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

Department of Computer Engineering

C/Embedded programming technologies
The program of the academic discipline (Syllabus)

1. Details of the academic discipline

Cycle of Higher Education	***First cycle of higher education (Bachelor's degree)***
Field of Study	*12 Information technologies*
Speciality	*123 Computer Engineering*
Education Program	*Computer Engineering*
Type of Course	*Selective*
Mode of Studies	*Full-time education*
Year of studies, semester	*3 year (1 semester)*
ECTS workload	*4 credits (120 hours)* *Full-time education: Lectures 18 (36), Laboratory 9 (18), Independent work (66)* *Extramural studies: Lectures 18 (8), Laboratory 9 (8), Independent work (104)*
Testing and assessment	*Semester credit*
Course Schedule	*According to the schedule for the spring semester of the current academic year at http://rozklad.kpi.ua*
Language of Instruction	*Ukrainian*
Course Instructors	Lecturer: Artem Kaplunov, art.kaplunov@gmail.com Laboratory: Artem Kaplunov, art.kaplunov@gmail.com
Access to the course	Google Workspace for Education code: https://classroom.google.com/c/NTI1OTI0NTg5Mzg5?cjc=jey2wfd

2. Program of academic discipline

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

The purpose of the discipline "C/Embedded Programming Technologies" is to study the theoretical grounds, functional capabilities and principles of interaction of the main components of embedded systems. The subject of study of the credit module "C/Embedded Programming Technologies" is the architecture and functionality of modern embedded systems; input/output systems; means and mechanisms of data transmission and storage; introduction to the low-level C language and features of programming for embedded systems.

The discipline enhances the following general and professional competencies:

- ЗК2 - ability to learn and acquire modern knowledge;
- ФК1 - ability to apply the legislative and regulatory framework, as well as national and international requirements, practices and standards in order to carry out professional activities in the field of computer engineering.
- ФК2 - ability to use modern methods and programming languages for development of algorithms and software.

- ФК5 - ability to use the tools and systems of design automation to development of components of computer systems and networks, Internet applications, cyber-physical systems, etc.
- ФК9 - ability to systematically support, use, adapt and operate existing information technologies and systems.
- ФК13 - ability to solve problems in the field of computer and information technologies, determine the limitations of these technologies.
- ФК14 - ability to design systems and their components taking into account all their aspects. life cycle and task, including creation, configuration, operation, maintenance and disposal.
- ФК16 - ability to design, implement and maintain the high-performance parallel and distributed computer systems and their components using FPGAs, modules and CAD systems.

In accordance with the above, strengthened general and professional competencies will provide the following learning outcomes:

- knowledge of fundamental concepts, paradigms and basic principles of functioning of embedded systems;
- the ability to use theoretical, logical and arithmetic foundations for the development of drivers and software for embedded systems and the ability to apply them in solving professional tasks;
- the ability to develop drivers for individual components of embedded systems, including using modern design of automation systems;
- ability to develop and use drivers of architecture-dependent elements based on knowledge of general principles of organization and functioning of embedded systems;
- the ability to participate in team work on the design of drivers and software for embedded systems;
- the ability to form and provide requirements for the reliability of embedded systems in accordance with customer requirements, specifications and standards.

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 "C/Embedded Programming Technologies", it is advisable to use the knowledge gained during the study of previous disciplines: «Introduction to the Linux operating system», «Computer logic», «Computer electronics», «Computer architecture. Part 1. Control and arithmetic devices», «Programming».

The discipline is basic for courses from the F-catalogue of optional disciplines included in the certificate program: «Technologies of programming on FPGA» (elective), «Technologies of designing intelligent systems» (elective), «Technologies of testing (QA) of embedded systems» (elective), «Management of infrastructure IT projects» (elective), and is also useful for studying regulatory disciplines: «Computer architecture. Part 2. Processors», «Computer architecture. Part 3. Microprocessor devices», «Computer architecture. Course work», «System programming».

3. Structure of the credit module

Introduction. C as a programming language for embedded systems
Section 1. The C Preprocessor
Topic 1.1. File Inclusion
Topic 1.2. Macro Substitution
Topic 1.3. Conditional Inclusion
Section 2. Operators
Topic 2.1. Arithmetic Operators
Topic 2.2. Relational and Logical Operators
Topic 2.3. Conversion of Types
Topic 2.4. Increment and Decrement Operators
Topic 2.5. Bitwise Operators
Topic 2.6. Assignment Operators and Expressions

Topic 2.7. Conditional Expressions
Topic 2.8. Precedence and Order of Evaluation

Section 3. Structures

Topic 3.1. Basics of Structures
Topic 3.2. Structures and Functions
Topic 3.3. Arrays of Structure
Topic 3.4. Typedef

Section 4. Strings

Topic 4.1. Strings in C
Topic 4.2. Declaring strings
Topic 4.3. Working with strings
Topic 4.4. String arrays
Topic 4.5. String conversion
Topic 4.6. Row processing
Topic 4.7. Character Constants
Topic 4.8. String Literals

Section 5. Pointers

Topic 5.1. Pointers and Addresses
Topic 5.2. Pointers and Function Argument
Topic 5.3. Pointers and Arrays
Topic 5.4. Address Arithmetic
Topic 5.5. Character Pointers and Function
Topic 5.6. Pointer Arrays; Pointers to Pointers

Section 6. 'Hello World' in Embedded

Topic 6.1. Features of the microcontroller.
Topic 6.2. Problems with the IDE.
Topic 6.3. Compiler settings on STM32CubeIDE.
Topic 6.4. The process of creating an embedded project.

Section 7. Embedded C code analysis

Topic 7.1. Microcontroller program memory.
Topic 7.2. Analysis of an ELF file using GNU tools.
Topic 7.3. Disassembler.

Section 8. Working with LEDs

Topic 8.1. Blinking an LED Tutorial.
Topic 8.2. Enabling peripheral devices.
Topic 8.3. Calculation of addresses of peripheral registers.

4. Educational resources and materials

4.1. Basic literature

1. Brian W. Kernighan, Dennis M. Ritchie (1988) The C Programming Language, 2nd ed. Pearson.
2. Artem Kaplunov, Iryna Klymenko (2022) C/Embedded programming technologies. Course for the bachelors of the educational program "Computer systems and networks" in the specialty 123 "Computer engineering" – Kyiv: Igor Sikorsky KPI – 91 p.
3. Tarasenko-Kliatchenko O., Kliatchenko Y., Mykhailiuk O. (2021) The C Programming Language. Tasks from the credit module "Programming-1. Fundamentals of programming" : Course for the bachelors of specialty 123 "Computer engineering" – Kyiv: Igor Sikorsky KPI – 43 p.

4.2. Additional literature

4. Mark Siegesmund (2014) Embedded C Programming: Techniques and Applications of C and PIC MCUS. Newnes.
5. K. N. King (2008) C Programming: A Modern Approach, 2nd ed. W. W. Norton & Company
6. Manuel Bermudez (1998) Study Guide for C Programming: A Modern Approach. W. W. Norton & Company

7. Randal Bryant and David O'Hallaron (1994) Computer Systems: A Programmer's Perspective (2nd Edition). Addison Wesley
8. Stephen Kochan (2013) Programming in C (3rd Edition). Pearson.
9. Richard Heathfield and Lawrence Kirby (2000) C Unleashed. Sams.

4.3. Information resources

1. Online Course on "C/Embedded programming technologies" via platform "Sikorsky" using Google Workspace for Education: https://classroom.google.com/c/NTI1OTI0NTg5Mzg5?cjc=jey2wfd
2. Fastbit Embedded Brain Academy – YouTube. – https://www.youtube.com/channel/UCa1REBV9hyrzGp2mjJCagBg

5. Laboratory work

The purpose of the laboratory work is to acquire the skills and abilities to apply in practice the principles of designing and developing software for embedded systems and their individual functional units. Microcontroller programming tools (vim, Eclipse, Cube IDE), hardware (BeagleBone Black, STM StarterKit GlobalLogic) are used to perform laboratory work.

Topics of laboratory works:

Laboratory work 1. The C Preprocessor.
Laboratory work 2. Operators.
Laboratory work 3. Structures.
Laboratory work 4. Strings.
Laboratory work 5. Pointers.
Laboratory work 6. STM32F4-DISCO. Environment and software deployment.
Laboratory work 7. GL Starter Kit Firmware. STM32CubeIDE & Qemu.
Laboratory work 8. Turn LED On and Off With Push Button.

6. Independent work of the student

Types of independent work (66 hours):

- preparation for classroom classes (0,5 hours x 18 lectures = 9 hours);
- performance of an individual task for laboratory work, solving problems, drawing up a protocol, placing the results on GitHub (2 hours x 8 laboratory works = 16 hours);
- execution of module's tests (4 hours x 2 Module's Tests = 8 hours);
- preparation for express tests (4 hours);
- working out topics for independent work, downloading and setting up software for laboratory work (29 hours).

6.1. Topics for self-study (Full-time education)

Section 2. Operators
Topic 2.3. Conversion of Types
Topic 2.6. Assignment Operators and Expressions
Topic 2.7. Conditional Expressions
Topic 2.8. Precedence and Order of Evaluation
Section 4. Strings
Topic 4.5. String conversion
Topic 4.6. Row processing
Topic 4.7. Character Constants
Topic 4.8. String Literals
Section 5. Pointers
Topic 5.4. Address Arithmetic
Topic 5.5. Character Pointers and Function

Topic 5.6. Pointer Arrays; Pointers to Pointers

7. The method of teaching the discipline in the extramural studies

7.1. The content of lectures and self-study

Lecture 1. Introduction to “C/Embedded Programming Technologies”

Theme 1.1. Basic aspects of embedded systems in modern technologies.
Theme 1.2. Trends in the development of modern embedded systems.

Topics for self-study:

Theme 1.3. Overview of STM32-based embedded systems.

Lecture 2. C programming language in the context of Embedded Systems

Theme 2.1. Features of the C language when working with embedded systems
Theme 2.2. C Language. Functions.
Theme 2.3. C Language. Pointers.
Theme 2.4. C Language. Complex types.
Theme 2.5. C Language. Dynamic memory.

Topics for self-study:

Theme 2.6. C Language. Keywords.
Theme 2.7. C Language. Constants.
Theme 2.8. C Language. Arrays.
Theme 2.9. C Language. Strings.
Theme 2.10. C Language. Structures.

Lecture 3. Discovery Kit “STM32F407G-Disk1” Features

Theme 3.1. Functional composition of the core.
Theme 3.2. A review of functional block diagrams of the processor, controller and SoC.
Theme 3.3. Overview of the interaction of "hardware" with "software".

Lecture 4. Development and testing of an embedded device

Theme 4.1. ARM architecture. Evolution of ARM. ARM Cortex family.
Theme 4.2. Comparison of the STM32F4 series.
Theme 4.3. Discovery board pinouts.

Topics for self-study:

Theme 4.4. STM32F4 User Guide and Reference manual.

7.2. Subjects of laboratory works for self-fulfillment

Laboratory work 1. Consolidation of knowledge of the topic “The C Preprocessor”.
Laboratory work 2. Consolidation of knowledge of the topic “Operators”.
Laboratory work 3. Consolidation of knowledge of the topic “Structures”.
Laboratory work 4. Consolidation of knowledge of the topic “Strings”.
Laboratory work 5. Consolidation of knowledge of the topic “Pointers”.

7.3. Subjects of laboratory work for classroom work

Classroom classes 1:

Defense of laboratory works 1-5.

Classroom classes 2:

Laboratory work 6. Joining the self-hosted Gitea KPI service. Deployment of environment and software for working with STM32F4-DISCO.

Classroom classes 3:
Laboratory work 7. Development of firmware for GL Starter Kit. Running it on Qemu.

Classroom classes 4:
Laboratory work 8. Development of firmware to turn LED On and Off via Push Button.

Types of independent work for correspondence education (104 hours):

- preparation for classroom classes (1,5 hours x 4 lectures = 6 hours);
- preparation for express tests (4 hours);
- self-fulfillment of laboratory works 1-5 (3,5 hours x 5 laboratory works = 17,5 hours);
- preparation for laboratory works (6 – 8), drawing up a protocol, placing the results on GitHub (1,5 hours x 3 laboratory works = 4,5 hours);
- execution of module's tests (4 hours x 2 Module's Tests = 8 hours);
- preparation for the semester credit (4 hours);
- working out topics for independent work, downloading and setting up software for laboratory work (60 hours).

3. Policy and Assessment

8. Course policy

The grade that a student can receive for each laboratory work and for each module's test is given in table 1 of semester work evaluations, chapter 8 of the syllabus.

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

Deadlines are set for laboratory work. Performance of laboratory work outside the set time is accompanied by penalty points, which are deducted from the grade for the protocol.

Penalty points and hard deadlines are not introduced during martial law.

Module's Test is performed independently according to an individual task, Module's Test is not accepted beyond the set deadline. Module's Test are not rewritten in case of a negative grade, a negative grade for the Module's Test (less than 9 points (<60%)) is equal to 0 points, in this case the Module's Test is not counted.

Individual lecture topics are accompanied by short express tests (for 20 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. Current tests are not retaken.

Types of independent work for Thus, the minimum grade that a student can receive to qualify for a course = 60 points, the maximum is 100 points for the completion of all types of tasks during the semester.

Students who have fulfilled all admission requirements (completed all laboratory works) and have a rating of less than 60 points, as well as applicants who wish to improve their rating, have the opportunity to take the semester test at the last class on the schedule.

In the case of performance of the semester test, the rating is defined as the sum of points for semester test and points for individual semester tasks.

The individual work of the student is related to the performance of laboratory work. The maximum number of points for individual work per semester is 60 points. The maximum mark for the test is 40 points. In this way, the student has the opportunity to increase his rating by writing a semester test and adding additional points to the number of points received for individual work during the semester.

After completion of the semester test, if the grade for the semester test is higher than the rating, the student receives a grade based on the results of the semester test. If the grade for the semester test is lower than the rating, the student's previous rating (with the exception of points for the individual task) is canceled and he receives a grade based on the results of the semester test. This option forms a responsible attitude of the student towards making a decision on the completion of the semester test, forces him to critically assess the level of his training and carefully prepare for the credit.

9. Monitoring and grading policy

Distribution of study time by types of classes and tasks in the discipline according to the working study plan. The credit module is allocated 120 hours and 4 credits.

The student's semester rating from the credit module is calculated based on a 100-point scale. The rating consists of points that the student receives for performing 8 laboratory works R_{LW}, two Module's Tests R_{MT} and two express tests R_{ET}.

The maximum number of points for all laboratory works is 60 points, i.e. $R_{LW} = 60$.

The criteria for evaluating laboratory works are as follows:

- timeliness of drawing up the protocol for the laboratory work, completeness of the theoretical or practical task in the protocol, its timely upload: 0 - 2 points;
- correct functioning of the developed models on software or hardware, demonstration of own repository on GitHub with materials of performed laboratory work: 0 - 4 points;
- survey on the topic of laboratory work for crediting the practical part of the work, defense of the results obtained in the work, answers to additional theoretical questions of the teacher: 0 - 2 points.

The maximum number of points per Module's Test $R_{MT} = 2 \times 15 = 30$ points.

The mark for Module's Test decreases by:

- incorrect design of work on the remote repository, errors in the build files;
- lack of comments in the program code and design of algorithms;
- absence of comments and explanations during calculations.

The maximum number of points for express tests $R_{ET} = 2 \times 5 = 10$ points, the tests are conducted in the form of automated testing.

Table 1. Detailing of points for current works for the semester

Task name	**Form of control**	**Number of points**	**Admission to automatic credit**	**Total points**
Laboratory work 1	Task completed	4	5	8
	Answers on questions	2		
	Protocol	2		
Laboratory work 2	Task completed	4	5	8
	Answers on questions	2		
	Protocol	2		
Laboratory work 3	Task completed	4	5	8
	Answers on questions	2		
	Protocol	2		
Laboratory work 4	Task completed	4	5	8
	Answers on questions	2		
	Protocol	2		
Laboratory work 5	Task completed	4	5	8
	Answers on questions	2		
	Protocol	2		
Laboratory work 6	Task completed	2	2	4
	Protocol	2		
Laboratory work 7	Task completed	4	5	8
	Answers on questions	2		
	Protocol	2		
Laboratory work 8	Task completed	4	5	8
	Answers on questions	2		
	Protocol	2		
Individual work	-	-	-	60
Express tests	2 x 5	10	5	10
Module's Test	Module's Test 1	15	9	15
	Module's Test 2	15	9	15
Total points		100	60	100

The maximum number of points for credit work is equal to R_C = 40 points.

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

Evaluation criteria for each question at four levels:

- correct and meaningful answer: 9 – 10 points;
- correct answer, incomplete explanations: 6 – 8 points;
- the answer contains errors: 3 – 5 points;
- no answer or the answer is incorrect: 0 points.

The grade for the semester test is calculated according to table 2. An unsatisfactory grade does not give the right to credit additional points. Additional points to the semester grade are also not credited if the grade for semester test is lower than the student's current semester grade (according to Table 2). The semester grade can be increased to the minimum number of points of the semester grade (see Table 3), which corresponds to the grade received for the semester test (see Table 2).

Table 2. Determining the grade for credit work

Score	*Grade*
40-38	Excellent
37-34	Very good
33-30	Good
29-26	Satisfactory
25-24	Sufficient
below 24	Not Graded

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, "Not Graded" is displayed in the certification information.

A necessary condition for a student's admission to credit 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

$$RC = R_{LW} + R_{MT} + R_{ET}$$

The maximum number of points per semester does not exceed RC = 100.
Taking into account the received sum of points, the final grade is determined by table 3.

If a student writes a semester test, the number of points the student receives per semester is determined by the formula

$$RC = R_{LW} + R_{C}$$

The maximum number of points per semester does not exceed RC = 100.
Taking into account the received sum of points, the final grade is determined by table 3.

Table 3. Determination of the semester grade

Score	*Grade*
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
below 60	Fail
Course requirements are not met	Not Graded

The program of the academic discipline (syllabus):
Made by assistant of the department of CE, Artem Kaplunov.
Approved by the Department of Computer Engineering (protocol No. 10 dated 05.25.2022).
Agreed by the methodical commission of FICT (protocol No. 10 dated 06.09.2022).