UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria		
ACADEMIC YEAR	2023/2024		
BACHELOR'S DEGREE (BSC)	ROBOTICS ENGINEERING		
INTEGRATED COURSE	ELECTRONIC CALCULATORS - INTEGRATED COURSE		
CODE	18794		
MODULES	Yes		
NUMBER OF MODULES	2		
SCIENTIFIC SECTOR(S)	ING-INF/05		
HEAD PROFESSOR(S)	CHELLA ANTONIO	Professore Ordinario	Univ. di PALERMO
OTHER PROFESSOR(S)	PIRRONE ROBERTO CHELLA ANTONIO	Professore Ordinario Professore Ordinario	Univ. di PALERMO Univ. di PALERMO
CREDITS	12		
PROPAEDEUTICAL SUBJECTS			
MUTUALIZATION			
YEAR	1		
TERM (SEMESTER)	Annual		
ATTENDANCE	Not mandatory		
EVALUATION	Out of 30		
TEACHER OFFICE HOURS	**CHELLA ANTONIO** Monday 09:00 11:00 DICGIM, edificio 6, III piano **PIRRONE ROBERTO** Wednesday 11:30 13:00 Studio del docente, Edificio 6, terzo piano, stanza 3025		

DOCENTE: Prof. ANTONIO CHELLA

PREREQUISITES	Basic knowledge of math and science learned in high school.
LEARNING OUTCOMES	Knowledge and comprehension skills At the end of the course, the student will acquire a good understanding of techniques for representing information and processing it using sequential and combinatorial networks and a good understanding of optimization techniques for such systems. The student will learn the syntax and techniques of C language programming, data structures, and fundamental algorithms. Finally, the student will gain a basic understanding of robot programming in a simulated environment using the C language. Ability to apply knowledge and comprehension The student will apply the techniques studied to design combinatorial and sequential networks on a logical and functional level and analyze their operation. In addition, the student will investigate simple use cases related to applying combinatorial and sequential networks to the control problem of simple reactive robots. The student will be able to use tools and development environments for programming in the C language and know how to implement software solutions. For example, the student will devise and write simple robot control programs in C language within the WeBots simulator. Autonomy of judgment The student will analyze a problem and design an appropriate software solution from a verbal description. They will be able to evaluate the quality of a software solution in terms of simplicity, readability, efficiency, and reusability. In addition, they will be able to understand the principles of computer operation. Communication abilities The student will learn to communicate and express issues about the course's subject matter. First, they will be able to describe with appropriate terminology a logic circuit. Then, using appropriate language, they will explain the processes of analysis and synthesis of software solutions.
ASSESSMENT METHODS	The course will be assessed through two written tests "in itinere" (in-progress), insisting on the two-course modules and an optional final oral examination. A student who does not participate in the in-progress tests will have to take a final written test on the contents of the two modules. The written test covering the topics of the Logic Networks module will consist of answering closed or open-ended questions and sequential and combinational circuit design questions. The test will analyze a) The knowledge acquired. b) The ability to design circuits that meet given specifications. c) The ability to organize and discuss technical knowledge. The maximum score is obtained when the verification ascertains full possession of the following three aspects: the ability to perform the questions concerning the representation of information, the ability to design or optimize logic circuits that fulfill specific tasks, and the ability to describe and compare different circuit solutions. The written test covering the topics of the Fundamentals of Programming module is taken by computer. The test consists in writing one or more programs in C language according to given technical specifications. The test will analyze a) Knowledge of the C programming language. b) The ability to find solutions to simple problems typical of the discipline. c) The ability to create working software. The maximum score will be obtained if the verification ascertains full possession of the following three aspects: a) The use of constructs of the language in a syntactically correct manner. b) The ability to compose constructs to solve problems concerning acquiring, processing, and storing information. c) The creation of software executed without malfunctioning. The final grade will be calculated as the average of the evaluations of the two tests. The oral examination, which the student will have optional access to only upon prior instruction from the lecturers, will cover the topics of both modules and will consist of a general discussion of all aspects of Logical Networks and C-Language Programming. The overall assessment is based on the following criteria: (a) excellent (30 - 30 cum laude): very good knowledge of the topics of the course, very good capability of using appropriate language, good analytical ability, the student is able to apply the acquired knowledge to solve the proposed problems; (b) very good (26 - 29): good knowledge of the topics of the course, good knowledge of the appropriate language, the student is able to apply the acquired knowledge to solve proposed problems;

	(c) good (24 - 25): basic knowledge of central topics, fair knowledge of the appropriate language with limited ability to independently apply knowledge to solve proposed problems; d) fair (21 - 23): limited mastery of the main topics of the proper, fair knowledge of the correct language, fair ability to independently apply the acquired knowledge; (e) sufficient (18 - 20): minimal basic knowledge of the course's main topics and the technical language, poor ability to independently apply the acquired knowledge. (f) insufficient: does not possess minimum acceptable content knowledge of the topics covered in the course.
TEACHING METHODS	Lectures, tutorials.

MODULE
COMPUTER ARCHITECTURES

Prof. ANTONIO CHELLA

SUGGESTED BIBLIOGRAPHY

Mano M. Morris, Charles R. Kime, Tom Martin, Reti Logiche - quinta edizione, Pearson Education Italia (2019), ISBN: 889190581X

AMBIT	50283-Matematica, informatica e statistica
INDIVIDUAL STUDY (Hrs)	96
COURSE ACTIVITY (Hrs)	54

EDUCATIONAL OBJECTIVES OF THE MODULE

The student will be able to apply the methodologies studied in different contexts and learn analysis and synthesis processes related to logic circuits, including those involved in simple robotic problems.

SYLLABUS

Hrs	Frontal teaching
6	Introduction to the course. The concept of information and its logarithmic nature. Representation of integers in a generic base, binary numbers. Conversion of numbers from one base to another. Representation of floating-point numbers and negative numbers. Numeric representation of pictures and sounds, ASCII code.
6	Introduction to Boolean algebra. Logic gates, Boolean functions, negated logic, theorems and properties of algebra, normal forms. Karnaugh maps, minimization of Boolean functions. Synthesis of combinatorial networks. Integrated circuits (full adder, summers, multiplexers, decoders).
4	Sequential circuits. Mealy's and Moore's models. SR latch, typeD latch, JK, T, D, SR flip flops. Master slave, variation-sensitive flip flops. Analysis of sequential synchronous networks. Concept of state, state diagram. Characteristic equations of flip-flops.
4	Synthesis procedures of synchronous sequential networks: encoding of states. Synthesis of sequential networks. Synthesis with flip-flop D. Verification of operation of sequential networks, Simulation of sequential networks; Synthesis with flip-flop D, flip-flop T, and flip-flop JK.
4	Synthesis of sequence recognizers, Synthesis of sequence networks from the state diagram. Registers, Counters, ROM, PAL, PLA.
6	Computer architecture, CPU, ALU, RAM, BUS, von Neumann machine. Concept of program. Processor. CISC systems, RISC systems. Wired logic, micro-programmed logic.
6	Applications and case studies to the control of simple robotic systems.

Hrs	Practice
3	Information representation. Boolean algebra.
3	Analysis and Synthesis of Combinatorial Networks, Representation on the Karnaugh Map; Conjunctive and Disjunctive Canonical Forms.
3	Analysis and Synthesis of Sequence Networks, Sequence Recognizers
3	Use cases of computer architectures
3	Problems of the analysis and synthesis of combinatorial and sequential networks for the control of simple robots.
3	Analysis and synthesis of combinatorial and sequential networks for controlling simple robots: use cases

MODULE
PRINCIPLES OF PROGRAMMING

Prof. ROBERTO PIRRONE

SUGGESTED BIBLIOGRAPHY

1.Jeri R. Anly – Elliot B. Koffman, Problem solving e programmazione in C, Apogeo, ISBN-10: 8838786410
2.J. Glenn Brookshear - Stephen G. Kochan, Fondamenti di informatica e programmazione in C, Pearson, ISBN-10: 8865183691

AMBIT	50283-Matematica, informatica e statistica
INDIVIDUAL STUDY (Hrs)	96
COURSE ACTIVITY (Hrs)	54

EDUCATIONAL OBJECTIVES OF THE MODULE

After attending the course, the student will be able to evaluate, analyze, communicate, and implement possible software solutions for simple problems using the C Programming Language. Moreover, he will be aware of the WeBots robot simulator along with the possibility to program robot controllers in C.

SYLLABUS

Hrs	Frontal teaching
2	Introduction to the course. Data processing. Computer architectures, Operating systems, Computer Networks. Programming Languages: historical perspective, program translation, machine independence.
2	Algorithms: definition, representation, pseudo-code, and flowcharts. Programming basics. Compiling and executing the very first C program.
2	Introducing the C programming language. Variables, data types, arithmetic expressions. Variables and constants. Assignment. Operators and their priority.
2	Definition of sub-program. c functions. Return values. Functions with and without parameters. Modular software design: top-down techniques, and structure diagrams. Understanding and simulating algorithms/programs.
2	Selection structures (if-else) and multiple selection structures (switch-case). Flowcharts for selection structures. Understanding and simulating algorithms/programs
2	Iterative programs in C. Iterative structures (for, while, and do-while). Flowcharts for iterative structures. Understanding and simulating algorithms/programs.
2	Iterative algorithms: efficency, and correctness. Computational complexity basics.
4	Functions in deep. Local variables. Variables' visibility and scope. Globaò variables, static and register. Introduction to pointers and their use in functions' parameters.
2	Recursion. Memory management. Implementation in C.
3	Introduction to data structures. Array declaration and indexing. Multidimensional arrays. enum type. Array with enum indexes. Iterations and arrays. Implementation of stack dta structures using arrays.
2	Strings. char arrays, strings with varying length. escape sequences.
4	Pointers in deep: pointers and arrays. Pointers math. Structures and union. Implementation of user dat astructures. Implementation of lists and queues. Searching in, and sorting data collections.
2	I/O management in C. Files.
2	Preprocessor directives. Including personal libraries.

Hrs	Practice
4	Desiging simple algorithms using flowcharts. Moving to C code. Very first C prorgams: compiling, linking and running C programs. Simple programs for managing numeric data types.
2	.Designing algorithms with selections and iterations using flowcharts. Implementing the corresponding C programs
3	Designing modular software through functions. Designing and implementing recursive algorithms.
6	Design and implementation of algorithms using strings, vectors, and matrices. Using pointers. Implementation of lists, stacks, and queues.
3	Implementation of programs using file I/O. Discussion of the arrangement of the written examination.
3	Introduction to Webts. Writing the very first C robot controller.