

COMP9032: Microprocessors and Interfacing

Course Outline

(Updated in July 2015)

Aims:

The aims of this course are

- to introduce students the basic concepts and major components of the microprocessor and microcontroller, and
- to provide students with knowledge and skills for solving problems with the microprocessor/microcontroller.

Learning Outcomes:

If successfully completing this course, students will

- know the fundamentals of microprocessors and microcontrollers,
- understand how C programs are converted into machine instructions,
- be proficient in AVR assembly language programming,
- understand typical interface designs between hardware and software,
- know how serial communication works,
- understand how analog signals are converted into digital signals and vice versa, and
- develop basic development skills for microprocessor/microcontroller applications.

Assumed Knowledge:

Before commencing this course, students should:

- be able to program in C,
 - know binary numbers, and
 - have some knowledge of digital circuits.
-

Units of Credit: 6

Co-requisite: COMP9021. **Exclusion:** COM9282

Lecturer in Charge: Hui (Annie) Guo K17-501F Ext. 57136 huig@cse.unsw.edu.au

Teaching Rationale

This course teaches both theory and skills needed to design and implement systems utilising microprocessors. Passing this course involves keeping up with the theory and putting in the time to complete the lab exercises. The lab work brings the theory alive and clarifies deeper issues. Without the theory, the lab work is difficult to master. Each therefore complements the other, and both require your time and effort. If you allocate sufficient time each week, you won't have much problems passing, and it can be a very rewarding experience.

It is our belief that certain activities make a world of difference to your learning experience at the university. You should:

- actively participate in lectures - if something is unclear, ask for clarification. If you don't understand something, try to figure it out when you reflect upon the material during private study, and if you still don't get it, raise it during the next class;
- do not imagine you are the only one having trouble with something; chances are someone else is having trouble as well. However, lecturers or tutors aren't necessarily aware of your problems, so please let them know when something is troubling you; do not leave the problem too late till it is hard to fix.
- prepare tutorials beforehand - know which problems you have difficulty solving – so you will have more understanding of the problems and effectively appreciate the solutions.
- prepare for labs by reading up what is expected well before your lab session - sometimes you will need to complete some preparatory exercises - lab time is too valuable to waste on preparations that could have been completed beforehand.

Teaching Strategies

- Concepts will be introduced during lectures. Students are expected to review these and read related material in order to form a theoretical understanding.
- Practical skills will be developed through laboratory exercises. Staff will provide these beforehand with sufficient time to prepare for them.
- Class test will be used to encourage students to study the subject and to determine the level of understanding achieved.
- A project will be given to you to complete. This project will use most of the concepts learned in class.

Teaching Plan

Lectures:

Each week there is a three-hour lecture. The following topics will be covered:

- Instruction set architecture
- Number systems (a brief review)
- AVR assembly programming and assembly process
- Interrupts
- Buses and I/O
- Serial communication
- Analog/digital and digital/analog conversions.

Laboratories:

The first lab starts in Week 2. Each week you are required to attend a compulsory two-hour laboratory.

You work in the laboratory in a group of TWO students. So if you intend to work with a partner of your choice, let your partner select the same laboratory session as yours. Otherwise, you will be randomly paired with another student if you haven't chosen a partner by the time of the first laboratory session.

There will be five experiments this session. Each lab exercise consists of a number of tasks. After you finish a task, ask the tutor to mark your work.

You will use the AVR Studio to develop, debug and simulate your AVR assembly programs and use the AVR lab board to run your program. The AVR board will be used in Experiment 3 onwards. The lab tutor will assist you with the lab work. However, you cannot ask the tutor for the solution to an exercise.

You are strongly recommended to download AVR Studio and install it on your PC or laptop. For each lab, you need to write programs at home. Otherwise, you will be probably unable to finish it on time.

Tutorials:

There will be no formal tutorial sessions. However, you will be given some questions based on the lecture material. It is recommended that you try your best to solve these questions. Solutions to some of questions will be discussed in the following lecture class.

Assignments:

There will be one project in this session. You are required to complete it on your own. The assignment will be released in Week 9.

Class Test:

During the lecture in week 6, there will be a 60-minute class test.

Examination:

There will be a formal two-hour exam at the end of this course. It is a closed book exam, covering materials from lectures, labs, tutorials and the assignment.

Assessment: To pass this course, you **MUST** get at least 50% of the full marks in lab, return the lab board, and achieve at least 40/100 in the final exam and 50/100 in the final result. Your final result is calculated based on:

- Class test 20%
- Lab exercises 20%
- Assignments 15%
- Final exam 45%

Late penalties:

You are expected to complete each lab on time. The maximum lateness allowed for a lab is ONE WEEK and a 20% late penalty will be applied. For assignment, the late penalty is 10% off per day.

Supplementary Exam:

You will be given a supplementary exam **ONLY IF**

- you have not been caught for plagiarism, and
- your final result is at least 45, or you are on the special list due to illness and other unexpected circumstances.

Note that all requests for special consideration must be put in writing **within one week** of the event at the Student Centre.

Plagiarism:

- **Plagiarism is strictly prohibited** - a mark of ZERO will be given for all works detected where it cannot be shown that it was not copied. For details, please read [CSE Plagiarism Policy](#).

Main References:

- Fredrick M. Cady: Microcontrollers and Microcomputers —Principles of Software and Hardware Engineering, Oxford University Press, 2nd Edition, 2010
- AVR data sheet and instruction manual (on line)

Other References:

- Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, 2005
 - Ronald J. Tocci and Frank J. Ambrosio, Microprocessors and Microcomputers, 6th Edition
 - Steve Furber, ARM System-on-Chip Architecture, Second Edition, Addison-Wesley.
 - Randy H. Katz, Gaetano Borriello, Contemporary Logic Design, Second Edition, Prentice Hall.
-

Provisional Schedule:

Week	Topic	Laboratory	Assignment & Test
1	<i>Introduction to the Microprocessor and Application System</i>	Lab 0	
2	<i>AVR & Programming (I)</i>	Lab 1	
3	<i>AVR & Programming (II)</i>	Lab 1	
4	<i>AVR & Programming (III)</i>	Lab 2	
5	<i>Parallel Input/Output</i>	Lab 2	
6	<i>Interrupts (I)</i>	Lab 2&3	Class Test
7	<i>Interrupts (II)</i>	Lab 3	
8	<i>I/O Devices</i>	Lab 3	
9	<i>Analog Input/Output</i>	Lab 4	Project is released
	<i>break</i>		
10	<i>Serial Communication</i>	Lab 4	
11	<i>More on Microprocessors</i>		
12	<i>Review</i>		Project is due

Course Evaluation and Development

This course is evaluated using the CATEI system. Your feedback (through CATEI and/or other methods) is highly appreciated.

CRICOS Provider Number: 00098G