COURSE DESCRIPTION CARD - SYLLABUS

Course name
Programming of AVR microcontrollers

Course

Field of study
Electronics and Telecommunications

Year/Semester
2/3

Area of study (specialization)

Profile of study
general academic

Level of study
First-cycle studies

Course offered in
English

Form of study
full-time

Requirements
elective

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	30	
Tutorials	Projects/seminars	

Number of credit points

5

Lecturers

Responsible for the course/lecturer:
dr inż. Sławomir Michalak (PhD Eng.)

slawomir.michalak@put.poznan.pl

Responsible for the course/lecturer:

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

Knowledge

Knows the principles of programming the AVR microcontroller, can design digital circuits with the microcontroller. Is able to program selected resources of the microcontroller, such as: memory, counters, communication interfaces.Has structured and practical knowledge in the field of analysis of technical documentation of microprocessors/microcontrollers, the use of programming tools and IDE environments, implementation, compilation and launch of complex programs for selected microcontrollers, performance control of hardware and program operation.

Skills

Is able to use the learned tools for code writing, assembly and compilation. Can run written code in the target runtime environment. Can use microcontroller resources, internal and external interrupts, send and receive data via serial bus, process acquired digital data, and control external devices.

Social competences

Has a sense of importance and responsibility for microcontroller programs in electronic and telecommunications systems, and is aware of potential threats to other people or society in the event of improper use of programs and components.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Knowledge acquired as part of the lecture, supplemented with practical skills acquired in the laboratory, is verified by passing the final test on your own (at the last lecture). The test sheet contains 10 questions, each question has four answers. Students must choose the right one. Students can get a maximum of 10 points, the scale of grades is as follows: 0-4 grade 2, 5-6 grade 3, 7-8 grade 4, 9-10 grade 5. After the grade is issued, until it is approved in the e-proto system, students also have the option of individual consultation and verification of the assessment (oral answer or writing a short program code).

Laboratory classes are awarded on the basis of student reports (in writing). The report (program code with description) is created after each laboratory unit (performing the assigned exercise). The semester grade from the laboratory is determined on the basis of all reports (arithmetic average). The correctness and scope of the program implementation are assessed (obligatory and additional tasks). Students have the option of individual consultation, verification of the grade (oral answer or additional tasks) and obtaining a higher grade

Programme content

• Decimal and binary numbers, hexadecimal numbers, assembler, tools for assembly programming, simple calculations adding, subtracting and comparing

• 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	125	5,0
Classes requiring direct contact with the teacher	70	3,0
Student's own work (literature studies, preparation for laboratory classes, preparation for exam test)[1]	55	2,0

[1] delete or add other activities as appropriate