University of the West of England
BRISTOL

ACADEMIC SERVICES

PROGRAMME SPECIFICATION

Part 1: Basic Data		
Awarding Institution	UWE	
Teaching Institution	UWE	
Delivery Location	UWE	
Faculty responsible for programme	Environment and Technology	
Department responsible for programme	Computer Science and Creative Technologies	
Modular Scheme Title	FET	
Professional Statutory or Regulatory Body Links	British Computer Society	
Highest Award Title	BEng(Hons) Computing for Embedded Systems	
Default Award Title		
Interim Award Titles	BEng Computing for Embedded Systems Dip HE Computing for Embedded Systems Cert HE Computing for Embedded Systems	
UWE Progression Route	n/a	
Mode(s) of Delivery	Full time, Sandwich, Part time	
Codes	**UCAS: G402**	**UCAS:**
	ISIS2: G502	**ISIS2:**
Relevant QAA Subject Benchmark Statements	Computing	
CAP Approval Date	16 November 2016; 31 Jan 2017 v2; 30 June v3	
Valid From	September 2017	
Valid until Date	September 2023	
Version	3	

Part 2: Educational Aims of the Programme

The BEng(Hons) Computing for Embedded Systems has the following general aims:

1. To prepare students for entry into the computing profession with the challenges of a profession where relevant technology has, and will continue to rapidly change for the foreseeable future.
2. To inculcate in students effective problem-solving and other valuable career skills.
4. To continue the development of personal study skills that will enable students to become enthusiastic, lifelong learners.

The BEng(Hons) Computing for Embedded Systems has the following specific aims:

1. To consolidate and extend the base of students' knowledge and expertise in the core areas of Computing Technology, Embedded Systems Development, and Systems Administration.
2. To strengthen and expand students' practical expertise in the design and implementation of

Part 2: Educational Aims of the Programme

programmable systems so as to become more effective and proficient as practitioners.

3. To gain a theoretical knowledge of a variety of computer-related technologies so as to adapt and move more readily between them during their future careers.
4. To obtain the knowledge and understanding to provide a sound basis for a successful career within the field of digital technology and associated service provision.
5. To widen students' skill-base, and so enable them to interact more effectively with colleagues, and clients.
6. To more fully understand the relevance of academic study to the world of commerce and industry.

Part 3: Learning Outcomes of the Programme

The award route provides opportunities for students to develop and demonstrate knowledge and understanding, qualities, skills and other attributes in the following areas:

Learning Outcomes	**Teaching, Learning and Assessment Strategies**
A Knowledge and Understanding	
A Knowledge and understanding of 1. The concepts, including mathematical concepts, underpinning microprocessor technology and related cross-development techniques 2. The interface between digital hardware and software systems to enable the development of embedded systems can be developed 3. Program design and implementation concepts, methods, and notations, 4. The architecture and main components of computers. 5. The concepts underpinning distributed systems and networks. 6. Professional, ethical, and social values 7. The commercial context of software development	**Teaching/learning methods and strategies:** The award is designed to introduce the main concepts and practicalities of Computer Systems, encompassing both hardware and software. This includes not simply selecting from the wide range of available components, but also the design and implementation of novel components for a particular application. The resulting complex technical problems, involving the interaction of VLSI hardware, operating systems and application code, demand a broad range of knowledge and understanding. Frequently high speed data communications will also be required to link several compute platforms. A practical, exploratory, lab-based approach is central to this course. Students are introduced at each level to topics that encourage them to develop an increasingly sophisticated understanding of the main concepts of Embedded Computer Systems. At level-1, small examples are presented as challenges for the students so that they are immediately exposed to realistic problems of relevance to the theoretical side of the course. Level-2 leads forward with more in-depth understanding of large, complex scenarios frequently found in modern commercial products. Level-3 offers more opportunity for analysis, design and implementation of larger systems, often working within small groups. The solutions may now be completely novel, not simply following accepted schemas. At level-1, the students assemble a knowledge and understanding of key topics, such as: the principles of serial communications, use of data structures to model a designated problem, the

Part 3: Learning Outcomes of the Programme	
	design and construction of modern electronic circuits, the use of high level languages to devise algorithms and solve problems, the appreciation of the fundamental operation of a CPU. This allows students to explore the general concepts, components and issues, positioning them in the computing environment. At level-2 the knowledge and understanding of systems integration continues with an expansion into broader and larger issues, such as, the interaction of operating systems with other system elements. By level-3, students have then built up a confident understanding of the technologies used within the hierarchy of provision from low-level hardware up to the highest application level. They are required to work in groups to consider complex, novel problems, and develop and demonstrate practical solutions. On all modules, at all levels, the learner is encouraged to undertake independent reading both to supplement and consolidate what is being taught/learnt and to broaden their individual knowledge of the subject.
	Assessment: Most of the knowledge and understanding outcomes are assessed by both examination and coursework. Much of the coursework is practical in nature. In addition, a variety of other assessment instruments are used to assess these outcomes, including the following: • Individual assignment project Group assignment project • Extended individual project • Tutor appraisal • In class test • Critical review • Essay Presentation • Portfolio of practical work
B Intellectual Skills	
B Intellectual Skills 1. Critical Thinking 2. Analysis 3. Synthesis of different types of information 4. Evaluation 5. Problem Solving 6. Appreciate problem contexts 7. Balance conflicting objectives 8. Construction of logical arguments 9. Discussion and debate about technical subjects with peers	**Teaching/learning methods and strategies:** At all levels students are required to bring together knowledge and skills acquired in several modules and hence determine new ways of working. As the student progresses, the need to synthesise (3) ever-greater volumes of information and approaches into a coherent approach is developed and consequently so is their critical thinking (1) as well as their ability to discuss and debate technical subjects with peers (9). At level-1 Analysis (2), Evaluation (4) and Problem Solving (5) are developed on small-scale problems in various programming activities in a number of modules. Here the focus is on understanding the problem and then solving it free from the environmental implications of real-world problems

	and without the need to examine alternatives and to balance conflicting goals. At level-2 there is a move away from small-scale problems to the design of larger scale systems. With this comes the need to evaluate (4) alternative methods and designs and to balance conflicting objectives (7). Level-3 sees the move to specific application examples and with it the need to appreciate problem contexts (6) is developed as well as striking the right balance when facing conflicting objectives (7). The skill of constructing a logical argument is inculcated in students in part when they develop cases to support decisions they have made to resolve conflicting objectives (8).
	Assessment: Programming of complex software requires demonstration of all of the intellectual skills. At level-1 the focus in programming coursework assessment, undertaken in a number of modules, is on the skills of Analysis (2), Evaluation (4) and Problem Solving (5). At level-2 and level-3 this branches out to include all the remaining skills. Many of the coursework assessments and exam papers include elements of programming work. Independent reading is used to enable students to focus on their own areas of interest and, in the process, subsequent reports, essays, and examinations allow skills 1 – 4 to be assessed. Design-work, even when not implemented in a programming language, requires demonstration of skills 1, 2, 5, 6, 7 and a number of coursework assessments and exam questions are devoted to such work. Many of the assignments and the individual project require students to express logical arguments, 8. Finally, all of the examinations assess skills 1-4 whist skills 5-7 are covered in many exams.
C Subject, Professional and Practical Skills	
C Subject, Professional and Practical Skills On completion of the programme students will be able to 1. Actively participate in the development and introduction of efficient and effective information systems. 2. Apply appropriate tools, techniques and frameworks in all areas of systems development and use. 3. Use of a range of programming languages , including hardware description languages. 4. Install and configure of software packages, particularly at the hardware./software interface. 5. Interpret design specifications, and implement them in a suitable programming language, 6. Specify the purchasing of software tools 7. Produce technical literature at a variety of levels,	Teaching/learning methods and strategies: Throughout the program, the skills listed are developed through a combination of the following devices: • Theoretical discussion • Practical laboratory-based work • Classroom-based tutorial exercises • Directed self-study Many of the skills listed are introduced at level-1 and then drawn into sharper focus at level-2, and deepened at level-3. The general teaching/learning method is to impart these practical/professional skills by a process of moving from an overview of what is required to a specific application of an individual skill at a higher level.
	Assessment:

Part 3: Learning Outcomes of the Programme	
8. Plan and deliver technical presentations for colleagues and clients.	The two main forms of assessment of the subject, professional and practical skills are as follows: • Extended individual project • Practical component of individual assignment projects In addition, other assessment instruments are used to assess some of the skills: • Examination • Portfolio of practical work • Group assignment project
D Transferable Skills and other attributes	
D Transferable Skills and other attributes 1. Communication skills: to communicate orally or in writing. 2. Self-management skills: to manage one's own time; to meet deadlines; to work with others. 3. IT skills in context: to use software tools in the context of application development. 4. Logical reasoning skills: To undertake analysis and interpretation of information in the context of the Computing discipline. 5. Problem formulation: To express problems in appropriate notations. 6. Progression to independent learning: To gain experience of, and to develop skills in, learning independently of structured class work. For example, to develop the ability to use on-line facilities to further self-study. 7. Comprehension of professional literature: to read and to use literature sources appropriate to the discipline to support learning activities.	**Teaching/learning methods and strategies:** 1. Communication skills are developed through a variety of methods and strategies including the following: • Students maintain laboratory log books • Students participate in electronic conferences, workshops, and groupwork sessions. • Students participate in discussion tutorials • Students present research topic findings in tutorials • Students participate in individual tutorials 2. Self-management skills are developed through a variety of methods and strategies including the following: • Students conduct self-managed practical work • Students participate in practically-oriented tutorial laboratory sessions • Students work through practical work-sheets in teams • Students practice design and programming Students participate in electronic conferencing tutorials • Students participate in electronic group-working tutorials 3. IT skills in context are developed through a variety of methods and strategies including the following: • Students conduct self-managed practical work • Students participate in experimental investigation tutorials • Students work through practical work-sheets in teams • Students make use of online teaching materials • Students use a range of system development tool, methods, and packages • Students are encouraged to practice programming to extend their skills • Students make sustained use of the internet 4. Logical reasoning skills are developed through a variety of methods and strategies including the following: • Students develop problem-solving

Part 3: Learning Outcomes of the Programme	
	programs • Case-Studies are used to explore design issues with students • Students practice design and programming • Students sketch designs of larger systems 5. Problem formulation skills are developed through a variety of methods and strategies including the following: • Students develop problem solving programs • Students practice design and programming 6. Progression to independent learning is developed through a variety of methods and strategies including the following: • Students are encouraged to practice programming to extend their skills • Students are encouraged to research relevant topics • Students are encouraged to use the library, the internet and other online facilities to discover information and broaden knowledge • Students are encouraged to articulate and reflect upon their own ideas and experiences • Students negotiate the content and structure of their individual projects with tutors 7. Comprehension of professional literature is developed through a variety of methods and strategies including the following: • Students are encouraged to access online material • Material is recommended to the students in module syllabi and by tutors • Students are required to research and refer to appropriate literature in assignments and the individual project
	Assessment: 1. Communication skills are assessed mainly by examination, but also by in-class tests, essays, presentations and poster presentations. 2. The other skills are assessed through a number of similar instruments including the following: • Individual and group projects • Practical assignments • Portfolio of exercises 3. In addition self-management skills are assessed by both peers and tutors through APT sessions and generally throughout the course.

Part 4: Programme Structure

This structure diagram demonstrates the student journey from Entry through to Graduation for a **full time student**, including: level and credit requirements, interim award requirements, module diet, including compulsory and optional modules

ENTRY

Year 1	UFCF93-30-1 Computer and Network Systems UFCFB3-30-1 Web Programming UFCFGL-30-1 Programming in C++ UFMFF8-30-1 Digital Principles	Optional Modules None	Interim Awards Cert HE in Computing for Embedded Systems 120 credits, of which not less than 100 are at Level 1 or above.
Year 2	UFCFW5-30-2 Mobile and Embedded Devices UFCFLC-30-2 Secure Computer Networks UFCFWK-15-2 Operating Systems UFCFVK-15-2 Internet of Things	Optional Modules UFCFV4-30-2 Data, Schema and Applications UFMFE8-30-2 Digital Design UFCFW4-30-2 Design and Analysis of Data Structures and Algorithms	Interim Awards Dip HE in Computing for Embedded Systems 240 credits, of which not less than 100 are at Level 2 or above and a further 120 are at Level 1 or above.

Year Out: *Students on the sandwich route complete a placement year. For students on placement, there is an opportunity to complete a professional experience or international experience module and be awarded 15 level 3 credits. The professional experience and international experience modules are shown in the option list for year 3 but is actually completed during the year out.*

	Year 3	Compulsory Modules UFCF6Y-30-3 Embedded Systems Development UFCFJ4-15-3 Building and Porting Embedded Operating Systems UFCFXK-30-3 Digital Systems Project UFCFFL-15-M Parallel Computing	Optional Modules – Students must take 15 credit from UFCFE6-15-3 Professional Experience (studied during placement year) OR UFCFWJ-15-3 International Experience (studied during placement year) OR UFCFVJ-15-3 Professional Development OR UFCFB5-15-3 Ethical and Professional Issues in Computing and Digital Media Students must also take 15 credits from UFCFX4-15-3 Designing and Developing Device Drivers OR UFMFV8-15-3 Group Design and Integration Project OR UFCF95-15-3 Entrepreneurial Skills OR UFCFT4-15-3 Cryptography	Interim Awards BEng Computing for Embedded Systems A student passing 300 credits can be awarded a non-honours degree 300 credits with at least 60 credits at level 3, plus a further 100 credits at level 2 or above and a further 120 credits at level 1 or above. Highest Award BEng(Hons) Computing for Embedded Systems 360 credits, of which at least 100 must be at Level 3 or above, at least a further 100 at Level 2 or above and a further 140 at Level 1 or above.

GRADUATION

Part 5: Entry Requirements

The University's Standard Entry Requirements apply.

Tariff points as appropriate for the year of entry - up to date requirements are available through the courses database.

Part 6: Assessment

A: Approved to University Regulations and Procedures

Part 7: Student Learning

Teaching, learning and assessment strategies to enable learning outcomes to be achieved and demonstrated

Part 7: Student Learning

At UWE, Bristol there is a policy for a minimum average requirement of 12 hours/week contact time over the course of the full undergraduate programme. This contact time encompasses a range of face:face activities as described below. In addition a range of other learning activities will be embedded within the programme which, together with the contact time, will enable learning outcomes to be achieved and demonstrated.

On the BEng(Hons) Computing for Embedded Systems teaching is a mix of scheduled, independent and placement learning. There is a significant emphasis on learning through practical exercises.

Scheduled learning includes lectures, seminars, tutorials, project supervision, demonstration, practical classes; external visits. Scheduled sessions may vary slightly depending on the module choices made.

Independent learning includes hours engaged with essential reading, case study preparation, assignment preparation and completion etc. These sessions constitute an average time per module as indicated in the module specifications. Scheduled sessions may vary slightly depending on the module choices made.

Placement learning: Students are strongly encouraged to undertake a placement year. They are required to take the Professional Experience module whilst on placement. They may alternatively take a study year abroad, in which case they take the International Experience module

Description of Distinctive Features and Support

Class-based Activities The particular mode of delivery of a module is determined by its Module Leader, and typically involves a combination of lectures, practical sessions, individual and group activities and group project work. Many modules involve significant practical work and therefore a proportion of the student's contact time for that module, usually 50%, is spent in the computer labs.

Academic Support Academic advice and support is the responsibility of the staff delivering the module. Outside of normal timetabled hours, advice and guidance on matters relating to the material being taught and on its assessment can be obtained either by arranging an appointment with academic staff or during published "surgery" hours. Appointments are most commonly arranged by email.

In addition all students are allocated Academic Personal Tutor (APT) to whom they can turn for general academic advice related to their studies. From time to time students can expect their APT to invite them to meet to discuss their progress.

As a supplement to this formal academic support, all modules at level 1 (i.e. first year modules) include timetabled Peer-Assisted Learning (PAL) sessions. These classes are extra to the sessions timetabled with academics and provide new students with a significant additional resource, over and above the normal 12 hours contact time. PAL sessions are led by trained PAL leaders; second and final year students who are able to use their experience during the first year to help the newer students overcome barriers to success in their studies.

On-line Academic Support Extensive on-line support for this programme is provided through the University portal (myUWE). This provides access to the University's e-library, which allows students to read academic journals and study-skills material. Of particular interest to students of this programme is access to the ACM, IEEE and British Standards Online databases. The portal also gives entry to UWE's Virtual Learning Environment (Blackboard) which is used by academics to make available general information about the module delivery, handbooks, lecture notes and other materials. In addition, the portal publishes individual student timetables, marks and other aspects of the operation of the programme and University life.

Pastoral Support Pastoral care is provided through the University-wide Student Advisers, a team of staff who provide comprehensive, full-time student support service. Advisers are trained to provide advice on matters commonly of concern, including regulatory and other matters; the Adviser will, when necessary, direct the student to specialist professional services including the University's counselling service, careers, financial services etc..

Independent Study
All modules require students to carry out independent study, such as preparation for classes, research for projects and completion of assignments, and a full range of facilities are available at all sites to help students with these. The philosophy is accordingly to offer students both guided support and

Part 7: Student Learning

opportunities for independent study. Guided support is mainly in the form of timetabled sessions. Students are expected to attend all sessions on their timetable.

The habits and practice of independent study is then developed through the support offered in individual modules. Typically, module leaders will provide a plan for the module indicating the activities to be carried out and the forms of learning to be undertaken during the delivery of the module, with a view to encouraging students to plan ahead and to take responsibility for managing their time and resources.

Computing Facilities In 2012 the Faculty has undertaken a major new build f computing facilities in which it offers a specialised computing facility alongside the general University provisions. There are multiple computing laboratories of 20 plus seats running Windows, Linux and dual-boot systems required for this program. Computers within the specialist laboratories include the standard University build augmented by software resources and hardware equipment necessary for the delivery of the modules. For example, the specialist Forensic and Security laboratory runs virtual machine and industry-standard specialist software.

In addition, one of the most popular areas within the Faculty is the Open Access laboratory. This area is never timetabled and gives students the opportunity to access machines at all times during opening hours.

Part 8: Reference Points and Benchmarks

Description of ***how*** the following reference points and benchmarks have been used in the design of the programme:

The latest QAA benchmark statement for Computing was published in 2007 and has informed the design of the programme.. To the extent that the 2010 Engineering benchmark statement defines Engineering as concerned with “developing, providing and maintaining infrastructure, products, processes and services for society”, that too is applicable to this programme and has been used as a check on the both the structure of the programme and its mode of delivery.

The Computing benchmark statement declares that Computing is concerned with the “understanding, design and exploitation of computation and computer technology”. This programme has taken each of these three strands of endeavour and applied them to each of “computation” and “computer technology”. In terms of understanding computation, this has been taken to refer both to the mathematical underpinnings and to the notion of the execution of algorithms. Both of these areas are covered by the programme. The “design” of computation has is seen as being central to the programme in so far as it is largely concerned with the interface between hardware and software – the point at which “computation” is “designed” into real, useable computing machinery. The “exploitation” of computation, and computer technology, is realised within the programme by the inclusion of a number of modules that focus on programming and though the largely self-driven projects that students complete in their final year. Here they are encouraged to use the technologies they have encountered during their studies and apply them in a novel fashion.

“Computer technology” is also at the heart of the programme. Its understanding is amply covered by the inclusion of modules with an electronics flavour and by the study of the low-level computing functions. This is coupled very literally with “design” through the study of VHDL, computer and network architectures and operating systems. Consideration of the “exploitation” angle has led to the inclusion of mobile and embedded systems as well as more professionally-facing modules in the final year. Finally, the design of the programme has been assessed against the 3 key ideas that can be expected to characterise an honours degree in Computing. (see http://www.qaa.ac.uk/Publications/InformationAndGuidance/Documents/computing07.pdf , p. 4 section 2.9). We believe that the programme has precisely incorporated these ideas.

In terms of the Engineering benchmark statement, (http://www.qaa.ac.uk/Publications/InformationAndGuidance/Documents/Engineering10.pdf) p. 3 section 3 has been used as an additional metric in assessing the likely outcomes for graduates of this programme.

University strategies and policies

The development of this programme reflects well institutional policies and is fully consistent with the University’s commitment to ‘make a positive difference to our students, business and society’.

Part 8: Reference Points and Benchmarks

Employer interaction and feedback

The content of the proposed programme has been very much shaped by feedback from employers (both of placement students and of graduates) and by past graduates of BSc ((Hons) Computer Systems Integration and its predecessors, BSc (Hons) Computing for Real-Time Systems/BSC (Hons) Computer Systems Engineering The programme team has extensive links local employers and alumni. Feedback on the new structure has been gained through the Bristol Embedded Systems Consortium which continues to support the programme. Graduates of Computing for Embedded Systems are greatly sought after, particularly in the South West.

FOR OFFICE USE ONLY

First CAP Approval Date	16 November 2016			
Revision CAP Approval Date *Update this row each time a change goes to CAP*		Version	1	
	31 January 2017		2	Link to RIA (ID 3763)
	30 June 2017		3	Link to RIA (ID 4397)
Next Periodic Curriculum Review due date	2022			
Date of last Periodic Curriculum Review				