ELEC3145
Real Time Instrumentation

Course Outline – Semester 2, 2015

Course Staff

Course Convener:	Dr. Alex von Brasch, Room EE338, a.vonbrasch@unsw.edu.au
Laboratory Contact:	Ahmad Baranzadeh, a.baranzadeh@unsw.edu.au

Consultations: You are encouraged to ask questions on the course material, after the lecture class times in the first instance, rather than via email. Lecturer consultation times will be advised during lectures. You are welcome to email the tutor or laboratory demonstrator, who can answer your questions on this course and can also provide you with consultation times. ALL email enquiries should be made from your student email address with ELEC3145 in the subject line, otherwise they may not be answered.

Keeping Informed: Announcements may be made during classes, via email (to your student email address) and/or via online learning and teaching platforms – in this course, we will use Moodle https://moodle.telt.unsw.edu.au/login/index.php. Please note that you will be deemed to have received this information, so you should take careful note of all announcements.

Course Summary

Contact Hours

The course consists of 2 hours of lectures, a 1-hour tutorial, and a 3-hour laboratory session each week. The tutorials will start in Week 2, and the Laboratory in Week 3.

Lectures	**Day**	**Time**	**Location**
	Wednesday	2pm - 4pm	Quad G040
Tutorials	Wednesday	4pm – 5pm	Quad G040

The laboratories are held weekly, Thursday 10-1pm, in EE201.

Context and Aims

This subject is offered in response to observations that real-time computing now plays a dominant part in the realisation of most systems developed by electrical engineers in all sub-disciplines, and to insistence from industry that our graduates should be adequately equipped to deal with real-time systems.

At the end of the course, students should be equipped with a set of skills and tools to be able to undertake a simple to moderately complex instrumentation project. To this end, the purpose of the course is to:

- Provide an understanding of what real-time is, and its importance in many diverse areas of engineering.
- Teach students operating system concepts such as interrupts, multitasking, and data communication.

- Ensure the familiarity with the fundamentals of discrete-time systems, and their significance and representation on digital computers using C programming.
- Provide a basic understanding of physical instrumentation devices, such as A/D and D/A converters.
- Provide an understanding of fundamental systems theory concepts, including differential equations, transfer functions, state-space, numerical integration, and simple feedback (PID) control.
- Allow students to gain practical experience in dealing with the various parts of a simple real-time instrumentation and control system, using the real-time operating system **RTAI**.

Indicative Lecture Schedule

Period	Summary of Lecture Program
Week 1	Real Time and Discrete Time Systems
Week 2	Control Systems Modelling
Week 3	State-Space Techniques
Week 4	Numerical Methods
Week 5	PID Control **(Quiz 1)**
Week 6	Digital Controller Realisations
Week 7	Real Time Systems and Microprocessors
Week 8	Interrupts and Task Switching
Week 9	RTAI and Task Scheduling
Break	
Week 10	Inter-Process Communication **(Quiz 2)**
Week 11	A/D and D/A Conversion
Week 12	Sampling and Scaling; Analog Circuits for Instrumentation

Indicative Laboratory Schedule

Period	Summary of Laboratory Program
Week 3	Lab 1 - MATLAB for Discrete-Time Systems
Week 4	Lab 2 – Numerical Methods in C and MATLAB
Week 5	Lab 3 – PID Control for a Servo-Motor
Week 6	Lab 4 – Linux and the GNU C compiler **(Checkpoint 1)**
Week 7	Lab 5 – RTAI and Real-Time Clock
Week 8	Lab 5 continued
Week 9	Lab 6 – Discrete-Time Filters in RTAI
Break	
Week 10	Lab 6 continued

Week 11	Lab 7 – Real Time Data Acquisition
Week 12	Lab 7 continued **(Lab Exam)**
Week 13	Lab 8 – Multithreading in RTAI **(Checkpoint 2)**

Assessment

Laboratory Checkpoints (2 x 5%)	10%
Lab Exam	10%
Mid-Session Quizzes (2 – 4% and 6%)	10%
Assignment	10%
Final Exam (3 hours)	60%

Course Details

Credits

This is a 6 UoC course and the expected workload is 10–12 hours per week throughout the 13 week semester.

Relationship to Other Courses

This is a 3rd year course in the School of Electrical Engineering and Telecommunications. The course is an elective in the Systems and Control discipline, focused on practical embedded digital control systems.

In ELEC2142, the emphasis is on dedicated real-time systems, which are connected as instrumentation and control devices to other electrical circuits. A later subject, ELEC4633, deals with systems which require increasing sophistication in software design and realisation, particularly in the design of executives, operating systems, and embedded systems.

Pre-requisites and Assumed Knowledge

A satisfactory performance in either COMP1911: Computing 1A or COMP1917: Computing 1, or equivalent, is a required pre-requisite for basic programming skills. Either ELEC2141: Digital Circuit Design, MTRN3200: Elements of Mechatronic Systems, or COMP3222: Digital Circuits and Systems are required pre-requisites for basic background in Digital Logic and Embedded Systems.

Basic competency in First Year Mathematics is assumed. In addition, an introductory knowledge of C-programming will be required for the Laboratory Component.

Following Courses

Students wishing to Real-Time Operating System design and Embedded Control Systems should consider subsequently taking ELEC4633: Real Time Engineering, as a final year technical elective.

Learning outcomes

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

1. Demonstrate an understanding of basic real-time operating system concepts, including interrupt processing, multitasking, inter-process communication.

2. Demonstrate an ability to undertake simple high-level real-time software design, specifically transforming a design specification into a description of software processes needed to support the design.

3. Demonstrate an understanding of, and an ability to effectively use, the RTAI operating system.

4. Use difference equations and discrete-time transfer functions as a means of describing discrete-time systems, and to be able to determine their stability.

5. Demonstrate the use of A/D and D/A converters, and understand their operation.

6. Use transfer functions, state-space, and block diagrams to describe and manipulate continuous time systems.

7. Describe how to use numerical methods such as Runge-Kutta integration and operator substitution for solving differential equations on a digital computer.

8. Demonstrate an understanding of what PID control is, how it is used effectively, and how it is implemented in a digital computer.

This course is designed to provide the above learning outcomes which arise from targeted graduate capabilities listed in ***Appendix A***. The targeted graduate capabilities broadly support the UNSW and Faculty of Engineering graduate capabilities (listed in ***Appendix B***). This course also addresses the Engineers Australia (National Accreditation Body) Stage I competency standard as outlined in ***Appendix C***.

Syllabus

Real Time Instrumentation aims to equip students with the necessary and additional computing and hardware skills to be able to work with, and design real time computer systems which are connected as instrumentation and control devices to other electrical and mechanical circuits. The course is problem-based so that students will address the issues associated with, and concepts behind, building a simple real time computer system. The course revises the concepts of interrupts and introduces the concept of real-time computing, and discussing why time is important and how it is incorporated into a design , multitasking and multithreading and simple inter-process communication. Students will learn about, and be exposed to various devices providing an interface between a computer and the environment. Fundamental signal processing and control will be covered, including discrete-time processing, signal filtering and conditioning, state machines, PID control, and numerical integration. Although the course will exercise analytical skills, there is a strong emphasis on practical implementation using a Real Time Operating System, and using both the C programming language and embedded MATLAB toolboxes to interface to, and control, real hardware.

Teaching Strategies

Delivery Mode

The teaching in this course aims at establishing a good fundamental understanding of the areas covered using:

- Formal face-to-face lectures, which provide you with a focus on the core analytical material in the course, together with qualitative, alternative explanations to aid your understanding;
- Tutorials, which allow for exercises in problem solving and allow time for you to resolve problems in understanding of lecture material;
- Laboratory sessions, which are the central learning environment for this course, and will also provide you with practical construction, measurement and debugging skills.

Learning in this course

You are expected to attend all lectures, tutorials, labs, and mid-semester exams in order to maximise learning. You must prepare well for your laboratory classes and your lab work will be assessed. In addition to the lecture notes, you should read relevant sections of the recommended text. Reading additional texts will further enhance your learning experience. Group learning is also encouraged. UNSW *assumes* that self-directed study of this kind is undertaken in addition to attending face-to-face classes throughout the course.

Lectures

The lectures run every week, and focus on the analytical and background concepts. The topics covered in the lectures align with the laboratory program, and is intended to give the students a deeper perspective in order to get the most out of the practical learning experience.

Tutorial classes

You should attempt all of your problem sheet questions in advance of attending the tutorial classes. The importance of adequate preparation prior to each tutorial cannot be overemphasized, as the effectiveness and usefulness of the tutorial depends to a large extent on this preparation. Group learning is encouraged. Answers for these questions will be discussed during the tutorial class and the tutor will cover the more complex questions in the tutorial class. In addition, during the tutorial class, 1-2 new questions that are not in your notes may be provided by the tutor, for you to try in class. These questions and solutions may not be made available on the web, so it is worthwhile for you to attend your tutorial classes to gain maximum benefit from this course.

Laboratory program

You are required to attend laboratory from Week 3 to Week 13. Laboratory attendance WILL be kept, and you MUST attend at least 80% of labs.

EE201 is the laboratory. This is equipped with computers each supporting the Linux Operating System, together with the RTAI. Arrangements will be made for students to access the lab outside of the scheduled class times. You are encouraged, if you are confident enough, to install Linux and RTAI on your own machines which will enable you to undertake much of the problem-based learning at home. The University however accepts no responsibility should you decide to do this, and given the number and complexity of different operating systems and PC platforms available can only provide limited support in this. An alternative would be to purchase a bootable CD from the school containing the Knoppix Linux distribution, with the RTAI patch already installed. This will allow you to use Linux and RTAI, without the need to install it on your hard drive. The cost of the Knoppix CD is \$5, which can be paid at the EET School Office.

Most of the work in this course is undertaken as problem-based learning within the laboratory program. The lecture program simply supports the problem-based learning.

The laboratory program is based on building up a complete real-time instrumentation and control system for the purpose of controlling the position of a DC motor. Each lab exercise builds on the previous one, where finally a real-time PID controller will be in place to control the DC motor.

Emphasis within the lab program is on digital filters, real-time concepts and instrumentation and control, NOT on learning how to use the Linux operating system and the C programming language. These concepts are in fact essentially treated as assumed knowledge. As such, the Linux OS, C and makefiles will not be taught in lectures and will be the responsibility of students to learn. The preliminary lab exercise addressing the Linux OS and C language

(Lab 4) is provided as an exercise to prepare students for the following laboratory exercises. It is recognized that some students will have less, or little experience in some or all of these areas, and so this exercise is particularly important. Further work will be required for these students initially so that they too feel confident in using these computer skills for the remainder of the laboratory program. It is expected that the preliminary exercise will be completed quickly.

The labs will be available at the start of session, and students are encouraged to move through the laboratory exercises at their own pace. Additional lab access will be arranged outside of class times for any students wishing for additional time in the labs.

Laboratory Exemption

There is no laboratory exemption for this course. Regardless of whether equivalent labs have been completed in previous courses, all students enrolled in this course for Semester 2, 2015 must take the labs. If, for medical reasons, (note that a valid medical certificate must be provided) you are unable to attend a lab, you will need to apply for a catch-up lab during another lab time, as agreed by the laboratory coordinator.

Assessment

The assessment scheme in this course reflects the intention to assess your learning progress through the semester. Ongoing assessment occurs through the lab checkpoints, lab exam and the mid-semester exams.

Laboratory Assessment

Laboratories are primarily about learning, and the laboratory assessment is designed mainly to check your knowledge as you progress through each stage of the laboratory tasks. You are required to maintain a lab book for recording your observations. A lab book is an A4 size notebook containing a mix of plain pages and graph sheets. You have to purchase your own lab book from any stores.

Assessment of the laboratory component will consist of a mark given for two (2) workbook checkpoint(s). The checkpoints will be marked no later than week 6 for checkpoint 1, and week 13 for checkpoint 2 – however, a student may have a checkpoint assessed at an earlier time, when the necessary work has been completed. Lab books will be periodically inspected by lab supervisors to check that labs are being sufficiently documented and for advice. Once again, **insufficient and illegible documentation in your workbooks is not acceptable, and will not be marked. Please refer to the attached appendix for more information.**

Laboratory Exam

To check that you have achieved the practical learning outcomes for the course, you will be examined in the laboratory. Laboratory Exams are closed book practical exams that include programming and analytical calculations. The exam questions will be based on what you have learned in your laboratory classes and lectures, and marks will be awarded for the correct understanding of practical and relevant theoretical concepts, correct execution of your code, and correct interpretation of the resultant behaviour.

The laboratory exam will be held toward the conclusion of session in scheduled lab times in Week 12. Students will attend the lab exam at their assigned laboratory time. The lab exam will test your basic understanding of the lab exercises. The laboratory exam will be taken individually by each student under strict University exam conditions. More information will be given about the exam closer to the time.

Mid-Semester Quizzes

The mid-session quizzes test your general understanding of the course material, and are designed to give you feedback on your progress through the analytical components of the course. Questions may be drawn from any course material from the lectures, tutorials, or laboratories. Marks will be assigned according to the correctness of the responses.

There will be two quizzes. The first will be undertaken in either Week 5 and will address the material covered in the first part of the course. This quiz will be in the form of a take-home exam, with an intended duration of one hour, and is worth 4\%. The Quiz will be available on Moodle after the Tutorial on Wednesday afternoon, and is due before 5pm on the Friday.

The second quiz will also be a take-home exam in Week 10, following a similar format. This quiz is worth 6\%.

More details of the quizzes will be supplied closer to the dates.

Assignment

The assignment allows self-directed study leading to the solution of partly structured problems. Marks will be assigned according to how completely and correctly the problems have been addressed, the quality of the code written for the assignment (must be attached to the report), and the understanding of the course material demonstrated by the report.

The assignment will be due at the end of week 13, and will focus on the design of a discrete-time controller for a target system, most likely using the Simulink toolbox in MATLAB. More details of the assignment will be provided closer to the date.

Late reports will attract a penalty of 10% of the maximum attainable mark per day (including weekends). For example, if the report is submitted one day late the highest mark it could achieve is 90%, and a report submitted could achieve no higher than 80% - even if the raw mark for the assignment two days late was 87%, say, the final mark for that report would be 80%.

Final Exam

The exam in this course is a standard closed-book 3 hour written examination, comprising four compulsory questions. University approved calculators are allowed. The examination tests analytical and critical thinking and general understanding of the course material in a controlled fashion. Questions may be drawn from any aspect of the course (including laboratory), unless specifically indicated otherwise by the lecturer. Marks will be assigned according to the correctness of the responses. *Please note that you must pass the final exam in order to pass the course.*

Relationship of Assessment Methods to Learning Outcomes

	Learning outcomes							
Assessment	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Laboratory practical assessments	✓	✓	✓	✓	✓	✓	✓	✓
Lab exam	✓	✓	✓	✓	✓	✓	-	-
Mid-semester quizzes	-	✓	-	✓	-	✓	✓	✓
Assignment	-	✓	-	✓	✓	✓	✓	✓
Final exam	✓	✓	✓	✓	✓	✓	✓	✓

Course Resources

Textbooks

Prescribed textbook

- None

None of the texts gives an authoritative coverage of material in this course. However, there are several books that will be helpful for particular parts of the course. The first two will be useful for ELEC4633, and provide useful background material in real-time concepts, however will not be useful for those parts of the course dealing with digital filters and systems theory. Students may also consider purchasing a suitable C/C++ reference book.

Reference books

- Alan C. Shaw, **Real-Time Systems and Software**, Wiley, 2001.
- Phillip A. Laplante, **Real-Time Systems Design and Analysis – An Engineer's Handbook**, IEEE Press, 1992.
- K. J. Astrom, B. Wittenmark, **Computer-Controlled Systems: Theory and Design**, 2nd edition, Prentice-Hall, 1990.
- Cay Horstmann, **Computing Concepts with C++ Essentials**, 3rd edition, Wiley, 2003.

On-line resources

Moodle

As a part of the teaching component, Moodle will be used to disseminate teaching materials, host forums and occasionally quizzes. Assessment marks will also be made available via Moodle: https://moodle.telt.unsw.edu.au/login/index.php.

Class notes will be available for each major topic in the course on Moodle.

Mailing list

Announcements concerning course information will be given in the lectures and/or on Moodle and/or via email (which will be sent to your student email address).

Other Matters

Academic Honesty and Plagiarism

Plagiarism is the unacknowledged use of other people's work, including the copying of assignment works and laboratory results from other students. Plagiarism is considered a form of academic misconduct, and the University has very strict rules that include some severe penalties. For UNSW policies, penalties and information to help you avoid plagiarism, see https://student.unsw.edu.au/plagiarism. To find out if you understand plagiarism correctly, try this short quiz: https://student.unsw.edu.au/plagiarism-quiz.

Student Responsibilities and Conduct

Students are expected to be familiar with and adhere to all UNSW policies (see https://student.unsw.edu.au/guide), and particular attention is drawn to the following:

Workload

It is expected that you will spend at least **ten to twelve hours per week** studying a 6 UoC course, from Week 1 until the final assessment, including both face-to-face classes and *independent, self-directed study*. In periods where you need to need to complete assignments or prepare for examinations, the workload may be greater. Over-commitment has been a common source of failure for many students. You should take the required workload into account when planning how to balance study with employment and other activities.

Attendance

Regular and punctual attendance at all classes is expected. UNSW regulations state that if students attend less than 80% of scheduled classes they may be refused final assessment.

General Conduct and Behaviour

Consideration and respect for the needs of your fellow students and teaching staff is an expectation. Conduct which unduly disrupts or interferes with a class is not acceptable and students may be asked to leave the class.

Work Health and Safety

UNSW policy requires each person to work safely and responsibly, in order to avoid personal injury and to protect the safety of others.

Special Consideration and Supplementary Examinations

You must submit all assignments and attend all examinations scheduled for your course. You should seek assistance early if you suffer illness or misadventure which affects your course progress. All applications for special consideration must be **lodged online through myUNSW within 3 working days of the assessment**, not to course or school staff. For more detail, consult https://student.unsw.edu.au/special-consideration.

Continual Course Improvement

This course is under constant revision in order to improve the learning outcomes for all 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. You can also provide feedback to ELSOC who will raise your concerns at student focus group meetings. As a result of previous feedback obtained for this course and in our efforts to provide a rich and meaningful learning experience, we have continued to evaluate and modify our delivery and assessment methods.

Two aspects that will be improved upon this session, based on recent student feedback, are:

- Closer integration of the lecture content with the laboratory program throughout the session.
- Student access to the laboratory outside of the scheduled laboratory class times.

Administrative Matters

On issues and procedures regarding such matters as special needs, equity and diversity, occupational health and safety, enrolment, rights, and general expectations of students, please refer to the School and UNSW policies:
http://www.engineering.unsw.edu.au/electrical-engineering/policies-and-procedures
https://my.unsw.edu.au/student/atoz/ABC.html

Appendix A: Targeted Graduate Capabilities

Electrical Engineering and Telecommunications programs are designed to address the following targeted capabilities which were developed by the school in conjunction with the requirements of professional and industry bodies:

- The ability to apply knowledge of basic science and fundamental technologies;
- The skills to communicate effectively, not only with engineers but also with the wider community;
- The capability to undertake challenging analysis and design problems and find optimal solutions;
- Expertise in decomposing a problem into its constituent parts, and in defining the scope of each part;
- A working knowledge of how to locate required information and use information resources to their maximum advantage;
- Proficiency in developing and implementing project plans, investigating alternative solutions, and critically evaluating differing strategies;
- An understanding of the social, cultural and global responsibilities of the professional engineer;
- The ability to work effectively as an individual or in a team;
- An understanding of professional and ethical responsibilities;
- The ability to engage in lifelong independent and reflective learning.

Appendix B: UNSW Graduate Capabilities

The course delivery methods and course content directly or indirectly addresses a number of core UNSW graduate capabilities, as follows:

- Developing scholars who have a deep understanding of their discipline, through lectures and solution of analytical problems in tutorials and assessed by assignments and written examinations.
- Developing rigorous analysis, critique, and reflection, and ability to apply knowledge and skills to solving problems. These will be achieved by the laboratory experiments and interactive checkpoint assessments and lab exams during the labs.
- Developing capable independent and collaborative enquiry, through a series of tutorials spanning the duration of the course.
- Developing digital and information literacy and lifelong learning skills through assignment work.

Appendix C: Engineers Australia (EA) Professional Engineer Competency Standard

	Program Intended Learning Outcomes	
PE1: Knowledge and Skill Base	PE1.1 Comprehensive, theory-based understanding of underpinning fundamentals	✓
	PE1.2 Conceptual understanding of underpinning maths, analysis, statistics, computing	✓
	PE1.3 In-depth understanding of specialist bodies of knowledge	✓
	PE1.4 Discernment of knowledge development and research directions	
	PE1.5 Knowledge of engineering design practice	✓
	PE1.6 Understanding of scope, principles, norms, accountabilities of sustainable engineering practice	
PE2: Engineering Application Ability	PE2.1 Application of established engineering methods to complex problem solving	✓
	PE2.2 Fluent application of engineering techniques, tools and resources	✓
	PE2.3 Application of systematic engineering synthesis and design processes	
	PE2.4 Application of systematic approaches to the conduct and management of engineering projects	
PE3: Professional and Personal Attributes	PE3.1 Ethical conduct and professional accountability	
	PE3.2 Effective oral and written communication (professional and lay domains)	✓
	PE3.3 Creative, innovative and pro-active demeanour	✓
	PE3.4 Professional use and management of information	✓
	PE3.5 Orderly management of self, and professional conduct	
	PE3.6 Effective team membership and team leadership	

Appendix D: 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.

- 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.
- Make sure all entries are sequential in chronological order. Do not mix up recorded lab material with lecture or tutorial (or other) notes.
- Ensure that the material is recorded in a legible fashion, so that others can read it and use it to reconstruct your experiment.
- Number pages sequentially.
- Do not obliterate errors, 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.