Toronto Metropolitan University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Outline (W2025)

ELE709: Real-Time Computer Control Systems

Instructor(s)	Dr. Mohamad Shahab [Coordinator] Office: ENG451 Phone: TBA Email: mshahab@torontomu.ca Office Hours: 2-4 PM on Mondays or by appointment
Calendar Description	This course deals with practical techniques for the specification, design and implementation of real-time computer control systems. Topics include: overview of computer control strategies; introduction to real-time systems; hardware and software requirements; implementation of digital control algorithms; design of real-time computer control systems; design analysis; considerations for fault detection and fault tolerance. The lab work and project require solid background in C programming.
Prerequisites	ELE 639 or MEC 830
Antirequisites	None
Corerequisites	None
Compulsory Text(s):	1. ELE709 Course Slides 2. ELE709 Laboratory Manual
Reference Text(s):	1. Real-Time Concepts for Embedded Systems, Q. Li and C. Yao, CRC Press, 2003. Note: According to the book's publisher, a print or an e-book version can be purchased for around C$80. 2. Advanced Linux Programming, M. Mitchell, J. Oldham and A. Samuel, New Riders Publishing, 2001. Note: The book can be downloaded subject to the license mentioned in the following website: https://sourcerytools.github.io/advancedlinuxprogramming/
Learning Objectives (Indicators)	At the end of this course, the successful student will be able to: 1. Learn Concepts of Computer Control. Learn the different classes of industrial process control systems, such as sequence control, control loop, and supervisory control. understand the classification for real-time systems time constraints and the classification of computer programs. Learn the characteristics and requirements of real-time operating systems. Understand scheduling algorithms and their impact on real-time performance. Understand C-coding using Pthread coding technique. Understand Real time operating system (RTOS) building components. Understand and learn three different types of real-time control and their applications. Identify classes of industrial process control systems. Learn properties and requirements for real-time Control systems. Learn the Hardware and software Requirements for designing Real-Time control System application. Learn the difference between General Purpose Operating System (GPOS) verses Real-Time Operating Systems (RTOS) and how both are being used with building a real-time control

system application. Learn Computer Languages for RT applications. Learn the concept for Concurrent Programming with Pthreads (POSIX thread) coding methodology. Learn Thread synchronization and communication. (**1c**)

2. Develop mathematical models of physical systems for control purposes. Explore the differences between analog and digital control systems. Learn about the sampling process and its effects on system performance. Keep up with the latest trends and advancements in real-time control systems, such as the Internet of Things (IoT) and edge computing. Learn different c-coding control protocols to avoid real-time control system failure. Learn different real-time operating system techniques for different task scheduling techniques. Calculate P gain for proportional controller using ultimate gain sensitive method. Apply Anti-wind up technique to improve PID controller performance. Learn and implement Typical digital control system design. Learn the advantages and disadvantages of digital controllers. Learn transferring analog systems into discrete system using three different digital control technique (Forward Rectangular rule, Backward Rectangular rule, and trapezoidal rule). Learn the mathematical models for difference equation, numerical integration, discrete time integrator, Z-transform, discrete transfer function, stability analysis, steady state error, and the relationship between S domain and Z domain. Understand the design considerations for real-time software. Learn Cyclic Execution Approach and scheduling algorithms. (**1d**)
3. Study Pthread (Proxix Thread) C-Programming coding technique in working with multiple tasks, threads, and processor to run concurrent programming. Understand how to apply Pthread codes and how they work. Understand when a thread needs to be joined, detached or terminated within C program. When a mutex is needed to be acquired by the thread and when it has to be released. When a condition variable is used for a specific program and when it is not needed. (**4b**)
4. Study communication protocols and interfaces for connecting system components. Understand the importance of safety in real-time control systems and techniques for ensuring system reliability. Learn about the importance of timing and latency in real-time control systems. Understand techniques to minimize delays and ensure timely execution of control tasks. Understand how quad-core processors can handle pthread work load using multiple threads. C Programing Review. Learn how to implement time and clock codes with in C programming. Build C program using POSIX threads and Concurrent Programming. Learn how to build c-code for resource sharing and coordination between threads. Learn how to design task synchronization and communication. (**5a**)
5. Laboratory and project performance through group work. Work with peers to design and implement real-time control systems using a team-based approach. Contribute effectively to group discussions, brainstorming sessions, and design meetings. Practice planning and managing team projects, including setting goals, establishing timelines, and allocating tasks. Work as a team to test and validate the real-time control system c-code, ensuring it meets specified requirements. Collaborate on troubleshooting and optimizing the system for optimal performance. Answering project related question presenting group members. (**6b**)

NOTE:*Numbers in parentheses refer to the graduate attributes required by the Canadian Engineering Accreditation Board (CEAB).*

Course Organization

3.0 hours of lecture per week for 13 weeks
1.0 hours of lab per week for 12 weeks
0.0 hours of tutorial per week for 12 weeks

Teaching Assistants

Somayeh Barzegar (sbarzegar@torontomu.ca)
Mohsen Ensafjoo (mensafjoo@torontomu.ca)

Course Evaluation

Theory	
Midterm Exam	25 %
Final exam (theory questions)	40 %
Laboratory	
Lab work + Project	25 %

Final exam (lab questions)	10 %
TOTAL:	100 %

Note: In order for a student to pass a course, a minimum overall course mark of 50% must be obtained. In addition, for courses that have both **"Theory and Laboratory"** components, the student must pass the Laboratory and Theory portions separately by achieving a minimum of 50% in the combined Laboratory components and 50% in the combined Theory components. Please refer to the **"Course Evaluation"** section above for details on the Theory and Laboratory components (if applicable).

Examinations	The mid-term exam is on Wednesday, 26 February 2025, during lecture time. The final exam is during the university's final examination period.
Other Evaluation Information	Lab work: Lab submissions are required no more than one week after a lab session. Project: Various parts of the project must be demonstrated, and results must be submitted according to the guidelines in the project document.
Teaching Methods	Lectures will be held in-person in the designated classroom. Lab attendance is mandatory. Course materials and announcements will be posted to the course shell on D2L (https://courses.torontomu.ca).
Other Information	In accordance with the Policy on TMU Student E-mail Accounts (Policy 157), the university requires that any electronic communication by students to TMU faculty or staff be sent from their official university email account.

Course Content

Week	Hours	Chapters / Section	Topic, description
Week 1	3	Lecture slides (and Ch. 1 of Li & Yao)	Introduction to computer control systems, concepts of computer control, classes of industrial process control systems Introduction to real-time systems, classification for real-time systems time constraints, characteristics and requirements of real-time systems
Week 2	3	Lecture slides (and Ch. 4 of Li & Yao)	Hardware and software requirements: general-purpose computers, specialized processors, external interfaces, A/D and D/A conversion, data transfer techniques, data communications techniques, real-time operating systems, computer languages for real-time applications

Week 3	3		Hardware and software requirements (continued)
Week 4	3	Lecture slides (and Ch. 5-8 & 15 of Li & Yao)	Concurrent programming: process and threads, process/thread life cycle, multi-threaded programming with POSIX threads (Pthreads), thread synchronization and communication, semaphores, mutexes, condition variables
Week 5	3		Concurrent programming (continued)
Week 6	3	Lecture slides	Digital control systems: design and implementation of digital controllers, review of discrete-time signal sampling, difference equations, discrete transfer function, z-transform, PID controller design and digital implementation, saturation and integrator wind-up, discretization of continuous-time controllers, control loop synchronization, choice of sampling period, effects of latency and timing jitters on control performance, quantization effects
===		===	No classes during the week of 17 Feb (Study/reading week)
Week 7	3		Digital control systems (continued) The mid-term exam is on Wednesday, 26 February 2025, during lecture time.
Week 8	3		Digital control systems (continued)
Week 9	3	Lecture slides (and Ch. 16 of Li & Yao)	Scheduling of real-time control tasks: cyclic executives, basic rate monotonic scheduling, earliest deadline first, basic response-time analysis, task blocking, transitive blocking, priority inversion, priority inheritance, priority ceiling, immediate priority ceiling, extended rate monotonic scheduling, response-time analysis with blocking starvation, deadlock
Week 10	3		Scheduling of real-time control tasks (continued)
Week 11	3	Lecture slides	Real-time application interface programming, real-time task creation, periodic and aperiodic tasks, interrupt service, routine scheduling policies

Week 12	3	Lecture slides	Design of real-time computer control systems, software life cycle planning analysis and specifications, approaches to real-time software design, tasking design
Week 13	3	Lecture slides	Introduction to reliability and fault-tolerance in computer control systems, reliability types of faults, failure modes, fault prevention: avoidance and removal, fault-tolerance: hardware and software redundancy Review

Laboratory(L)/Tutorials(T)/Activity(A) Schedule

Week	L/T/A	Description
Week 1	No Lab	No Lab
Week 2	Lab 1	C Review
Week 3	Lab 2	Time and Clocks
Week 4	Lab 3	POSIX Threads and Concurrent Programming
Week 5	Lab 3	POSIX Threads and Concurrent Programming (continued)
Week 6	Lab 4	Resource Sharing and Coordination
===	===	No labs during the week of 17 Feb (Study/reading week)
Week 7	Lab 5	Task Synchronization and Communication
Week 8	Lab 5	Task Synchronization and Communication (continued)
Week 9	Project	Real-Time Digital PID Controller Design and Implementation

Week 10	Project	Real-Time Digital PID Controller Design and Implementation (continued)
Week 11	Project	Real-Time Digital PID Controller Design and Implementation (continued)
Week 12	Project	Real-Time Digital PID Controller Design and Implementation (continued)

University Policies & Important Information

Students are reminded that they are required to adhere to all relevant university policies found in their online course shell in D2L and/or on the Senate website

Refer to the Departmental FAQ page for furhter information on common questions.

Important Resources Available at Toronto Metropolitan University

- The Library provides research workshops and individual assistance. If the University is open, there is a Research Help desk on the second floor of the library, or students can use the Library's virtual research help service to speak with a librarian.
- Student Life and Learning Support offers group-based and individual help with writing, math, study skills, and transition support, as well as resources and checklists to support students as online learners.
- You can submit an Academic Consideration Request when an extenuating circumstance has occurred that has significantly impacted your ability to fulfill an academic requirement. You may always visit the Senate website and select the blue radio button on the top right hand side entitled: **Academic Consideration Request (ACR)** to submit this request.

  For Extenuating Circumstances, Policy 167: Academic Consideration allows for a once per semester ACR request without supporting documentation if the absence is less than 3 days in duration and is not for a final exam/final assessment. Absences more than 3 days in duration and those that involve a final exam/final assessment, require documentation. Students must notify their instructor once a request for academic consideration is submitted. See Senate Policy 167: Academic Consideration.
- If taking a remote course, familiarize yourself with the tools you will need to use for remote learning. The Remote Learning Guide for students includes guides to completing quizzes or exams in D2L Brightspace, with or without Respondus LockDown Browser and Monitor, using D2L Brightspace, joining online meetings or lectures, and collaborating with the Google Suite.
- Information on Copyright for Faculty and students.

Accessibility

- Similar to an accessibility statement, use this section to describe your commitment to making this course accessible to students with disabilities. Improving the accessibility of your course helps minimize the need for accommodation.
- Outline any technologies used in this course and any known accessibility features or barriers (if applicable).
- Describe how a student should contact you if they discover an accessibility barrier with any course materials or technologies.

Academic Accommodation Support

Academic Accommodation Support (AAS) is the university's disability services office. AAS works directly with incoming and returning students looking for help with their academic accommodations. AAS works with any student who requires academic accommodation regardless of program or course load.

- Learn more about Academic Accommodation Support.
- Learn how to register with AAS.

Academic Accommodations (for students with disabilities) and Academic Consideration (for students faced with extenuating circumstances that can include short-term health issues) are governed by two different university policies. Learn more about Academic Accommodations versus Academic Consideration and how to access each.

Wellbeing Support

At Toronto Metropolitan University, we recognize that things can come up throughout the term that may interfere with a student's ability to succeed in their coursework. These circumstances are outside of one's control and can have a serious impact on physical and mental well-being. Seeking help can be a challenge, especially in those times of crisis.

If you are experiencing a mental health crisis, please call 911 and go to the nearest hospital emergency room. You can also access these outside resources at anytime:

- **Distress Line:**24/7 line for if you are in crisis, feeling suicidal or in need of emotional support (phone: 416-408-4357)
- **Good2Talk:**24/7-hour line for postsecondary students (phone: 1-866-925-5454)
- **Keep.meSAFE:** 24/7 access to confidential support through counsellors via My SSP app or 1-844-451-9700

If non-crisis support is needed, you can access these campus resources:

- **Centre for Student Development and Counselling:** 416-979-5195 or email csdc@torontomu.ca
- **Consent Comes First - Office of Sexual Violence Support and Education:** 416-919-5000 ext 3596 or email osvse@torontomu.ca
- **Medical Centre:** call (416) 979-5070 to book an appointment

We encourage all Toronto Metropolitan University community members to access available resources to ensure support is reachable. You can find more resources available through the Toronto Metropolitan University Mental Health and Wellbeing website.