Tunisian Republic
Private Higher School of Engineers of Gafsa
Private Higher Education Institution,
State-approved under N° 05-2013

Course Title:	**Embedded systems**
Course Code:	CSE421
Program:	Master Degree In Computer Engineering
Department:	Computer Engineering
Course coordinator:	Dr. Dhikra Chermiti
Institution:	Private Higher School of Engineers of Gafsa (ESIP)

A. Course Identification

1. **Credit hours: 3 (1.5-0.5-1)**

2. Course type

a. College □ Department ■ Others □

b. Fundamental ■ Transversal □ Optional □

3. **Level/year at which this course is offered:** 2.2/3

4. **Pre-requisites for this course :** (CSE131), (CSE122), (CSE231), (CSE242), (CSE232), (CSE311), (CSE341), (CSE342)

1. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Self-study	Total workload
1	**Traditional classroom**	…..	33	78
2	**Blended**	45		
3	**E-learning**	…...		
4	**Distance learning**	…...		
5	**Other ()**	…...		

2. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	**Lecture**	30
2	**Laboratory/Studio**	11
3	**Tutorial**	4
4	**Others** (specify)	-
	Total	45

B. Course Objectives and Learning Outcomes

Course Description

The "Introduction to Embedded Systems" course is designed for 2nd-year engineering students in computer science. It provides a comprehensive understanding of embedded systems, covering fundamental concepts, hardware and software architecture, design, and applications.

Students will explore:

- Characteristics and constraints of embedded systems
- Microcontrollers, microprocessors, memory, and peripherals
- Embedded programming with C/C++
- Communication interfaces (UART, SPI, I2C, wireless protocols)
- Real-time operating systems (RTOS), task scheduling, and resource management

The course emphasizes practical applications, preparing students for careers in IoT, automotive, robotics, aerospace, and industrial automation where embedded systems are essentiels.

Course Main Objective

At the end of the module, the student should be able to:

- ✓ Develop a deep understanding of embedded system fundamentals, their characteristics, and their significance across industries.
- ✓ Understand hardware architecture, including microcontrollers, memory management, and peripheral components.
- ✓ Gain practical skills in embedded programming using C/C++, including software design, debugging, and optimization.
- ✓ Analyze and implement communication interfaces, including UART, SPI, I2C, and wireless protocols.
- ✓ Understand embedded operating systems (RTOS), task scheduling, and resource management in real-time environments.

1. Course Learning Outcomes

CLOs		Aligned PLOs
1	**Knowledge and Understanding**	
2.1	Understand the principles of embedded operating systems, with an emphasis on real-time systems. They must be able to manage tasks, scheduling and resource management in an embedded environment.	PLO.K2
2.2	Understand the architecture of embedded systems, including hardware components such as microprocessors, microcontrollers, memory, and peripherals, as well as associated programming logic and data structures.	
2	**Skills**	
2.1	Develop an in-depth knowledge of the basic concepts of embedded systems, including their definition, characteristics and importance in different fields of application.	PLO.S2

CLOs		**Aligned PLOs**
6.1	Develop practical programming skills for embedded systems, using languages such as C/C++. They must be able to design, develop and debug software suitable for embedded systems.	PLO.S6
7.1	Acquire an in-depth knowledge of the different communication interfaces used in embedded systems, as well as commonly used protocols such as UART, SPI, I2C and wireless protocols.	PLO.S7

C. Course Content

No	List of Topics	Contact Hours
1	Chapter 1: Introduction to Embedded Systems and SoC Definition, architecture, applications	3
2	Chapter 2: Microcontroller Architecture and Programming Cortex-M, memory, registers, low-level programming	3
3	Chapter 3:Inter-Chip Communication Protocols (SPI, I2C, UART, CAN, USB, Ethernet)	3
4	Chapter 4:SoC: Bus Architectures and Network-On-Chip (NoC) AMBA, Wishbone, Avalon, NoC, and interconnections	3
5	Chapter 5: Real-Time Operating Systems (RTOS) RTOS vs General-Purpose OS, scheduling, multitasking	3
6	Chapter 6: Application: Wireless Sensor Network (WSN) Wireless protocols (LoRa, ZigBee, BLE), network design	3
Total		18

C.1Tutorial Content

No	List of Topics	Contact Hours
1	T1: Introduction to embedded systems (Definition, architecture, applications)	2.5
2	T2: Microcontrollers and communication (Architecture ARM Cortex-M, protocoles (SPI, I2C, UART))	2.5
3	T3: SoC and interconnection	2.5
4	T3: RTOS and Embedded Systems FreeRTOS Principles, Zephyr, Scheduling and Multitasking	2.5
5	Cross-compilation and security	2
Total		12

C.2Practical work Content

No	List of Topics	Contact Hours
1	STM32CubeIDE Setup and Blink LED	3
2	Interrupt Handling and Timers	3
3	Serial Communication (UART, SPI, I2C)	3
4	Implementing FreeRTOS on STM32	3
5	Deploying LoRa/BLE with STM32	3

Total	15

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	**Knowledge and Understanding**		
	✓ Understand the principles of embedded operating systems, with an emphasis on real-time systems. They must be able to manage tasks, scheduling and resource management in an embedded environment. ✓ Understand the architecture of embedded systems, including hardware components such as microprocessors, microcontrollers, memory, and peripherals, as well as associated programming logic and data structures.	▪ Lecturing ▪ Practical work ▪ Class discussions	▪ Reports ▪ Exercises and participation throughout the course ▪ Exams
2.0	**Skills**		
	✓ . Develop an in-depth knowledge of the basic concepts of embedded systems, including their definition, characteristics and importance in different fields of application.	▪ Lecturing ▪ Practical work ▪ Class discussions	▪ Reports ▪ Exercises and participation throughout the course ▪ Exams
	✓ Develop practical programming skills for embedded systems, using languages such as C/C++. They must be able to design, develop and debug software suitable for embedded systems.		
	✓ Acquire an in-depth knowledge of the different communication interfaces used in embedded systems, as well as commonly used protocols such as UART, SPI, I2C and wireless protocols.		

2. Assessment Tasks for Students

	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Practical Work (written or oral)	Monthly	25 %
2	Quizzes, Homework assignments	Random	00%

	Assessment task*	Week Due	Percentage of Total Assessment Score
3	First mid Term	8	25%
4	Final Exam	16	50 %

E. Student Academic Counselling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice:

1- Office hours
2- Blackboard interface

F. Learning Resources and Facilities

1. Learning Resources

Required Textbooks	1. David E. Simon. An Embedded Software Primer. Addison-Wesley, 1999. ISBN: 978-0201615692 2. Frank Vahid & Tony Givargis. Embedded System Design: A Unified Hardware/Software Introduction. Wiley, 2001. ISBN: 978-0471386780. 3. Jonathan W. Valvano. Embedded Systems: Introduction to ARM Cortex-M Microcontrollers. 5th Edition, 2019. ISBN: 978-0997925968. 4. Michael Barr & Anthony Massa. Programming Embedded Systems in C and C++. O'Reilly Media, 2006. ISBN: 978-0596009830.
Essential References Materials	▪ STM32CubeIDE & Keil uVision ▪ STM32f407
Electronic Materials	1. STMicroelectronics (STM32): Technical Documentation & Development Tools (www.st.com) 2. Coursera & edX: Embedded Systems & Real-Time Programming Courses 3. YouTube Channels: "Embedded Systems with STM32" (Shawn Hymel, Phil's Lab, Great Scott!) 4. GitHub Repositories: Open-source embedded software, STM32 projects (www.github.com)
Other Learning Materials	**STM32CubeIDE & Keil uVision:** *IDE for STM32 development and debugging* FreeRTOS Documentation: Real-time OS for embedded applications

2. Facilities Required

Item	Resources
Accommodation	**Classroom board** **Computer lab with the necessary software** **Internet access**
Technology Resources	**Data projector**

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching and assessment.	Students, course coordinator, Alumni, Employers	Direct/Indirect
Extent of achievement of course learning outcomes.	Faculty, Program Leaders, quality department	Direct
Quality of Learning resources	Faculty, Program Leaders,	Direct, Indirect
Teaching and learning quality and effectiveness.	Students, Faculty Program Leaders,	Direct, Indirect

H. Specification Approval Data

Council / Committee	Computer Engineering Council
Date	07/02/2024