CM0605 Embedded systems engineering

RTOS Image

This module is designed to provide an in-depth understanding of the critical issues in the engineering of embedded systems. It treats the theory and practical techniques required to implement such systems on both uni-processor and distributed systems. The work will cover aspects of concurrent programming, embedded programming design patterns, safe programming language subsets, reliability, fault tolerance, scheduling and resource management, distributed real-time design and implementation concepts, and the testing of embedded systems. The module will be assessed by means of a portfolio of activities, designed to enable the student to demonstrate achievement of the module learning outcomes. Typically there will be a variety of activities, e.g. a small research activity, programming, pen and paper exercises etc. as appropriate.
The main ideas are communicated in a series of 24 lectures. The laboratory sessions are used by students to put these ideas into practice in a variety of exercises.

1. News

Assignment Specification Published 4th November 2011. Please address any queries about the specification to David Kendall.

2. Module Team

Module tutor
Ian Bradley
Lecturer
Michael Brockway
Lecturer
David Kendall

3. Teaching Arrangements

Lecture Mon 11.00 - 12.00 EB A009
Lecture Mon 12.00 - 13.00 EB A009
Lab/Seminar Tue 09.00 - 11.00 PB S2
Lab/Seminar Fri 09.00 - 11.00 PB S2

You should attend both lecture sessions and one of the lab/seminar sessions every week.

4. Synopsis

The aim of this module is to develop a critical understanding of the engineering issues and principles of uni-processor and distributed embedded systems and, moreover, to develop an in-depth knowledge of the techniques required to develop safe, reliable software capable of responding to real-time changes in the environment.

On completion of this module, students will be able to:

1. Assess the issues relating to software engineering development of embedded systems and apply their knowledge in the creation of such systems.
2. Appraise and produce reliable, fault tolerant real-time software.
3. Evaluate the scheduling and resource management requirements of real-time systems and make effective use of the associated algorithms.
4. Assess the problems involved in developing distributed real-time systems and construct solutions to these problems

5. Teaching Plan

The following is a *provisional* guide to the organisation of *this part* of the module for this year. These arrangements are subject to change during the course of the module. Notice that Ian Bradley and Michael Brockway will provide information about their teaching on this module elsewhere.

Week	W/c	Lecture	Practical
1	26-Sep	Introduction. Key concepts: embedded, real-time, time-triggered, event-triggered. [Slides] *Reading:* [PON10] Chp. 1, [KOP93]	Introduction/review of development tools/environment. The timer. Simple interrupt handler. [Lab]
2	03-Oct	Time-triggered foundations [Slides]	IMB Practical 01
3	10-Oct	Implementing a time-triggered scheduler [Slides]	Time-triggered implementation. [Lab]
4	17-Oct	Using a time-triggered scheduler. Moving to pre-emptive scheduling. uC/OS-II	IMB Practical 02

		review [Slides]	
5	24-Oct	Shared memory communication and synchronisation [Slides]	ucOS-II lab [Lab]
6	31-Oct	Software execution time: measurement [Slides]	IMB Practical 03
7	07-Nov	Software execution time: analysis [Slides]	Software execution time calculation [Lab]
8	14-Nov	Methods, standards and tools: Agile, MISRA, Mercurial [Slides]	MB Practical 01
9	21-Nov	Distributed embedded systems; Predictable communication networks; Controller Area Network (CAN) [Slides]	CAN practical [Lab]
10	28-Nov	CAN message response time analysis [Slides]	MB Practical 02
11	05-Dec	Distributed scheduling analysis 1 [Slides]	Assignment lab/surgery [Lab]
12	12-Dec	Distributed scheduling analysis 2 [Slides]	MB Practical 03

Note:
In addition to the taught sessions, you are expected to undertake independent and directed learning. On average, you should be spending about 8 hours per week on this part of the module.

6. Assessment

Summative assessment comprises:

- A portfolio of work relating to the syllabus topics, carried out over the semester and submitted in week 12. Typically one exercise per major topic which may involve: a small research activity, programming, or a pen and paper exercise as appropriate.

Formative assessment comprises a variety of theoretical and practical exercises with opportunities for discussion with tutors and colleagues.

Feedback on formative assessment will be given during the seminar and laboratory sessions. Additionally the eLearning Portal will be used for more formal feedback on summative assessment.

- Assignment Specification

7. Recommended Reading

Books

- **[LS11]** Introduction to Embedded Systems, A Cyber-Physical Systems Approach, http://LeeSeshia.org, ISBN 978-0-557-70857-4, 2011.
  A free text book by Edward A. Lee and Sanjit A. Seshia that provides a modern introduction to embedded systems.
- **[PON10]** Patterns for time-triggered embedded systems (Local copy)A free text book by Michael Pont that gives a comprehensive coverage of a variety of time-triggered patterns for the 8051 processor. The patterns are clearly described and easy to adapt to other processors.
- **[DOW08]** The Little Book of Semaphores (Local copy) A free text book by Allen Downey that introduces a variety of interesting synchronisation problems and their solution using semaphores.

Papers

- **[DBB07]** Davis, R. et al., Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised, Journal of Real-time Systems, 35:239-272, 2007 [PDF]
- **[KOP93]** Kopetz, H. *Should responsive systems be event-triggered or time-triggered?*, RR 16/93, TUW, Vienna, 1993 [PDF]

8. Resources

- **IAR**
  - Embedded Workbench User Guide
  - C/C++ Development Guide
  - uCOS-II-CSPY User Guide
- **Micrium (site)**
  - uC/OS-II Datasheet
  - uC/OS-II Quick Reference Chart
  - uC/OS-II Reference Manual
  - uC/OS-II Configuration Manual
  - Readme for uC/OS-II on LPC2378
  - Application note for uC/OS-II on LPC2378
- **LPC2378**

  - Olimex LPC2378-STK Documentation for the Olimex development prototype board
  - LPC2377_78 Product Specification (external)
  - LPC23xx User Manual (external)
  - The Insider's Guide to the NXP LPC2300/2400 based Microcontrollers
  - LPC2378 Errata Sheet (external)
- **Miscellaneous**
  - EE Times Embedded Design Centre for Electrical Engineers.
  - Embedded Gurus A blog from a variety of experts on embedded software.
  - The Ganssle Group Jack Ganssle's page has lots of useful embedded systems development information and links to other information sources.
  - Sticky Bits Niall Cooling's blog on developing software for real-time and embedded systems.