SUBJECT OUTLINE

48434 Embedded Software

Course area	UTS: Engineering
Delivery	Autumn 2018; 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. Research and evaluate 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)
- 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)
- 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)
- 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)

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	14 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	16 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	21 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	23 Mar	Lab work
3A	28 Mar	***4 - Memory*** Flash memory. EEPROM. RAM. Special function registers. Memory-mapped peripherals. **Notes:** Study the readings as specified in the Learning Guide.
3B	30 Mar	***Public Holiday***
4A	4 Apr	**Assessment Task 1 - Lab 1 Due** **Notes:** Read Lab 2.

4B	6 Apr	***5 - Interrupts*** Interrupts. Interrupt service routines. Hardware interrupts. Interrupt vectors and priority. Exceptions. Threads. Foreground and background threads. Re-entrant programming.
5A	11 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	13 Apr	Lab work.
6A	18 Apr	**Assessment Task 1 - Lab 2 Due** **Notes:** Read Lab 3.
6B	20 Apr	***7 - Concurrent Software*** Threads. Schedulers. Operating systems. The semaphore. Mutual exclusion with semaphores. Synchronisation with semaphores. The producer / consumer problem with semaphores.
S1A	25 Apr	***Public Holiday***
S1B	27 Apr	***8 - Interfacing*** Input switches and keyboards. Analog to digital conversion. Digital to analog conversion. Serial Peripheral Interface (SPI). Inter-Integrated Circuit (I2C) interface. **Notes:** Study the readings as specified in the Learning Guide.
7A	2 May	**Assessment Task 1 - Lab 3 Due** **Notes:** Read Lab 4.
7B	4 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.

8A	9 May	**Assessment Task 2 - Quiz (Topics 1-8 inclusive)**
8B	11 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.
9A	16 May	**Assessment Task 1 - Lab 4 Due** **Notes:** Read Lab 5.
9B	18 May	***Embedded Software Project*** Overview of project. **Notes:** Read the project specficiation.
10A	23 May	***Embedded Software Project*** Project work.
10B	25 May	***Embedded Software Project*** Project work.
11A	30 May	**Assessment Task 1 - Lab 5 Due**
11B	1 Jun	***Embedded Software Project*** Project work.
12A	6 Jun	***Embedded Software Project*** Project work.
12B	8 Jun	***Embedded Software Project*** Project work.
S2A	13 Jun	***Final StuVac***
S2B	15 Jun	***Final StuVac***
A1A	20 Jun	
A1B	22 Jun	

A2A	27 Jun	**Assessment Task 3 Project Due**
A2B	29 Jun	

Assessment

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, 4 and 5

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

A.1, B.2, B.4, B.5, B.6 and C.1

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).

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

Due: Week 3 to Week 11
in Class

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	20	1, 2	A.1
Functionality of design	20	1, 2, 3, 4	B.2, B.4, C.1
Correctness of application of theory	20	1, 2, 3, 4	B.5
Correctness of design	20	1, 2, 3, 4	B.5, B.6
Correct use of research and resource evaluation	20	5	B.6

SLOs: subject learning objectives
CILOs: course intended learning outcomes

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.2, B.4, B.5, B.6 and C.1

Type: Quiz/test

Groupwork: Individual

Weight: 20%

Task: Design and write software modules to perform a variety of embedded system tasks, for real-time applications. The quiz will include:

- utilisation of serial I/O

- data structures to support producer/consumer processes

- real-time processing and interfacing

- scenario-based multiple choice questions that test an understanding of the principles of C programming in an embedded environment.

Students will be assessed individually.

Due: Week 8
In Class (Activity 8B)

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	20	1, 2	A.1
Functionality of design	20	1, 2, 3	B.2, B.4, C.1
Correctness of application of theory	20	1, 2, 3	B.4, B.5
Correctness of design	20	1, 2, 3	B.5, B.6
Correct use of research and resource evaluation	20	5	B.6

SLOs: subject learning objectives
CILOs: course intended learning outcomes

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, 4 and 5

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

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

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.

Students will be assessed individually.

Due: < Due Date/ Time >

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	20	1, 2	A.1
Functionality of design	20	1, 2, 3, 4	B.2, B.4, C.1
Correctness of application of theory	10	1, 2, 3, 4	B.5
Correctness of design	10	1, 2, 3, 4	B.2, B.5
Correct use of research and resource evaluation	20	5	B.6
Evidence of benchmarking	20	5	F.1

SLOs: subject learning objectives
CILOs: course intended learning outcomes

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 FEIT Graduate Attributes webpage.

Assessment: faculty procedures and advice

Extensions

When, due to extenuating circumstances, you are unable to submit or present an assessment task on time, please contact your subject coordinator before the assessment task is due to discuss an extension. Extensions may be granted up to a maximum of 5 days (120 hours). In all cases you should have extensions confirmed in writing.

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.

Late Penalty

Work submitted late without an approved extension is subject to a late penalty of 10 per cent of the total available marks deducted per calendar day that the assessment is overdue (e.g. if an assignment is out of 40 marks, and is submitted (up to) 24 hours after the deadline without an extension, the student will have four marks deducted from their awarded mark). Work submitted after five calendar days is not accepted and a mark of zero is awarded.

For some assessment tasks a late penalty may not be appropriate – these are clearly indicated in the subject outline. Such assessments receive a mark of zero if not completed by/on the specified date. Examples include:

a. weekly online tests or laboratory work worth a small proportion of the subject mark, or
b. online quizzes where answers are released to students on completion, or
c. professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date, or
d. take-home papers that are assessed during a defined time period, or
e. pass/fail assessment tasks.

Querying marks/grades and Final Results

If a student disagrees with a mark or a final result awarded by a marker:

- where a student wishes to query a mark, the deadline for a query during teaching weeks is 10 working days from the date of the return of the task to the student
- where a student wishes to query a final examination result, the deadline is 10 working days from the official release of the final subject result.

Further information can be found at Academic advice.

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 discuss their situation with an accessibility consultant at the Accessibility 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:

Dr Nahm Tran
telephone +61 2 9514 4468

Statement about assessment procedures and advice

This subject outline must be read in conjunction with the policy and procedures for the assessment for coursework subjects.

Statement on copyright

Teaching materials and resources provided to you at UTS are protected by copyright. You are not permitted to re-use these for commercial purposes (including in kind benefit or gain) without permission of the copyright owner. Improper or illegal use of teaching materials may lead to prosecution for copyright infringement.

Statement on plagiarism

Plagiarism and academic integrity

At UTS, plagiarism is defined in Rule 16.2.1(4) as: 'taking and using someone else's ideas or manner of expressing them and passing them off as ... [their] own by failing to give appropriate acknowledgement of the source to seek to gain an advantage by unfair means'.

The definition infers that if a source is appropriately referenced, the student's work will meet the required academic standard. Plagiarism is a literary or an intellectual theft and is unacceptable both academically and professionally. It can take a number of forms including but not limited to:

- copying any section of text, no matter how brief, from a book, journal, article or other written source without duly acknowledging the source
- copying any map, diagram, table or figure without duly acknowledging the source
- paraphrasing or otherwise using the ideas of another author without duly acknowledging the source
- re-using sections of verbatim text without using quote marks to indicate the text was copied from the source (even if a reference is given).

Other breaches of academic integrity that constitute cheating include but are not limited to:

- submitting work that is not a student's own, copying from another student, recycling another student's work, recycling previously submitted work, and working with another student in the same cohort in a manner that exceeds the boundaries of legitimate cooperation
- purchasing an assignment from a website and submitting it as original work
- requesting or paying someone else to write original work, such as an assignment, essay or computer program, and submitting it as original work.

Students who condone plagiarism and other breaches of academic integrity by allowing their work to be copied are also subject to student misconduct Rules.

Where proven, plagiarism and other breaches of misconduct are penalised in accordance with UTS Student Rules Section 16 – Student misconduct and appeals.

Avoiding plagiarism is one of the main reasons why the Faculty of Engineering and IT is insistent on the thorough and appropriate referencing of all written work. Students may seek assistance regarding appropriate referencing through UTS: HELPS.

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).

Further information about avoiding plagiarism at UTS is available.

Retention of student work

The University reserves the right to retain the original or one copy of any work executed and/or submitted by a student as part of the course including, but not limited to, drawings, models, designs, plans and specifications, essays, programs, reports and theses, for any of the purposes designated in Student Rule 3.9.2. Such retention is not to affect any copyright or other intellectual property right that may exist in the student's work. Copies of student work may be retained for a period of up to five years for course accreditation purposes. Students are advised to contact their subject coordinator if they do not consent to the University retaining a copy of their work.

Statement on UTS email account

Email from the University to a student will only be sent to the student's UTS email address. Email sent from a student to the University must be sent from the student's UTS email address. University staff will not respond to email from any other email accounts for currently enrolled students.