IMPERIAL

# Module Specification

## Basic details

| | | | |
|---|---|---|---|
| UID | | Cohorts covered (Earliest cohort) | |
| Long title | Embedded Systems | | |
| New code | ELEC60013 | New short title | |
| Brief description of module *(approx. 600 chars.)* | This module is a practical exploration of embedded systems, with a focus on the Internet of Things and real-time programming. A full-stack approach is taken to reflect the modern trend for embedded devices to form part of larger systems, and the coursework provides opportunities to work on hardware, low-level programming, real time operating systems, lightwight internet communication and back-end/front-end integration. | | |
| Available as a standalone module/ short course? | No | | |

## Statutory details

| | ECTS | CATS | Non-credit | |
|---|---|---|---|---|
| Credit value | 5 | 10 | N | HECOS codes |
| FHEQ level | 6 | | | |

## Allocation of study hours

| | Hours | |
|---|---|---|
| Lectures | 20 | |
| Group teaching | 0 | *Incl. seminars, tutorials, problem classes.* |
| Lab/ practical | 0 | |
| Other scheduled | 10 | *Incl. project supervision, fieldwork, external visits.* |
| Independent study | 95 | *Incl. wider reading/ practice, follow-up work, completion of asses* |
| Placement | 0 | *Incl. work-based learning and study that occurs overseas.* |
| Total hours | 125 | |
| ECTS ratio | 25.00 | |

## Project/placement activity

| | |
|---|---|
| Is placement activity allowed? | No |

## Module delivery

| | | | |
|---|---|---|---|
| Delivery mode | Taught/ Campus | Other | |
| Delivery term | Term 2 | Other | |

## Ownership

| | |
|---|---|
| Primary department | Department of Electrical and Electronic Engineering |
| Additional teaching departments | |
| Delivery campus | South Kensington |

## Collaborative delivery

| | |
|---|---|
| Collaborative delivery? | No |
| External institution | |
| External department | |
| External campus | |

## Associated staff

| Role | CID | Given name | Surname |
|---|---|---|---|
| Module leader | | Edward | Stott |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## Learning and teaching

### Module description

| | |
|---|---|
| Learning outcomes | Upon successful completion of this module, you will be able to:<br>1. Identify the particular requirements of computer systems that interact directly with the physical world<br>2. Analyse the constraints imposed by time-critical applications and prove that they are met<br>3. Set up low-level communication between software, hardware devices and the wider world<br>4. Use interrupts and threading to guarantee that time constraints are met<br>5. Write complex, low-level software that is reliable<br>6. Configure and use a variety of hardware platforms and development toolchains |
| Module content | Programming with embedded Python and C++<br>System interfacing using low-level communication;<br>Lightweight internet communication for embedded systems;<br>Real time constraint analysis and validation;<br>Interrupts and multi-threading with a real-time operating system. |
| Learning and Teaching Approach | This module is focussed on practice, with all assessment by coursework. Background theory is explored in interactive lectures (1 lecture per week) and supervised lab sessions (1 session per week) provide an introduction to the coursework assignments. All the lab and coursework activities are supported by a 'Lab-in-a-Box' approach, allowing students to work flexibly. Approximately 50% of the lectures are delivered through video, with the contact time used for Team-Based Learning (TBL) quizzes - an individual readiness test (multiple choice) followed by the opportunity to refine answers in a group. The TBL exercises have a small weighting to encourage participation. |

| | |
|---|---|
| Assessment Strategy | The module is primarily assessed by coursework (94%). The first coursework is a group project to invent an IoT project based on a selection of interesting sensor devices. The deliverable is a complete product concept, including back-end and user interface components, a marketing website and a promotional video.<br>The second coursework is to produce real-time software for a music synthesiser using a real-time operating system. The synthesiser has a number of functions that must be completed concurrently with strict timing requirements. There are many possibilities for groups to integrate their own innovative feature. Assessment is based on the functional code, plus documentation of the real-time specifications and performance of the code.<br>6% of the module mark is earned through participation in the TBL exercises. |
| Feedback | Formative verbal feedback is provided on demand during scheduled lab sessions. Summative feedback on coursework deliverables is given in written format, two weeks after submission. |
| Reading list | |

## Quality assurance

| | |
|---|---|
| Date of first approval | July 2021 |
| Date of last revision | June 2026 |
| Date of this approval | |
| Module leader | Dr Edward Stott |

## Office use only

| | |
|---|---|
| QA Lead | |
| Department staff | |
| Date of collection | |
| Date exported | |
| Date imported | |

| | |
|---|---|
| Notes/ comments | |

Template version

## Assessment details

| Grading method | Numerical | Pass mark | 40% |
|---|---|---|---|

## Assessments

| Assessment type | Assessment description | Weighting | Pass mark | Must pass? |
|---|---|---|---|---|
| Coursework | Internet of Things product concept | 47% | 40% | N |
| Coursework | Real-time software for music synthesiser | 47% | 40% | N |
| Coursework | TBL Participation | 6% | 40% | N |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | 100% | | |