Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of Computer Engineering

Computer architecture 3 "Microprocessor systems"

Work program of the academic discipline (Syllabus)

Details of the academic discipline

Level of higher education	*First (bachelor's degree)*
Branch of knowledge	*12 Information technologies*
Specialty	*123 Computer engineering*
Educational program	*Computer Systems and Networks*
Discipline status	*Normative*
Form of education	*intramural(daytime)*
Year of education, semester	*3rd year, spring*
Scope of the discipline	*5 credits, 150 hours,* *Lectures 18 (36 hours), Laboratory 9 (18 hours)*
Semester control/ control measures	*Examination*
Lessons schedule	*According to the schedule for the spring semester of the current academic year at http://rozklad.kpi.ua*
Language of teaching	*Ukrainian*
Information about head of the course / teachers	Lecturer: *Ph.D., docent, Tkachenko Valentina Vasylivna; 050 356 42 32 mail: Tkavalivas@gmail.com* Laboratory: *Artem Volodymyrovych Kaplunov, Nikolsky Serhii Serhiovych*
Placement of the course	Lecture material: *https://bbb.comsys.kpi.ua/b/*

Program of the academic discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

The purpose and main tasks of the course are to study the theoretical foundations, principles and methods of building microprocessor systems (MPS) based on microcontrollers, the formation of knowledge about the architecture and practical programming skills of modern microcontrollers.

The purpose of the academic discipline is to form students' skills:

- perform an analytical review of technical solutions;
- to develop algorithms for the operation of MPS components, the organization of input and output processes, as well as the exchange of information between MPS units;
- develop structural, functional and principle schemes of MPS using microcontrollers;
- work with software development environments for microcontrollers STM32 Cube MX, Keil MDK-ARM, STM32 ST-LINK Utility, architecture simulators;
- program for STM32 microcontrollers;
- work with Hard Ware, detractors and flash boards.

As a result of studying the discipline, the student should obtain the following:

Knowledge:

- actual problems of the theory of microcontrollers, MPU, MPS, basic terms and definitions;
- trends in the life cycle of the development of a modern element base, Fabless model of the organization of the electronic industry, system-on-a-crystal (SoC) technology, FPGA programming;
- MPS design tools on a modern element base, including using microcontrollers and FPGAs,
- principles of construction and operation of MPS microprocessor systems;
- methods of information exchange between MPS components;
- ways of organizing information input and output processes in MPS;
- ways of organizing MPS multi-level memory;
- the current state, trends and prospects for the development of 32-bit microcontrollers for the implementation of embedded systems and in the IoT infrastructure;
- architecture and software model of the ARM CORTEX M4 processor;
- interfaces for communication with peripheral devices;
- principles of data conversion using ADC and DAC;
- tools for developing programs for STM32;
- systems research methods, comparative analysis;
- methods of finding optimal solutions;

Skills:

- use modern mathematical apparatus to solve engineering and scientific problems that arise during the development and research of computer equipment;
- use modern mathematical apparatus to solve engineering and scientific problems, in particular, those that arise during the development and research of MPS;
- carry out calculations necessary for the design and use of MPS.
- use languages of different levels to describe hardware and software;
- choose the type of microcontroller to solve the given applied problem;
- program STM32 using modern automation tools;
- work with tools when developing programs for STM32;
- debug programs for STM32 using the JTAG interface;
- to be able to develop and implement units and components of MPS on a modern software element base (FPLS).

Experience:

The design of MPS should begin with the analysis of the algorithm for solving the given problem. Initially, the command system includes commands that provide data input and output, work with subprograms, program disclosure, and enable and disable interrupts. After that, the system of commands is distributed by commands that, together with the existing ones, ensure the implementation of the given algorithm. The program model of the processor is compiled, the memory organization is specified, and functional micro algorithms for command execution are developed.

Next, the issues of information exchange with external devices (priority interrupt mode, direct memory access, software exchange mode) are considered. Based on the analysis of the developed micro algorithms of the commands and the selected means of interaction with external devices (software, firmware, hardware), the structural diagram of the MPS is specified.

Practical experience related to programming for a modern 32-bit STM32F4 microcontroller based on the ARM CORTEX-M4 processor in assembler and the C programming language using software emulators of the ARM Cortex M4 processor; development of software and hardware on the STM32F407VG Discovery board using STM32 Cube MX, Keil MDK-ARM microcontroller programming environments.

Studying the discipline provides the following general and professional competencies:

- GC2. Ability to learn and master modern knowledge.

- PC1. Ability to apply legislative and regulatory frameworks, as well as national and international requirements, practices and standards, to implementation of professional activities in Computer Engineering field.
- PC5. Ability to use automation design tools and systems for the development of components of Computer Systems and Networks, Internet applications, Cyber-Physical Systems also.
- PC7. Ability to use and implement new technologies, including Smart, Mobile, Green and Secure Computing Technologies, to take part in the modernization and reconstruction of Computer Systems and Networks, a variety of Embedded and Distributed applications, in particular with the aim of increasing their efficiency.
- PC13. Ability to solve problems in the field of Computer and Information Technology, to determine the limitations of these Technologies.
- PC14. Ability to Design Systems and their Components taking into account all aspects of their life cycle and task, including creation, configuration, operation, maintaining and disposal.
- PC16. Ability to design, implement and administer High Performance parallel and distributed computer systems and their components using FPGA modules and Systems of Automated Design.

In accordance with aforecited, students will receive the following program learning outcomes:

- PLO1. Know and understand the scientific principles that underpin the functioning of Computer, Systems and Networks.
- PLO3. Know the latest technologies in Computer Science Engineering.
- PLO4. Know and understand the impact of Technical Solutions in a social, economic, social and environmental context.
- PLO5. Have knowledge of the basics of Economics and Project Management.
- PLO7. Be able to solve speciality-specific problems of methods of analysis and synthesis.
- PLO10. Be able to develop Software for Embedded and Distributed Applications, Mobile and Hybrid Systems; evaluate, operate specialty-specific equipment.
- PLO11. Be able to search for information in various sources to solve problems of Computer Engineering.
- PLO12. Be able to work effectively both individually and in a team.
- PLO13. Be able to identify, classify and describe the operation of Computer Systems and their components.
- PLO14. Be able to combine Theory and Practice, as well as make decisions and develop a Strategy for solving speciality-specific problems, taking into account Universal Values, Social, State and Industrial Interests.
- PLO15. Be able to perform experimental research on professional topics.
- PLO19. Ability to adapt to new situations, justify, make and implement relevant decisions.
- PLO22. Determine parameters of individual blocks of Computers, Computer Systems, Computer Networks.

PLO24. Build, configure, and use Linux-like operating systems.2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

When studying the discipline «Computer architecture. Part 3. Microprocessor systems» it is advisable to use the knowledge obtained during the study of previous disciplines: «Computer logic. Part 1. Computer logic», «Computer logic. Part 2. Computer arithmetics», «Computer circuitry», «System programming», «Computer electronics», «Computer architecture. Part 1. Arithmetical and controlling devices».

The discipline is basic for the courses: «Computer architecture. Course work», «Computer systems», «System software»; courses of elective disciplines of F-catalogue, «Testing and quality control (QA) of embedded systems», «FPGA programming technologies», «Designing technologies of intellectual systems».

3. The structure of the credit module

Introduction

Chapter 1. Introduction. The purpose and tasks of the discipline. Classification and purpose of microprocessor systems (MPS). General architectural and functional features.

Topic 1.1. Basic concepts and definitions of microprocessor technology. Classification of MPS depending on the area of destination. Classification of microprocessors.

Topic 1.2. Modern concepts and comparative features of MPS architecture, MPU, controllers, microcontrollers, features of use and principles of operation.

Topic 1.3. Architecture of universal microprocessors. The concept of assembler. Command systems and operand addressing methods. Features of the command system. Team formats.

Chapter 2. Practical fundamentals of programming for a 32-bit STM32F4 microcontroller based on the ARM CORTEX-M4 processor.

Topic 2.1. Overview of ARM processors. Features of CISC and RISC architecture, advantages and disadvantages for the implementation of processor cores. Development trends and a line of modern ARM processors. Modern ARM architecture profiles - Cortex A, Cortex R, Cortex M processors. Use and features of ARM Cortex M processors.

Topic 2.2. Classification of STM32 chip series by architecture and functional characteristics. A comparative overview of architecture and performance. Familiarity with the information space of manufacturers of products based on STM 32, documentation sets for developers and microcontroller programming. Overview of tooling environments for development and debugging.

Topic 2.3. Concept of system on a crystal (SoC). Overview of ARM CORTEX-M4 CPU SoC architecture, performance enhancements, reliability and interrupt handling. Overview of STM32F4 SoC microcontroller architecture and major peripherals. STM32F407VG Discovery Development Board Review.

Topic 2.4. Operating modes of the Cortex M4 processor and software access levels. Organization of the processor stack.

Topic 2.5. Cortex M4 processor registers. Model of the programmer. Instructions for accessing special registers.

Topic 2.6. The concept of load/store machines architecture is the separation of memory access instructions and data processing. Memory access instructions in the Cortex M4 processor.

Topic 2.7. Address space of the Cortex-M processor.

Topic 2.7. Data types. Operations with registers in the Cortex M4 processor. A set of arithmetic and logical instructions. Instructions for performing conditional transitions.

Chapter 3. Software development on the STM32F407VG Discovery board.

Topic 3.1. Overview of STM32F4 SoC interfaces and peripherals.

Topic 3.2. Signal converters ADC/ADC, DAC/DAC.

Topic 3.3. Peripherals: Power Control, POR, DMA, I^2S, LCD, PWM, WDT.

Topics 3.4. Interfaces and buses: CANbus, DCMI, EBI/EMI, Ethernet, I^2C, SPI, UART/USART, USB OTG.

Topic 3.5. Memory architecture. SRAM, FLESH memory modules. Memory management module MPU (Memory protection unit). Organization of the memory cache.

Topic 3.6. Work with input/output ports and interrupt handling.

Topic 3.7. Connection of sensors, displays, keyboard.

Topic 3.8. Software development in the STM32 Cube MX, Keil MDK-ARM microcontroller programming environment.

Chapter 4. Theoretical foundations of MPS design on microcontrollers.

Topic 4.1. Design stages of microprocessor systems. Examples of architectural design. Means and methods of designing and autonomous debugging of microprocessor system equipment.

Topic 4.3. Control and synchronization tools. Formation of control signal delays (short and long duration) at the port outputs during transmission to the control device. Programming the time delay of microcontrollers.

Topic 4.3. Interaction of processor and memory devices during execution of commands. Connecting external programs and data memory to the microcontroller.

Topic 4.4. Organization of information input and output in MPS. Connecting additional ports.

Topic 4.5. Organization of interruptions in microprocessor systems. Interrupt and direct access controllers of microprocessor sets. Types of interrupt controllers.

Topic 4.6. Trends in the development of MPS architecture. Organization and features of programming INTEL, PIC, AVR Atmega microcontrollers. RISC microprocessors.

Chapter 5. Modern technologies of design and development of microprocessors on field-programmable gate arrays (FPGA).

Topic 5.1. Modern technologies of design and development of digital systems – system on a chip technology (SoC). Main aspects of field-programmable gate arrays (FPGA) usage for digital systems development.

Topic 5.2. Hardware design and development lifecycle. Unified design flow of digital circuits. Modern CAD systems overview. Hardware description languages overview.

Topic 5.3. The concept of functional modelling and structural synthesis of a digital circuit. Practical aspects of digital circuits' functional modelling cycle in modern CAD systems.

Topic 5.4. Overview of Mentor Graphics (Siemens) ModelSim CAD system for functional simulation of hardware description languages. Verilog hardware description language programming fundamentals.

Topic 5.5. Historical aspects and tendencies of element base development. Modern digital systems development tendencies in the conditions of the 4th industrial revolution.

4. Educational resources and materials

4.1. Basic literature

1. Architecture of computers 3. Microprocessor tools. Part 1: Programming for the Cortex M4 processor. Study guide for degree candidates: Laboratory workshop [Electronic resource] : study guide for the student's educational program "Computer Systems and Networks" in specialty 123 "Computer Engineering" / I. A. Klymenko, V. V. Tkachenko, A. V. Kaplunov; Igor Sikorsky Kyiv Polytechnic Institute - Electronic text data (). - Kyiv : Igor Sikorsky Kyiv Polytechnic Institute, 2022. - 50 p.
2. Architecture of computers 3. Microprocessor tools. Part 2: Programming for STM32 microcontrollers. Study guide for degree candidates: Laboratory workshop [Electronic resource] : study guide for the student's educational program "Computer Systems and Networks" in specialty 123 "Computer Engineering" / I. A. Klymenko, V. V. A. Taraniuk, V. Tkachenko, A. V. Kaplunov; Igor Sikorsky Kyiv Polytechnic Institute - Electronic text data (). - Kyiv : Igor Sikorsky Kyiv Polytechnic Institute, 2022. - 125 p.Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 1 від 02.09.2022 р.).
3. FPGA Programming Technologies : Laboratory workshop [Electronic resource]: training . help _ for studies _ of the educational program "Computer systems and networks" by specialty 123 "Computer engineering" / O. A. Verba, V. I. Zhabin , I. A. Klymenko, V. V. Tkachenko; KPI named after Igor Sikorsky. – Electronic text data (1 file: 8.64 MB). – Kyiv: KPI named after Igor Sikorskyi, 2019. – 110 p.
4. Sergiyenko A.M. Computer Architecture (Архітектура комп'ютерів: підручник англійською мовою). - КПІ ім. Ігоря Сікорського, 2022. – 395 с.

4.2. Additional literature

1. Sarah L. Harris, David Harris Digital Design and Computer Architecture / Elsevier Science & Technology, 2022. – 720 p.

2. Computer Architecture 2. Processors: Theory and Practicum [Electronic resource]: tutorial for bachelor's study degree applicants under the educational program "Computer systems and networks" specialty 123 "Computer engineering" / I. A. Klymenko, A. V. Kaplunov, V. A. Taraniuk, V. V. Tkachenko; Igor Sikorsky KPI. – Electronic text data (1 file: XX MB). – Kyiv: Igor Sikorsky KPI, 2022. – 50 p.

3. Hamacher C., Vranesic Z., Zaky S. COMPUTER ORGANIZATION: 5th edition. – 2022. [Електронний ресурс] Hamacher:ComputerOrganization (mhhe.com), Computer Organization By Carl Hamacher 5th Edition | lulabi.live.

4. Tanenbaum A.S. Structured Computer Organization: 6th Edition – 2013. - [Електронний ресурс]

5. Комп'ютерна схемотехніка. Лабораторний практикум : навчальний посібник для студентів освітньої програми «Комп'ютерні системи та мережі» за спеціальністю 123 «Комп'ютерна інженерія» / КПІ ім. Ігоря Сікорського ; уклад.: О. А. Верба, В. І. Жабін, І. А. Клименко, В. В. Ткаченко. – Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 1 від 26.09.2019 р..). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 110 с. – https://ela.kpi.ua/handle/123456789/29747

6. Архітектура комп'ютерів – 1. Арифметичні та управляючі пристрої. Практикум : навчальний посібник для студентів спеціальності 123 «Комп'ютерні системи та мережі», спеціалізацій «Комп'ютерні системи та мережі» та «Технології програмування для комп'ютерних систем та мереж» / КПІ ім. Ігоря Сікорського ; уклад.: В. І. Жабін, І. А. Клименко, В. В. Ткаченко. – Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол №3 від 15.11.2018р.). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 53 с. – https://ela.kpi.ua/handle/123456789/29525

4.3. Informational resources

1. Video lectures course – https://bbb.comsys.kpi.ua/b/iry-ped-qe9
2. Remote courses on remote learning platform «Sikorsky» in Google Workspace for Education environment: Computer architecture. Part 2. Processors. https://classroom.google.com/c/NDY4OTYwNDM1MTYw?cjc=biw7yns
3. STM32 MCU & MPU Eval Tools - STMicroelectronics
4. Discovery kit with STM32F407VG MCU. Стислий опис (Data brief): STM32F4DISCOVERY - Discovery kit with STM32F407VG MCU * New order code STM32F407G-DISC1 (replaces STM32F4DISCOVERY) – STMicroelectronics
5. Discovery kit with STM32F407VG MCU. UM1472 User manual: Discovery kit with STM32F407VG MCU - User manual
6. STM32F407VG Product overview. https://www.st.com/en/microcontrollers-microprocessors/stm32f407vg.html#overview
7. Download Documentation: https://www.st.com/en/microcontrollers-microprocessors/stm32f407vg.html#documentation
   - STM32F407VG: Datasheet
   - STM32 Reference Manual (RM0090)
   - STM32 Cortex®-M4 Programming Manual (PM 0214):
8. Arm Cortex-M4 - Microcontrollers - STMicroelectronics

Educational content

5. Methodology of mastering an educational discipline (educational component)

Distribution of study time by types of classes and tasks in the discipline according to the working study plan. 150 hours and 5 credits are allocated to the credit module.

To achieve the goal of the educational discipline, the lecture material should focus on the features of the construction of the functional level of the computer, processor and other components. Particular attention should be paid to the features of computer design using a modern element base.

The purpose of the laboratory work is to acquire skills and abilities of practical application of the design principles of microprocessor systems and their individual functional nodes. Laboratory works are related to programming for a modern 32-bit microcontroller STM32F4 based on ARM CORTEX-M4 processor in assembler and C programming language using software emulators of ARM CORTEX-M4 processor; development of software and hardware on STM32F407VG Discovery board using STM32 Cube MX and Keil MDK-ARM microcontroller programming environments. To create your own projects for working with sensors, the GL Starter Kit STM32 board is used.

Topics of laboratory works:

Laboratory work 1. Deployment of the software environment for performing laboratory work on the Qemu MCU STM32F407VG emulator.

Laboratory work 2. Development of assembler programs for the CORTEX M4 architecture processor.

Laboratory work 3. Study of the command system of the CORTEX M4 processor. Development of data processing programs.

Laboratory work 4. Study of the memory architecture and operating modes of the CORTEX M4 processor. Development of own bootloader.

Laboratory work 5. Deployment of the STM32Cube IDE software environment (Keil). Development of an educational project in STM32Cube IDE for working with sensors on the STM32F4 Discovery board.

Laboratory work 6. Development of your own project for working with sensors on the GL Starter Kit STM32 board. Flashing of the GL Starter Kit STM32 board and visualization of the device work results.

6. Independent work of a student of full-time higher education

Types of independent work for students of full-time education (96 hours):

- preparation for auditorium classes, work on current homework and study of lecture materials (0.5 hour x 18 lectures = 9 hours);
- implementation of individual tasks for laboratory works, exercise solving, protocol formatting, preparation and processing of calculations based on primary data obtained in practical classes, formatting of laboratory work report (1.5 hours x 6 laboratory works = 9 hours);
- preparation and completion of module test (4 hours);
- preparation for the examination (30 hours);
- self-study of theoretical topics, programing environment deployment and software source code downloading for laboratory works (44 hours).

Topics for self-study (full-time education)

Chapter 1. Introduction. The purpose and tasks of the discipline. Classification and purpose of microprocessor systems (MPS). General architectural and functional features.

Topic 1.2. Modern concepts and comparative features of MPS architecture, MPU, controllers, microcontrollers, features of use and principles of operation.

Chapter 2. Practical fundamentals of programming for a 32-bit STM32F4 microcontroller based on the ARM CORTEX-M4 processor.

Topic 2.2. Classification of STM32 chip series by architecture and functional characteristics. A comparative overview of architecture and performance. Familiarity with the information space of manufacturers of products based on STM 32, documentation sets for developers and microcontroller programming. Overview of tooling environments for development and debugging.

Topic 2.7. Data types. Operations with registers in the Cortex M4 processor. A set of arithmetic and logical instructions. Instructions for performing conditional transitions.

Chapter 3. Software development on the STM32F407VG Discovery board.

Topic 3.3. Peripherals: Power Control, POR, DMA, I²S, LCD, PWM, WDT.

Topics 3.4. Interfaces and buses: CANbus, DCMI, EBI/EMI, Ethernet, I²C, SPI, UART/USART, USB OTG.

Topic 3.7. Connection of sensors, displays, keyboard.

Topic 3.8. Software development in the STM32 Cube MX, Keil MDK-ARM microcontroller programming environment.

Chapter 4. Theoretical foundations of MPS design on microcontrollers.

Topic 4.1. Design stages of microprocessor systems. Examples of architectural design. Means and

Topic 4.6. Trends in the development of MPS architecture. Organization and features of programming INTEL, PIC, AVR Atmega microcontrollers. RISC microprocessors.

7. Методика викладання дисципліни на заочній формі навчання.

Лекція 1.

Розділ 1. Вступ. Мета і задачі дисципліни. Класифікації та призначення мікропроцесорних систем (МПС). Загальні архітектурні та функціональні особливості.

Тема 1.1. Основні поняття і визначення мікропроцесорної техніки. Класифікація МПС в залежності від області призначення. Класифікація мікропроцесорів.

Теми на самостійне опрацювання

Тема 1.3. Сучасні поняття та порівняльні особливості архітектури МПС, MPU, контролерів, мікроконтролерів, особливості використання та принципи функціонування.

Тема 1.4. Архітектура універсальних мікропроцесорів. Поняття асемблеру. Системи команд та способи адресації операндів. Особливості системи команд. Формати команд.

Розділ 2. Практичні основи програмування для 32 розрядного мікроконтролера STM32F4 на базі процесора ARM CORTEX-M4.

Тема 2.1. Поняття система на кристалі (SoC). Огляд архітектури SoC ARM CORTEX-M4 CPU, засоби підвищення продуктивності, надійності та обробки переривань. Огляд архітектури мікроконтролера SoC STM32F4 та основних периферійних пристроїв. Огляд плати розробника STM32F407VG Discovery.

Тема 2.2. Режими роботи процесора Cortex M4, та рівні доступу програмного забезпечення. Організація стеку процесора.

Тема 2.3. Регістри процесора Cortex M4. Модель програміста. Інструкції доступу до спеціальних регістрів.

Тема 2.4. Поняття архітектури *load/store machines* – розділення інструкції звернення до пам’яті та обробки даних. Інструкції доступу до пам’яті в процесорі Cortex M4.

Теми на самостійне опрацювання

Тема 2.5. Огляд процесорів ARM. Особливості CISC та RISC архітектури, переваги та недоліки для реалізації процесорних ядер. Тенденції розвитку та лінійка сучасних ARM процесорів. Сучасні профілі архітектури ARM - процесори Cortex A, Cortex R, Cortex M. Використання та особливості ARM процесорів Cortex M.

Тема 2.6. Класифікація серій мікросхем STM32 за архітектурою та функціональними характеристиками. Порівняльний огляд архітектури й продуктивності. Знайомство з інформаційним простором виробників продуктів на базі STM 32, комплектами документації для розробники та програмування мікроконтролерів. Огляд інструментальних середовищ для розроблення та налагоджування.

Тема 2.7. Типи даних. Операції з регістрами в процесорі Cortex M4. Набір арифметичних та логічних інструкцій. Інструкції для виконання умовних переходів.

Тема 2.8. Адресний простір процесора Cortex-M.

Лекція 2.

Розділ 3. Розроблення програмно-апаратного забезпечення на платі STM32F407VG Discovery.

Тема 3.1. Огляд інтерфейсів та периферійних пристроїв SoC STM32F4.

Тема 3.2. Перетворювачі сигналів ADC/АЦП, DAC/ЦАП.

Тема 3.3. Периферійні пристрої: Контроль живленням, POR, DMA, I^2S, LCD, PWM, WDT.

Теми 3.4. Інтерфейси та шини: CANbus, DCMI, EBI/EMI, Ethernet, I^2C, SPI, UART/USART, USB OTG.

Тема 3.5. Архітектура пам’яті. Модулі SRAM, FLESH пам’яті. Модуль керування пам’яттю MPU (Memory protection unit). Організація КЕШ пам’яті.

Тема 3.6. Робота портами введення/виведення й обробка переривань.

Тема 3.7. Підключення датчиків, дисплеїв, клавіатури.

Тема 3.8. Розроблення програмного забезпечення в інструментальному середовищі програмування мікроконтролерів STM32 Cube MX, Keil MDK-ARM.

Лекція 3.

Розділ 4. Теоретичні основи проектування МПС на мікроконтролерах.

Тема 4.1. Взаємодія пристроїв процесора і пам’яті під час виконання команд. Підключення зовнішньої пам’яті програм та даних до мікроконтролеру.

Тема 4.2. Організація вводу-виводу інформації в МПС. Підключення додаткових портів.

Тема 4.3. Організація переривань в мікропроцесорних системах. Контролери переривань та прямого доступу мікропроцесорних комплектів. Різновиди контролерів переривань.

Теми на самостійне опрацювання

Тема 4.4. Етапи проектування мікропроцесорних систем. Приклади проектування архітектури. Засоби та методи проектування та автономного налагодження апаратури мікропроцесорних систем.

Тема 4.5. Засоби керування та синхронізації. Формування затримок управляючих сигналів(малої та великої тривалості) на виходах портів при передачі на пристрої управління. Програмне формування часової затримки мікроконтролерів.

Тема 4.6. Тенденції розвитку архітектури МПС. Організація та особливості програмування мікроконтролерів INTEL, PIC, AVR Atmega. RISC мікропроцесори.

Лекція 4

Розділ 5. Сучасні технології проектування та розроблення мікропроцесорних засобів на програмовних логічних інтегральних схемах (ПЛІС).

Тема 5.1. Життєвий цикл процесу проектування та розроблення апаратурних засобів. Уніфікований потік проектування цифрових схем (Design Flow). Огляд сучасних САПР. Огляд мов опису апаратури.

Тема 5.2. Поняття функціонального моделювання та структурного синтезу цифрової схеми. Практичні аспекти циклу функціонального моделювання цифрових схем в сучасних САПР.

Тема 5.3. Огляд системи автоматизації проектування Mentor Graphics (Simens) ModelSim для функціонального моделювання мов опису апаратури. Основи програмування на мові опису апаратури Verilog.

Теми на самостійне опрацювання

Тема 5.4. Сучасні технології проектування та розроблення цифрових систем – технологія системи на кристалі (SoC). Основні аспекти використання програмовних логічних інтегральних схемах (ПЛІС) для розроблення цифрових систем.
Тема 5.5. Історичні аспекти та тенденції розвитку елементної бази. Тенденції розроблення сучасних цифрових систем в умовах 4-ї Індустріальної революції

Тематика лабораторних робіт:

Лабораторні роботи для самостійного виконання

Лабораторна робота 1. Розвертання програмного окружения для виконання лабораторних робіт на емуляторі Qemu MCU STM32F407VG
Лабораторна робота 2. Вступ до розроблення та налагодження програм на асемблері для процесора архітектури CORTEX M4

Лабораторні роботи для аудиторного виконання

Лабораторна робота 3. Вивчення архітектури та системи команд процесора CORTEX M4. Розроблення програм обробки даних.
Лабораторна робота 4. Вивчення архітектури пам'яті та режимів роботи процесора CORTEX M4. Розроблення власного завантажувальника.
Лабораторна робота 5. Розвертання програмного середовища STM32Cube IDE (Keil). Розроблення навчального проєкту в STM32Cube IDE для роботи з датчиками на платі STM32F4 Discovery
Лабораторна робота 6. Розроблення власного проєкту для роботи з датчиками на платі GL Starter Kit STM32. Прошивка плати платі GL Starter Kit STM32 та візуалізація результатів роботи пристрою.

Види самостійної роботи для студентів заочної форми навчання (134 годин):

– підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (1,5 годин х 4 лекції = 6 годин);
– виконання індивідуального завдання до лабораторних робіт, розв'язок задач, оформлення протоколу, підготовка та оброблення проведення розрахунків за первинними даними, отриманими на лабораторних заняттях, оформлення звіту до лабораторної роботи (1,5 годин х 6 лабораторних робіт = 9 годин);
– підготовка та виконання поточної контрольної роботи (4 години);
– підготовка до екзамену (30 годин);

самостійне опрацювання теоретичного матеріалу та виконання лабораторних робіт, розвертання програмного оточення та завантаження вихідних кодів програмного забезпечення для виконання лабораторних робіт (44 години). (85 годин).

Policy and control

8. Policy of academic discipline (educational component)

Deadlines are set for the completion of laboratory work and modular control work.

Completion of laboratory work outside of established deadlines is accompanied by penalty points, which are deducted from the grade for the protocol. Module test is not accepted outside of established deadlines.

Module Test is carried out independently according to an individual task.

Each laboratory work is preceded by the completion of an individual task and its preparation in the form of a protocol. A student who came to class without a formatted protocol is not allowed to do laboratory work. At the first stage, a student defends the results received while working on the individual task for the laboratory work, at the second stage – defends theoretical knowledge by the means of oral questioning or testing. The majority of laboratory works are accompanied with test for evaluation of theoretical and practical material studied for the laboratory work. Points received for the completion of a laboratory work, for the test and protocol are included in the score for the laboratory work. Testing is conducted during the laboratory class after checking the results of completing laboratory works. A student who did not complete the individual task is not allowed to the test and laboratory work.

Individual lecture topics are accompanied by short express tests (for 15 minutes), which include the material of the studied topic and questions that are asked for independent study. The points obtained for the test are included in the semester rating. Recurring tests are not retaken.

The module test is conducted during the auditorium class without the usage of auxiliary means (mobile phones, tablets etc.); the result is sent to the corresponding directory of Google Drive through a Google Form.

The module test is not rewritten in case of a negative mark, a negative mark on the Module Test (less than 9 points (<60%)) is equaled to 0 points, in which case the Module Test is not passed.

Completion of laboratory works is mandatory for admission to the semester exam. The condition for admission to the exam is the completion of all laboratory work and a starting rating of at least 30 points.

The points for the exam work are added to the points for the laboratory work and Module Test and make up the semester rating.

Bonus points are awarded for: active participation in lectures; completing recurrent homework, keeping lecture notes, preparing a presentation on one of the self-study topics of the discipline, etc. Maximum available bonus points are no more than 6.

Penalty points can be issued for untimely completion of laboratory works. Maximum penalty points are no more than 10. During the period of martial law in the country penalty points are not implemented.

9. Types of control and rating system for evaluating learning outcomes (RSE)

The student's semester rating for the discipline is calculated based on a 100-point scale. The semester rating consists of the starting (current) grade R_S and examination R_E. Starting rating consists of points which a student receives for completing of 6 laboratory works R_L, module test R_{MT} and points for express-tests and bonus points.

The maximum number of points for laboratory works is 40 points, i.e., $R_L = 40$.

The criteria for evaluating laboratory works are as follows:

– timeliness of preparation of the protocol for the laboratory session, completeness of the theoretical or practical task in the protocol, the protocol is posted on GitLab on time: 0 - 2 points (penalty points can be deducted from the grade for late delivery of the protocol 0-2 points);
– correct functioning of the developed models on software or hardware, demonstration of own repository on GitLab with laboratory work materials and presence of commits: 0 - 2 points;
– a survey/testing on the subject of laboratory work for accepting the practical part of the work, defense of the results obtained in the work, answers to additional theoretical questions of the teacher, completeness of the report/protocol on the work on GitLab: 0 – 3(4).

The maximum number of points for Module Test R_{MT} = 15 points.

The criteria for evaluating Module Test at four levels:

– correct and meaningful answer with explanations in the terms of the subject area: 13 - 15 points;
– correct answer, incomplete explanations: 11 - 12 points;
– the answer contains mistakes: 9 - 10 points;
– the answer contains significant mistakes, there are no explanations: 4-8 points;
– no answer: 0 points.

The score for MKR is reduced by:

– incorrect design of schemes and drawings, lack of digitization of registers, buses, non-compliance with GOST standards;

- lack of comments in the program code and design of algorithms;
- absence of comments and explanations during calculations.

The details of the points for the current works for the semester are given in the following table.

The name of the class	**Form of control**	**Scores**	**Admission to the exam by automatic evaluation**	**Total points**
Laboratory work 1	Protocol on GitLab	2	5	7
	Task completion	2		
	Survey/test (Linux introduction, Git)	3		
Laboratory work 2	Protocol on GitLab	2	4	7
	Task completion	2		
	Survey/test	3		
Laboratory work 3	Protocol on GitLab	2	4	7
	Task completion	2		
	Survey/test	3		
Laboratory work 4	Protocol on GitLab	2	4	7
	Task completion	2		
	Survey/test	3		
Laboratory work 5	Task completion	2	2	4
	Protocol on GitLab	2		
Laboratory work 6	Task completion	2	5	8
	Protocol on GitLab	2		
	Survey/test on laboratory works 5 - 6	4		
Module Test		15	9	15
Express-tests		5		5
Total points		60	30	60

The maximum number of points for the exam is **R_E = 40 points**.

The examination ticket contains 4 tasks (one theoretical and three practical) on the subject of lectures and laboratory work performed during the semester. Each question is evaluated from 0 to 10 points.

The criteria for evaluating each question at four levels:

- correct and meaningful answer: 9 - 10 points;
- correct answer, incomplete explanations: 7 - 8 points;
- the answer contains mistakes: 5 - 6 points;
- no answer or the answer is incorrect: 0 points.

Calendar certification of students (for 8 and 14 weeks of semesters) in the discipline is carried out according to the value of the student's current rating at the time of certification. If the value of this rating is at least 50% of the maximum possible at the time of certification, the student is considered certified. Otherwise, the attestation information is marked as "uncertified".

A necessary condition for a student's admission to the exam is the completion and defense of all laboratory work with a total of at least 30 points.

The number of points a student receives per semester is determined by the formula

$$R = (R_L + R_{MT}) + R_E = R_S + R_E$$

The maximum number of points per semester does not exceed RS = 100.

Taking into account the received sum of points, the final grade is determined by the following table:

Scores	*Rating*
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

Work program of the academic discipline (syllabus):
Made by, Ph.D., docent, professor of the department of CE Klymenko Iryna Anatoliivna
Approved by the Department of Computing Engineering (Protocol No.10 dated 25.05.2022).
Agreed by the methodical commission of FICS (protocol No.10 dated 09.06.2022).