SCHOOL OF ELECTRICAL ENGINEERING AND TELECOMMUNICATIONS
ELEC4633: Real Time Engineering
S1 - 2009

1 Course Staff and Consultation

Course convenor/Lecturer:	Dr Ray Eaton	EE Rm 207	R.Eaton@unsw.edu.au
Tutorials:	Anton Delprado		
Laboratory teaching staff:	TBA		

Consultation time for this subject will be Wednesday 9am-11am in my office, room 207. Please make sure that you restrict your enquiries to this time.

Consultation via email is also possible, provided that all email enquiries come from your student email address, otherwise there is no guarantee that emails will be answered.

All email enquiries will be answered in bulk at a couple of chosen times during the week, rather than at the time each email arrives.

2 Course Details

Elec4633 is a 6 UoC course; expected workload is 8-10 hours per week during the 12 week session.

Teaching Methods

Lectures = 2hr/week; Tutorials = 0.5hr/week; Labs = 1.5hrs/week

Lectures, Tutorials and Laboratories

Lecture (all)	Tues	10-12	RedC M030
Tutorial (odd weeks)	Wed	14-15	HutD10 G03
Laboratory (odd & even - only one)	Wed	15-18	EE 201

Tutorial and Laboratory classes officially start in weeks 1 and 2 respectively.

3 Course Overview

Real Time Engineering is concerned with the design and implementation of computer-based real time systems, and deals with the hardware and software issues associated with ensuring they work in a practical and real time sense. Broadly speaking, a system is said to be real time if it adheres to constraints in time. The real time system may be one centred around applications in control systems, signal processing, instrumentation, simulation, and many more. A real time system may be governed by a single stand-alone computer, or a computer embedded within the application itself, and hence known as an *embedded system*.

Assumed Knowledge

The subject follows on from material covered in introductory courses in real time instrumentation (ELEC2042 Real Time Instrumentation) and embedded systems (ELEC2142 Embedded Systems Design). It is assumed that students will have a basic understanding of real time concepts, will be able to program in the C language, and will have had a small exposure in the use of embedded systems.

4 Course Objectives

The objective of this course is to equip students with the necessary skills to analyse, design and implement computer-based real time systems, as well as critically evaluate their performance. With this course and some further experience, students should be able to convert a worded problem specification into a functional and reliable real time solution which satisfies all requirements. Although real time systems encompass a very broad range of application areas, a central theme in this course is the application of real time computing for the purpose of signal processing and control.

The course aims to give students fundamental knowledge in real time operating systems, including scheduling, kernels, and inter-process communication, as well as skills in the effective use of embedded computers.

5 Learning Objectives

At the successful completion of this course, you should be able to:

- develop high-level design strategies for a given real time design problem statement, and to be able to objectively assess the strengths and weaknesses of each strategy,
- demonstrate an ability to think about embedded system design issues, and to design and implement simple real time embedded systems from problem specification to code generation,
- demonstrate an understanding of real time operating system concepts, such as scheduling, multitasking and task/context switching, inter-process communication, and have a knowledge of the different types of real time kernels,
- demonstrate, an understanding of, and a basic competence in, such real time and embedded system aspects as input/output device programming, interrupt programming, and programming of microcontrollers,
- demonstrate an understanding of the fundamentals of real-time embedded systems, including basic knowledge of simple microcontrollers and the more advanced NI Compact Field Point,
- demonstrate an understanding of the RTAI real-time kernel, and be able to use it to construct simple to moderately complex real-time programs.

6 Graduate Attributes

Graduate attributes are those which the University and/or Faculty of Engineering agrees students should develop during their degree. Further information can be obtained in the document, http://www.ltu.unsw.edu.au/content/userDocs/GradAttrEng.pdf. This course aims to contribute to students attaining the following graduate attributes:

- The ability to engage in independent and reflective learning, which is addressed by the laboratory program.
- The skills of effective communication, which are addresses by tutorial problems and assessed in the final exam.
- The capacity for enterprise, initiative and creativity, which is addressed by the laboratory program.
- The capacity for analytical and critical thinking and for creative problem-solving, which is addressed by tutorial and lab exercises.

7 Teaching Strategies

A problem-based learning approach is employed in delivering this course. For each section of the course, different examples of real systems are introduced in lectures as a way of explaining the concepts. The analysis and sometimes design for each example will lend themselves in assisting students satisfy the learning objectives above. The lab program aims to support the lecture program, and provides the student with hands-on software design in interacting with, and controlling real hardware. The laboratory program is challenging, and as such, students must come prepared for each laboratory in order to complete the lab program on time.

The tutorial sessions will also be problem based, with examples of real systems introduced as problem specifications for the students to design a solution. To help facilitate the solutions, tutorials will often involve class discussion.

Students will be required to do home-based work, which will include the self-guided tutorial questions, related to course material covered in lectures, as well as an assignment in the second half of session allowing students to research and report on one of several real time issues or case studies.

8 Assessment

Final Examination	50%
Short tests (2)	10%
Assignment	20%
Laboratory Work (checkpoints)	20%

Final Examination The final examination will be a 3 hour, closed book exam dealing with material in texts, systems engineering concepts, and requiring simple program specification and coding examples. A sample examination paper will be distributed beforehand to compensate for the fact that access to previous relevant examination papers is limited.

Short tests There will be a total of two short tests during the session, to be held in lecture times. The tests will be held in Weeks 5 and 10. The tests will generally consist of analysis and design type question. Marks will be awarded based on the correct answer being supported by correct working out, or in the case of design outline of the choices made and the justification of such choices. The aim of the test questions will be to specifically address one, or more, of the learning outcomes stated above.

Assignment Each student will be required to write a report related to an area of real-time systems and engineering, drawing on information/knowledge gained in the laboratories, lectures, as well as some individual research. Specific topics will be provided, but in previous years, have included such topics as comparing real-time operating systems, comparing scheduling algorithms, or writing a formal lab report on one of the lab exercises.

Laboratory Component Laboratory work commences in Week 2 and all experiments are done in Room 201. Students are only required to attend labs once per fortnight. Each student will carry out 3 lab experiments, and each experiment is made up of a series of checkpoints, some of which are bonus checkpoints. This component of the assessment is worth 20 marks. A signature is required for each checkpoint. Each signature will be grudgingly given, based on completion of the checkpoint, a sufficient understanding of the work done and concepts explored, and a satisfactory record (via a lab book) of the checkpoint completion; unsatisfactory records will not be rewarded, and the decision of the tutor will be final. Tutors may be asked for advice on what is necessary to obtain a signature. More details will be contained in the laboratory handouts. Laboratory work is an essential component of this subject. ***A satisfactory performance in the laboratory is necessary to pass this subject***.

Assessment submission guidelines Hand-in times for the assignment/report will be strictly enforced. A report may attract up to 20 marks; a 10% (2.0 marks) penalty will be imposed for each day or part of a day that the report is late. The report will be due on the Monday of week 12 of session.

The assignment submission should be handed into the assignment box outside G12A, clearly labeled with your name, student number, course name AND lecturers name. A cover sheet will be provided for this purpose and MUST be included.

It is hoped that assignment submissions will be marked and returned within 1 week of the due date.

9 Syllabus

Not all of the topics listed below will be covered in the lecture program, but a majority will.

- An introduction to real time and embedded systems. Sound embedded system design.
- Real time operating systems. Software architectures. Concurrent process. Tasks and threads. Real time kernels. State machines. Linux and RTAI. Scheduling. Dynamic and static scheduling. Rate-Monotonic scheduling. Context switching. Task management.
- Inter-process communication. Data buffering. Semaphores. Critical regions. Mutual exclusion. Message passing. Memory management and global memory. Priority inversion.

- Real time systems programming. Input/output device programming. Interrupts programming. Working with registers and hardware. Numerical issues. Data communication.
- Real time systems analysis. Execution time prediction. Markov chains.
- Design case study.
- Embedded system development. Microcontrollers. The NI Compact Field Point and/or Motorola MC68HC11 microprocessor. Software design issues for embedded systems. Real time embedded operating systems.

10 Recommended Textbooks and Reading Material

The following is a list of books used as references during the course. Each book tends to place an emphasis on different areas of real time systems design, and as such, there is no one book that can be prescribed a stand alone text.

????

Steven F. Barrett, Daniel J. Pack , *Embedded Systems: Design and Applications with the 68HC12 and HCS12*, Prentice Hall, 2005.

Jane W. S. Liu , *Real-Time Systems*, Prentice Hall, 2000.

Alan C. Shaw , *Real-Time Systems and Software*, Wiley, 2001.

Daniel W. Lewis , *Fundamentals of Embedded Software: Where C and Assembly Meet*, Prentice Hall, 2002.

Phillip A. Laplante , *Real-Time Systems Design and Analysis - An Engineer's Handbook*, IEEE Press, 1992.

11 The Laboratory Program

The laboratory work will take place in EE201. This lab is equipped with Pentium machines, each supporting the Linux Operating System, together with the RTAI kernel. Some arrangements may be made for work outside the formal laboratory times, but other classes do use the lab, and will take priority. You are encouraged, if confident enough, to install Linux and RTAI on your own machines which will enable you to undertake much of the laboratory work at home. The University however accepts no responsibility should you decide to do this.

All practical work is to be recorded in a bound workbook. A bound student exercise book would be suitable for the purpose. The workbook will be perused and signed by laboratory supervisors periodically through the session, as discussed below (see Assessment). The workbook should be a diary of your thoughts as you undertake the labs and assignments. Program specifications should be clearly noted, together with clever implementation ideas, results of tests, and suggested extensions. Documentation which does not appear in the programs should be in the notebook. Please see the attached appendix for more explanation of keeping lab notebooks.

All laboratory experiments will be described in simple documents which detail the work to be undertaken, but which leave considerable flexibility in the methods by which this might be achieved. There are three laboratory experiments, however each experiment will work towards the common goal of designing and implementing a real time embedded control system. The expectation is that you will have to work at times other than the assigned laboratory periods to complete the work. Each laboratory experiment will be nominally completed in two laboratory sessions.

11.1 Experiments

The goal of the laboratory program is to design and implement a real time embedded control system. The bench process to be controlled will be a DC motor. At the successful completion of the lab program, the position of the DC motor will be controlled directly and automatically by an embedded microcontroller/computer, with supervision and operator interface being supplied by the RTAI kernel driven desktop PC. The overall task is simplified by breaking it down into smaller tasks. The lab experiments are listed below. The philosophy is to design the software system firstly on the desktop PC, and then once tested, port it, bit-by-bit to the microcontroller.

Multitasking and RT system analysis .

Real-time PC based motor control .

Embedded computer use .

12 Software

A real-time operating system will be used throughout the course. The OS is RTAI, written as a real-time "extension" to the Linux operating system.

Linux is a full-featured POSIX-like OS (so that it "looks" like UNIX), and is becoming an increasingly popular choice of OS. Its source is available as freeware, or alternatively, a distribution can be purchased at a small cost through a variety of distributors.

RTAI is designed to give real-time capabilities to Linux. It does not replace Linux, but simply allows real-time processes to run while Linux operates fully in the background as a low priority process. RTAI is also available as freeware, and can be downloaded. Instructions on how to install the RTAI kernel will be provided online.

The Linux and RTAI kernel versions are relatively new. You may install Linux/RTAI on your home PC's if you are sufficiently confident. Alternatively, you may obtain a bootable Linux/RT-Linux CD from the electronics workshop (G17) for $5 (to cover costs...pay at the School office). Note, the CD does not contain the RTAI kernel, but the RT-Linux kernel. Although different, RTAI and RT-Linux work similarly, and you may use RT-Linux to test some of your lab work, with small changes necessary for compatibility with the lab machines. The CD is installed with the *Knoppix* Linux distribution as well as RT-Linux.

13 WebCT Vista

All material available in electronic format, will be available in WebCT:

http://vista.elearning.unsw.edu.au

Each student enrolled will be granted access to the ELEC4633 subject page in WebCT Vista, where your login is your standard UniPass login.

WebCT Vista is quite versatile, and will be used for such things as:

- displaying/posting notices/messages;
- posting lecture notes / tutorial handouts / lab exercises / short quizzes;
- hosting discussions (only related to the subject) between class/teacher class/class etc;
- posting grades as they become available.

It is encouraged that students seeking advice/help on matters related to the course material seek help from other students, either in person, or via the discussion board in WebCT.

Note: The discussion board is not to be used as a chat forum for subject matter not related to the subject.

14 Continual Course Improvement

Students are advised that the course is under constant revision in order to improve the learning outcomes of its students. Please forward any feedback (positive or negative) on the course to the course convener or via the Course and Teaching Evaluation and Improvement Process.

Appendix on Laboratory Notebooks

According to Barrett [Barrett et.al. 2005], "Lab notebooks are used to record the process of scientific discovery, project evolution, design rationale, steps in engineering analysis, procedures followed, and raw data collected. ... Furthermore, a carefully maintained notebook allows for adequate reconstruction of original work years from the original entry ...".

Barrett also offers some guidelines that should be followed in maintaining a "good" lab book. The following points are based on the guidelines.

1. Ensure the book is bound so that individual pages cannot be lost or removed. By the same token, do not record material on loose sheets of paper, unless they are properly bound.
2. Make sure all entries are sequential in chronological order. Do not mix up recorded lab material with lecture or tutorial (or other) notes.
3. Ensure that the material is recorded in a legible fashion, so that others can read it and use it to reconstruct your experiment.
4. Number pages sequentially.
5. Do not obliterate error, cross them out with a single line.

Importantly, the lab notebooks should NOT be written as formal lab reports where much of the elements in the quote above are not included.

Reference

Barrett S. F., and Pack D. J., *"Embedded Systems: Design and Applications with the 68HC12 and HCS12"*, Prentice Hall, New Jersey, 2005.

Appendix on Plagiarism

The following is an official, now mandatory inclusion in all UNSW course handouts.

"Plagiarism is the presentation of the thoughts or work of another as one's own. [1] Examples include:

- direct duplication of the thoughts or work of another, including by copying work, or knowingly

  permitting it to be copied. This includes copying material, ideas or concepts from a book, article, report or other written

  document (whether published or unpublished), composition, artwork, design, drawing, circuitry, computer program or

  software, web site, Internet, other electronic resource, or another person's assignment without appropriate

  acknowledgement

- paraphrasing another person's work with very minor changes keeping the meaning, form and/or

  progression of ideas of the original;

- piecing together sections of the work of others into a new whole;

- presenting an assessment item as independent work when it has been produced in whole or part in

  collusion with other people, for example, another student or a tutor; and,

- claiming credit for a proportion a work contributed to a group assessment item that is greater than that

  actually contributed. [2]

Submitting an assessment item that has already been submitted for academic credit elsewhere may also be considered

plagiarism.

"The inclusion of the thoughts or work of another with attribution appropriate to the academic discipline does not amount

to plagiarism.

"Students are reminded of their Rights and Responsibilities in respect of plagiarism, as set out in the University

Undergraduate and Postgraduate Handbooks, and are encouraged to seek advice from academic staff whenever

necessary to ensure they avoid plagiarism in all its forms.

"The Learning Centre website is the central University online resource for staff and student information on plagiarism

[1] Based on that proposed to the University of Newcastle by the St James Ethics Centre. Used with kind permission from the University of Newcastle.

[2] Adapted with kind permission from the University of Melbourne.

and academic honesty. It can be located at:
`www.lc.unsw.edu.au/plagiarism`
The Learning Centre also provides substantial educational written materials, workshops, and tutorials to aid students,

for example, in:

- correct referencing practices;
- paraphrasing, summarising, essay writing, and time management;
- appropriate use of, and attribution for, a range of materials including text, images, formulae and

  concepts.

Individual assistance is available on request from The Learning Centre.

"Students are also reminded that careful time management is an important part of study and one of the identified causes

of plagiarism is poor time management. Students should allow sufficient time for research, drafting, and the proper

referencing of sources in preparing all assessment items.