



Title:	Embedded Systems Engineering	APPROVED
Long Title:	Embedded Systems Engineering	
Module Code:	COMP8049	Duration: 1 Semester
Credits:	5	
NFQ Level:	Advanced	
Field of Study:	Computer Science	
Valid From:	Semester 1 - 2019/20 (September 2019)	
Module Delivered in	2 programme(s)	
Module Coordinator:	Sean McSweeney	
Module Author:	Paul Davern	
Module Description:	This module gives an introduction to the engineering of software architectures for embedded and cyber physical systems. The focus of the module is on tools and design modelling for embedded and cyber-physical systems software development. The learner will apply these engineering principles to practical case studies of embedded and cyber-physical systems. The compilation and debugging focus is on the development of efficient high level language constructs for embedded systems.	
Learning Outcomes		
<i>On successful completion of this module the learner will be able to:</i>		
LO1	Evaluate an embedded system platform and runtime environment.	
LO2	Discuss the role of a compiler in the optimisation of an embedded system.	
LO3	Apply modeling techniques to embedded system design.	
LO4	Evaluate the techniques and algorithms used in embedded and cyber-physical system design.	
LO5	Develop efficient programming language constructs for an embedded system.	
Pre-requisite learning		
Module Recommendations		
<i>This is prior learning (or a practical skill) that is strongly recommended before enrolment in this module. You may enrol in this module if you have not acquired the recommended learning but you will have considerable difficulty in passing (i.e. achieving the learning outcomes of) the module. While the prior learning is expressed as named CIT module(s) it also allows for learning (in another module or modules) which is equivalent to the learning specified in the named module(s).</i>		
12689	SOFT7019	C Programming
Incompatible Modules		
<i>These are modules which have learning outcomes that are too similar to the learning outcomes of this module. You may not earn additional credit for the same learning and therefore you may not enrol in this module if you have successfully completed any modules in the incompatible list.</i>		
No incompatible modules listed		
Co-requisite Modules		
No Co-requisite modules listed		
Requirements		
<i>This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed. You may not enrol on this module if you have not acquired the learning specified in this section.</i>		
No requirements listed		



Module Content & Assessment

Indicative Content

Embedded System Runtime Environment

Platforms; Bare metal runtime environment; Bootloaders; Using Virtual Platforms; Simulation e.g. QEMU; Debugging embedded systems.

Modelling Embedded Systems

Embedded systems modelling languages e.g. SystemC; Modelling computational machines; Communication models; Automata; Finite State Machines.

Compiling a language

Phases of a Compiler; Symbol tables; Executable file layout; Linkers; Stack and heap layout; Analysis of high level language and generated machine code; The Abstract Syntax Tree. Case study e.g. how the Clang AST models a C program.

Cyber Physical System Design

Sensors and Actuators; Tasks; Concurrency; Communication; Input/Output; Case study e.g. quadcopter firmware design.

Software Synthesis

Implementing a design of an embedded architecture in C; Target machine model; Portability; Optimisation; Firmware architecture e.g. the STM32Cube MCU Package. Case study e.g. the Cleanflight flight controller software architecture.

Assessment Breakdown

%

Course Work

100.00%

Course Work

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Practical/Skills Evaluation	Labs based on lectures, which will re-enforce the students learning. For example: a) the development of a bare metal runtime for an embedded application board. b) modification of the firmware for a quadcopter flight controller.	1,2,3,4,5	30.0	Every Week
Essay	Essay on a topic related to embedded system design.	1,2,3,4,5	20.0	Week 9
Project	Development of a embedded systems design model with software synthesis for an embedded platform e.g. design and development of a firmware for a quadcopter flight controller.	1,2,3,4,5	50.0	Week 13

No End of Module Formal Examination

Reassessment Requirement

Coursework Only

This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.

The institute reserves the right to alter the nature and timings of assessment

Module Workload

Workload: Full Time				
<i>Workload Type</i>	<i>Workload Description</i>	<i>Hours</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	Lectures delivering the theory underpinning the learning outcomes.	2.0	Every Week	2.00
Lab	Lab supporting the learning outcomes and content delivered in lectures.	2.0	Every Week	2.00
Independent Learning	Independent study.	3.0	Every Week	3.00
Total Hours				7.00
Total Weekly Learner Workload				7.00
Total Weekly Contact Hours				4.00

Workload: Part Time				
<i>Workload Type</i>	<i>Workload Description</i>	<i>Hours</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	Lectures delivering the theory underpinning the learning outcomes.	2.0	Every Week	2.00
Lab	Lab supporting the learning outcomes and content delivered in lectures.	2.0	Every Week	2.00
Independent Learning	Independent Study.	3.0	Every Week	3.00
Total Hours				7.00
Total Weekly Learner Workload				7.00
Total Weekly Contact Hours				4.00

Module Resources

Recommended Book Resources

- Peter Marwedel 2018, *Embedded System Design*, 3rd Edition Ed., Springer [ISBN: 9783319560434]
- Aho, Lam, Sethi, and Ullman 2006, *Compilers: Principles, Techniques, and Tools*, 2nd Ed., Addison Wesley [ISBN: 9780321486813]
- Edward A. Lee and Sanjit A. Seshia 2017, *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, 2nd Edition Ed., MIT Press [ISBN: 9780262533812]

Supplementary Book Resources

- Jonathan Valvano 2011, *Embedded Systems: Real-Time Interfacing to ARM Cortex M Microcontrollers*, 2nd Edition Ed., CreateSpace Independent Publishing Platform [ISBN: 9781463590154]
- Joseph Yiu 2010, *The Definitive Guide to the ARM Cortex-M3 and Cortex-M4 Processors*, 3rd Edition Ed., Newnes [ISBN: 9780124080829]

Recommended Article/Paper Resources

- Dömer, Rainer, et al. 2008, *System-on-chip environment: a SpecC-based framework for heterogeneous MPSoC design*, EURASIP Journal on Embedded Systems, vol. 5
- Andreas Gerstlauer, Shuqing Zhao, Daniel D. Gajski and Arkady M. Horak 1999, *Design of a GSM Vocoder using SpecC Methodology*, UC Irvine, Technical Report ICS-TR-99-11

Other Resources

- Website: *Arm reference manuals and documentation*
<http://infocenter.arm.com>
- Website: *Open Virtual Platforms*
<http://www.ovpworld.org>
- Website: *Workshop on Software and Compilers for Embedded Systems (SCOPES)*
<http://www.scopesconf.org>
- Website: *modeling of heterogeneous systems*
<http://ptolemy.eecs.berkeley.edu/ptolemy II/>
- Website: *LLVM toolkit*
<http://llvm.org/>
- Website: *SystemC library for virtual prototyping of multi-processors system on chip*
<http://www.soclib.fr/trac/dev>
- Website: *Modelica a system for modeling complex physical systems*
<https://www.modelica.org/>

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KSDEV_8	<u>Bachelor of Science (Honours) in Software Development</u>	7	Mandatory
CR_KDNET_8	<u>Bachelor of Science (Honours) in Computer Systems</u>	7	Mandatory