COURSE DESCRIPTION CARD - SYLLABUS

Course name
Tools and Software for Industrial Control Systems [S1AiR2>PO2-NiOdPSS]

Course

Field of study
Automatic Control and Robotics

Year/Semester
3/5

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
elective

Number of hours

Lecture
15

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr hab. inż. Stefan Brock prof. PP
stefan.brock@put.poznan.pl

Lecturers

Prerequisites

The student has a structured knowledge of programming PLCs and digital controllers. He/She knows and understands the principles of developing programs for PLCs in selected programming languages. Has a structured knowledge of selected algorithms and data structures as well as procedural and object-oriented programming methodologies and techniques. Knows and understands in the basics of mathematical description of control systems. He/She is able to use English at a level that enables him/her to understand datasheets, user manuals and descriptions of IT tools.

Course objective

In-depth understanding of the practical problems of programming industrial control systems, including PLCs, especially in procedural and object-oriented programming, as well as learning the tools that support the programming of industrial control systems

Course-related learning outcomes

Knowledge
1. has a structured knowledge of programming techniques and tools for industrial control systems
2. has the knowledge necessary to design and commission an industrial control system

3. has a structured and theoretically based knowledge of procedural and object-oriented programming of industrial control systems
Skills
1. is able to design a simple control and monitoring system using PLCs
2. is able to program an industrial control system using procedural and object-oriented programming methods
3. is able to commission and diagnose a simple industrial control system
Social competences
1 He/she is ready to critically evaluate his/her knowledge. Understands the need for and knows the possibilities of continuous training - improving professional, personal and social competence, is able to inspire and organize the learning process of others.
2. He/she is ready to think and act in a business-like manner.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture
- Evaluation of knowledge and skills demonstrated on the written credit examination
Laboratory classes
- Current assessment of knowledge and skills during the classes

Programme content

The course content covers selected advanced PLC programming topics, including error handling and pointers, the basics of state machines, proper code documentation, procedural programming and design patterns, functions and function blocks, passing data to and from functions, creating modular code, and the basics of object-oriented programming (OOP). Part of the course covers commissioning and debugging PLC code, and the design, creation and use of custom code libraries.
Attention will also be given to the principles of software engineering for PLCs

Course topics

L1. Advanced Structured Text - error handling and pointers, basics of state machines and proper code documentation.

L2. Debugging PLC code - print debugging, forcing variables, using built-in debugging tools. Variables: elementary data types, structured and enumerated data types, user defined data types

L3. Procedural programming and design patterns for PLCs: function and function blocks, passing data to and from functions, making code modular and maintainable

L4. Reducing, Reusing, and Recycling Code - basic object-oriented programming (OOP) - function blocks, methods, and getter and setter methods.

L5. Object-oriented programming for PLCs - Encapsulation versus abstraction, Inheritance, polymorphism

L6. Libraries - design, development and use.

L7. Software Engineering for PLCs, Software Development Life Cycle, rules of SOLID programming.

Teaching methods

A) Lecture:
multimedia presentations (slides) illustrated with examples analyzed on the blackboard and fragments of program code implementing selected content described during the lecture
B) Lab:
the class will be conducted using a problem-solving approach. The student will receive an introduction to the lab, where the connection of the class topic to the lecture content will be described. Then, using the assistance of the instructor, he/she will solve subsequent problems that will be posed to him/her.

Bibliography

Basic
1. White M. T. : Mastering PLC Programming: The software engineering survival guide to automation programming
2. Antonsen, Tom Mejer: PLC Controls with Structured Text (ST), V3: IEC 61131-3 and best practice ST programming
3. Kwaśniewski, Janusz: Język tekstu strukturalnego w sterownikach SIMATIC S7-1200 i S7-1500
4. Kwaśniewski, Janusz: Sterowniki SIMATIC S7-1200 i S7-1500 w zaawansowanych systemach sterowania
Additional
1. Martin R. C. : Czysty kod. Podrecznik dobrego programisty .
2. Pratt, Gary: The Book of CODESYS: The ultimate guide to PLC and Industrial Controls programming with the CODESYS IDE and IEC 61131-3
3. Siemens: Programming Guideline for S7-1200/S7-1500

Breakdown of average student's workload

	Hours	ECTS
Total workload	90	3,00
Classes requiring direct contact with the teacher	45	1,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	45	1,50