UNSW
THE UNIVERSITY OF NEW SOUTH WALES

FACULTY OF ENGINEERING

SCHOOL OF MECHANICAL AND MANUFACTURING ENGINEERING

MTRN3500

COMPUTING APPLICATIONS IN MECHATRONIC SYSTEMS

SESSION 2, 2009

Contents

MTRN 3500 COMPUTING APPLICATIONS IN MECHATRONIC SYSTEMS

COURSE OUTLINE

COURSE STAFF

A/Prof. Jayantha Katupitiya
Room M9
Mezzanine Level
Mechanical Engineering Laboratory Building
Tel (02) 9385 4096

Email: J.Katupitiya@unsw.edu.au

Dr Ray Eaton
Room 207
Level 2, Electrical Engineering Building
Tel (02) 9385 4059

Email: R.Eaton@unsw.edu.au

Demonstrators: TBA

Consultation times:

Dr J. Katupitiya	**Monday 5-7pm**
Dr R. Eaton	**Tuesday 9-11am**

Consultation is through direct contact at the office of the staff member named above. If you wish to make an appointment outside these times, please call his office number.

COURSE DETAILS

Units of credit

This is a 6 units-of-credit (UoC) course and involves 6 hours per week face-to-face contact.

There will be two 2 hour lecture and two 1 hour /tutorial laboratory classes a week for each student.

UNSW expects that you will put in on average 40 h/w for 24 UoC (including both in-class and out-of-class time) for an effective 14 weeks of the session (12 weeks plus stuvac plus one effective exam week) for an average student aiming for a credit grade. Various factors, such as your own ability, your target grade, etc., will influence the time needed in your case. Some students spend much more than 40 h/w, but you should aim for not less than 40 h/w on coursework for 24 UoC.

This means that you should aim to spend not less than 10 h/w on this course, i.e. an additional 4 h/w of your own time. This should be spent in making sure that you understand the lecture material, practice tutorial questions, further reading about the course material, and revising and learning for the examinations.

Parallel Teaching

There is no parallel teaching in this course.

COURSE AIMS

Description: The first part of this course will develop the programming skills of the students, especially for the purpose of software interfacing with hardware used in the development of Mechatronic Systems. Among these are speed sensors, position sensors, laser range finders, serial interface devices, cameras, tilt sensors and inertial navigation systems. The second part of the course aims to give the students an understanding of real-time systems and equip the students with the necessary skills to develop a complete small to moderate real-time instrumentation system.

The course has laboratory session that will enable the student to train the programming of a specially developed interfacing system that carries a number of experiments.

How the course relates to other course offerings and overall program(s) in the discipline

The courses in the Mechatronics discipline are built up on four different areas. They are; mechanical design, computing, electronics and microprocessors, and control systems. The latter three areas are interrelated and this course forms a corner stone of the fundamental courses on which the Mechatronic Engineering course at UNSW is built upon. A high level of programming skills is necessary to develop customised interface routines to communicate/control various elements of Mechatronic systems. This knowledge is essential in programming control systems and developing software modules for the interface of various hardware elements together to form complete

Mechatronic Systems. As can be seen the contributions from this course to the Mechatronic Engineering degree program is absolutely essential and vital.

STUDENT LEARNING OUTCOMES

By the end of this course, it is expected that you will:

- be thoroughly conversant with structured and modular programming using C/C++.
- be able to interface hardware system to computers through the development drivers/libraries.
- have appreciated the use of software to communicate with sensors to gather information.
- be able to understand how to command actuators through a computer program to effect control action.
- be able to develop prototype user interfaces to assist in the development of controlled Mechatronic systems.
- have developed a fundamental knowledge of what “real-time” means and how it is important for Mechatronic systems.
- have a fundamental knowledge of the systems theory/control aspects required to implement a real-time Mechatronic control system.
- have a fundamental knowledge of the use of numerical methods used in the computer implementation of differential equations.
- have the necessary skills and be able to develop and implement simple real-time programs within the RT-Linux/RTAI operating system.

THE RATIONALE BEHIND YOUR APPROACH TO LEARNING AND TEACHING

Our primary goal is to provide the students a learning atmosphere within which knowledge dissemination by the lecturer and knowledge extraction by the student is facilitated. Within this atmosphere the student will be given a sound theoretical basis for the subject matter. They will also be provided stimuli and resources that they can use to extract further knowledge outside the classroom. This will further be enhance by the compulsory laboratory sessions in which the students will be guided to develop their own (i) software modules to interact with a purpose built interface board that has a number of sensors and actuators, (ii) ability to develop user interfaces for Mechatronic system.

Students learn most effectively when they are confronted with practical systems that relate to lecture content. This will be achieved in this course by requiring compulsory laboratory exercises.

Effective learning requires a system that demands problem solving by students rather than problems solved by the teacher. To facilitate the development of problem solving, the students are provided ample guidance to develop solutions to problems. The essential attributes are the desire to continuously develop the programming skills, especially directed at interfacing.

TEACHING STRATEGIES

Teaching of this course is through lectures, tutorials and laboratory sessions. All laboratory work is individual work and attendance is compulsory.

ASSESSMENT

Continuous assessment on hardware programming/programming (Each task must be fully programmed to be functional, a thorough explanation must be given to a demonstrator and must answer all questions asked by the demonstrator at the time of assessment)	40 marks
Mark required to pass the laboratory component	**30 marks**
End of session examination	60 marks
Mark required to pass the end examination component	**20 marks**
Mark required to pass the course	**50 marks**

Academic honesty and plagiarism

- The following extract must appear in *all* course outlines:

What is Plagiarism?

Plagiarism is the presentation of the thoughts or work of another as one's own.*
Examples include:

- direct duplication of the thoughts or work of another, including by 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.†

For the purposes of this policy, submitting an assessment item that has already been submitted for academic credit elsewhere may be considered plagiarism.

Knowingly permitting your work to be copied by another student may also be considered to be plagiarism.

Note that an assessment item produced in oral, not written, form, or involving live presentation, may similarly contain plagiarised material.

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

The Learning Centre website is main repository for resources for staff and students on plagiarism and academic honesty. These resources can be located via:

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.

* Based on that proposed to the University of Newcastle by the St James Ethics Centre. Used with kind permission from the University of Newcastle
† Adapted with kind permission from the University of Melbourne.

COURSE SCHEDULE

Lectures: **Wednesday, 9 - 11 am, in Mech. Eng. 203**
Tutorial classes: **Wednesday, 11 - 12 pm, in Mech. Eng. 203**
Prerequisites: **COMP1911** Computing I

Week	Topics
1	**Introduction**: Programming needs of Mechatronic Systems. Introduction to the course. Software platforms used for programming within the course.
2	**C Programming Revision:** Editing, Compiling, Linking, Debugging, Loading and executing. Basic building block – the function. Declaration and definitions, calling a function, modular programming, header file formation and definition file formation. Revision of program constructs
3	**Hardware Interfacing – Digital I/O:** Input/output, digital inputs and outputs, reading status of switches and lighting up LEDs. H-bridges and stepper motor control. PWM signals and DC motor control. Reading encoders through polling. Direction discrimination. Position determination
4	**Hardware Interfacing – Analog I/O:** Analog inputs and analog outputs, programming data acquisition systems. Reading potentiometers and driving DC brush type motors.
5	**Writing a driver**: Real-time usable serial port driver. Programming the UART. Using the serial driver to program sensors; laser range finder, tilt sensor, Inertial measurement unit.
6	**Firewire Interface:** Programming imaging devices and simple image processing.
7-8	**Introduction to real-time:** Definitions. Requirements of a real-time operating system. Multitasking. RT-Linux and/or RTAI and how to program in real-time.
9-10	**Numerical methods:** Discrete solution to differential equations. Approximations to the derivative operator. Euler, Tustin's formula. Runge-Kutta integration.
11	**Systems theory:** Differential and difference equations for real-time control. State-space systems. PID control and implementation
12	Revision

Resources for students

As part of the course some handouts will be made available to the students during the course.

Recommended Textbooks

- Dorsey, J., “Continuous and Discrete Control Systems”, McGraw Hill
- Golten, J. and A. Verwer, “Control System Design and Simulation” McGraw Hill

Additional Readings

A number of additional documents will be available on the web. Some materials from earlier years may also be available at WebCT Vista’s MTRN9211 Home page

WebCT

Under MTRN3200 a number of documents including the complete 68HC11 technical data book and other data sheets are be available.

Recommended websites

http://ocw.mit.edu/OcwWeb/Mechanical-Engineering/2-737Spring1999/Syllabus/index.htm **- MIT Open Courseware**

Library

info.library.unsw.edu.au/web/services/services.html

COURSE EVALUATION AND DEVELOPMENT

This course is continuously developed based on the students’ feedback. Whenever possible, the course participates in Course and Teaching Evaluation and Improvement (CATEI) Process. Students are invited to provide any feedback or comments at other times.

ADMINISTRATIVE MATTERS

Information about each of the following matters is presented in a School handout, *Administrative Matters for All Courses*, available from the School office separately or as part of *The Guide*.

It is essential that you obtain a copy, read it carefully and become familiar with the information, as it applies to this course and to each of the other courses in which you are enrolled. Among them are:

> Expectations of students (including attendance at lectures and tutorials/laboratory classes/seminars; and computer use, for example, in the use of email and online discussion forums)
>
> Procedures for submission of assignments and the School's policy concerning late submission Information on relevant Occupational Health and Safety policies and expectations:
>
> www.hr.unsw.edu.au/ohswc/ohs/ohs_home.html
>
> Examination procedures and advice concerning illness or misadventure.

Equity and diversity Students who have a disability that requires some adjustment in their teaching or learning environment are encouraged to discuss their study needs with the course convener prior to, or at the commencement of, their course, or with the Equity Officer (Disability) in the Equity and Diversity Unit (9385 4734 or www.studentequity.unsw.edu.au/content/default.cfm?ss=0). Issues to be discussed may include access to materials, signers or note-takers, the provision of services and additional exam and assessment arrangements. Early notification is essential to enable any necessary adjustments to be made.

A/Prof. Jay Katupitiya
1 July 2009