SUBJECT OUTLINE

48434 Embedded Software

Course area	UTS: Engineering
Delivery	Autumn 2017; City
Subject classification	Field of practice: Electrical Engineering and ICT Engineering majors
Credit points	6cp
Requisite(s)	48430 Fundamentals of C Programming
Result type	Grade and marks

Recommended studies: Knowledge of the C language and digital systems is essential for this subject.

Subject coordinator

Dr Peter McLean
Email: Peter.McLean@uts.edu.au
Room: CB11.11.403
Phone: +61 2 9514 2339

Subject description

This subject develops the student's technical knowledge of the design, implementation and testing of software modules and application frameworks for embedded systems.

Students develop their ability to interpret and evaluate a set of software specifications and work in small groups to write software modules and applications for an embedded system. Students are introduced to abstracting hardware functionality into software modules and researching and implementing software data structures.

Students develop their ability to test and modify their software to ensure compliance with the application specifications and be introduced to reviewing and evaluating their own and others software.

The technical content is contextualised in a project in which students analyse the requirements of an embedded system and design the software to meet those requirements. Skills in debugging software are also developed through the practice-based nature of the subject.

Subject learning objectives (SLOs)

Upon successful completion of this subject students should be able to:

1. Design, write and test a variety of software modules found in modern embedded systems, such as: hardware abstraction layers; data structures; and interrupt service routines.
2. Design, write and test an embedded application that is modular, hierarchical, responsive to real-time requirements, and tightly constrained by time, size and cost.
3. Utilise a variety of software tools to write, execute and test embedded software applications.
4. Test software performance in an embedded system by selecting and using appropriate laboratory equipment.
5. Take responsibility for seeking out and evaluating knowledge from many sources.

Course intended learning outcomes (CILOs)

This subject also contributes specifically to the development of the following faculty Course Intended Learning Outcomes (CILOs) and Engineers Australia (EA) Stage 1 competencies:

- Identify, interpret and analyse stakeholder needs, which is linked to EA Stage 1 Competencies: 1.2, 2.3, 2.4 (A.1)
- Identify and apply relevant problem solving methodologies, which is linked to EA Stage 1 Competencies: 1.1, 2.1, 2.2, 2.3 (B.1)
- Design components, systems and/or processes to meet required specifications, which is linked to EA Stage 1 Competencies: 1.3, 1.6, 2.1, 2.2, 2.3 (B.2)
- Synthesise alternative/innovative solutions, concepts and procedures, which is linked to EA Stage 1 Competencies: 1.1, 3.3 (B.3)
- Apply decision making methodologies to evaluate solutions for efficiency, effectiveness and sustainability, which is linked to EA Stage 1 Competencies: 1.2, 2.1 (B.4)
- Implement and test solutions, which is linked to EA Stage 1 Competencies: 2.2, 2.3 (B.5)
- Demonstrate research skills, which is linked to EA Stage 1 Competencies: 1.4, 2.1 (B.6)
- Apply abstraction, mathematics and/or discipline fundamentals to analysis, design and operation, which is linked to EA Stage 1 Competencies: 1.1, 1.2, 2.1, 2.2 (C.1)
- Manage own time and processes effectively by prioritising competing demands to achieve personal goals, which is linked to EA Stage 1 Competencies: 3.5, 3.6 (D.1)
- Reflect on personal and professional experience to engage independent development beyond formal education for lifelong learning, which is linked to EA Stage 1 Competencies: 3.3, 3.5 (D.2)
- Work as an effective member or leader of diverse teams within a multi-level, multi-disciplinary and multi-cultural setting, which is linked to EA Stage 1 Competencies: 2.4, 3.2, 3.6 (E.2)
- Be able to conduct critical self-review and performance evaluation against appropriate criteria as a primary means of tracking personal development needs and achievements, which is linked to EA Stage 1 Competency: 3.5 (F.1)
- Appreciate ethical implications of professional practice, which is linked to EA Stage 1 Competency: 3.1 (F.2)

Teaching and learning strategies

This subject uses a problem-based learning strategy that allows students to research and develop their own solutions to complex design challenges. Most assessment tasks are practice-based and are designed to reflect current industry practice. A series of staged laboratory tasks allows students to build up a complete software system in a step-by-step hierarchical manner, culminating in a framework which is used for a final project. Face-to-face class time occurs twice per week: one 2 hour lab and one 3 hour lab/assessment.

Student learning is supported in the following way:

1. Prior to each lab, students will be required to study the Notes and associated readings and prepare questions relating to the weekly content and the assesment tasks they are working on.

2. In the lab, students will work in groups of 2 on their laboratory tasks. At the beginning of the lab, academic staff will discuss with the entire group the aims of the lab and the overall challenges they are facing. Groups faced with similar challenges will be prompted to come together to facilitate collaborative discussions.

3. Academic staff are available in each lab to review work and provide immediate feedback.

4. For the final individual project, lab time will be used for one-on-one consultations on particular aspects of a student's work.

Content (topics)

The content covered is divided into the following sections:

1. Embedded Systems
2. Embedded C
3. Microcontroller Architecture
4. Memory
5. Interrupts
6. Timing Generation and Measurement
7. Concurrent Software
8. Interfacing
9. Fixed-Point Processing
10. Real-Time Operating Systems
11. Design Project

Each of these sections addresses an important aspect in modern embedded systems. The intention is that, as you work your way through the subject, your learning will be cumulative. That is, the content you cover in one section should directly help you to understand the topics that follow. A weekly learning schedule, based on a recommended study sequence of the sections, is given in the Program. For each of the above sections, a separate list of topics and suggested reading is also provided in the Study Guide.

Below is a brief summary of the content that is later covered in detail in the Notes.

Prerequisite knowledge

You are expected to have successfully completed subjects in the C language and Introductory Digital Systems.

1 Embedded Systems

An overview of embedded systems is given, before a specific example is treated in detail. The architecture of a popular 32-bit microcontroller is given, in terms of hardware modules and a programming model. Various features of the microcontroller are highlighted, including its architecture, memory map, universal serial bus, serial peripheral interface, enhanced capture timers, analog-to-digital converter, pulse width modulator, and non-volatile Flash memory. Schematics for the hardware platform will be given, showing various pieces of peripheral and interfacing hardware that will be used in the laboratory.

2 Embedded C

Aspects of quality programming, self-documenting code, modular software development and layered software systems will be covered. Special attention is given to the application of the C language to microcontrollers with limited resources.

3 Microcontroller Architecture

The microcontroller architecture will be examined. Specific attention will be given to clock generation and distribution. Microcontrollers have a wealth of built-in peripherals. Some of these peripherals will be examined in depth. The encapsulation of microcontroller peripheral functions in software using “device drivers” will form part of the laboratory program.

4 Memory

Modern 32-bit microcontrollers have many types of memory, such as internal Flash, SRAM and external SDRAM; as well as special function registers that are memory-mapped peripherals. Utilising these different memories and registers requires special software techniques in both C and assembly language.

5 Interrupts

Interrupts are the key to building real-time embedded systems. The interrupt structure and hardware support for interrupts on a 32-bit microcontroller will be examined. Special compiler support for interrupt service routines will be highlighted.

6 Timing Generation and Measurement

The enhanced capture timer module of a 32-bit microcontroller will be examined to see how periodic and aperiodic interrupts are generated as well as how external event capturing can be used to simplify software tasks.

7 Concurrent Software

Foreground and background threads will be covered, as well as multithreaded applications and the basis for real-time operating systems. The concept of shared resources and some mechanisms for accessing them using semaphores and critical sections will be discussed. The concept of thread scheduling will be reviewed.

8 Interfacing

Standard parallel digital interfacing of external devices such as input switches and keyboards, liquid crystal displays and output LEDs is looked at in terms of hardware and software. A microcontroller often needs to handle analog data, such as an automatic control system, or a measurement system. Methods for obtaining, operating on, and producing analog data at the required rate will be reviewed. Peripherals such as an analog-to-digital converter and a pulse width modulator (PWM) will be looked at in detail. The inter-integrated circuit (I2C) interface will be examined – it is used to connect to chips such as analog-to-digital converters, accelerometers, Flash memory and many other special purpose chips.

9 Fixed-Point Processing

Microcontrollers are limited in their data handling capabilities, as they often need to process data in real-time and most do not possess hardware floating-point capabilities. Finite word length effects will be given and methods to overcome them will be examined. Some numerical methods will be presented for the integer evaluation of difficult results such as the square root.

10 Real-Time Operating Systems

An overview of a real-time operating system (RTOS) is given. The process of thread scheduling, pre-emption and thread switching is examined in detail with a simple implementation shown for a priority-based pre-emptive operating system. The design of application software for use in a system with an RTOS is discussed.

11 Design Project

This section brings all the other sections together in a project that requires the analysis and design of an embedded system. You will be required to interpret specifications and come up with sound engineering designs using a variety of methods. The designs will be implemented and experimentally verified.

Program

Week/Session	Dates	Description
1A	13 Mar	***1 - Embedded Systems*** Overview of Embedded Systems. Overview of Tower board. NXP K70F120M architecture. Lab safety. **Notes:** Download and peruse the readings as specified in the Learning Guide.
1B	15 Mar	***2 - Embedded C*** Review of the C language. Kinetis Design Studio. Initializing and accessing I/O ports. Memory allocation. Self-documenting code. Modular software development. Layered software systems. Debugging. **Notes:** Study the readings as specified in the Learning Guide. Download, install and familiarise yourself with the Kinetis Design Studio. Read Lab 1. Prepare questions to ask in Week 2A on any of the prepatory content.
2A	20 Mar	***3 - Microcontroller Architecture*** Clock generation and distribution. UART. PC USB Interface. FIFOs. Polling. Tower serial protocol. **Notes:** Study the readings as specified in the Learning Guide.
2B	22 Mar	Lab work
3A	27 Mar	***4 - Memory*** Flash memory. EEPROM. RAM. Special function registers. Memory-mapped peripherals. **Notes:** Study the readings as specified in the Learning Guide.
3B	29 Mar	Assessment: **Lab 1 Due** **Notes:** Read Lab 2.

4A	3 Apr	***5 - Interrupts*** Interrupts. Interrupt service routines. Hardware interrupts. Interrupt vectors and priority. Exceptions. Threads. Foreground and background threads. Re-entrant programming.
4B	5 Apr	Lab work
5A	10 Apr	***6 - Timing Generation and Measurements*** Timer module. Periodic timer. Output compare. Input capture. Pulse accumulator. **Notes:** Study the readings as specified in the Learning Guide.
5B	12 Apr	Assessment: **Lab 2 Due** **Notes:** Read Lab 3.
6A	17 Apr	***Public Holiday***
6B	19 Apr	Lab work.
S1A	24 Apr	***7 - Concurrent Software*** Threads. Schedulers. Operating systems. The semaphore. Mutual exclusion with semaphores. Synchronisation with semaphores. The producer / consumer problem with semaphores.
S1B	26 Apr	***Mid-Session StuVac***
7A	1 May	***8 - Interfacing*** Input switches and keyboards. Analog to digital conversion. Digital to analog conversion. Inter-Integrated Circuit (I2C). **Notes:** Study the readings as specified in the Learning Guide.
7B	3 May	Assessment: **Lab 3 Due** **Notes:** Read Lab 4.

8A	8 May	***9 - Fixed-Point Processing*** *Q*-notation. Other notations. Fixed-point calculations. Square-root algorithm for a fixed-point processor. **Notes:** Study the readings as specified in the Learning Guide.
8B	10 May	Assessment: **Quiz** Topics 1-8 inclusive.
9A	15 May	***10 - Real-Time Operating Systems*** Real-time kernel concepts. Re-entrancy. Thread priority. Mutual Exclusion. Synchronization. Inter-thread communication. Interrupts. Memory requirements. Advantages and disadvantages of real-time operating systems.
9B	17 May	Assessment: **Lab 4 Due** **Notes:** Read Lab 5.
10A	22 May	***Embedded Software Project*** Overview of project. **Notes:** Read the project specficiation.
10B	24 May	***Embedded Software Project*** Project work.
11A	29 May	***Embedded Software Project*** Project work.
11B	31 May	Assessment: **Lab 5 Due**
12A	5 Jun	***Embedded Software Project*** Project work.
12B	7 Jun	***Embedded Software Project*** Project work.
S2A	12 Jun	***Public Holiday***

S2B	14 Jun	***Final StuVac***
A1A	19 Jun	
A1B	21 Jun	
A2A	26 Jun	
A2B	28 Jun	Assessment: **Project Due**

Additional information

Repeated Failure in this Subject

The Faculty takes repeated failures in a subject seriously and enforces Rule 10.6 of the University's Student and Related Rules. You should read these rules and be aware of the consequences of failure.

If you have failed **twice** before in this subject, then:

(i) You must seek advice from the Subject Coordinator. You will be asked to draw up and submit a study plan that outlines your strategy for passing this subject on the third attempt. A signed copy of this study plan will be kept by the Faculty for internal records.

(ii) If you do not seek advice from the Subject Coordinator by Week 2, then you do not have the Faculty's permission to enrol in the subject. If you stay enrolled in the subject then you will be breaking Rule 10.6.2 (1) of the University's Student and Related Rules.

(iii) You need to be aware that if you fail this subject for a third time, you will need to seek permission from the Deputy Head of School (Teaching & Learning) for any further enrolment in this subject (see below).

If you fail this subject for a **third** time, then:

(i) The Subject Coordinator will deny permission for any further enrolment unless you can produce documentary evidence of extenuating circumstances that require special consideration. In such cases, the Subject Coordinator will refer the matter to the Deputy Head of School (Teaching & Learning), who will grant or deny enrolment for a fourth or subsequent attempt based on a student's overall performance in the course and the extent to which extenuating circumstances have contributed to one or more of the failures.

(ii) If you are granted permission for a fourth or subsequent attempt at this subject, then you must seek continuing assistance throughout this semester from the Subject Coordinator.

Assessment

Late Submission of Assessment Tasks

Unless otherwise specified, late submission of an assessment task will attract a 20% penalty per working day, up to a maximum of 5 working days. If late submission of an assessment item is due to extenuating or special circumstances beyond your control, then you should contact the Subject Coordinator.

Assessment task 1: Labs

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2 and E.2

Type: Laboratory/practical

Groupwork: Group, group and individually assessed

Weight: 40%

Task: Write software that uses:

serial communication to transfer information between the Embedded Hardware and a PC;

non-volatile-memory and system clocks;

interrupts and timers;

serial protocols and other digital interfaces to acquire analog signals;

a human-machine interface or a Real-Time Operating Systems (RTOS).

Due: Week 3 to Week 11

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	17	1, 2	A.1
Functionality of design	17	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	17	1, 2, 3	B.3
Correctness of design	17	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	17	5	D.1, D.2
Efficiency of task performance	15	1, 2, 3	E.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed in a group of 2, and will be awarded the same mark.

Assessment task 2: Quiz

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2 and E.2

Type: Quiz/test

Groupwork: Individual

Weight: 20%

Task: Write a program to utilise non-volatile-memory and system clocks

Due: Week 8
Activity 8B

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	17	1, 2	A.1
Functionality of design	17	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	17	1, 2, 3	B.3
Correctness of design	17	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	17	5	D.1, D.2
Efficiency of task performance	15	1, 2, 3	E.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed individually.

Assessment task 3: Project

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2, E.2, F.1 and F.2

Type: Project

Groupwork: Individual

Weight: 40%

Task: Write a substantial embedded program that uses interrupts, timers, analog I/O and various other microcontroller peripherals to create a real-time, responsive system.

Due: Assessment Week 2

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	14	1, 2	A.1
Functionality of design	14	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	14	1, 2, 3	B.3
Correctness of design	14	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	14	5	D.1, D.2
Efficiency of task performance	14	1, 2, 3	E.2
Evidence of benchmarking	16	1, 2, 3	F.1, F.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed individually.

Assessment feedback

Labs: individual detailed feedback, formative and summative

Quiz: returned work, summative with feedback

Project: returned work, summative with feedback

Required texts

McLean, P., *48434 Embedded Software Notes*, UTS, 2016.

References

Valvano, J.W., *Embedded Systems: Introduction to ARM® CortexTM-M Microcontrollers, 5th Ed.*, CreateSpace Independent Publishing Platform, 2012. ISBN-13: 978-1-47-750899-2

Valvano, J.W., *Real-Time Interfacing to ARM® CortexTM-M Micro-controllers, 5th Ed.*, CreateSpace Independent Publishing Platform, 2015. ISBN-13: 978-1-46-359015-4

Yiu, J.*: The Definitive Guide to ARM® Cortex®-M3 and ARM Cortex®-M4 Processors*, Newnes, 2014. ISBN-13: 978-0-12-408082-9

Graduate attribute development

For a full list of the faculty's graduate attributes and EA Stage 1 competencies, refer to the Student Guide.

Assessment: faculty procedures and advice

Special Consideration

If you believe your performance in an assessment item or exam has been adversely affected by circumstances beyond your control, such as a serious illness, loss or bereavement, hardship, trauma, or exceptional employment demands, you may be eligible to apply for Special Consideration.

Academic integrity

Work submitted electronically may be subject to similarity detection software. Student work must be submitted in a format able to be assessed by the software (e.g. doc, pdf (text files), rtf, html).

For information about avoiding plagiarism see:

https://avoidingplagiarism.uts.edu.au
www.gsu.uts.edu.au/rules/student/section-16.html#r16.2

Academic liaison officer

Academic liaison officers (ALOs) are academic staff in each faculty who assist students experiencing difficulties in their studies due to: disability and/or an ongoing health condition; carer responsibilities (e.g. being a primary carer for small children or a family member with a disability); and pregnancy.

ALOs are responsible for approving adjustments to assessment arrangements for students in these categories. Students who require adjustments due to disability and/or an ongoing health condition are requested to consult a disability services officer in the Special Needs Service before speaking to the relevant ALO.

The ALO for undergraduate students is:

Chris Wong
telephone +61 2 9514 4501

The ALO for postgraduate students is:

Associate Professor Rob Jarman
telephone +61 2 9514 2368

Disclaimer

This outline serves as a supplement to the Faculty of Engineering and Information Technology Student Guide. On all matters not specifically covered in this outline, the requirements specified in the Student Guide apply.