MODULE SPECIFICATION

Name of Module		**Embedded Systems**					
Parent School/Dept		**Computer Science**					
Programme(s) where module is offered		BSc Computer Science with Electrical Engineering; BSc Computer Science with Economics; BSc Computer Science with Business; BSc Computer Science with International Relations; BSc Computer Science with Political Science;					
Status (core, option, free choice)		**Core**		**Pre-Requisite Modules or Qualifications**		**CS110; CS160; CS280**	
FHEQ Level	**6**	**Unit Value**	**6 ECTS**	**Module Code**	CS480	**Module coordinator**	**Dr. Zanin Vejzovic**
Term taught		**Spring**		**Applicable From**		**2019**	

Educational Aims of the Module

Mobile phones, printers, smart devices, robots, drones, routers, cars, microwave ovens and refrigerators are some examples of devices we use which are controlled by embedded microprocessors (also called processors, controllers or embedded computers). Actually, over 90% of electronic devices use embedded microprocessors. Such processors and its associated resources (memory, input/output interfaces, etc.) are called an embedded processor system. The software development for such embedded processor systems different from the typical PC software development. This is due to the restrictions of the available processor performance, RAM, I/O, etc. This course will help the student to understand embedded systems, how to design suitable system as well as how to develop efficient and reliable software for the target application. The course will also address the requirement of embedded middleware.
The main goal of the course is to teach the programming concepts, PIC (peripheral interface controller) architecture and peripheral interfacing. The course focuses on how to write program and peripheral interfacing of PIC microcontroller and develop the applications.

Module Outline/Syllabus

1. Embedded system introduction
    a. Introduction to embedded system,
    b. Embedded system architecture,
    c. classifications of embedded systems,
    d. challenges and design issues in embedded systems,
    e. fundamentals of embedded processor and microcontrollers,
    f. CISC vs. RISC,
    g. fundamentals of Vonneuman/Harvard architectures,
    h. types of microcontrollers,
    i. selection of microcontrollers
2. PIC architecture
    a. Introduction to PIC microcontrollers,
    b. PIC architecture,
    c. comparison of PIC with other CISC and RISC based systems and microprocessors
3. I/O programming
    a. PIC I/O ports,
    b. I/O bit manipulation programming,
    c. timers/counters,
    d. programming to generate delay and wave form generation,
    e. I/O programming,
    f. LEDs, LCD and Keypad interfacing.

Student Engagement Hours			
Type	**Number per Term**	**Duration**	**Total Time**
Lectures	30	2 hours	60 hours
Laboratory sessions	15	2 hours	30 hours
		Total Guided/Independent Learning Hours	**60**
		Total Contact Hours	**90**
		Total Engagement Hours	**150**

Assessment Method Summary				
Type	**Number Required**	**Duration / Length**	**Weighting**	**Timing/Submission Deadline**
Final exam	1	180 minutes	50%	End of semester
Project (Group)	1	~2000 words	30%	Begin W 3 Deadline W 14
Assignments	2	~1000 words	20%	Throughout semester

<table>
<tr><th colspan="3">Module Outcomes</th></tr>
<tr><td rowspan="2">Intended Learning Outcomes:

At the end of the Course, students should be able to gain:
1. Hardware design skills: know embedded processor system hardware and how to design appropriate system for the target application.

2. Programming skills: program embedded processors to handle registers, memory access, control interfaces, and other functionalities within the boundary of the resources available.

3. System design skills: use PC based emulator tools to develop software for various embedded systems.

4. Real-time systems requirements and development skills: programming embedded systems</td><td>→</td><td>Teaching and Learning Strategy:

All outcomes will be achieved by the following
1. Teaching strategy: to teach concepts, principles, design methods and development techniques by using example application scenarios.
2. Learning strategy: to learn the course materials by extensive reading of the textbooks, handouts, actual tool demonstrations, discussion via tutorials, exercises and implementations of case studies.</td></tr>
<tr><td>→</td><td>Assessment Strategy

1. All 4 outcomes will be embedded in the exam questions (midterm and final) to assure theoretical and design methodology clearly understood.
2. Outcomes (2), (3) and (4) are embedded in the assignment work. The assignment is to design, develop and implement a controller for a real-time application such an intruder alarm controller</td></tr>
<tr><td rowspan="2">Transferable Skills

At the end of the Course, students should be able to:
1. Acquire and organise information from primary and secondary sources, using both library and internet resources.
2. Plan and deliver a structured, organised, critical design and report.
3. Present, including hypothesis and ideas.
4. Use of computers for writing assignments, analysing data and literature searches.</td><td>→</td><td>Teaching and Learning Strategy:

The emphasis of real-time actual embedded system used every-day and commercially available shall help the student accomplish all the 4 transferable skills.
Skill (2) is achieved by the practical set to implement an embedded system using the embedded emulator.</td></tr>
<tr><td>→</td><td>Assessment Strategy

The exam will be held after the completion of the taught course. It will assess skills (1), (3) and (4) via question tailored to address these issues.</td></tr>
<tr><td rowspan="2">Practical Skills

1. Communication skills: present technical solutions and approaches in both written and verbal formats
2. Ability to design and carry out a solution for the practical problems
3. Program embedded systems
4. Ability to perform electronics troubleshooting
5. Formal report writing
6. Ability to blend multiple sources of information</td><td>→</td><td>Teaching and Learning Strategy:

1. Laboratory sessions (TS: 1-6)</td></tr>
<tr><td>→</td><td>Assessment Strategy

1. Assignment (TS: 1-6)
2. Project (TS: 1-6)</td></tr>
</table>

Key Texts and/or other learning materials

Set Text

- Barr, M. and Massa A. "Programming Embedded Systems: With C and GNU Development Tools", 2nd Edition, O'Reilly Media, 2006.
- Popovici, K., Jerraya, A., Rousseau, F., Wolf, M. "Embedded Software Design and Programming of Multiprocessor System-on-Chip: Simulink and System C Case Studies", Springer, 2010

  - Arduino Project Book (Basic)
  - Exploring Arduino by Jeremy Blum
  - Arduino Workshop: A Hands-On Introduction with 65 Projects by John Boxall
  - The Official Raspberry Pi Guide
  - Raspberry Pi Projects Book (Russel Barnes et al)

Links to be used during course:

- Arduino Course on Programming Electronics Academy (https://programmingelectronics.com)
- Raspberry Pi and the Internet of Things (https://www.udemy.com/from-0-to-1-raspberry-pi/)
- Lot of courses and explanations (https://www.sparkfun.com/)
- Arduino simulator (https://library.io/)
- Server for IoT projects (https://thingspeak.com/)

Please note: This specification provides a concise summary of the main features of the module and the learning outcomes that a typical student might reasonably be expected to achieve and demonstrate if he/she takes full advantage of the learning opportunities that are provided. More detailed information on the learning outcomes, content and teaching, learning and assessment methods of each module and programme can be found in the departmental or programme handbook. The accuracy of the information contained in this document is reviewed annually by the University of Buckingham and may be checked by the Quality Assurance Agency.

Date of Production	Spring 2019
Date approved by School Learning and Teaching Committee	
Date approved by School Board of Study	
Date approved by University Learning and Teaching Committee	
Date of Annual Review	