Engineering Design with Embedded Systems
ECE155, Winter 2011

Patrick Lam*

Brief Overview

This course is a medley of three different topics: software design, project management, and the underlying theme of embedded systems. We will be programming LEGO Mindstorms in this course; they are a friendly introduction to embedded systems and it should be fun to work with them. Also, there are a lot of miscellaneous topics, relating to software development, that electrical and computer engineers ought to know about. This course will help introduce you to some of the most important ones.

Course Description. "Introduction to embedded systems, review of engineering design and analysis principles, software development life cycle, integrated development environments, use of software requirements and specifications, unified modeling language and documentation, event handling, simulation, project management, project scheduling, testing, verification, and maintenance considerations."

Objectives. More specifically, after this course you will be able to:

1. describe the engineering design process (and its application to software);
2. use state-of-the-art integrated development environments;
3. specify, design, analyze, test, and verify software for an embedded system; and
4. work as part of an engineering design team.

General Information

Course Web Page: Information will be posted on UW-ACE.
Schedule:

Regular lectures	MF 13:30-14:20			RCH302
	T 14:30-15:20			RCH302
Extra lectures	Th 16:30-17:20		01/06, 01/20, 02/03, 02/17, 03/10, 03/24	RCH302
Midterm	13:00-14:30		02/14	
Tutorials	T 10:30-11:20	TUT101		MC 4040
	Th 10:30-11:20	TUT102		MC 4040
	F 15:30-16:20	TUT103		DWE 3518
Lab Hours	W 8:30-11:20	LAB201	01/05, 01/19, 02/02, 03/09, 03/23, 04/04	E2 1310
	F 8:30-11:20	LAB202	01/07, 01/21, 02/04, 03/11, 03/21, 03/25	E2 1310

Regular lectures are cancelled for midterm week, February 14-18, and for reading week, February 21-25.

*Thanks to Dr. Bill Bishop and Dr. Rudy Seviora for the overall course design and course materials.

Instructor:

Prof. Patrick Lam
Office: DC2534
Office Hours: Mondays 14:30-15:30, or by appointment
Email: p.lam@ece.uwaterloo.ca
Phone: Use email instead!

I encourage you to see me in my office hours or to communicate with me by email. I'm here to help you learn, but you have to contact me first, and we'll figure out how to best proceed. If my office hours don't work for you, send me an email to set a mutually convenient appointment time.

Lab Instructor:

Sanjay Singh
Office: E2 3343
Phone: x36165
Email: ssingh@uwaterloo.ca

Students may visit the lab instructor to inquire about the laboratory studies in the course and to arrange additional time with the laboratory equipment. An individual appointment with a lab instructor may be booked by contacting the lab instructor via email.

Teaching Assistants:

Khaled Hatem Almotairi
Office: EIT4139
Email: khalmota@bbcr.uwaterloo.ca

Scott (Wei-Ting) Chen
Office: EIT3136
Email: w25chen@uwaterloo.ca

Karim El-Rayes
Office: DC3722
Email: kelrayes@uwaterloo.ca

Yun-Qian (Mike) Miao
Office: E5 5124
Office Hours: Friday 15:00-16:00
Email: yqmiao@uwaterloo.ca

Students may visit the teaching assistants to discuss course materials, tutorial materials, lab materials, examination materials, and assignment solutions. An individual appointment with a teaching assistant may be booked by contacting a teaching assistant via email.

Required Textbooks

- J. Chegwidden and T. Gaddis, Starting Out with C#, Addison-Wesley, February 2005.
- S. Morgan, Programming Microsoft Robotics Studio, Microsoft Press, 2008.
- A. Stellman and J. Greene, Applied Software Project Management, O'Reilly Media, 2006.

Other References of Potential Interest

- S. McConnell, Code Complete, 2nd Edition, Microsoft Press, 2004.
- Microsoft Corporation, The C# Language, http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx, September 2009.

- W. Bishop, Lecture Notes. "ECE 155: Engineering Design with Embedded Systems", Lecture Notes, University of Waterloo, Winter 2010.

Lecture Topics

Here is a detailed list of topics, the objective they contribute to (in parentheses), and estimated lecture hours for each topic.

Introduction to Embedded Systems	2
Integrated Development Environments (2)	2
Event-Driven Programming (3)	3
Software Development Life Cycle (1,3)	2
Engineering Design and Analysis (1)	3
Planning and Estimation (3)	2
Requirements Specifications (3)	2
Unified Modelling Language (3)	2
Project Scheduling (3)	2
Version Control Systems (2,4)	2
Refactoring (2,3)	3
Reviews (3,4)	2
Simulation (2,3)	2
Testing (3)	3
Verification (3)	1
Maintenance (3)	1
Process Improvement (3)	2

The number of lecture hours per topic is approximate, and may vary based on the needs of the class. The number of lecture hours provided is based on reported past experience in teaching the course. Some topics will be introduced early in the course and revisited later in the course.

Labs

This course includes labs, to be done in randomly-assigned groups of 5 or less. The 6 lab sessions last approximately 3 hours each. Lab 0 will introduce the equipment you'll be using in this course. The remaining lab sessions will make up 4 labs. All labs will be in E2 1310. Here is a list of the labs:

		LAB 201	LAB 202
Lab 0	Introduction to the Lab Equipment	January 5	January 7
Lab 1	Colour Detection	January 19	January 21
Lab 2	Following Straight Lines	February 2	February 4
Lab 3	Following Curved Lines	March 9	March 11
Lab 4	Searching for an Object in a Maze	March 23	March 21
		April 4	March 25

In the labs, you will gain experience designing embedded control software for robots using cross-compilation techniques and the Microsoft Robotics Developer Studio. You will write the control software in C#. You will also get to simulate and test your software. We will mark each of the four graded labs based on the quality and performance of your software designs. You must demonstrate working software to a teaching assistant during the lab session and then submit the software online using ACE.

After you have completed all of the labs, you will submit a lab report on ACE. This report is an opportunity for you to practice your technical writing skills. It will document your design process and your experience doing the labs.

Grading of Labs. We will assign each lab submission a grade out of 5, according to a rubric that we'll develop and distribute. We will provide feedback (through ACE) about the quality of your solution. All of the 4 lab submissions contribute to your mark for the course.

Lateness, Collaboration, Extra Credit, etc.

Here are my policies on late submissions, collaboration, extra credit and attendance.

Lateness. In this course, any late deliverable will be penalized 20% of the maximum grade for the deliverable for each business day that the deliverable is submitted late. That is,

- Deliverables handed in 1 day late will be penalized 20% of the maximum grade.
- Deliverables handed in 2 days late will be penalized 40% of the maximum grade.
- Deliverables handed in 3 days late will be penalized 60% of the maximum grade.
- Deliverables handed in 4 days late will be penalized 80% of the maximum grade.
- Deliverables handed in 5 or more days late will be assigned a grade of 0.

Deliverables that are submitted 5 or more days after the submission deadline will be assessed for acceptability. Failure to submit an acceptable attempt at a deliverable will result in an overall course grade of INC (Incomplete) which reverts to a grade of 32 if the deliverable is not completed within 4 months of the end of the course.

Collaboration. I encourage collaboration, but I condemn plagiarism: copying penalizes students who do the work. I will therefore be reporting any cases of plagiarism that I detect.

You are expected to collaborate within your group. Also, you may discuss ideas, design alternatives, and help each other debug small fragments of code. Each group must submit their own, independently-developed, code for each lab. A good heuristic is "look, but don't write:" you can look at other groups' code, but don't do that anywhere that you might be writing your own code.

To be precise, groups are not permitted to share code electronically or in written form, unless such sharing has been clearly documented and acknowledged in the receiving work. An acknowledged fragment will not be considered while grading the assignment (but will not trigger disciplinary penalties). The receiving group will not be further penalized. Acknowledgements must include the name of the providing group and the date of the collaboration.

Extra Credit. In ECE 155, there are no opportunities for extra credit work. Makeup assignments and makeup examinations will not be offered to students under any circumstances.

Attendance. Although I encourage you to attend class, attendance is not graded. If you feel ill, you should immediately seek medical attention. If you miss an exam for health reasons, you need a verification of illness form.

- Forms can be completed by the physicians at Health Services.
- Forms should be presented to the First-Year Engineering Office for verification.

If you anticipate missing an assignment deadline or an examination for a non-medical reason, you should contact me as soon as you are aware of the problem. Given sufficient notice, alternate arrangements may be possible. Alternate arrangements are rare and subject to my discretion.

Readings

Lectures usually go much better if you've read the material beforehand. Here is a list of readings, which will help you understand what's going on at lecture.

Week	Topic	Pages
1, Jan 3	A First Look at Windows Applications	SOWC# pp. 587–658
	Introduction to Microsoft Visual C#.NET	SOWC# pp. 731–736
2, Jan 10	Overview of Robotics	PMSRS pp. 1–18
3, Jan 17	Understanding Services	PMSRS pp. 19–44
4, Jan 24	Autonomous Roaming	PMSRS pp. 125–150
5, Jan 31	Software Project Planning	ASPM pp. 15–32
		ASPM pp. 33–51
6, Feb 7	Software Requirements	ASPM pp. 97–130
7, Feb 14	Project Schedules	ASPM pp. 53–72
	Reviews	ASPM pp. 73–96
8, Feb 21	Version Control Systems	ASPM pp. 131–149
	Refactoring	ASPM pp. 149–155
9, Feb 28	Simulation	PMSRS pp. 71–100
10, Mar 7	Unit Testing	ASPM pp. 156–170
	Software Testing	ASPM pp. 171–199
11, Mar 15	Maintenance	ASPM pp. 203–229
12, Mar 22	Process Improvement	ASPM pp. 277–293

- ASPM: Applied Software Project Management, A. Stellman and J. Greene.
- PMSRS: Programming Microsoft Robotics Studio, S. Morgan
- SOWC#: Starting Out with C#, J. Chegwidden and T. Gaddic

Required inclusions

Academic Integrity: In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. [Check `www.uwaterloo.ca/academicintegrity/` for more information.]

Grievance: A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4, `www.adm.uwaterloo.ca/infosec/Policies/policy70.htm`. When in doubt please be certain to contact the departments administrative assistant who will provide further assistance.

Discipline: A student is expected to know what constitutes academic integrity [check `www.uwaterloo.ca/academicintegrity/`] to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, `www.adm.uwaterloo.ca/infosec/Policies/policy71.htm`. For typical penalties check Guidelines for the Assessment of Penalties, www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm.

Appeals: A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals) `www.adm.uwaterloo.ca/infosec/Policies/policy72.htm`.

Note for Students with Disabilities: The Office for Persons with Disabilities (OPD), located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the OPD at the beginning of each academic term.