical robotic systems. The following areas are recommended as starting points for concept development:

- **Robotic warehouse logistics:** automated storage and retrieval systems, inventory management with mobile robots, smart shelving solutions.
- **Autonomous transport systems:** self‑driving delivery vehicles, automated guided vehicles (AGVs) in industrial environments, last‑mile delivery robots.
- **Intelligent production lines:** collaborative robot (cobot) workcells, adaptive assembly systems, quality control with vision‑guided robots.
- **Robotics quality control systems:** automated inspection using machine vision, defect detection with AI algorithms, precision measurement systems.
- **Industrial process monitoring and control systems:** predictive maintenance with sensor networks, condition monitoring of machinery, energy management in smart factories.

- **Robotics infrastructure maintenance systems:** inspection drones for power lines, robotic systems for building facade maintenance, automated utility monitoring.
- **Agricultural robotics:** autonomous crop monitoring, precision farming with robotic platforms, automated harvesting systems.
- **Healthcare robotics:** telepresence robots, automated medication delivery, rehabilitation robotics.

It is possible to choose another field, subject to agreement with the instructor. The chosen area must demonstrate a clear need for integration between physical processes and digital control systems.

3. General structure of the course project

The project should contain the following sections:

1. Engineering problem

Description of the problem to be solved using a cyber-physical system. You must specify:

- context of the problem (industry, process, location);
- existing limitations and inefficiencies;
- consequences of the problem for the enterprise or users;
- potential benefits of solving the problem.

2. Stakeholder analysis

Identification of the main categories of system users:

- operators (front-line users);
- service engineers (maintenance personnel);
- managers (decision-makers);
- system integrators (implementation specialists);
- end users (beneficiaries of the solution).

For each group, describe:

- their interaction with the system;
- specific needs and expectations;
- level of technical expertise;
- decision-making authority.

3. User needs

Formulation of user needs that the system must satisfy. Examples include:

- automation of repetitive or dangerous operations;
- improving safety for human workers;
- reduction in the time it takes to complete operations;
- increasing the accuracy and consistency of processes;
- enabling remote monitoring and control;
- providing real-time data for decision-making.

4. System usage scenarios

Description of main operational scenarios reflecting system operation in real conditions. Each scenario should include:

- user actions (inputs to the system);
- system actions (responses and operations);
- data flows and communication;
- expected result (output or outcome);
- error conditions and recovery procedures.

5. System requirements

Based on user needs, a set of requirements is formed. The section should include:

Functional requirements:

- specific tasks the system must perform;
- performance metrics (speed, accuracy, capacity);
- interface specifications (human-machine, system-to-system);
- safety functions and emergency procedures.

Non-functional requirements:

- reliability (uptime, mean time between failures);
- safety (risk mitigation, fail-safe modes);
- performance (response time, throughput);
- scalability (ability to handle increased load);
- maintainability (ease of repair and upgrades);
- security (data protection, access control).

6. Cyber-physical system architecture

Development of an architectural model of the system. Highlight the main subsystems:

- sensory system (types and placement of sensors);
- computing infrastructure (edge/fog/cloud components);
- control system (algorithms, decision logic);
- actuators (types and specifications);
- communications infrastructure (wired/wireless networks).

Describe data flows between subsystems, including:

- sensor data acquisition;
- data processing and decision-making;
- command execution;
- status reporting and logging.

7. Robotic components of the system

Determination of the type of robotic system:

- manipulator (articulated arms, SCARA, delta);
- mobile robot (wheeled, tracked, legged);
- robotic module (specialized units for specific tasks).

Describe:

- robot functions (tasks to be performed);
- structure of the drive system (motors, transmissions);
- sensors used (position, force, vision, etc.);
- power supply and energy management;
- communication interfaces.

8. Information architecture of the system

Development of a data exchange scheme between system components. Describe:

- data sources (sensors, user inputs, external systems);
- data processing (filtering, transformation, analysis);
- data transmission channels (protocols, network topology);
- interaction between systems (APIs, message formats);
- data storage (databases, cloud services);
- user interfaces (dashboards, mobile apps, alerts).

9. Technologies used

Specification of technological solutions required for system implementation:

- sensor technologies (types, accuracy, range);
- actuators (specifications, control interfaces);
- control algorithms (PID, fuzzy logic, machine learning);
- software platforms (ROS, PLCs, cloud services);
- artificial intelligence technologies (computer vision, predictive analytics);
- communication protocols (MQTT, OPC UA, Modbus);
- safety systems (emergency stops, collision avoidance).

---

4. Step-by-step procedure for project implementation

Work on the project is carried out in stages throughout the semester:

Stage 1. Formulation of the engineering problem

- The student selects a project area and formulates the problem that will be solved by the system.
- Stage result: a document describing the problem, its context, and potential benefits of a solution.

Stage 2. Stakeholder analysis

- The system users and their roles are defined.
- Stage result: stakeholder map with user categories, needs, and interactions.

Stage 3. Defining user needs

- A list of user needs is being formed based on stakeholder analysis.
- Stage result: structured description of user needs, prioritized by importance.

Stage 4. Formulation of requirements

- User needs are translated into system requirements (functional and non-functional).
- Stage result: requirements table with clear, measurable specifications.

Stage 5. Development of system architecture

- The architecture of the cyber-physical system is being developed, including hardware and software components.
- Stage result: architectural diagram of the system with data flows and interfaces.

Stage 6. Definition of the technological concept

- Technological solutions and system components are selected based on requirements and architecture.
- Stage result: description of technologies and system components with justification for selection.

Stage 7. Formation of the final concept of the system

- Based on all the previous stages, a holistic concept of the system is formed, integrating all components and processes.
- Stage result: final document of the course project, including all sections and supporting materials.

5. Using LLM in project work

Large language models (LLM) can be used by students as a tool to support various stages of the course project. The use of LLM is instrumental — it is not a substitute for engineering thinking, but a means to enhance productivity and creativity.

Key areas of LLM application in project work:

Problem analysis and refinement:
  - generating questions for stakeholder interviews;

- identifying potential risks and limitations of existing solutions;
- formulating the engineering problem from different perspectives;
- searching for analogous solutions in other industries.

Requirements engineering:

- transforming user needs into technical requirements;
- suggesting additional requirements based on industry standards;
- checking completeness and consistency of the requirements list;
- creating templates for requirements documentation.

Architectural design:

- generating initial architectural concepts based on functional requirements;
- proposing alternative architectural patterns (centralized vs. distributed, event-driven, etc.);
- describing data flows between system components;
- creating drafts of architectural diagrams and UML-like representations.

Technology selection:

- comparing different sensor technologies for specific use cases;
- analyzing communication protocols suitable for the system;
- researching available software platforms and frameworks;
- evaluating AI/ML algorithms for specific tasks (object detection, anomaly prediction).

Documentation support:

- drafting sections of the project report;
- creating user manuals and operating procedures;
- preparing presentation materials and speaker notes;
- translating technical documentation into other languages.

Testing and validation:

- generating test cases based on requirements;
- creating checklists for system verification;
- suggesting edge cases and failure scenarios for robustness testing;
- documenting test results and findings.

6. Final defense of the project

The defense of the course project is carried out in the form of a presentation of the system concept. The student must demonstrate a comprehensive understanding of the developed solution and its engineering rationale.

Deliverables for defense:

Engineering problem statement with context and significance.
Stakeholder analysis and user needs.
System requirements (functional and non-functional).
CPS architecture (diagrams and descriptions).
Robotic system components and their integration.
Technological solutions with justification.
LLM usage log — a record of how LLM was used in the project, including:

- specific tasks assigned to LLM;

  - prompts used;
  - results obtained;
  - verification methods applied to LLM output;
  - changes made based on critical analysis of LLM suggestions.

7. Defense format

The defense includes:

- **Project presentation** (10–12 minutes):
  - clear structure (problem → solution → architecture → technologies → benefits);
  - visual support (architectural diagrams, flowcharts, mockups);
  - demonstration of key system features (simulations or prototypes if available);
  - emphasis on engineering decisions and trade-offs.
- **Q&A session** (5–7 minutes):
  - answers to technical questions from instructors;
  - discussion of alternative solutions and their limitations;
  - justification of technology choices;
  - explanation of LLM usage and verification procedures.

8. Evaluation criteria

The project evaluation is determined by the following criteria:

Criterion	Weight	Description
Problem formulation	15%	Clarity, relevance, and depth of the engineering problem analysis
Requirements quality	20%	Completeness, consistency, and measurability of system requirements
Architecture logic	25%	Soundness of architectural decisions, data flows, and component integration

Technology justification	20%	Appropriateness of selected technologies, consideration of alternatives, cost-benefit analysis
LLM usage and verification	10%	Effectiveness of LLM utilization, critical analysis of model output, proper verification procedures
Presentation quality	10%	Clarity of communication, visual materials, ability to answer questions

Grading scale:

Excellent (90–100%): All criteria fully met, innovative solutions, deep understanding of engineering principles, effective and responsible LLM usage.

- Good (75–89%): Most criteria met, solid engineering solution, appropriate LLM usage with basic verification.

- Satisfactory (60–74%): Basic requirements met, functional solution, limited LLM usage or insufficient verification.

- Unsatisfactory (< 60%): Major criteria not met, incomplete solution, lack of engineering rationale, or irresponsible LLM usage without verification.

9. Connection of the project with the next course

The course project serves as a preparatory stage for the next discipline «Integrated robotic complexes and information systems». In the following semester, students use the developed system concept as a basis for:

- detailed design and component specification;
- implementation of software and hardware components;
- system integration and interoperability testing;
- development of user interfaces and monitoring systems;
- demonstration of a working prototype in a simulated or real environment.

This continuity ensures a smooth transition from conceptual design to practical implementation, building on the foundation established in this course.