

Syllabus for the Course «Industrial Internet of Things (IIoT)»

Duration: 16 academic weeks

Classwork: 4 academic hours per week (2 lectures + 2 labs from week 7)

Independent work: 44 hours

Form of midterm assessment: exam (defense of a course project with a level assessment)

Format: System Engineering + AI-Assisted Lifecycle Modeling

1. General characteristics of the discipline

The Industrial Internet of Things (IIoT) course is part of the professional module of the Master's program and is implemented in the first semester as an introductory engineering course on the design of cyber-physical systems.

The course volume is 3 ECTS credits.

This course provides a comprehensive understanding of the development of distributed cyber-physical systems, from identifying needs and formalizing requirements to implementing a prototype, conducting tests, and assessing the technology readiness level (TRL).

The course lays the methodological foundation for subsequent courses on IIoT architecture, industrial platforms, cybersecurity, and distributed systems.

2. The place of the discipline in the program structure

The discipline serves the following functions:

- providing an introduction to systems engineering in the master's program;
- acting as an integration module between programming and systems architecture;
- establishing a methodological basis for project activities.

The course develops:

- a culture of navigating the life cycle of an engineering solution;
- skills in development documentation;
- an understanding of the technological maturity spiral;
- a foundational understanding of distributed cyber-physical systems, IIoT architecture, and industrial digital platforms.
-

3. Objectives of mastering the discipline

The objectives of mastering the discipline are:

- developing systems engineering thinking in the field of cyber-physical systems;
- mastering the logic of the IIoT product life cycle;
- developing skills in formalizing requirements and design documentation;

- mastering the basic principles of digital control, management, and data transmission;
- fostering a culture of engineering maturity and TRL assessment;
- mastering a conscious model of interaction with LLM as a tool for supporting the development life cycle.

4. Tasks within the discipline

Within the framework of the discipline, students will complete the following tasks:

- identifying needs and creating a stakeholder map;
- forming functional and non-functional requirements;
- developing a conceptual project;
- preparing technical specifications;
- designing the testing program and methodology;
- developing a prototype of a cyber-physical system on a training base (Arduino + Python);
- implementing basic data transfer between devices;
- integrating a local node into a distributed model;
- assessing the level of technological readiness;
- developing skills in using LLM for planning, documentation, and engineering reflection.

5. Planned learning outcomes

As a result of mastering the discipline, the student must:

Know:

- the stages of the life cycle of a cyber-physical system;
- the structure of technical specifications and technical design;
- the principles of requirements formation;
- the basic principles of digital control and data transmission;
- the concept of TRL and the technological maturity spiral;
- the limitations and risks of distributed systems.

Be able to:

- formalize a problem into an engineering statement;
- identify and structure requirements;
- justify the choice of technical solutions;
- develop a testing program;
- implement a prototype of a cyber-physical node;
- integrate devices through local and network channels;
- critically analyze the results obtained using LLM.

Have proficiency in:

- methodology for completing the product life cycle;
- skills in preparing design documentation;
- basic microcontroller programming skills;
- skills in integrating devices into a distributed system;

- culture of engineering argumentation;
- practice of using LLM as a tool to support engineering activities.

6. Methodological concept of the discipline

6.1. Discipline as a model for passing the life cycle

The course is structured as a model of the upward spiral of engineering maturity:

Need → Requirements → Concept → Specification → Prototype → Testing → TRL → Integration

The student completes the entire cycle through a course project (working in pairs), which simulates a real engineering environment of distributed responsibility.

6.2. The principle of “engineering maturity before architectural complexity”

In the 1st semester, priority is given to:

- the logic of systems engineering;
- formalization and documentation;
- a culture of testing;
- responsibility for decisions made.

Architectural depth (edge/fog/cloud, industrial cybersecurity, etc.) is transferred to subsequent disciplines.

6.3. The TRL spiral as a didactic framework

TRL is used as a tool:

- for project maturity assessment;
- to structure development stages;
- for reflecting on the achieved level.

By the end of the semester, the project reaches the level of a demonstration prototype (TRL 5–6).

7. The role and contribution of LLM in the educational process

The use of LLM in the discipline is instrumental and methodologically meaningful. LLM is not a subject of study and is not used as a source of ready-made technical solutions.

7.1. LLM as a life cycle management tool

LLM is used for:

- emulating stakeholder positions;
- structuring requirements;
- identifying contradictions;

- generating alternative technical solutions;
- preparing draft versions of design documentation;
- forming test checklists;
- developing user instructions;
- planning the project stages.

7.2. LLM as an intellectual opponent

At the design and defense stage:

- the model asks clarifying questions;
- identifies logical gaps;
- generates alternative approaches;
- provokes a check of the correctness of the formalization.

7.3. Principle of mandatory engineering responsibility

Any result obtained using LLM:

- is subject to independent verification;
- is accompanied by argumentation;
- is assessed for compliance with the project requirements and constraints.

The use of LLM does not relieve the student of responsibility for the correctness of the technical solution.

7.4. Formation of professional autonomy

The key pedagogical effect of using LLM:

- development of the skill of critically checking automatically generated solutions;
- development of engineering reflection;
- strengthening the ability to reasonably defend a project.

LLM is not seen as a substitute for engineering thinking, but as a tool to enhance it.

Calendar and thematic plan (1st semester)

Volume: 32 hours of lectures/seminars, 32 hours of laboratory work, 44 hours of independent work.

Form of assessment: exam (defense of course project + theory + documentation).

SDS — Self-Directed Study

Module 1. Lifecycle of IIoT Cyber-Physical Systems (Weeks 1–6)

The module is aimed at developing systems engineering thinking and understanding the life cycle of a cyber-physical product.

Know: stages of the life cycle of a cyber-physical system, principles of requirements formation, structure of the technical specifications and technical project, basics of TRL.

Be able to: identify needs, formalize requirements, justify technical solutions, develop a test program.

Master: methods of engineering decomposition of tasks, preparation of design documentation, and use of LLM as a tool for supporting the development life cycle.

Week	Topic	Lect.	Lab.	SDS
1	Introduction to IIoT and cyber-physical systems. Differences between IoT and IIoT. Examples of system classes. Product lifecycle. Introduction to TRL. LLM as a tool for engineering reflection.	2	0	4
2	Needs analysis and stakeholder mapping. Problem identification methods. LLM as a customer and operator emulator. Formulation of a problem statement for the project proposal.	2	0	4
3	Forming functional and non-functional requirements. Acceptance criteria. Using LLM to structure requirements and identify inconsistencies.	2	0	4
4	Generation and selection of technical solutions. Criteria for comparing alternatives. Conceptual design. LLM as a tool for comparative analysis.	2	0	4
5	Technical specifications and technical design. Documentation structure. LLM as an aid in structuring and verifying the logical integrity of documents.	2	0	4
6	Test program and methodology. Protocols, checklists. User documentation. TRL spiral. LLM as a tool for formalizing testing and assessing maturity.	2	0	4

Module 2: On-Premise Cyber-Physical Solutions (Weeks 7–12)

The module is aimed at mastering the basic technical principles of digital control and management on an educational platform (Arduino + Python).

Know: the operating principles of microcontrollers, sensors, and actuators.

Be able to: implement control algorithms, ensure data processing, formalize test results.

Possess: skills in developing and debugging local cyber-physical nodes and using LLM for planning, testing, and documentation.

Week	Topic	Lect.	Lab.	SDS
7	Microcontroller system architecture. Sensors and signals.	2	2	3
8	Actuators and measure-decision-act logic.	2	2	3
9	Control algorithms. Threshold circuits and state logic.	2	2	3
10	Data transfer to a PC. Processing and visualization in Python.	2	2	3
11	Reliability and error handling. Designing security algorithms.	2	2	3
12	A mini-project of a local system. Testing. Report generation. LLM as a tool for creating checklists and reports.	2	2	4

Module 3: Communication and Distributed IIoT Solutions (Weeks 13–16)

The module is aimed at mastering the principles of data exchange and integration of subsystems.

Know:

the basic principles of network interaction;

- the client-server model;
- data transfer methods.

Be able to:

- implement data transfer between devices;
- organize remote control.

Possess:

- skills in integrating local nodes into a distributed system;
- skills in using LLM to design interaction scenarios and operating procedures.

Week	Topic	Lect.	Lab.	SDS
13	Networking basics. Local Area Networks (LAN). Client-server model.	2	2	3
14	Wireless data transfer (Wi-Fi). Exchange via the Internet.	2	2	3
15	Data transfer to a remote database. Notifications and incidents (Telegram/email).	2	2	3
16	System integration. Demonstration of a distributed prototype. LLM as a tool for preparing user manuals and pilot operation procedures.	2	2	4

Final logic of the course

Stage	Result
Weeks 1–6	Concept, requirements, technical specifications, test program (TRL 2–3)
Weeks 7–12	Locally functioning prototype (TRL 4–5)
Weeks 13–16	Integrated distributed prototype (TRL 5–6)
Exam	Defense of a course project (2 people), demonstration of the system, a set of documentation, and a theoretical interview