Toronto Metropolitan University

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

Course Outline (F2023)

COE538: Microprocessor Systems

Instructor(s)	Dr. Vadim Geurkov [Coordinator] Office: ENG430 Phone: (416) 979-5000 x 556088 Email: vgeurkov@torontomu.ca Office Hours: TBD
Calendar Description	This course introduces students to small microprocessor-based systems, with an emphasis on embedded system hardware and software design. Topics will include microprocessor architecture and structure, with an overview of 8- 16- and 32-bit systems, assembly language programming and the use of high-level languages. Basic input/output including parallel communications with and without handshaking and serial protocols. Hardware and software timing. Using interrupts and exceptions. Overview of single-chip microprocessors and controllers with an emphasis on the Freescale HCS12. The internal structure and design of peripheral devices. Memory system design and analysis. The use and structure of development tools such as (cross) assemblers or compilers, monitor programs, simulators, emulators, etc.
Prerequisites	CEN 199 and COE 328 and ELE 404 and MTH 314
Antirequisites	None
Corerequisites	None
Compulsory Text(s):	1. HCS12/9S12: An Introduction to Software and Hardware Interfacing, Huang, H.-W., Delmar Cengage Learning, 2010. 2. Microprocessor Systems: Selected Course Notes, Lecture slides at D2L.
Reference Text(s):	1. Microcontrollers and Microcomputers: Principles of Software and Hardware Engineering, Cady, F., Oxford University Press, 2010. 2. Introduction to Embedded Systems: Interfacing to the Freescale 9S12, Valvano J., Cengage Learning, 2010. 3. HCS12 Microcontrollers and Embedded Systems, Mazidi, M., Causey, D., and Mazidi, J., Prentice Hall, 2009.
Learning Objectives (Indicators)	At the end of this course, the successful student will be able to: 1. Use technical knowledge on microprocessor architecture, I/O interface and peripherals, assembly/C language programming and debugging methodology. Use design tools and related resources, microprocessor peripherals, assemblers, compilers, and monitor programs. (**4a**) 2. Learn good practices in structuring a microprocessor control program. Apply the programming principles, including top-down programming, bottom-up programming, and functional programming to define an accurate programming problem statement. Recognize that good problem definition assists the program design process. Describe differences between the various approaches to solving a programming problem using assembly/C

language. Select one specific approach. When it fails, analyze the cause of failure using standard programming and debugging methodologies. Based on the analysis, improve the existing approach. Integrate the new suggestions into the existing design plan. Judge the completeness and quality of the generated solutions. (**4b**)

3. Describe the iterative process of programming/debugging assembly/C programs. Use debugging tools to generate information on the current state of a program. Use it to modify/improve the solution as needed. Incorporate and integrate feedback from the instructors and generate new knowledge about the programming problem. (**4c**)
4. A student manages own time and processes effectively to achieve personal and team goals - Assessment of laboratory and project assignments on the correctness and quality of design, decomposing project into key tasks, managing project to meet timeline, code quality, language and technical quality (**6b**)
5. Produce lab and project reports using appropriate format, grammar, and citation styles for technical and non-technical audiences. (**7a**)
6. Illustrate concepts including the structure of assembly/C language programs and obtained experimental results. (**7b**)
7. - Understanding and establishment of project scope - Planning tasks, allocating responsibilities, and setting timelines to meet project goals - Identifying assumptions that may affect project success - Communicating key project deliverables in a clear, concise manner - Displaying a basic understanding of the issues in managing the implementation of the project - Understanding task inter-relationships and managing projects accordingly to time deadlines - Allocating tasks to team members and coordinating dynamically as problems or opportunities emerge (**11b**)

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
2.0 hours of lab per week for 12 weeks
0.0 hours of tutorial per week for 12 weeks

Teaching Assistants

TBA

Course Evaluation

Labs and Quizzes	25 %
Project	10 %
Midterm Exam	25 %
Final Exam	40 %
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

Midterm exam in Week 7, one hour and fifty minutes, closed book (covers Weeks 1-5).
Final exam, during the exam period, three hours, closed-book (covers Weeks 1-12).

Other Evaluation Information

Lab Project
The lab assignments and the project (originally developed by Prof. Peter Hiscocks) involve a robot. The project is to program the eebot mobile robot with a navigation system that can find its way through a maze, reverse, and back its way out again. A possible variation on this is that the

	robot first learns the maze. Then it is started again at the beginning and should navigate the maze without errors. The project must be demonstrated during the demonstration week. The project report must be submitted on or before the end day of the semester. At the time of demonstration, students will also be required to submit the project source code electronically. Lab Management Labs will be graded 8 marks maximum for each lab, to a maximum of 40 marks which will be scaled to 20% of the final mark. And there will be lab quizzes at the end of labs 2, 3, 4 and 5, accounting for a maximum of 5% of the final mark. Credit for labs will be based on the quality of how well the project works (demonstration) and how well the student can answer questions about the lab. If answers to these questions are inadequate, the lab will be marked as 0, although the student will be given an opportunity to rectify his or her preparation. Partial marks may be assigned at the discretion of the instructor. The Lab Project accounts for 10% of the final mark. The project must be demonstrated during the Demonstration Week. The project report must be submitted on or before the end day of the semester. It must include: - A formal description of the work (at least 2 pages, no more than 5 pages) - An appendix containing a hard copy of all source code (.asm file) The proper report description should address the following: - Overall approach and description of performance - Main design decisions - Problems encountered and their solutions - Recommendations:how you would continue the project to make it even better and how you would try to fix any remaining bugs. At the time of demonstration, you will also be required to submit your source code electronically. (You will be told how to do this.) The Project Evaluation will be done according to the following: Evaluation of Lab Project (8%): - 3.5% Basic functionality - 3.5% Code quality - 1.0% Extra functionality Evaluation of Project Report (2%): - 1.0% Report English quality - 1.0% Report technical quality All the labs are done individually. The lab project is done in groups of 3 students. Each student must also keep a complete and continuous record of the year's lab activities. Equipment should not be moved during the lab; if you believe equipment to be defective, report it to the lab instructor who will take care of the problem. Labs are conducted using a Motorola HCS12-based microprocessor board and computer-aided design tools from Freescale, specifically "Special Edition: CodeWarrior for HCS12(X) Microcontrollers (Classic)". To obtain a passing grade in the course, a student must obtain at least 50% in both the lab and theory portions of the course.
Other Information	None

Course Content

Week	Hours	Chapters / Section	Topic, description
1	3	1, 2	Introduction to COE 538 - Scope and objectives, management Introduction to the HCS12 Microcontroller Chapter 1: - 1.3 Computer Hardware Organization - 1.5 Memory system Operation - 1.6 Program Execution - 1.8 The HCS12 CPU Registers - 1.9 HCS12 Addressing Modes - 1.11 A Sample of HCS12 Instructions HCS12 Assembly Programming Chapter 2: - 2.2 Assembly Language Program Structure - 2.3 Assembly Directives
2	3	2, 3, 4	HCS12 Assembly Programming Chapter 2: - 2.5 Writing Programs to Do Arithmetic - 2.6 Program Loops - 2.7 Shift and Rotate Instructions - 2.8 Boolean Logic Instructions - 2.9 Bit Test and Manipulate Instruction - 2.10 Program Execution Time HW/SW Development Tools Chapter 3: - 3.2 Development Tools for the HCS12 - 3.8 Using CodeWarrior Advanced Assembly Programming Chapter 4: - 4.10 Intro to Parallel I/O Port & Devices - 4.11 Simple I/O Devices
3	3	4, 7	Advanced Assembly Programming Chapter 4 - 4.3 Stack - 4.4 What Is a Subroutine? - 4.5 Issues related to Subroutine Calls - 4.6 The Stack Frame - 4.9 Subroutines for Creating Time Delay Advanced Parallel I/O Chapter 7: - 7.5 The HCS12 Parallel Ports - 7.7 Liquid Crystal Displays (LCDs) - 7.8 The HD4478U LCD Controller - 7.9 Interfacing Parallel Ports to a Keypad

4	3	6, 12	Interrupts Chapter 6 - 6.2 Fundamental Concepts of Interrupts Analog-to-Digital Converter Chapter 12 - 12.2 Basics of A/D Conversion - 12.3 The HCS12 A/D converter - 12.4 The Functioning of the ATD Module - 12.5 Procedure for Performing A/D Conversion
5	3	6, 8	Interrupts, Clock Generation and Operation Modes Chapter 6 - 6.3 Resets - 6.4 HCS12 Exceptions - 6.6 Clock and Reset Generation Block - 6.7 Real-Time Interrupt - 6.11 HCS12 Operation Modes Timer Functions Chapter 8 - 8.3 Standard Timer Module - 8.4 Timer Counter Register
6	3	8	Timer Functions Chapter 8 - 8.5 Input-Capture Function - 8.6 Output-Compare Function - 8.7 Pulse Accumulator - 8.8 Modulus down Counter
7	2	1-4, 6, 7, 12	Midterm Covers all material up to week 5 (excluding chapter 8).
8	3	5	C Language Programming Chapter 5 - 5.3 Types Operators and Expressions - 5.4 Control Flow - 5.5 Input and Output - 5.6 Functions and Program Structure - 5.7 Pointers Arrays Structures and Unions - 5.8 Writing C Programs to Perform Simple I/O - 5.11 Using the CodeWarrior to Develop C Programming

9	3	9	Serial Communication Interface Chapter 9 - 9.3 The RS-232 Standard - 9.4 The HCS12 SCI - 9.5 SCI Baud Rate Generation - 9.6 The SCI Operation - 9.9 Interfacing SCI with TIA-232
10	3	10	The SPI Function Chapter 10 - 10.2 Introduction to the SPI Function - 10.3 Registers Related to the SPI Subsystem - 10.4 SPI Operation - 10.5 SPI circuit connection - 10.6 Configuration / Data Transfer in SPI - 10.8 The 74HC595 Shift Register
11	3	11	Inter-Integrated Circuit (I2C) Interface Chapter 11 - 11.2 The I2C Protocol - 11.3 An Overview of the HCS12 I2C Module - 11.4 Registers for I2C Operation - 11.5 Programming the I2C Module
12	3	14	Internal Memory Configuration and External Expansion Chapter 14 - 14.3 Internal Resource Remapping - 14.4 Expanded Memory Mapping - 14.7 HCS12 External Memory Interface - 14.9 Memory Devices - 14.10 Example of External Memory Expansion for the HCS12
13	3	1-12,14	Review and Catch Up
Exam Period	3	1-12,14	Final Exam - Covers material up to the end of week 13

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

Week	L/T/A	Description
2	ENG411	Lab 1: Using the CodeWarrior IDE and Introduction to Assembly Language Programming

3-4	ENG411	Lab 2: Programming the I/O Devices
5-6	ENG411	Lab 3: Battery and Bumper Displays
7	ENG411	Lab 4: Motor Control & Using the Hardware Timer
8-9	ENG411	Lab 5: Robot Roaming Program
10-12	ENG411	Project: Robot Guidance Challenge

University Policies

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

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.