The University Of Sheffield.

Electronic & Electrical Engineering.

EEE226 ENGINEERING SOFTWARE DESIGN

Credits: 10

Course Description including Aims

This module builds on the C programming learned in year 1 by exploring both the higher level issues of programming, modelling, and embedded programming. The aim is to develop in students the habits of object orientation (e.g. modularity, data hiding, etc.) using C and MATLAB/Simulink, both commonly used industry standard tools, and writing software for embedded systems. This is done in the belief that these are skills that a 'normal' Electronic Engineer should possess. Four mini projects using C and MATLAB and drawn from across the department are used as a focus for the various activities and to enable students to demonstrate achievement of the module outcomes.

Outline Syllabus

Program structure and organisation;
Approaches to design (e.g. waterfall, spiral, concurrent, agile) and associated tools and methodologies;
Verification and validation;
Modelling;
Hierarchical design: Object orientation (e.g. modularity, data hiding/scope);
Libraries/Toolboxes/interfaces and reuse;
Embedded concepts (relevant to C) e.g. interrupts, hardware support. Real-time software and operating systems;
Projects: small problems looking at modeling aspects (i.e. MATLAB) or embedded design targeted towards microcontroller-based systems accentuating interfacing with sensors and control of a larger system.

Time Allocation

48 hours in total with 1 lecture and 1 x 2 hour practical class per week for 12 weeks (including case studies), and additional set of 12 hours of practical classes for final project. There is the assumption of a further 36 hours of independent practical work and 8 hours of software design based work.

Recommended Previous Courses

Knowledge equivalent to first year C programming (part of EEE160).

Assessment

Assessment will be via three mini projects with a 5% allocation for attendance in the labs.

Recommended Books

The 8051 Microcontroller - A systems approach, M. A. Mazidi, J. G. Mazidi, R. D. McKinlay, 2013

C for Engineers, Bramer, B & S, Anrold 1997

C programming for the absolute beginner, Vine, Michael A., Boston, MA : Thompson Course

Technology, 2007

C for Electronic Engineering: with applied Software Engineering, Buchanan, W, Prentice Hall, 1995

C programming for embedded microcontrollers, Smith, Warwick A., Elektor International Media BV

Exploring C for microcontrollers : a hands on approach, Dordrecht, London : Springer, c2007

Objectives

By the end of the unit, a candidate will be able to:

1 Use standard languages in a practical context, confidently (as evidenced by a student's ability to address the final year project)

2 Design and implement a well-structured program for a 'small' hardware/software system mindful of real-time constraints.

3 Use Matlab/Simulink effectively for modeling problems congruent to various aspects of electronics

Detailed Syllabus

Lecture	Topic
1	Software design and engineering: Problems that can arise and the need for methodologies. Case studies;
2	Design methodologies: waterfall, spiral, concurrent, agile. advantages and disadvantages;
3…4	Embedded system programming concepts: software/hardware delays, switch de-bouncing, pulse counting, considerations for real-time critical functions, compare/capture, loop timeouts, timers, timer based interrupts, external hardware interrupts
5…6	Verification and validation: unit testing, integration testing, system an acceptance testing, example of tools; splint, gcov, cunit
7…8	Introduction to Simulink: building models, using solvers effectively, introduction to basic blocks, case study, linking data to MATLAB workspace, automating simulations, validation of data
9	Diagramming: techniques, Jackson structured charts
10…11	Object Oriented Programming: introduction to classes, instantiation, encapsulation, constructors/destructors, inheritance, polymorphism

UK-SPEC/IET Learning Outcomes

Outcome Code	Supporting Statement
SM1p/SM1m	The underlying principles of good software design and engineering underpin this module. This will be assessed via the organisation and design of the software written in the mini-projects. Assessed by assignment.
SM2p/SM2m	The modelling aspects and problems set will embody various mathematics relevant to the part of the discipline from which problems are drawn. Assessed by assignment.
SM3p/SM3m	The modelling aspects and problems set will embody various principles and knowledge from the part of the discipline from which problems are drawn. Assessed by assignment.
EA1p	Comprehension of and the ability to apply aspects of software 'engineering' will be key to and assessed in the mini-projects – including requirements, design, modelling, test, and documentation. Assessed by assignment.

EA1m	Understanding of engineering principles and the ability to apply them to undertake critical analysis of key engineering processes. Assessed by assignment.
EA2p	Some of the problems will relate to real-time systems and meeting the constraints set will relate critically to design decisions made by students. This aspect will be assessed via the mini-projects. Assessed by assignment.
EA2m	The students will design a model of a component to analyse its performance using Simulink. Assessed by assignment.
EA3p/EA3m	Matlab will be used to create models to solve engineering problems within the mini-projects. This aspect will be assessed via the mini-projects. Assessed by assignment.
EA4p/EA4m	Students will be expected to take a system-level approach to design via the mini-projects. This aspect will be assessed via the mini-projects. Assessed by assignment.
D1p / **D1m**	Students will be expected to create sets of user requirement to underpin the design within the mini-project. Assessed by assignment.
D2m/D2p	The student will be expected to apply their knowledge of design processes in unfamiliar situations through the mini-projects. Assessed by assignment.
D3p / D3m	The students will be given a descriptive specification for their final mini-project and will be expected to make assumptions based on some uncertainties. Assessed by assignment.
D4i	Creativity is a core aspect of software design.
D5p / **D5m**	Whilst the projects are small, the need for management of the overall process is still important – to meet deadlines, for example. Assessed by assignment.
EP2p / **EP2m**	Students will use design tools, languages, and libraries and are expected to become proficient users. Assessed by assignment.
EP4p / **EP4m**	Students will be expected to use relevant manuals and literature to inform their understanding. Assessed by assignment.