Poznan University of Poland

Nr	**Faculty**	**coordinator**	**Topic**	**Numer of students**
1	WARiE	dr hab. inż. Stefan Brock, prof. PP	Selected methods of programming for programmable controllers	Min: 6 Max: 24
2	WARiE	dr hab. inż. Stefan Brock, prof. PP	FPGA Design	Min: 6 Max: 30
3	WTCh	dr hab. inż. Magdalena Regel-Rosocka, prof. PP dr inż. Martyna Rzelewska-Piekut	"Formation of nanoparticles of platinum group metals"	Only 4

POZNAN UNIVERSITY OF TECHNOLOGY
Faculty of Chemical Technology
Institute of Chemical Technology and Engineering ul. Berdychowo 4, 60-965 Poznan, POLAND

Mini project

"Formation of nanoparticles of platinum group metals"

SUPERVISORS

Magdalena Regel-Rosocka (D.Sc., Assoc. Prof.)
magdalena.regel-rosocka@put.poznan.pl

Martyna Rzelewska-Piekut (Ph.D, Adjunct)
martyna.rzelewska-piekut@put.poznan.pl

Pre-requisites

The student should be:

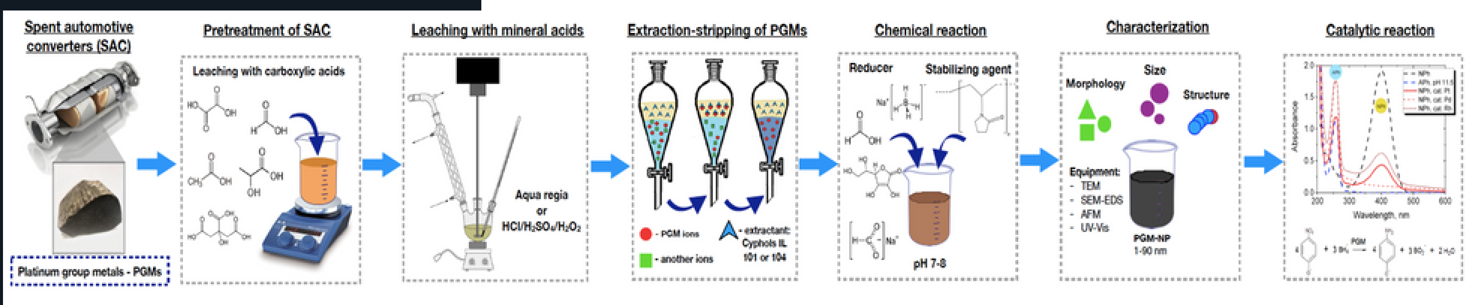
- familiar with the fundamentals of working in a chemical/ environmental laboratory
- interested in spent materials recycling
- open-minded and willing to learn
- communicative in English

General description

Pharmaceuticals pass through water treatment and could be detected in streams miles downstream from wastewater-treatment plants. Thus, the idea of catalytically active nanoparticles is to synthesize and characterize nanoparticles of platinum group metals dispersed on support which will be effective in the degradation of pharmaceuticals such as ibuprofen or antibiotics.

Tasks

A student will produce on a laboratory scale nanoparticles of platinum group metals by reduction reactions in aqueous solutions to study the effects of various parameters on the size of the particles and their catalytic properties in the reactions of degradation of pharmaceuticals, such as ibuprofen, in water.



Selected methods of programming for programmable controllers

Summary

In lectures and practical exercises, students will learn the principles of PLC application and programming. The course will cover both the popular ladder logic language and the modern structured text language. Practical classes will be held using the CodeSys simulator system, as well as physical objects and Siemens controllers in the TIAPortal environment.

PLCs are basic control and automation systems used in various industries, including Renewable energy, Robotics and Food technology.

Program content:

1. **An introduction to PLC programming**: This module provides a comprehensive overview of Programmable Logic Controllers (PLCs), their architecture, and how they function. It covers the basics of PLC programming and the different types of PLCs available in the market.
2. **Implementation of simple logic functions in ladder logic language**: This section focuses on the practical implementation of basic logic functions such as AND, OR, NOT, etc., using the ladder logic language. It includes hands-on exercises to reinforce learning.

3. **Implementation of simple control systems in ladder logic language**: This module extends the previous one by teaching students how to implement simple control systems, such as motor control circuits, using ladder logic.
4. **Using text language to program controllers: data types and structures, array operations, program structures**: This part of the course introduces text-based PLC programming languages. It covers different data types and structures, array operations, and various program structures used in these languages.
5. **Algorithms of continuous processes control, modifications of elementary PID algorithm. Methods of controller tuning**: This module delves into the algorithms used for controlling continuous processes. It includes a detailed study of the Proportional-Integral-Derivative (PID) algorithm and its modifications, along with various methods for tuning controllers for optimal performance.
6. **Practical issues of using controllers for various technological objects**: This section discusses the practical aspects of using controllers for different technological objects. It includes case studies and real-world examples to illustrate the concepts.
7. **The use of dedicated libraries (OpenPLC Control, OpenPLC MotionCotrol) and the use of programming environments (CodeSys, TIAPortal)**: The final module of the course introduces students to dedicated libraries like OpenPLC Control and OpenPLC MotionControl. It also covers the use of popular PLC programming environments like CodeSys and TIAPortal.

15 h of lectures and 30 h of laboratory classes monitored by the supervisor.

Prerequisite:

A student beginning this course should have basic knowledge of discrete process control, especially programmable controllers and industrial networks, basics of automation and metrology in the field of measurements of various physical quantities. They should be able to solve basic problems in designing automatic control systems, including: selection of the structure and parameters of controllers, selection of measurement sensors. He/she should also have the ability to acquire information from indicated sources, understand the necessity of broadening his/her competence and be ready to start cooperation within a team.

FPGA Design

An FPGA, or Field-Programmable Gate Array, stands as an integrated circuit embedded with configurable logic blocks and adaptable features that users can program and reprogram. At its core, FPGA technology hinges on adaptive hardware, possessing the distinctive capability to undergo modification post-manufacture. Arrays of configurable hardware blocks, interconnected as necessary, facilitate the creation of highly efficient, application-specific architectures. Despite its versatile potential, many designers remain unfamiliar with FPGA capabilities, evolution, and utilization. This course aims to explore FPGA fundamentals and showcase example solutions from major providers.

Programming an FPGA involves using a Hardware Description Language (HDL) to manipulate circuits based on the desired device functionalities. Unlike programming a GPU or CPU in a sequential manner, FPGA programming entails HDL utilization to design circuits and physically alter hardware configurations to meet specific requirements.

Central to any FPGA's functionality is its programmable fabric, presented as an array of programmable logic. These logic blocks house a collection of elements, including a look-up table (LUT), multiplexer, and register, all configurable to suit specific needs. The FPGA's programmable fabric can accommodate a diverse range of logical functions, scaling up to include processor cores or even multiple 'soft cores' implemented within the programmable fabric.

FPGAs find application across various domains, proving particularly valuable for implementing intelligent interfacing functions, motor control, algorithmic acceleration, high-performance computing (HPC), image and video processing, machine vision, artificial intelligence (AI), machine learning (ML), deep learning (DL), radar, beamforming, base stations, and communications. A simple example involves employing FPGAs to serve as intelligent interfaces between devices using different interface standards or communication protocols.

Schedule:

1. Programmable logic structures. History and Architectures.
2. Digital Logic – completely new application approach.
3. FPGA Design Flow. An Example designs.
4. Programmable logic design using schematic (HDL) entry design tools.
5. Test and hardware validation
6. Introduction to VHDL.
7. Implementing structures in VHDL.
8. State Machines and Sequential Logic in VHDL
9. FPGA implementation structures. Optimization and Resource Utilization.
10. Debug high speed digital systems. Troubleshooting Strategies.
11. High-Level Synthesis and C-Based FPGA Design (additionally).