1. **Title of the module**

EENG8960 (EL896) - Computer and Microcontroller Architectures

1. **School or partner institution which will be responsible for management of the module**

Engineering and Digital Arts

1. **The level of the module (Level 4, Level 5, Level 6 or Level 7)**

Level 7

1. **The number of credits and the ECTS value which the module represents**

15 credits (7.5 ECTS)

1. **Which term(s) the module is to be taught in (or other teaching pattern)**

Autumn

1. **Prerequisite and co-requisite modules**

None

1. **The programmes of study to which the module contributes**

MSc/PDip in Advanced Digital Systems Engineering

MSc/PDip in Advanced Electronic Systems Engineering

MSc/PDip in Advanced Digital Systems Engineering (Integrated Circuit Design)

MSc/PDip in Advanced Digital Systems Engineering (Communications)

MSc/PDip in Advanced Communications Engineering (Wireless Systems and Networks) (option)

MSc/PDip in Advanced Communications Engineering (RF Technology and Telecommunications) (option)

1. **The intended subject specific learning outcomes.  
   On successfully completing the module students will be able to:**

1) Systematically and comprehensively understand of fundamental computer architectures and the basic building blocks (i.e. ALU, CPU, Registers, Program and Data memory) used to build them.

2) Appreciate and critically appraise alternative processor architectures such as RISC, CISC, VLIW, SIMD, MIMD and DSP. Application Specific Signal Processing (ASSP) and Multi-Core Processors.

3) Comprehensively understand the structure of a typical microcontroller and associated peripherals. An ability to make an informed decision about the choice of microcontroller for a particular application(s).

4) Compile and download code onto a microcontroller using commercial Integrated Development Environments.

5) Systematically and comprehensively understand Microcontroller Peripherals: ADCs and DACs, Timers and Input Capture. Communication using the IIC, SPI, UART, Displays. Interrupts and Interrupt Service Routines.

6) Program microcontrollers using C and C++ Programming.

7) Critically appraise software development and Software testing techniques.

1. **The intended generic learning outcomes.  
   On successfully completing the module students will be able to:**

1) Show an ability to deal with complex issues systematically and creatively and make judgements in the absence of complete data, and show that they are capable of self-direction and problem solving.

2) Use and understand a range of modern CAD tools and general ICT.

3) Demonstrate the ability to communicate complex ideas and concepts to specialist and non-specialist audiences.

4) Show that they are capable of learning independently, use critical thinking and analysis and demonstrate autonomy in time and resource management.

1. **A synopsis of the curriculum**

This module focuses on the basic principles of modern computer architecture and how they are mapped onto modern (32-bit) microcontrollers. The course uses the ARM processor core as an exemplar of a modern processor architecture that is now ubiquitous in embedded systems. The course will cover classic topics in architecture (CPU and ALU structure, Instruction sets, memory and memory) and performance metrics for evaluating the relative performance of different architectures such as RISC vs CISC and also VLIW, SIMD, MIMD, ASSP and DSP devices.

The NXP 1786 (mbed) microcontroller is used as an example microcontroller development platform and industry standard IDE's from Keil/IAR are used to program, test and debug them. The course includes a comprehensive presentation of typical microcontroller peripherals: ADCs and DACs, Timers and Input Capture, communication using IIC, SPI, UART. Displays. Interrupts and Interrupt Service Routines (ISRs).

The course also provides an introduction to the C and C++ programming languages and their use with microcontroller based systems. This material will include: Variables, data-types and arithmetic expressions. Strings, Loops, Arrays. Functions, Structures, Pointers, bit operators. The pre-processor. I/O operations in C. Debugging Programs. Object-Oriented Programming. The Standard C Library.

Issues such as software testing and testing strategies are discussed. Compiling and downloading code onto the mbed using commercial Integrated Development Environments such as Keil® and IAR®. GNU based toolchains for Microcontroller development.

1. **Reading list (Indicative list, current at time of publication. Reading lists will be published annually)**

* Martin, T. (2013) The Designers Guide to the Cortex-M Processor Family. Elsevier (Newnes) ISBN: 978-0-08-098296-0
* Yiu, J. (2010) The Definitive Guide to the ARM Cortex-M3. Elsevier (Newnes). ISBN: 978-1-85617-963-8
* Kochan, S. G. (2005) Programming in C: A Complete Introduction to the C Programming Language. Developers Library. ISBN: 978-0672326660
* Peckol, J. K. (2008) Embedded Systems: A Contemporary Design Tool. John Wiley ISBN: 978-0-471-72180-2
* Oshana, R and Kraeling, M. (2013) Software Engineering for Embedded Systems: Methods, Practical Techniques and Applications Elsevier (Newnes) ISBN: 978-0-12-415917-4
* Harris, David Money and Harris, Sarah L., 2012. Digital Design and Computer Architecture. San Francisco, Calif: Morgan Kaufmann. ISBN 9780123944245
* Hennessy, John L., Patterson, David A. and Asanović, Krste, Computer Architecture: A Quantitative Approach. Waltham, MA: Morgan Kaufmann/Elsevier. ISBN 012383872X
* Keller, Rainer, Kramer, David and Weiss, Jan-Philipp, 2010. Facing the Multicore-Challenge: Aspects of New Paradigms and Technologies in Parallel Computing. Berlin, Heidelberg: s.n. ISBN 9783642162329
* Nisan, Noam and Schocken, Shimon, 2008. The Elements of Computing Systems: Building a Modern Computer from First Principles. Cambridge, Mass: MIT. ISBN 9780262640688
* Patterson, David A. and Hennessy, John L., Computer Organization and Design: The Hardware/Software Interface. Amsterdam: Morgan Kaufmann. ISBN 9780123747501
* Stallings, William, Computer Organization and Architecture: Designing for Performance. Upper Saddle River, NJ: Prentice Hall. ISBN 0135064171

1. **Learning and teaching methods**

Total contact hours: 76

Private study hours: 74

Total study hours: 150

1. **Assessment methods**
   1. Main assessment methods

Workshop (10%)

Workshop (5%)

Workshop (5%)

Workshop (5%)

Workshop (5%)

Workshop (10%)

Examination (60%)

In order to obtain credit for this module on IET accredited programmes, the coursework mark and the exam mark must each be greater than or equal to 40% as well as achieving the pass mark for the module. This module will only be considered for compensation if the coursework mark and exam mark are each greater than 40%.

13.2 Reassessment methods

Like-for-like

1. **Map of module learning outcomes (sections 8 & 9) to learning and teaching methods (section12) and methods of assessment (section 13)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Module learning outcome** | *8.1* | *8.2* | *8.3* | *8.4* | *8.5* | *8.6* | *8.7* | *9.1* | *9.2* | *9.3* | *9.4* |  |
| **Learning/ teaching method** |  |  |  |  |  |  |  |  |  |  |  |  |
| Private Study |  |  |  |  |  |  |  |  |  |  |  |  |
| *Lectures* | **X** | **X** | **X** |  | **X** | **X** | **X** |  |  |  |  |  |
| *Workshops* |  |  | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  |
| *Laboratories* |  |  | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  |
| **Assessment method** |  |  |  |  |  |  |  |  |  |  |  |  |
| *Workshops* |  |  | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  |
| *Laboratories* |  |  | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  |
| *Examination* | **X** | **X** |  |  | **X** |  | **X** | **X** |  | **X** | **X** |  |

1. **Inclusive module design**

The School recognises and has embedded the expectations of current equality legislation, by ensuring that the module is as accessible as possible by design. Additional alternative arrangements for students with Inclusive Learning Plans (ILPs)/declared disabilities will be made on an individual basis, in consultation with the relevant policies and support services.

The inclusive practices in the guidance (see Annex B Appendix A) have been considered in order to support all students in the following areas:

a) Accessible resources and curriculum

b) Learning, teaching and assessment methods

1. **Campus(es) or centre(s) where module will be delivered**

Canterbury

1. **Internationalisation**

A number of internationally recognised hardware and software tools (e.g. Keil and MBED, C and C++) are used to present the material presented in this course.

**FACULTIES SUPPORT OFFICE USE ONLY**

**Revision record – all revisions must be recorded in the grid and full details of the change retained in the appropriate committee records.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date approved | Major/minor revision | Start date of the delivery of revised version | Section revised | Impacts PLOs (Q6&7 cover sheet) |
|  |  |  |  |  |
|  |  |  |  |  |

Revised FSO Jan 2018