University of Manitoba | Price Faculty of Engineering
Department of Electrical and Computer Engineering

Course Outline

ECE 3630 - Real-time Embedded Systems

Winter 2026

Instructor
- Ken Ferens, P.Eng.
  E1–544 EITC
  (204) 474–8517
  Ken.Ferens@umanitoba.ca

Office Hours
- By appointment

Teaching Assistant
- Neelofar Vahora
  vahoran@myumanitoba.ca

Contact Hours
- 4 credit hours
- Lectures: 3 hours per week
- Laboratories: 3 hours x 5 weeks

Prerequisites:
- ECE 3610 Microprocessing Systems
- ECE 3740 Systems Engineering Principles 1

Traditional Territories Acknowledgement

The University of Manitoba campuses and research spaces are located on original lands of Anishinaabeg, Ininiwak, Anisininewuk, Dakota Oyate, Dene and Inuit, and on the National Homeland of the Red River Métis.

UM recognizes that the Treaties signed on these lands are a lifelong, enduring relationship, and we are dedicated to upholding their spirit and intent. We acknowledge the harms and mistakes of the past and the present. With this understanding, we commit to supporting Indigenous excellence through active Reconciliation, meaningful change, and the creation of an environment where everyone can thrive. Our collaboration with Indigenous communities is grounded in respect and reciprocity and this guides how we move forward as an institution.

Course Objectives

The main objectives of this course are to: (i) Understand the meaning of abstraction and arbitration in the definition of an operating system (ii) Understand the differences between General Purpose Operating System (GPOS) and Real-time Operating System (RTOS); (iii) Understand the meaning of and differentiate between hard real-time and soft real-time software requirements; (iv) Organize the design of software as a collection of independent threads of execution; (v) Apply the FreeRTOS kernel and create multitasking applications on an embedded microcontroller; (vi) Understand and apply intertask communication mechanisms, including task notifications, queues, semaphores, and event groups; and (vii) and understand resource management in multitasking systems and apply methods used to safely share resources between tasks. Students are evaluated by their demonstration of the specified CEAB attributes and performance in the hands-on projects, midterm test, and final exam.

Course Content

The following topics will be covered:

- Operating Systems: Definition, Past, Present, And Future
  - Kernels and Processes, Concurrency, Memory Management, and Persistent Storage
  - The meaning of abstraction and arbitration as applied to a software operating system in computer systems.
- The difference between General Purpose Operating System (GPOS) and Real-time Operating System (RTOS).
- Real-time software requirements, including hard real-time and soft real software requirements.
- Protection, fault isolation, and security.
- Memory models and management for real-time embedded systems.
- Task and process management and multitasking on single and multi-core processors.
- Design and organization of software using the independent threads of execution approach (tasks and multitasking).
- Interrupt management and application to multitasking of a collection of independent threads of execution approach.
- ARM hardware support for RTOS & exception processing.
- Security for embedded systems: case study of stack buffer overflow.
- The FreeRTOS kernel real-time operating system for embedded systems.
- Resource and critical section management in multitasking systems and application of methods used to safely share resources between tasks.
- Intertask synchronization and communication mechanisms, including queues, binary semaphores, counting semaphores, mutexes, recursive mutexes, event groups and direct to task notifications.

Projects

- STM32F407 Discovery Kit Familiarization, Software Installation, and Porting of FreeRTOS Kernel onto the STM32F407.
- Independent Threads of Execution (Task) Based Design Approach to "Hello World".
- Resource Management 1: Critical sections and mutexes.
- Resource Management 2: Priority Inversion and Methods to Overcome It.
- Application Of Intertask Communication Mechanisms, Including Task Notifications, Queues, Semaphores, And Event Groups.

Textbook

Course notes available on UM Learn..

Mastering the FreeRTOS Real Time Kernel a Hands-on Tutorial Guide: FreeRTOS, “Free RTOS book and reference manual,” FreeRTOS, https://www.freertos.org/Documentation/RTOS_book.html (accessed Aug. 15, 2023).

Important Dates

- **Term Test**
  March 2nd, 2026
  6:00PM – 8:00PM
- **Voluntary Withdrawal Deadline**
  March 19th, 2026
- **Louis Riel Day**
  February 16th, 2026
  No classes or examinations
- **Spring Break**
  February 17th – 20th, 2026
  No classes or examinations
- **Good Friday**
  April 3rd, 2026
  No classes or examinations

Accreditation Details

Accreditation Units

- Mathematics: 0%
- Natural Science: 0%
- Complementary Studies: 0%
- Engineering Science: 50%
- Engineering Design: 50%

Graduate Attributes

KB: A knowledge base for engineering
PA: Problem analysis
IN: Investigation
DE: Design
ET: Use of engineering tools
IT: Individual and team work
CS: Communication skills
PR: Professionalism
IE: Impact of engineering on society/ environment
EE: Ethics and equity
EP: Economics and project management
LL: Life-long learning

Competency Levels

I - Introduced (Introductory)
D - Developed (Intermediate)
A - Applied (Advanced)

Other Resources

The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors by Joseph Yiu.

Operating Systems: Principles and Practice (Volume 1 of 4) by Thomas Anderson and Michael Dahlin

Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau.

Learning Outcomes

1. Identify and differentiate the abstraction and arbitration aspects of a software operating system.
2. Identify and differentiate between hard and soft real-time software requirements.
3. Port FreeRTOS to the STM32F407 Discovery Kit.
4. Design embedded software using the collection of independent threads (tasks) of execution approach (multitasking).
5. Analyze software to identify potential resource management problems, such as critical sections and priority inversion.
6. Demonstrate the ability to design intertask communication mechanisms, including task notifications, queues, semaphores, and event groups.
7. Create test plan and procedures for testing multitasking software.
8. Write design documentation.

Expected Competency Levels

Outcome	KB	PA	IN	DE	ET	IT	CS	PR	IE	EE	EP	LL
1	I	I					D					I
2	D	D	D				D					I
3	D	D	D		D							I
4	D	D	D	A	D							
5	D	D			D							
6	D	D	D	D	D							
7	D				D		D					I
8	D	D			D		D					I

Evaluation

The final course grade will be determined from a student's performance in the projects and on examinations. In order to receive a passing grade in this course:

- All projects must be completed and a passing grade must be achieved.
- A passing grade in the final exam must be achieved.

Component	Value (%)	Method of Feedback	Learning Outcomes Evaluated
Projects, Assignments, Laboratories	25	F, S	1, 2, 3, 4, 5, 6, 7, 8
Term Test	25	F, S	4, 5, 6, 7, 8
Final Examination	50	S	4, 5, 6, 7, 8

* Method of Feedback: **F** - Formative (written comments and/or oral discussion), **S** - summative (numerical grade)

CEAB Graduate Attributes Assessed

DE.1 – Understands the complexities of an open-ended engineering design problem and defines appropriate objectives and constraints.

ET.1C – Uses hands-on tools to complete engineering activities.

Grading Scale

Letter	Mark
A+	95–100
A	85–94
B+	80–84
B	70–79
C+	65–69
C	55–64
D	45–54
F	< 45

Note: These boundaries represent a guide for the instructor and class alike. Provided that no individual student is disadvantaged, the instructor may vary any of these boundaries to ensure consistency of grading from year-to-year.

Student Absences

Attendance in lectures, tutorials, and laboratories is mandatory. For short-term absences due to illness or other extenuating circumstances of 120 hours (5 days) or less, students are required to complete a *Self-Declaration Form for Brief or Temporary Absence* available on the University website.🔗 This form must be submitted to the course instructor within 48 hours of the absence. (No additional documentation is required.)
Note that students are responsible to complete any missed work and must consult with the instructor to make appropriate arrangements.
For absences longer than 120 hours, students must contact the instructor and ECE Undergraduate Advisor, Tammy Holowachuk (Tammy.Holowachuk@umanitoba.ca) for further instructions.

Deferred Final Examinations

Students who miss the regular scheduled writing of a final examination, for valid medical or compassionate reasons, may be given the opportunity to write a deferred examination, subject to approval by the Associate Dean (Undergraduate). All requests for a deferred examination must be made within 48 hours of the missed examination, and must follow the procedure described on the Faculty website, without exception. Course instructors do not have the discretion to grant deferred final examinations.
(https://umanitoba.ca/engineering/student-experience#engineering-student-policies)

Academic Integrity

Students are expected to conduct themselves in accordance with the highest ethical standards of the Profession of Engineering and evince academic integrity in all their pursuits and activities at the university. As such, in accordance with the *General Academic Regulations on Academic Integrity* 🔗, students are reminded that plagiarism or any other form of cheating in examinations, term tests, assignments, projects, or laboratory reports is subject to serious academic penalty (e.g. suspension or expulsion from the faculty or university). This includes the unauthorized use of AI when preparing course deliverables. A student found guilty of contributing to cheating by another student is also subject to serious academic penalty. Integrity also applies to respecting copyrighted course content, which should not be distributed without the creator's permission. Uploading content for the purpose of transcription or other AI-enabled features is commonly a violation of the copyright holder's rights.

Copyright Notice

Requirements and Regulations

- Attendance at lectures and laboratories is essential for successful completion of this course. Students must satisfy each evaluation component in the course to receive a passing final grade.
- It is the responsibility of each student to contact the instructor in a timely manner if they are uncertain about their standing in the course and about their potential for receiving a failing grade. Students should also familiarize themselves with the University's *General Academic Regulations* 🔗, as well as the Price Faculty of Engineering *Academic Regulations* 🔗 dealing with incomplete term work, deferred examinations, attendance and withdrawal.
- No programmable devices or systems (such as calculators, PDAs, smart phones, smart watches, wireless communication or data storage devices) are allowed in examinations unless approved by the course instructor.
- Students should be aware that they have access to an extensive range of resources and support organizations. These include Academic Resources, Counselling, Advocacy and Accessibility Offices as well as documentation of key University policies e.g. Academic Integrity, Respectful Behaviour, Examinations and related matters.

🔗 *Supplemental Resources*

Retention of Student Work

Students are advised that copies of their work submitted in completing course requirements (i.e. assignments, laboratory reports, project reports, test papers, examination papers, etc.) may be retained by the instructor and/or the department for the purpose of student assessment and grading, and to support the ongoing accreditation of each Engineering program. This material shall be handled in accordance with the University's *Intellectual Property Policy* and the protection of privacy provisions of *The Freedom of Information and Protection of Privacy Act (Manitoba)*. Students who do not wish to have their work retained must inform the Head of Department, in writing, at their earliest opportunity.