| Instructor(s)               | Vadim Geurkov [Coordinator]  
|                            | Office: ENG430  
|                            | Phone: (416) 979-5000 x 6088  
|                            | Email: vgeurkov@ryerson.ca  
|                            | Office Hours: TBA  
|                            | Prathap Siddavaatam  
|                            | Office: TBA  
|                            | Phone: TBA  
|                            | Email: prathap.siddavaatam@ryerson.ca  
|                            | Office Hours: TBA  |
| 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              | COE 328 and ELE 404 and MTH 314 and CEN 199  
| Antirequisites             | None  
| Corequisites                | None  
|                            | 2. Microprocessor Systems: Selected Course Notes, Clowes, K.  

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. 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. (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 solve 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. Produce lab and project reports using appropriate format, grammar, and citation styles for technical and non-technical audiences. (7a)

5. Illustrate concepts including the structure of assembly/C language programs and obtained experimental results. (7b)

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

<table>
<thead>
<tr>
<th>Course Organization</th>
<th>3.0 hours of lecture per week for 13 weeks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.0 hours of lab/tutorial per week for 12 weeks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Teaching Assistants</th>
<th>TBA</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Course Evaluation</th>
<th>Labs</th>
<th>35 %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Midterm Exam</td>
<td>25 %</td>
</tr>
<tr>
<td></td>
<td>Final Exam</td>
<td>40 %</td>
</tr>
<tr>
<td></td>
<td>TOTAL:</td>
<td>100 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examinations</th>
<th>Midterm exam in Week 8, one hour and fifty minutes, closed book (covers Weeks 1-5).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Final exam, during exam period, three hours, closed-book (covers Weeks 1-13).</td>
</tr>
</tbody>
</table>
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 10 marks maximum for the first lab and 15 marks for each subsequent lab, to a maximum of 70 marks which will be scaled to 35% 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 extra 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 and the project are done individually. 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: 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.
<table>
<thead>
<tr>
<th>Chapter Number</th>
<th>Page Number</th>
<th>Topics</th>
</tr>
</thead>
</table>
| 1              | 3           | Introduction to COE 538  
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           | 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           | 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           | 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 |
<table>
<thead>
<tr>
<th>Page</th>
<th>Week</th>
<th>Content</th>
</tr>
</thead>
</table>
| 5    | 3    | 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 |
| 6    | 3    | Timer Functions  
   Chapter 8  
   - 8.3 Standard Timer Module  
   - 8.4 Timer Counter Register  
   - 8.5 Input-Capture Function  
   - 8.6 Output-Compare Function  
   - 8.7 Pulse Accumulator  
   - 8.8 Modulus down Counter |
| 7    | 3    | 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 |
| 8    | 3    | Midterm  
   (Covers all material up to week 5) |
| 9    | 3    | 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    | 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    | 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 |
### 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

### Laboratory/Tutorials/Activity Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Lab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ENG411</td>
<td>Lab 1: Using the CodeWarrior IDE and Introduction to Assembly Language Programming</td>
</tr>
<tr>
<td>3-4</td>
<td>ENG411</td>
<td>Lab 2: Programming the I/O Devices</td>
</tr>
<tr>
<td>5-6</td>
<td>ENG411</td>
<td>Lab 3: Battery and Bumper Displays</td>
</tr>
<tr>
<td>7</td>
<td>ENG411</td>
<td>Lab 4: Motor Control &amp; Using the Hardware Timer</td>
</tr>
<tr>
<td>8-9</td>
<td>ENG411</td>
<td>Lab 5: Robot Roaming Program</td>
</tr>
<tr>
<td>10-13</td>
<td>ENG411</td>
<td>Project: Robot Guidance Challenge</td>
</tr>
</tbody>
</table>

### Policies & Important Information:

1. Students are required to obtain and maintain a Ryerson e-mail account for timely communications between the instructor and the students;
2. Any changes in the course outline, test dates, marking or evaluation will be discussed in class prior to being implemented;
3. Assignments, projects, reports and other deadline-bound course assessment components handed in past the due date will receive a mark of ZERO, unless otherwise stated. Marking information will be made available at the time when such course assessment components are announced.
4. If you have taken the course previously and are currently looking to get a laboratory exemption, then you must fill out this form: [http://www.ee.ryerson.ca/guides/ECE-LabExemptionForm.pdf](http://www.ee.ryerson.ca/guides/ECE-LabExemptionForm.pdf)
5. Refer to our [Departmental FAQ](https://www.ee.ryerson.ca/guides/Student.Academic.FAQ.html) page for information on common questions and issues at the following link:

### Missed Classes and/or Evaluations

When possible, students are required to inform their instructors of any situation which arises during the semester which may have an adverse effect upon their academic performance, and must request any consideration and accommodation according to the relevant policies as far in advance as possible. Failure to do so may jeopardize any academic appeals.

1. **Health certificates** - If a student misses the deadline for submitting an assignment, or the date of an exam or other evaluation component for health reasons, they should notify their instructor as soon as possible, and submit a Ryerson Student Health Certificate AND an Academic Consideration Request form within 3 working days of the missed date. Both documents are available at [https://www.ryerson.ca/senate/forms/medical.pdf](https://www.ryerson.ca/senate/forms/medical.pdf). If you are a full-time or part-time degree student, then you submit your forms to your own program department or school;
2. **Religious, Aboriginal and Spiritual observance** - If a student needs accommodation because of religious, Aboriginal or spiritual observance, they must submit a Request for Accommodation of Student Religious, Aboriginal and Spiritual Observance AND an Academic Consideration Request form within the first 2 weeks of the class or, for a final examination, within 2 weeks of the posting of the examination schedule. If the requested absence occurs within the first 2 weeks of classes, or the dates are not known well in advance as they are linked to other conditions, these forms should be submitted with as much lead time as possible in advance of the absence. Both documents are available at [www.ryerson.ca/senate/forms/relobservforminstr.pdf](http://www.ryerson.ca/senate/forms/relobservforminstr.pdf). If you are a full-time or part-time degree student, then you submit the forms to your own program department or school;
3. **Academic Accommodation Support** - Before the first graded work is due, students registered with the Academic Accommodation Support office (AAS - [www.ryerson.ca/studentlearningsupport/academic-accommodation-support](http://www.ryerson.ca/studentlearningsupport/academic-accommodation-support)) should provide their instructors with an Academic Accommodation letter that describes their academic accommodation plan.
Academic Integrity

Ryerson’s Policy 60 (the Academic Integrity policy) applies to all students at the University. Forms of academic misconduct include plagiarism, cheating, supplying false information to the University, and other acts. The most common form of academic misconduct is plagiarism - a serious academic offence, with potentially severe penalties and other consequences. It is expected, therefore, that all examinations and work submitted for evaluation and course credit will be the product of each student's individual effort (or an authorized group of students). Submitting the same work for credit to more than one course, without instructor approval, can also be considered a form of plagiarism.

Suspicions of academic misconduct may be referred to the Academic Integrity Office (AIO). Students who are found to have committed academic misconduct will have a Disciplinary Notation (DN) placed on their academic record (not on their transcript) and will normally be assigned one or more of the following penalties:

1. A grade reduction for the work, ranging up to an including a zero on the work (minimum penalty for graduate work is a zero on the work);
2. A grade reduction in the course greater than a zero on the work. (Note that this penalty can only be applied to course components worth 10% or less, and any additional penalty cannot exceed 10% of the final course grade. Students must be given prior notice that such a penalty will be assigned (e.g. in the course outline or on the assignment handout);
3. An F in the course;
4. More serious penalties up to and including expulsion from the University.

The unauthorized use of intellectual property of others, including your professor, for distribution, sale, or profit is expressly prohibited, in accordance with Policy 60 (Sections 2.8 and 2.10). Intellectual property includes, but is not limited to:

1. Slides
2. Lecture notes
3. Presentation materials used in and outside of class
4. Lab manuals
5. Course packs
6. Exams

For more detailed information on these issues, please refer to the Academic Integrity policy (https://www.ryerson.ca/senate/policies/pol60.pdf) and to the Academic Integrity Office website (https://www.ryerson.ca/academicintegrity/).

Important Resources Available at Ryerson

1. The Library (https://library.ryerson.ca/) provides research workshops and individual assistance. Inquire at the Reference Desk on the second floor of the library, or go to library.ryerson.ca/guides/workshops
2. Student Learning Support (https://www.ryerson.ca/studentlearningsupport) offers group-based and individual help with writing, math, study skills and transition support, and other issues.

Approved by: _______________________________                Date ________________________________

Course Instructor

Approved by: _______________________________                Date ________________________________

Associate Chair or Program Director