Dalhousie University
Department of Electrical and Computer Engineering

ECED 4402: Real Time Systems

Course Outline

Fall 2018

Instructor

Dr. Larry Hughes
Room C-369
Voice: 902.494.3950
Email: larry.hughes@dal.ca
URL: http://lh.ece.dal.ca/eced4402

Office hours

Tuesday	10am-1130am; 1pm-245pm
Thursday	10am-1130am; 1pm-5pm

Introduction

A *system* can be defined as "a group of interacting, interrelated, or interdependent elements forming, or regarded as forming, a collective entity".[1] In the case of a *computer system*, the elements consist of:

Hardware, typically a central processing unit (CPU), memory, some form of bus or buses, and devices for input and output.

Software, usually subdivided into smaller, more manageable elements; at a minimum, system software and application software.

A *real-time system* is a computer system employed in an environment where a number of events, usually external to the computer system, must be accepted and handled within a short period or within certain deadlines. There are two, broad definitions of real-time systems:

Hard real-time, in which each task must be completed before some fixed-time interval has elapsed. Service within this time must be guaranteed. It must make the correct response to external changes within a specific time or deadline, or a failure will be said to have occurred. A control system in a nuclear power plant is a classic example—missing a deadline in such an environment can be fault-*intolerant* if the system is not resilient.

Soft real-time, in which failure to meet a deadline is not considered an application or system failure. An automobile's cruise-control is an example of a soft real-time system—failure to recognize an event will not cause the system to fail, since changes in system state are typically small and repetitious, usually making the system fault-tolerant.

Some features of computer systems are often omitted from real-time systems; for example, if

[1] System. (n.d.) American Heritage Dictionary of the English Language, Fifth Edition. (2011). Retrieved August 9 2017 from http://www.thefreedictionary.com/system

processes are memory-resident, a system's memory management features can be kept to a minimum. Similarly, some embedded real-time systems have no need for a file system, and when file systems are present, they are typically designed for speed rather than optimal storage techniques. The Internet of Things (IoT) can be thought of as a group of embedded real-time systems on a sensor network.

ECED 4402 considers many issues relating to the design and implementation of a real-time system through an examination of ARM's Cortex M4 microprocessor architecture, the Texas Instrument's Tiva TM4C1294NCPDT microcontroller, a message-passing operating system and kernel, and several real-time applications.

Reading material, Software, and Hardware

Course notes will be made available on the course web site:

http://lh.ece.dal.ca/eced4402/

Other course material will also be posted here as well. A significant portion of ECED 4402 involves designing a real-time operating system kernel; accordingly, access to an IDE consisting of a compiler, linker, debugger, and downloader for the Tiva is available from the TI website.

Individual Tiva microcontrollers should be obtained from the electronics workshop (paid for as part of your tuition). Unless informed otherwise, the trainset will be stationed in C-238/239.

Course Objectives

ECED 4402 allows the undergraduate Computer Engineer to develop an understanding of the internals of TI's Cortex M4 and the Tiva microcontroller, how it can be programmed to handle read-time events, and some of the tradeoffs involved when designing an operating system kernel to support real-time events.

By the end of ECED 4402, the student will be able to meet the following objectives and corresponding CEAB attributes and indicators:

Objective	CEAB Attributes/Indicators
Design, implement, and test data structures for controlling real-time systems	Problem Analysis 2.C Investigation 3.C Design 4.B, 4.C Use of Engineering Tools 5.C Individual and Team Work 6.B, 6.B Impact of Engineering on Society 11.C
Create and employ real-time interrupt control software	
Evaluate, design, implement, and verify message-based operating systems	
Design and demonstrate user interface design	
Write and apply real-time testing techniques	
Develop and apply run-time fault detection and correction techniques	

Course Structure

ECED 4402 is a one-semester course, consisting of four components:

- A series of classroom lectures covering the architecture of the Cortex and Tiva, operating systems, real-time operating systems, and real-time applications.

- Three assignments that involve the design, implementation, and testing of various elements of real-time software.
- Two in-class quizzes, covering lecture material and assignments.
- A final examination.

The marking scheme is as follows:

Quizzes	15%
Assignments	45%
Final examination	40%
Total	100%

In order to pass the course, passing grades must be attained in the assignments and the final examination. The final letter-grade will be obtained from Dalhousie's grading system (listed below) using the percentage total of the quizzes, assignments, and final examination.

A+ 90%-100%	A 85%-89%	A- 80%-84%
B+ 77%-79%	B 73%-76%	B- 70%-72%
C+ 65%-69%	C 60%-64%	C- 55%-59%
D 50%-54%		
F <50%		

Course Contents

1. Introduction to course
2. The architecture and organization of the Cortex microprocessor and Tiva microcontroller:
   a) CPU structure
   b) Memory organization
   c) Devices (clock and I/O)
3. Building an operating system's kernel:
   a) Controlling and sharing the hardware
   b) Processes
   c) Devices and Interrupt Service Routines (ISRs)
   d) Inter-process communications
4. Extending the kernel to handle real-time events:
   a) Semaphores and critical regions
   b) Scheduling and priority
   c) Concurrency
   d) Synchronization
   e) Priority issues: inheritance, ceiling, inversion
5. Controlling a real-time application
   a) Virtual machines
   b) Reliable communications
   c) Representing real-time applications

   d) State diagrams
   e) Event handling
   f) Fault tolerance
6. Course Review
7. Final Examination