COURSE DESCRIPTION CARD - SYLLABUS

Course name
Programming of AVR microcontrollers [S2EiT1E-TIT>PMAVR]

Course

Field of study
Electronics and Telecommunications

Year/Semester
1/2

Area of study (specialization)
Information and Communication Technologies

Profile of study
general academic

Level of study
second-cycle

Course offered in
English

Form of study
full-time

Requirements
elective

Number of hours

Lecture
30

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

5,00

Coordinators

dr inż. Sławomir Michalak
slawomir.michalak@put.poznan.pl

Lecturers

Prerequisites

The student knows the basic symbols of digital and analog electronic components, has basic knowledge about digital electronic components and their characteristics, the basics of digital circuit theory. Shows basic knowledge about architecture of microprocessors. He knows the principles of programming and can create simple program algorithm. Carries out a check on the correct operation of the equipment and program. Uses programming tools and IDEs for selected microcontrollers. Implements, compiles and runs an extensive program for the selected microcontroller. Can use the catalog data of microcontrollers. Uses the computer to perform the assumed tasks. Demonstrates the ability to obtain information (catalog data) on the Internet. Capable of independent learning (textbooks, computer programs). Behaves actively in class, puts questions, consciously uses contacts with the teacher (e.g. as part of consultations).

Course objective

Providing students with knowledge of the basics and tools for programming AVR microcontrollers using the dedicated AVR Studio environment, providing knowledge about the next stages of design and commissioning of the microprocessor system. Developing students" skills in creating algorithms and programs in assembler language, running programs, finding and correcting errors, sending the program to the target device. Getting to know the possibilities of using the microcontroller in digital and analogue-digital systems, optimizing program code parameters and resources of the designed system.

Course-related learning outcomes

none

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The knowledge acquired during the lecture, supplemented with practical skills acquired in the laboratory, is verified by a final assessment in the form of a written answer or test. A student can obtain a maximum of 100% points, the grading scale is as follows: 0-49% grade 2, 50-59% grade 3, 60-69% grade 3.5, 70-79% grade 4, 80-89% grade 4.5 , 90-100 grade 5. After the grade has been issued and until it is approved in the system, students also have the option of individual consultation and verification of the grade (oral response or writing a short program code).
Credit for laboratory classes is based on students' reports (in writing). After each laboratory unit (performing the assigned exercise), a report (program code with description) is created. The semester grade for the laboratory is determined on the basis of all reports (arithmetic average). The correctness and scope of program implementation is assessed (mandatory and additional tasks). Students have the opportunity to receive individual consultations, verify their grades (oral answers or additional tasks) and obtain a higher grade

Programme content

The course program covers the following topics:
• Microcontroller architecture
• Development environment
• Assembler - programming language
• Program memory and data memory
• Built-in hardware resources
• Communication interfaces

Course topics

The lecture and laboratory program covers the following topics:
• Decimal and binary numbers, hexadecimal numbers, assembler, tools for assembly programming, simple calculations adding, subtracting and comparing.
• Signed numbers, Binary Code Digits BCD, packed BCD, ASCII format.
• Processor architecture, registers, SRAM, stack, stack pointer, bugs with the stack operation.
• Linear program execution and branches, macros and subroutines, jumping and branching.
• Signed numbers, Binary Code Digits BCD, packed BCD, ASCII format.
• Interrupts, interrupt vector addresses, internal and external interrupts.
• Timers and counters, 7-segment LED display, N-digit multiplexed LED display.
• Ports and peripherals, serial RS232C interface, USART registers, communication with terminal, echo.
• SPI interface, exchanging dates between SPI devices.
• I2C interface, communication with peripherals.
• D/A converters, signal generator, samples in data program memory.
• A/D converters, reference voltage, data acquisition, store in SRAM and EPROM.
• Multichannel A/D converters, free running and single conversion mode.
• Wireless communication, mini-robot controller, acquisition data from robot.
• Cooperation with GPS receiver, NMEA commands.
• SCPI commands, wireless data acquisition system with digital oscilloscope.

Teaching methods

1. Lecture: traditional lecture; multimedia presentation, illustrated with examples of assembler code programs.
2. Laboratory exercises: practical exercises at computer stations and STK500 development kit, performing tasks given by the teacher, supported by examples of solutions (multimedia presentations).

Bibliography

Basic
1. Timothy M.S., "Some assembly required : assembly language programming with the AVR microcontroller", CRC Press, 2012.
2. Crisp J., "Introduction to Microprocessors and Microcontrollers", Newnes, 2004.
Additional
1. Cluley J.C., "Minicomputer and Microprocessor Interfacing", Crane Russak, 1982.
3. Leahy W.F., "Microprocessor architecture and programming", John Wiley & Sons, 1977.
3. Furber S., "ARM System-on-Chip Architecture", Addison-Wesley Professional, 2000.

Breakdown of average student's workload

	Hours	ECTS
Total workload	0	0,00
Classes requiring direct contact with the teacher	0	0,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	0	0,00