MODULE SPECIFICATION

Code: UFEEHM-30-2	**Title: Embedded and Microprocessor Systems**	**Version: 2008**
Level: 2	**UWE credit rating: 30**	**ECTS credit rating: 15**
Module Type: Standard		
Owning Faculty: Bristol Institute of Technology	**Field: Computer Science**	
Valid from: 1st September 2008	**Discontinued From: 31st August 2009**	
Pre-requisites:	**UFEEHE-30-1 Programming in C OR UFEEHF-30-1 Computer Systems Architecture OR UFEE7A-20-1 Introduction to Software Development in C OR UFEETS-20-1 Programming in C OR UFEEHV-20-1 Computer Systems**	
Co-requisites:	**None**	
Excluded combinations:	**None**	

Learning Outcomes

On completion of this module a student will typically be able to:-		*Assessed in component(s):*
A. Show a detailed knowledge and understanding of		
i)	the uses of C as a high level language and as an alternative to an assembler.	B
ii)	the data structures used to implement high level languages and features such as local variables and parameter passing …	A, B
B. Demonstrate subject specific skills with respect to		
i)	developing further the students' knowledge of C as a language for cross development	B
ii)	Give the students practical experience of designing, building and booting a small micro-controller based systems.	B
iii)	Design, code and boot small systems using C.	B
C. Show cognitive skills with respect to		
i)	Understand the design and integration of a modern microprocessor based system	A, B
ii)	Understand the operation and programming of various peripheral devices such as serial ports, timers and io ports	A, B

D. Demonstrate key transferable skills in

i)	communication skills	A, B
ii)	self-management skills	A, B
iii)	IT skills in context	A, B
iv)	problem formulation and decision making	A, B

Syllabus Outline

Introduction to C: Basic types, control structures, operators; Data structures in C; Pointers; Scope and functions; Preprocessor; ANSI, POSIX and K&R standards.

Linking C and an assembler: Writing environment initialisation files; Mixed language programming; Passing parameters between HLLs and ALPs; Writing interrupt service routines.

Software tools for cross development: Understanding how compilers work; Using linkers and linker options; Assemblers and compilers; S-record and Intel Hex downloaders;

Compilation control programmes; Writing support code for compilers; How to select tools for cross-development; Make and debuggers.

Interfacing in C: Writing portable code for low level operations; Programming serial ports; Programming parallel ports; Programming serial ports; Programming timers; Introduction to writing device drivers; Writing small schedulers.

Teaching and Learning Methods

The course will include lectures but a large emphasis of the course will be practical, with students in practical labs and tutorials. Individual worksheet and group-oriented practical exercises are central to the students\' experience in order to reinforce and extend the lectures and associated readings. The laboratory work includes both hardware and software, at basic unit and higher system level. An extended case-study, supported by focussed tutorials and practicals, will allow the students to follow through an example application from design to implementation, and appreciate the relevance of all the component parts of the module syllabus. Some example subjects would be: a web cam pan and tilt control system based on the Infineon 80C164 for motor control, developing a microprocessor based home security system with swipe card access and various input and output controls and sensors, designing, developing and downloading a small protocol for data communications for use between standalone single board computers.

Indicative Reading List

The following list is offered to provide validation panels/accrediting bodies with an indication of the type and level of information students may be expected to consult. As such, its currency may wane during the life span of the module specification. However, CURRENT advice on readings will be available via other more frequently updated mechanisms.

Selected Papers from:Proceedings of the Symposia on Operating System Principles; IEEE International Conference on Distributed Computing Systems; ACM SIGOPS Operating Systems Review.

Yaghmour (2002). *Building Embedded Linux Systems*, O'Reilly

Massa (2002). *Embedded Software Development with Ecos*, Prentice Hall

Banahan M. (1992). *The C Book*, Addison-Wesley

Clements A. (1997). *Microprocessor Systems Design*, PCW

(1997). *Microprocessor System Manuals,*

Crossware (1996). *68000 Developer's Guide,*

Motorola (2001). *MC68307 Manual*, Motorola

(2001). *Minos Developers Manual, CMS System,*

Assessment

Weighting between components A and B A: 50% B: 50%

ATTEMPT 1
First Assessment Opportunity

Element Description	**% of Component**	**% of Assessment**
Component A (Controlled Conditions)		
Examination	100%	50%
Component B		
Coursework 1.	50%	25%
Coursework 2.	50%	25%