COURSE GUIDE

Dean, Prof. Daniela Tărniceriu

.....

1. Program info

1.1 Higher education institution	"Gheorghe Asachi" Technical University of Iaşi
1.2 Faculty / Department	Electronics, Telecommunications and Information Technology
1.3 Department	Telecommunications and Information Technologies
1.4 Field	Electronic Engineering, Telecommunications and Information Technology
1.5 Study level	Bachelor's Degree Studies
1.6 Study program / Qualification	Telecommunications Systems and Technologies

2. Course info

2.1 Course name: **MICROCONTROLLERS**						Code: EDIS 304	
2.2 Course organizer (lecturer)		Lecturer Ph.D. bioeng. Dobrea Monica-Claudia					
2.3 Teaching assistants		Lecturer Ph.D. bioeng. Dobrea Monica-Claudia					
2.4 Year of study	3	2.5 Semester	5	2.6 Assessement	E	2.7 Type of subject	MD

3. Estimated total time (hours per semester for teaching activities)

3.1 Number of hours per week	4	3.2 lecture	2	3.3 seminar/laboratory	2
3.4 Total number of hours in curricula	56	3.5 lecture	28	3.6 seminar/laboratory	28

Time distribution	Hours
Textbook, course support, references and course notes study	12
Library, electronic platforms and on site documentation	5
Seminar/laboratory preparation, homework, reports, portfolios and essays	12
Tutoring	7
Assessment	2
Other activities	2

3.7 Total individual study hours	40
3.9 Total hours per semester	96
3. 10 Number of credit points	4

4. Prerequisites (where applicable)

4.1 curricula type	Fundamental Electronic Circuits, Digital Integrated Circuits, Computer Programming and Programming Languages I, Computer Programming and Programming Languages II
4.2 competence type	Proper application of fundamental specific knowledge of (C/C++) programming.

5. Infrastructure (where applicable)

5.1. for lectures	Classroom equipped with multimedia equipment (PC/laptop, video projector, presenter - PowerPoint remote/laser, screen), table.
5.2. for laboratories	Laboratory room with: computer network, internet access, 10 workstations, 10 development boards Intel Galileo Gen 2, cables, sensors, LCDs, electronic components (LEDS, resistors, switches, servo motor, stepper motors etc.), GPS & GSM modules, web camera etc.

6. Specific competences

Professional competences	- Knowing the diference between microprocessors, microcontrollers, SoC, DSP etc. - Knowing the weaknesses and advantages of different architectures (IA-32, IA-64, ARM, MIPS and SH). - To know conceptually the role of each internal element within a processor and to quantify its influence on overall system's performance. - Be able to understand and use the Arduino integrated development menus and options. - Be able to configure, connect and communicate with a development system. - Be able to write simple programs in the Arduino environment, understanding the fundamental parts of the program. - Become familiar with the Intel Galileo Gen 2 board. - Have skills in connecting, interfacing, and programming devices connected to the GPIO lines. - Have skills in connecting, interfacing, and programming devices connected to the serial lines (RS232, I2C and SPI). - Understand GSM and GPS standards from a user's point of view that develops software for the microcontrollers connected to them.
Transversal competences	- Make effective use of information resources and computer-aided communication and training resources. - Possessing the learning skills that enable them to continue studying in a way that will be largely self-directed and autonomous. Demonstrate preoccupation for professional development by engaging critical thinking skills and refining training and education throughout their entire career development. - Develop interpersonal skills (teamwork skills, collaborative problem solving, communication skills), intrapersonal skills (self-discipline, perseverence and motivation, ability to learn independently) and critical and innovative thining (creativity, reflecting thinking, reasoned decison making); become easily familiar with electronics for embedded systems. - Develop the skills of both capitalizing on knowledge and adapting to the requirements of a dynamic and competitive technological environment, proving flexibility and adaptability. - Knowing to communicate their conclusions (and knowledge and last reasons that support them) to specialised and non-specialised audience in a clear way and unambiguously.

7. Course targets (as resulting from **6. Specific competences** table)

7.1 Course main target	The materials presented in classes and laboratories are designed to give students basic training in the microcontrollers' field; this knowledge is necessary to understand the organisation and use of microcontrollers in the development, implementation and control of various applications, mainly in the IoT emerging field.
7.2 Course specific targets	The objectives of this discipline reside in acquiring the concepts, terms and technical language specific to the field of microcontroller systems and, moreover, in using first and foremost these concepts in practice. After completing this course, students should be able to: understand the components of a microcontroller, differentiate between the different peripheral equipment interconnection standards (advantages/disadvantages), develop practical applications in the ANSI C language, understand and apply microcontrollers' specific programming techniques.

8. Contents

8. 1 Lectures	**Teaching methods**	**Notes**
I. Introduction **Hardware** a) What is a Microprocessor? b) What is a Microcontroller? (Evolution, Definition, Features, A typical microcontroller architecture, Examples) c) What is a System on Chip (SoC)? Introduction to Intel Quark X1000 processor System on Chip. d) What is a development board? Introduction to Intel Galileo Gen 2 board. e) Embedded systems - Basic concepts, General-purpose computers *versus* Embedded Systems f) Memory Technologies used in Embedded Devices **Software** a) Machine code, Assembly language, Higher-level languages b) Arduino IDE. Basic structure of Arduino programming (simple programs - sketches)	• Lectures, • Discussions, • Connections with information taught to other disciplines	• Video projector, • Table; • Intel Galileo Gen 2 board
II. General overview over the main architectures		

a) IA32/64, ARM, MIPS, PowerPC etc. b) The relation between different types of OS and microprocessor families c) IA32 architecture blueprint on the Intel Quark X1000 processor System on Chip	• Lectures, • Discussions, • Connections with information taught to other disciplines	• Video projector, • Table; • Intel Galileo Gen 2 board
III. Architectures - main concepts. Intel Galileo Gen 2 board – hardware, Arduino IDE and software 1. CPU a. Units in the CPU: *Control Unit* (CU), *Execution Unit – Arithmetic Logic Unit* (**ALU**), *Floating Point Unit* (**FPU**) –, *Status Register* (**SR**), *CPU Registers*, *Cache* (**L1/L2/L3**), buses, registers, and their widths); b. The Fetch, Decode, Execute Cycle c. Instructions: classical vs. SIMD. d. CPU in the Intel Quark X1000 SoC 2. Memory a. Introduction to the Main memory b. System Memory Map: Memory Space (DRAM, MMIO); I/O Address Space. c. Memory management unit (MMU) d. Addressing modes (immediate, register, memory location, I/O port) e. Memory specifications for the Intel Quark X1000 SoC 3. Microprocessor buses 4. Pipelining 5. General-purpose input/output (GPIO) – Main concepts; GPIO on Intel Galileo Gen 2 board (hardware & software). 6. Timers – Main concepts; Timers on Intel Galileo Gen 2 board (hardware & software). 7. Interrupt systems – Main concepts; Interruptions on Intel Galileo Gen 2 board (hardware & software). 8. Input-Output System: I/O organization, I/O Techniques (Programmed I/O; Interrupt-driven I/O; Direct memory access Direct Memory Access (DMA)).	• Lectures, • Discussions, • Connections with information taught to other disciplines	• Video projector, • Table; • Intel Galileo Gen 2 board
IV. Hardware and Software for Interfacing to External World 1. Parallel Interface 2. Interfacing to motors, relays and other high current devices 3. Analog Inputs and Outputs 4. RS-232C Serial I/O Interface 5. SPI Bus Interface 6. System Management Bus (SMBus) 7. I2C Bus Interface 8. The Universal Serial Bus (USB) 9. CAN and LIN Network Interfaces	• Lectures, • Discussions, • Connections with information taught to other disciplines	• Video projector, • Table; • Intel Galileo Gen 2 board

References:

[1] James O. Hamblen, Introduction to Embedded Systems Using Windows Embedded CE, ed. Georgia Institute of Technology, 2007

[2] David Murray, Microcontroller Systems: Lecture 2 - The CPU, Instruction Fetch & Execute, University of Oxford, 2014

[3] Manoel Carlos Ramon, Intel Galileo and Intel Galileo 2 - Arduino Linux, Apress Media, LLC New York (978-1-4302-6838-3), 2014

[4] Yan Luo, Chen Xu, and Peilong Li, Microprocessor II and Embedded Systems Design Course, Chapter 13 - Direct Memory Access and DMA-Controlled I/O, University of Massachusetts Lowell (UMass Lowell) - Fall 2014, https://piazza.com/uml/fall2014/16480552/resources

[5] H.S. Jamadagni, Full Intel® Galileo Curriculum: "*Intel® Galileo – An Experiential Learning*", Indian Institute of Science, 2013, http://educate.intel.com/download/Galileo/IntelGalileo_WorkshopPresentationCumManual_Bangalore.pdf

[6] Intel, The Pentium processor technical documentation (accessed on February 2011), www.intel.com/design/pentium/datashts/24199710.pdf

[7] http://vega.unitbv.ro/~romanca/C-Arc/10-IO-System.pdf

8. 2 Laboratory	Teaching methods	Notes
1. **Introduction to Intel Galileo Gen 2** (Galileo – relevant specifications, setting up the Intel® Galileo Gen 2, attachments & connectors, Arduino IDE, sketches, simple examples)	• Presentations • Discussions, • Software development, • Applications according to the laboratory written support	Materials used: Intel Galileo Gen 2 development boards, cables, sensors (PIR, TMP102, TMP112 etc.), LCDs, electronic components (LEDS, resistors, switches etc.), motor drivers, servo motors, stepper motors, keypads, LCD displays, GSM GPRS modules etc.
2. **Programming Techniques** (Arduino Headers on Intel Galileo Gen 2 – *Digital Input/Output pins,* applications – *The Passive Infrared (PIR) Motion Sensor*)		
3. **Programming Techniques** (Arduino Headers on Intel Galileo Gen 2: *Digital pins – improving the I/O speed.* Memory-Mapped I/O *versus* Port-Mapped I/O)		
4. **Programming Techniques** (Arduino Headers on Intel Galileo Gen 2: *Analog pins*; *PWM pins*; applications)		
5. **Arduino Language – Strings. Interfacing Keypad with Intel Galileo Gen 2** (applications)		
6. **Serial communication** (configuring the serial port, accessing the serial port, output in terminal window: Serial (CDC) communications over USB; UART TTL (5V/3.3V) serial communication, Console UART port providing access to the Linux serial console; *Serial*, *Serial1*, and *Serial2* Objects)		
7. **External (pin) and Internal (timer) Interrupts**		
8. **Interfacing with servo motors**		
9. **Interfacing with stepper motors**		
10. **Parallel Interface** (Interfacing with *Licquid Crystal Displays*, LCDs; applications)		
11. **I2C Communication** - practical example, interfacing with a temperature sensor (e.g., TMP112, TMP102)		
12. **Ethernet Communication**		
13. **Interfacing GSM GPRS module with the Galileo board** (practical examples of sending messages to cell phone)		
14. **Practical exam**		

References:

[1] Manoel Carlos Ramon, Intel Galileo and Intel Galileo 2 - Arduino Linux, Apress Media, LLC New York (978-1-4302-6838-3), 2014

[2] Intel® Galileo Gen 2 Board Schematic (file: galileo_fabh_schem_12-05-14.pdf), https://www.intel.com/content/www/us/en/support/boards-and-kits/intel-galileo-boards/000005911.html

[3] Prof. Dr. Djones Lettnin, Lab 2: Introduction to Intel Galileo Gen 2 - Programming Techniques, developed by Rafael Osmari e Lucas Lazare, Federal University of Santa Catarina, Florianópolis, 2015

[4] Prof. Dr. Djones Lettnin, Lab 3: Serial communication, developed by Rafael Osmari e Lucas Lazare, Federal University of Santa Catarina, Florianópolis, 2015

[5] ***Serial Terminal Basics***. Available at: https://learn.sparkfun.com/tutorials/terminal-basics/tera-term-windows

[6] Galileo Getting Started Guide, https://learn.sparkfun.com/tutorials/galileo-getting-started-guide/using-the-terminal

[7] RS-232 vs. TTL Serial Communication, <https://www.sparkfun.com/tutorials/215?_ga=2.93026393.535924108.1501855656-217888673.1501650949 >

9. Course contents corroboration with the expectations of the epistemic community representatives, professional associations and relevant employers in the field of the program

In determining the content of the Microcontrollers discipline, the programs used in other university centers from the country and abroad were consulted. The objectives of the discipline are in line with the curriculum: the transmission of information and the training of the specialists required in the field of electronics, telecommunication and information technology. The acquired competencies will be needed for employees working in the field of complex engineering systems controlled by the mirocontroller systems. The programming skills acquired within the faculty are strengthened and developed by connecting both the hardware and the software.

The discipline uses the knowledge and methods presented in the disciplines "Computer Programming and Programming of Language 1 and 2", thus proving that the Microcontrollers course is properly placed in the curriculum chronology.

Microcontrollers (MCUs) provide wide marketting area for embedded systems and IoT. In fact, microcontrollers are the heart of embedded systems, upgrading constantly the standard for this market.

10. Assessment

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Percentage of final grade

10.4 Lectures	• Correct answer and completeness of knowledge. • Logical coherence in expression and appropriate use of the notions presented.	**Written exam**	60%
10.5 Laboratory	• frequency and relevance of oral interventions; • the quality of the work done; • the systematic recording of significant student-generated information in the application group.	30%	40%
	• Final test at week 14	• Correctness and functionality of the developed program – 70%	
10.6 Minimum performance standard			
Obtaining a minimum score of 5 at each of the previous mentioned two criteria (lectures & laboratory).			

Completion date:
12.09.2019

Course organizer signature,
Lecturer Ph.D. bioeng. **Dobrea Monica-Claudia**

Teaching assistant signature,
Lecturer Ph.D. bioeng. **Dobrea Monica-Claudia**

Department approval date,

16. SEP. 2019

Department director signature,
Assoc. Prof. Ph.D. eng. Luminița Scripcariu