Subject Description Form

Subject Code	COMP 4438
Subject Title	Embedded Software
Credit Value	3
Level	4
Pre-requisite/ Co-requisite/ Exclusion	Pre-requisite: COMP 3438
Objectives	The objectives of this subject are: • To introduce students the definitions, scope and common properties of embedded systems and embedded software, and enable them to understand the duties and scope of an embedded software designer. • To provide students the knowledge about both theoretical and practical aspects of embedded software design, teaching them the methods and techniques for designing and implementing embedded software. • To train students in developing skills for writing embedded software with the aid of embedded software development platforms.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: Professional/academic knowledge and skills (a) have a good understanding of the role of embedded systems and embedded software programming and the scope of duties and tasks of an embedded software programmer; (b) grasp the concepts and principles, and be familiar with the approaches and methods of developing embedded software; (c) apply the knowledge and techniques learnt to develop solutions to real-world problems; (d) organize and manage embedded software built for deployment and demonstration; Attributes for all-roundedness (e) analyze requirements and solve problems using systematic planning and development approaches;
Subject Synopsis/ Indicative Syllabus	1. Introduction to embedded systems and embedded software design. Definitions, scope and common properties of embedded systems; performance metrics and technique challenges; design methodologies and issues.

2. Organizations and architectures of embedded systems. Processors; memory; buses; I/O; interrupts; storage systems; power supply systems.

3. Embedded software architecture and design platform. System software and application software; cross-platform development platform.

4. Design and optimization. Embedded operating systems; real-time operating systems; application software development.

5. Embedded software engineering. Embedded software models, software testing and verification.

Teaching/Learning Methodology

Lectures

In lectures, concepts, methodologies, architectures, operating systems and design flow will be explained with illustrative examples.

Tutorials/Labs

Tutorials and lab sessions help students understand concepts and improve their skills on solving problems.

Assignments

Assignments help develop students’ programming skills and critical thinking.

Assessment Methods in Alignment with Intended Learning Outcomes

Specific assessment methods/ tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)				
		a	b	c	d	e
1. Assignments	35%	✓	✓	✓	✓	✓
2. Mid-term	20%	✓	✓	✓		✓
3. Examination	45%	✓	✓	✓		✓
Total	100 %					

All three items are appropriate to evaluate the intended learning outcomes. Assignments are used to evaluate writing skills, critical thinking, and problem solving. Mid-term test and final examination can further help evaluate the above outcomes.

Student study effort expected	Class Contact:	
	Lecture	26 hours
	Tutorial/Lab	13 hours
	Other student study effort:	
	Assignments and self-study	80 hours
	Total student study effort	119 hours
Reading list and references	*Textbook:* *D. E. Simon, An Embedded Software Primer, MA: Addison Wesley, 1999.* *Reference Books:* *1. K. Qian, D. D. Haring, and L. Cao, Embedded Software Development with C, Springer, 2009.* *2. C. S. Rodriguez, G. Fischer and S. Smolski, The Linux(R) Kernel Primer: A Top-Down Approach for x86 and PowerPC Architectures, Prentice Hall, 2005.* *3. R. Kamal, Embedded Systems: Architecture, Programming and Design, McGraw-Hill, 2003.* *4. Q. Li and C.Yao, Real-Time Concepts for Embedded Systems, CMP Books, 2003.* *5. J. W. S. Liu, Real-Time Systems, Prentice Hall, 2000.* *6. A. S. Berger, Embedded Systems Design: An Introduction to Processes, Tools and Techniques, Lawrence, KA: CMP Books, 2001.* *7. T. A. Pender, UML Weekend Crash Course, New York, NY: Wiley, 2002.* *8. M. Barr, Programming Embedded Systems in C and C ++, O'Reilly and Associates, 1999.*	

Last update in July 2014