COURSE DESCRIPTION CARD - SYLLABUS

Course name
Hardware applications of microcontrollers and microcomputers [S1EiT1>ASMiM]

Course

Field of study
Electronics and Telecommunications

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 (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr inż. Krzysztof Arnold
krzysztof.arnold@put.poznan.pl

Lecturers

Prerequisites

The student has an ordered and mathematically based knowledge of the basics of circuits, necessary for the operation of electrical circuits. He mastered the news on the analog circuits and fired. He has knowledge of the basics of metrology and measuring equipment. He can obtain information from the literature in Polish and English. Right beliefs from the application of electronic components and circuits. He is able to measure typical electrical parameters myth. Understands the necessity to expand his own knowledge and is responsible. Being active in class, systematically solving problems in the team.

Course objective

Presentation of the basic applications of microcontrollers and microcomputers. Knowledge and understanding the mechanisms of cooperation of the microcontroller and the microcomputer with the environment. Getting to know the principles of operation of the microcontroller in the modes of viewing the state of I/O modules and external devices and handling interrupts. Mastering the skills of programming the operation of basic systems cooperating with microcontrollers. Mastering the ability to create simple microprocessor systems.

Course-related learning outcomes

Knowledge:
Has ordered knowledge of the architecture of microcontrollers and microcomputers. Has a basic knowledge of programming microcontrollers in assembly language, their hardware applications and microprocessor systems. He knows about the development trends of modern microcontrollers and microcomputers.

Skills:
He can use company datasheets of modern microcontrollers. He can use the programming environment and write simple programs in assembly language. He can analyze and put together typical systems, built with the use of microcontrollers and microcomputers.

Social competences:
Is able to implement team projects. Has a sense of responsibility for running microcontroller and microcomputer applications.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
Written and/or oral final test verifies knowledge and understanding of the content of the lecture. Contains open-ended problem questions with different scores. Final grade for the test: less than 50% of the number of points possible - 2.0; from 50% - 3.0; from 60% - 3.5; from 70% - 4.0; from 80% - 4.5; from 90% - 5.0.
The final laboratory grade is the arithmetic mean of the grades of completion of basic and additional tasks (preparation for subsequent tasks, behavior, commitment, skills consolidation) and grades for individual or team reports, closing tasks. Additional tasks verify skills when applying for a laboratory pass or an increase in grade. They may include a written or oral test. Scale for final grades: up to 2.75 inclusive - 2.0; more than 2.75 - 3.0; over 3.25 - 3.5; over 3.75 - 4.0; over 4.25 - 4.5; over 4.75 - 5.0. The laboratory retake includes a practical part and a written or oral test.

Programme content

Lecture: Microcontroller architecture. CPU principle. Development environment, programmers and target systems. ISP programming. Programming microcontrollers in assembler language. Organizing the stack and calling subroutines. The principle of operation, initialization and modes of operation of parallel ports. Interrupt system. Applications and handling of typical data input systems to the microcontroller. Basic applications of microcontrollers and microcomputers with support for output devices. Applications with viewing the status of parallel and serial ports. Communication of microcontrollers and microcomputers with a character terminal of a PC.

Laboratory: Introduction to development environment and the basics of assembly language programming. Using commands of microcontrollers. Working registers, memories and data transfer. Conditional and unconditional changes in the order of execution of instructions. Arithmetic and logical orders and programs. Program delays. Using the stack. Running the main program and subroutines using simulation tools. Programming the microcontroller in the target system. Working with the evaluation kit. Running parallel ports of the microprocessor and the microcomputer. Support for binary inputs in viewing mode. Support for buttons and keyboards. Applications of the port output mode to control external devices. Visualization of the status of parallel ports with the use of LEDs and displays. Launching embedded serial ports using measurement equipment. Starting communication of the slave station with a PC.

Course topics

Lecture: RISC microcontroller architecture with AVR core. CPU principle. Development environment, programmers and target systems. ISP programming. Programming AVR microcontrollers in assembler language. Organizing the stack and calling subroutines. The principle of operation, initialization and modes of operation of parallel ports. Interrupt system in AVR microcontrollers. Applications and handling of typical data input systems to the microcontroller. Basic applications of microcontrollers and microcomputers with support for output devices. Applications with viewing the status of parallel ports. The principle of operation and application of the embedded USART module. Communication of microcontrollers and microcomputers with a character terminal of a PC.

Laboratory: Introduction to the AVR Studio development environment and the basics of assembly language programming. Using commands of AVR microcontrollers. Working registers, memories and data transfer. Conditional and unconditional changes in the order of execution of instructions. Arithmetic and logical instructions and programs. Program delays. Using the stack. Running the main program and subroutines using simulation tools. Programming the microcontroller in the target system. Working with the evaluation kit. Running parallel ports of the microprocessor and the microcomputer. Support for binary inputs in viewing mode. Support for buttons and keyboards. Applications of the port output mode to control external devices. Visualization of the status of parallel ports with the use of LEDs and displays. Launching embedded USART modules using measurement equipment. Starting communication of the slave station with a PC.

Teaching methods

A lecture with a multimedia presentation, supported by a problematic discussion and examples on the blackboard, containing references to the gradual results of laboratory tasks.
Laboratory: implementation of practical problem tasks given by the teacher and verification of the results using the programming environment and development kits, comparative discussion of final solutions, possible implementation of tasks requiring cooperation of two or more teams.

Bibliography

Basic
1. Andrzej Pawluczuk: Sztuka programowania mikrokontrolerów AVR. Podstawy. Wyd. BTC, Warszawa 2006
2. Andrzej Pawluczuk: Sztuka programowania mikrokontrolerów AVR. Przykłady. Wyd. BTC, Warszawa 2007
3. Rafał Baranowski: Mikrokontrolery AVR ATmega w praktyce, Wyd. BTC, Warszawa 2005
4. ATmega16A. 8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash. Datasheet, Atmel Corporation 2014.

Additional
1. Paweł Hadam: Projektowanie systemów mikroprocesorowych, Wyd. BTC, Warszawa 2004
2. ATmega128A. 8-bit AVR Microcontroller Datasheet Complete. Atmel Corporation 2015
3. ATmega8A, mega AVR Data Sheet. 2020 Microchip Technology Inc.
4. Jacek Bogusz: Lokalne interfejsy szeregowe w systemach cyfrowych. Wyd. BTC, Warszawa 2004

Breakdown of average student's workload

	Hours	ECTS
Total workload	90	3,00
Classes requiring direct contact with the teacher	55	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	35	1,00