Birzeit University

Faculty of Engineering and Technology

Electrical & Computer Engineering Department

Course title code	Real-Time Applications and Embedded Systems	ENCS4330
Instructor office	Dr. Hanna Bullata	Masri219
How to reach me	**Email:** hanna@birzeit.edu **Skype:** hanna.bullata **Ritaj ID:** 2295	
Semester- Year	1st semester 2021/2022	
Compulsory/Elective	Compulsory	
Prerequisites	Operating Systems, Linux Lab	

Course Description	The goal of the course is to introduce students to building applications using the different inter-process communication techniques. In addition, the goal is to expose students to building multi-processing and multi-threading applications using the C-programming language. The other part of the course introduces the fundamentals of embedded system design and implementation, including specifications and modeling of embedded systems, hardware/software partition and codesign; validation and implementation; real-time operating system In the last part of the course, students will learn about FreeRTOS on embedded controllers in terms of installation, compilation and structuring their embedded applications into different tasks on both PIC microcontrollers and on Arduino boards.

Course Objectives	• Build multi-processing and multi-threading applications. • Understand concepts, analytical tools and design techniques of embedded systems. • Investigate topics from hardware/software partition, development tools selection, circuit and operating system design, to system verification and implementation. • Use FreeRTOS on embedded controllers.
Student Outcomes (SOs)	1. An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety,

	manufacturability, and sustainability (C) 2. An ability to identify, formulate, and solve engineering problems (E) 3. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (K)

	Course contents	SO1	SO2	SO3
1	Inter-process communication techniques (signals, pipes, fifos, semaphores, message queues, shared memory, sockets).	X		
2	Microcontroller architecture and the flow of data within a microcontroller system.	X		
3	Accomplish a given task using Assembly language on a microcontroller.		X	
4	Necessary steps and methods used to interface a microcontroller system to devices such as sensors, etc.			X
5	Interrupts and other advanced concepts related to microcontrollers.	X		
6	Design, development, programming, and testing of a PIC microcontroller based embedded system.			X
7	Design, development, programming, and testing of an Arduino board based embedded system.			X
8	Develop applications using FreeRTOS on Arduino-based controllers.			X

Textbook and References	• John Shapley Gray, ***Interprocess Communications in UNIX*** – Second Edition. Prentice Hall, 462 pages. • Michael Barr, ***Programming Embedded Systems in C and C++*** – O'Reilly. • Embedded System Design: An Introduction to Processes, Tools and Techniques, by A. Berger and, CMP books, 2001. ISBN: 1578200733 • Embedded System Design: A Unified Hardware/Software Introduction, by F. Vahid and T. Givargis, John Wiley and Sons, 2002. ISBN: 0-471-38678-2 • Bill Gallmeister, ***Programming for the Real World - POSIX.4*** – O'Reilly & Associates. • Alan Burns, Andy Wellings, ***Real-Time Systems and Programming Languages*** – Addison Wesley • Mastering the FreeRTOS™ Real Time Kernel – a hands on tutorial guide.

	• The FreeRTOS™ Reference Manual • www.FreeRTOS.org	
Assessment Criteria		**%**
	Quizzes	20
	Midterm exam	30
	Projects	20
	Final Exam	30

Schedule	
Week	**Subject**
1	Introduction to real-time systems, Real-Time Systems Concepts
2, 3, 4	Inter-process Communications • Signals • Pipes • Named pipes (Unix & win32) • Mailslots (win32) • Message queues • Shared memory(Unix & win32) • Sockets • Semaphores
5, 6	Threads • Threads: creating, exiting, management, attributes, scheduling. Using signals in threads. Threads Synchronization (mutex variables, condition variables, read-write locks) • Building multi-threaded applications
7	Introduction to embedded systems • Architecture • Applications • Design Methods
8	Introduction to Microcontrollers (PIC) • Architecture • Building PIC applications • Simulating and debugging • I/O ports • Oscillator • MPLAB

	• PICC • BOOT loaders
9, 10	Microcontrollers and Standard single-purpose processors: Peripherals • Timers • Comparators • PWM generator • Analog-to-digital conversion • Universal Synchronous Asynchronous Receiver Transmitter (USART)
11, 12, 13, 14	Introduction to embedded real-time operating systems: • Why choosing FreeRTOS for the embedded world • Downloading, installing and tuning FreeRTOS • Building applications on Arduino boards using FreeRTOS