COURSE DESCRIPTION CARD - SYLLABUS

Course name
Microprocessor systems [S1AiR1E>SM1]

Course

Field of study
Automatic Control and Robotics

Year/Semester
2/4

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
English

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture	Laboratory classes	Other
30	30	0
Tutorials	Projects/seminars	
0	0	

Number of credit points

4,00

Coordinators

dr inż. Dominik Łuczak
dominik.luczak@put.poznan.pl

Lecturers

Prerequisites

Knowledge: A student starting this subject should have basic knowledge of electronics and basic programming. Skills: The student should have the ability to solve basic problems in the field of digital signal processing and the ability to obtain information from specified sources. He should also understand the need to expand his competences and be ready to cooperate in a team. Social competences: In addition, in the area of social competences, the student must exhibit such qualities as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

1. To provide students with basic knowledge about the architecture and programming of microcontrollers. 2. Developing students" skills to solve problems related to data processing and communication using interfaces in microprocessor electronic systems. 3. Developing the importance of knowledge of standards and recommendations related to the construction and programming of microprocessor electronic devices in students.

Course-related learning outcomes

Knowledge:

Has a structured knowledge of computer architectures, computer systems and networks and operating systems including real-time operating systems [K1_W9 (P6S_WG)].
Knows and understands to an advanced degree the theory and methods in the architecture and programming of microprocessor systems, knows and understands selected high- and low-level microprocessor programming languages; knows and understands the principle of operation of basic peripheral modules and communication interfaces used in microprocessor systems [K1_W13 (P6S_WG)].
Knows and understands typical engineering technologies, principles and techniques of construction of simple automation and robotics systems; knows and understands the principles of selection of executive systems, computational units and measurement and control elements and devices [K1_W20 (P6S_WG)].
Skills:
Can interpret with understanding the design technical documentation and simple technological diagrams of automation and robotics systems [K1_U2 (P6S_UW)].
Be able to use selected rapid prototyping tools for automation and robotics systems [K1_U13 (P6S_UW)].
Is able to select the type and parameters of the measurement system, control unit and peripheral and communication modules for the selected application and integrate them in the form of the resulting measurement and control system [K1_U22 (P6S_UW)].
Is able to construct an algorithm to solve a simple measurement and control task and implement, test and run it in a selected programming environment on a microprocessor platform [K1_U27 (P6S_UW)].
Social competences:
Is ready to critically assess his/her knowledge; understands the need for and knows the possibilities of continuous training - improving professional, personal and social competence, is able to inspire and organize the learning process of others [K1_K1 (P6S_KK)].
The graduate is aware of the need for a professional approach to technical issues, meticulous familiarization with the documentation and environmental conditions in which the equipment and its components can operate. The graduate is ready to observe the rules of professional ethics and to demand it from others, to respect the diversity of opinions and cultures [K1_K5 (P6S_KR)].

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:
a) in the scope of lectures:
based on answers to questions about the material discussed in previous lectures,
b) in the scope of the laboratory:
based on assessment of knowledge and understanding of current issues presented in the course of the subject.
c) In both forms of classes, it is possible to use Problem-Based Learning (PBL) tasks that support the current research and technical needs of the course coordinator and are supervised by the instructor, taking into account the iterative and cyclical nature of task implementation, provided that they are consistent with the course content.

Summative rating:
a) in the scope of lectures, verification of assumed learning outcomes is carried out by:
i. assessment of knowledge and skills demonstrated during the exam in the form of a test
ii. discussion of exam results.
b) in the scope of laboratory, verification of assumed learning outcomes is carried out by:
i. assessment of student's preparation for individual classes,
ii. continuous assessment, during each class (oral answers) - rewarding the increase in the ability to use known principles and methods,
iii. assessment of reports prepared partly during classes and also after their completion.
c) The summative assessment may include the results of Problem-Based Learning (PBL) assignments developed for the research and technical needs of the course coordinator and supervised by the instructor, provided they are consistent with the course curriculum.
d) Obtaining additional points for activity during classes, in particular for:
i. independent construction of an electronic module with a microprocessor and preparation of documentation
ii. effectiveness of applying the acquired knowledge while solving a given problem
iii. comments related to the improvement of teaching materials.

Programme content

This subject covers a wide range of issues related to microprocessor systems, emphasizing their construction, programming and application. Students learn both the theoretical foundations and practical skills necessary to work with microprocessor systems in various fields.

During the lectures, students become familiar with digital logic, including basic logical operations, software and hardware implementation of combinational and sequential circuits, number coding systems, bit operations, as well as the structure of the user interface and the exchange of information between a microcontroller and another device. They will also learn to design electrical diagrams and PCB mosaics for digital and microprocessor circuits.

Laboratory classes will allow students to practically apply the knowledge acquired during lectures and develop the skills necessary to design and implement microprocessor systems.

The course program provides comprehensive preparation for the design and implementation of microprocessor systems. Students will acquire theoretical knowledge and practical skills necessary to work in various fields.

Course topics

The lecture program covers the following topics:
1. Logical Operations. Basic operations of Boolean algebra, software and hardware implementation, and applications in digital circuit design.
2. Small-Scale Integration (SSI) Combinational Circuits. Fundamentals of design and implementation of simple combinational circuits. Sum of Products (SOP) and Product of Sums (POS) forms using Karnaugh maps.
3. Bitwise Operations and Data Encoding. Bitwise operations, number encoding systems, bit masks, and multi-byte data manipulation. Utilization of all bitwise operators and byte extraction techniques from arbitrary data types, merging them into new structures, operating on data fragments using structures and unions, and accessing individual bits of a variable for both reading and modifying values.
4. Combinational Circuits – Hardware and Software Implementation. Medium-Scale Integration (MSI) circuits, design, and minimization of logical expressions. Definitions and properties of multiplexers and demultiplexers with typical applications; hardware realization of combinational circuits using multiplexers, demultiplexers, combined configurations, and memory. The scope also includes software implementation of combinational circuits in C using switch statements and memory-based data structures, including arrays, unions, and bit-field structures.
5. Sequential Circuits – Components, Definitions, Structures, and Applications. Definition of sequential circuits, state machines, and characteristics of Mealy and Moore structures with their properties and typical applications. Discussion of flip-flops (construction, parameters, and functions), including the SR flip-flop with switch debouncing, the D flip-flop used as a frequency prescaler, the JK flip-flop, flip-flops with enable inputs, and edge detection circuits. Definition, structure, properties, and applications of shift registers.
6. Sequential Circuits – Implementation and Hardware Counters. Practical aspects of implementing sequential circuits, including Mealy and Moore structures. Hardware realization using D flip-flops and software implementation in C. Operation of hardware counters, including methods for increasing bit resolution, counting rising and falling edges, and calculating the total number of pulses.
7. Data Types and Arithmetic – Integers, Floating-Point Numbers, and Data Structures. Standard integer types (8, 16, 32, and 64-bit), their bit representation, value interpretation, size in bytes, and properties of operations such as integer division and modulo. Floating-point types with a focus on arithmetic issues resulting from limited precision, methods of extracting integer and fractional parts, and modulo operations for floats. Application of unions and structures, analysis of size, memory alignment, and padding, and modifying alignment rules using the #pragma pack directive. Methods for converting and scaling variable values between different numerical ranges.
8. Text Data Conversions, Text Sizes, and Preprocessor Macros. String-to-number and number-to-string conversions, special characters, macros, and operations on data structures. Analysis of differences between strlen and sizeof in the context of text and data size, methods for converting variables to text representation, and string parsing into various data types. Techniques for creating and applying macros using token pasting and stringizing for code generation at the preprocessor stage.
9. Text-Based User Interface with LCD Display. Basics of UI creation and presentation of calculation results. Connecting an LCD to a microcontroller, transmitting commands and data in 4-bit mode, creating and using custom characters, sending single characters and strings, and implementing text page switching for complex user interfaces.

10. Gray Code – Conversions and Applications. Conversion of Binary-Coded Decimal (BCD) to Reflected Binary Code (RBC/Gray Code) in both hardware and software, and reverse conversion from RBC to BCD. Exemplary applications of RBC in measurement systems, position encoders, and circuits minimizing error risk during bit state transitions.
11. DSP (Digital Signal Processor) Expansion Card. Architecture, communication, and application of DSP cards in digital systems. Full electronic design workflow, including hardware interface preparation for incremental and absolute encoders considering signal and timing requirements. Designing expansion boards for DSP controllers, utilizing processor address lines with hardware address decoders, data bus handling, and control signals (read/write lines) for proper system bus integration.
12. Digital Signal Processor (DSP) – Structure and Operation. Integration levels of digital circuits and principles of interfacing a DSP with other electronic system components, with emphasis on the memory interface. Designing hardware address decoders, implementing data write operations to peripherals, and reading data via the memory bus. Address space issues, including the distinction between logical and hardware addresses and their significance for system integration.
13. Electronic Device Design – From Concept to Digital Circuit Integration. Full design workflow, including enclosure development, graphical interface layout, and electronic design. Hardware output interfaces (transistors and relays as control elements for digital signals) and optoisolators as input interfaces. Voltage level translation techniques in digital circuits, wiring principles, and connecting components with different integration levels, including analog outputs with parallel interfaces and quadrature counter integration.
14. Selected Applications of Digital and Microprocessor Systems. Examples of functional and applicational integration.
15. Course Summary. Review of content and summary of key topics.

The laboratory course program covers the following topics:
1. Organizational session – introduction to laboratory equipment and safety rules, introduction to design environments.
2. Software and hardware implementation and verification of logical operations. Direct manipulation of microcontroller registers. Execution-time analysis.
3. Software and hardware implementation and verification of combinational circuits.
4. Software implementation and verification of bitwise operations. Splitting and assembling multibyte numbers using bit masks, unions, and data structures. Introduction to mapping structures onto hardware registers.
5. Use of multiplexers, demultiplexers, flip-flops, and memory components to implement a selected digital circuit, e.g., an expansion board for an incremental/absolute encoder for a microprocessor system.
6. Software and hardware implementation of sequential circuits, including a state machine.
7. Analysis of arithmetic in finite-precision microprocessor systems – integers and floating-point numbers, use of the FPU, computation time and precision, alignment and organization of data structures.
8. Building parts of a user interface presenting computational results – number-to-text and text-to-number conversion, data formatting, and use of preprocessor macros.
9. Implementation of a user menu based on a state machine and control of an LCD display in 4-bit mode.
10. Analysis of the microcontroller address space and design of a logical memory map for system expansion (simulation of address decoding and peripheral integration).
11. Conceptual design of a digital circuit cooperating with a microcontroller (e.g., an absolute-encoder module), including an electrical schematic and voltage-level analysis.
12. Preparation of PCB artwork for digital circuits, e.g., an expansion board for an incremental and/or absolute encoder for a microprocessor system.
13. Extension of electrical schematics and PCB artwork with a microcontroller subsystem.
14. Presentation of the final project, e.g., a microcontroller-based simple user interface for a selected internal or external peripheral device.

Teaching methods

1. Lecture: multimedia presentation illustrated with computer simulations
2. Laboratory classes: the use of STM microprocessor development modules, IDE programming environments
3. Both forms of instruction offer the opportunity to incorporate elements of Problem-Based Learning (PBL), in which students work on problems and projects defined for the research and technical needs of the course coordinator and supervised by the instructor. This approach places particular emphasis on the

iterative nature of work, encompassing problem analysis, solution design, practical verification, and systematic refinement.

Bibliography

Basic
1. Geoffrey Brown, Discovering the STM32 Microcontroller, 2016
2. Donald S. Reay, Digital Signal Processing Using the ARM Cortex M4, 2015
3. Dogan Ibrahim, Microcontroller Based Applied Digital Control, 2006
4. W. Gay, Beginning STM32 Developing with FreeRTOS, libopencm3 and GCC, APRESS, 2018.
Additional
1. D. Łuczak, A. Wójcik, DSP implementation of state observers for electrical drive with elastic coupling , Przegląd Elektrotechniczny R.92 nr 5, s. 100-105, 2016.
2. M. Szumski, Mikrokontrolery STM32 w systemach sterowania i regulacji, BTC, 2018.
3. A. Kurczyk, Mikrokontrolery STM32 dla początkujących, BTC, 2019.
4. K. Paprocki, Mikrokontrolery STM32 w praktyce, BTC, 2009.
5. Łuczak, D. Machine Fault Diagnosis through Vibration Analysis: Continuous Wavelet Transform with Complex Morlet Wavelet and Time–Frequency RGB Image Recognition via Convolutional Neural Network. Electronics 2024, 13, 452, doi:10.3390/electronics13020452.

Breakdown of average student's workload

	Hours	ECTS
Total workload	120	4,00
Classes requiring direct contact with the teacher	60	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	60	2,00