COURSE DESCRIPTION CARD - SYLLABUS

Course name
Microprocessors systems [S1MNT1>J-SystM]

Course

Field of study
Mathematics of Modern Technologies

Year/Semester
3/6

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
30

Laboratory classes
15

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

4,00

Coordinators

dr inż. Michał Bołtrukiewicz
michal.boltrukiewicz@put.poznan.pl

Lecturers

Prerequisites

Basic knowledge of digital electronics and also in scope of C++ programming language. Can design a simple combinational circuit. Can write a simple program in C++. Observe the rules of ethics in scope use of software.

Course objective

Knowledge in scope of architecture and principles of operation of microprocessor system and also properties of microcontrollers, their programming languages and debugging tools.

Course-related learning outcomes

Knowledge:
• knows and understands issues in the field of technical sciences, including automation, robotics, electrical engineering and electronics to a sufficient degree [K_W04(P6S_WG)];
• knowsandunderstandstherelationshipbetweenmathematicsandmoderntechnologies[K_W05(P6S_WG)];
• knows and understands the techniques of measurement, acquisition, processing and analysis of data or signals to an advanced degree [K_W08(P6S_WG)].

Skills:
• can use mathematical tools to support and develop modern technologies used in engineering and technical sciences [K_U06(P6S_UW)];
• can choose the appropriate method and use measuring equipment to measure basic measurable quantities; can use the basic methods of processing and analyzing data or signals [K_U09(P6S_UW)];
• can use the acquired knowledge and appropriate methods and tools to solve typical engineering tasks [K_U12(P6S_UW)].

Social competences:
• is ready to deepen and expand knowledge to solve new technical problems [K_K02(P6S_KK)].

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lectures: currently estimation of student activity; final exam in a form of a test (passing over 50%);
Laboratory classes: current estimating of knowledge and skills; current estimation of ability to programming. Evaluation of prepared reports from laboratories.

Programme content

Internal architecture and principle of operation of a microprocessor and microcontroller. Memory map and microprocessor system architecture. Microprocessor addressing modes. Microprocessor programming languages. Design of microprocessor systems. External and internal peripheral systems of microcontrollers. Communication interfaces. IoT microcontrollers.

Course topics

Update: 20.06.20224
Lectures:
Internal microprocessor systems and its principle of operation. Microprocessor system configuration. Connecting and addressing memory and input/output systems. Design aspects of microprocessor systems - voltage and time standards of connected I/O devices. Single-chip microcomputers and microcontrollers with von Neumann, Harvard and RISC structures. Internal peripheral systems of microcontrollers: A/D and D/A converters, timer and counter systems, "watchdog", RTC systems, I2C, SPI and RS232C serial interfaces. Selected issues in the area of intelligent sensors and AFE systems. IoT microcontrollers. Popular runtime "platforms" with ATmega (ARDUINO) and STM32 (STM32 CUBE IDE) microcontrollers
Laboratory classes: microprocessor system configuration and programming of microcontrolers in asembler and C++ languages.

Teaching methods

Lectures: multimedia presentations expanded by examples shown on a board; activity of students is taken into consideration in final students evaluation; theoretical questions are presented in the exact reference to the practice;
Laboratory classes: detailed reviewing of particular exercises reports; realization of laboratory tasks in teams; performing experiments to familiarize with the operation of microprocessor systems; using tools to enable students to work from home; methods of education are orientated to students to motivate them to participate actively in education process by discussion and reports.

Bibliography

Basic:
• Baranowski R. Mikrokontrolery AVR AT MEGA w praktyce. Wydawnictwo BTC, Warszawa 2005;
• Kniat J. Programowanie obiektowe w języku C++. Wydawnictwo Politechniki Poznańskiej, Poznań 1995;
• Bogusz J. Lokalne interfejsy szeregowe w systemach cyfrowych. Wydawnictwo BTC, Warszawa 2004;
• Sibigtroth J.M. Zrozumieć małe mikrokontrolery, Wydawnictwo BTC, Warszawa 2003;
• Pełka R. Mikrokontrolery architektura, programowanie, zastosowania. WKiŁ, Warszawa 1999;
• Monk S., Arduino dla początkujących: podstawy i szkice, Helion, Gliwice 2014.

Additional:
• Hajduk Z., Mikrokontrolery w systemach zdalnego sterowania. Wydawnictwo BTC. Warszawa 2005;

• Horowitz P., Hill W., Sztuka elektroniki t.2. WKiŁ, Warszawa 1996;
• Mielczarek., Szeregowe interfejsy cyfrowe, Wydawnictwo Helion, Gliwice 1993;
• Tietze U., Schenk Ch. Układy półprzewodnikowe, WNT Warszawa 1996.

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	47	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	53	2,00