1. **Title of the module**

 EENG5600 (EL560) Microcomputer Engineering

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

 Engineering

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

 Level 5

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**

 Pre-requisite modules:

 EL313 Introduction to Programming or CO324 Computer Systems

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

 BEng Electronic and Communications Engineering

 BEng Electronic and Communications Engineering with a Year in Industry

 BEng Computer Systems Engineering

 BEng Computer Systems Engineering with a Year in Industry

 MEng in Electronic and Communications Engineering

 MEng in Electronic and Communications Engineering with a Year in Industry

 MEng in Computer Systems Engineering

 MEng in Computer Systems Engineering with a Year in Industry

 BEng/MEng Electronic and Computer Engineering and its variants

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

 1. demonstrate a working knowledge of a programming language for microcontrollers;

 2. demonstrate experience of developing microcomputer applications;

 3. demonstrate a working knowledge of software engineering principles.

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

1. demonstrate key skills in problem solving

2. demonstrate key skills in information technology.

1. **A synopsis of the curriculum**

This is a highly practical module that starts with a typical programming language environment suitable for microcontrollers, looks at software engineering issues, methods for the programming of a 32-bit microcontroller and concludes with the input/output of data using polling and interrupts. There are supporting practical (PC and Lab) sessions.

1. **Reading list (Indicative list, current at time of publication. Reading lists will be published annually)**
* Toulson, Rob (2017) Fast and effective embedded systems design applying the ARM mbed, Oxford: Newnes
1. **Learning and teaching methods**

 Total contact hours: 40

 Private study hours: 110

 Total study hours: 150

1. **Assessment methods**
	1. Main assessment methods
* Programming assignment 1: 25%
* Programming assignment 2: 25%
* Intro to microcomputers assignment: 25%
* Microcomputer laboratory assignment: 25%

13.2 Reassessment methods

Reassessment instrument: 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 | 9.1 | 9.2 |
| **Learning/ teaching method** |  |  |  |  |  |
| Private Study | **x** | **x** | **x** | **x** |  |
| Lectures  | **x** | **x** | **x** | **x** | **x** |
| Workshops  | **x** |  | **x** | **x** | **x** |
| Lab experiments  | **x** | **x** | **x** | **x** | **x** |
|  |  |  |  |  |  |
| **Assessment method** |  |  |  |  |  |
| Programming assignments  | **x** |  | **x** | **x** | **x** |
| Intro to microcomputers assignment | **x** | **x** | **x** | **x** | **x** |
| Microcomputer laboratory assignment | **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**

The module introduces standard internationally recognised software design tools and programming language techniques in wide international use.

**DIVISIONAL 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 delivery of revised version | Section revised | Impacts PLOs (Q6&7 cover sheet) |
|  |  |  |  |  |
|  |  |  |  |  |