

# The Hong Kong Polytechnic University

## Subject Description Form

<b>Subject Code</b>	COMP5586
<b>Subject Title</b>	Embedded Systems
<b>Credit Value</b>	3
<b>Level</b>	5
<b>Pre-requisite/ Co-requisite/ Exclusion</b>	C/C++ programming; basic computer architecture concepts.
<b>Objectives</b>	<p>The objectives of this subject are:</p> <ol style="list-style-type: none"><li>1. Equip students with advanced theoretical knowledge and practical skills required for designing, developing, and optimising embedded software systems in resource-constrained environments.</li><li>2. Provide hands-on experience with advanced embedded platforms for prototyping and deploying embedded systems.</li><li>3. Enable students to integrate user space tools/models/libraries, real-time operating systems (RTOS), and hardware-software co-design to build embedded systems.</li></ol>
<b>Intended Learning Outcomes</b>	<p>Upon completion of the subject, students will be able to:</p> <ol style="list-style-type: none"><li>a. Understand the unique characteristics, constraints, and applications of embedded systems, and apply advanced development methodologies to design optimised solutions.</li><li>b. Develop efficient and reliable software for embedded systems, while employing debugging and testing techniques to ensure performance, energy efficiency, and reliability.</li><li>c. Demonstrate mastery in using advanced embedded platforms for designing, prototyping, and deploying embedded systems.</li><li>d. Deploy user space tools/models/libraries in embedded systems, addressing resource constraints and ensuring real-world applicability.</li><li>e. Design innovative solutions to complex real-world problems in areas such as IoT, robotics, or AI, integrating hardware and software through project-based learning.</li></ol>
<b>Subject Synopsis/ Indicative Syllabus</b>	<p>The subject explores advanced concepts and practical techniques for designing, developing, and optimising embedded systems in resource-constrained environments. Students will gain expertise in hardware-software co-design, real-time operating systems (RTOS), and advanced programming techniques. Emphasis is placed on practical applications using advanced embedded platforms with a focus on integrating user space tools/models/libraries and solving real-world problems in areas such as IoT, robotics, or AI. The subject combines theoretical foundations with hands-on project work to prepare students for industry and research challenges in modern embedded systems.</p>

	<p>The following topics will be covered:</p> <ol style="list-style-type: none"> <li>1. Foundations of Embedded Systems <ul style="list-style-type: none"> <li>- Concepts, characteristics, constraints, and applications of embedded systems.</li> <li>- Hardware-software integration and co-design principles.</li> </ul> </li> <li>2. Real-Time Operating Systems (RTOS) <ul style="list-style-type: none"> <li>- Task scheduling, multitasking, and interrupt handling.</li> <li>- Timing constraints and real-world performance optimisation.</li> </ul> </li> <li>3. Programming and Debugging for Embedded Systems <ul style="list-style-type: none"> <li>- Embedded software development with programming languages like C and Assembly.</li> <li>- Debugging and testing techniques for performance and reliability.</li> </ul> </li> <li>4. Project-Based Applications <ul style="list-style-type: none"> <li>- Design and implementation of real-world systems to solve complex problems in areas such as IoT, robotics, or AI.</li> <li>- Integration of hardware and software components using advanced embedded platforms.</li> </ul> </li> </ol>
<p><b>Teaching/Learning Methodology</b></p>	<p>The teaching and learning activities for this subject are structured to ensure students achieve a deep understanding of both theoretical concepts and practical applications of embedded systems. Lectures will focus on delivering core concepts, theories, and methodologies, providing students with a solid foundation in embedded systems design and development.</p> <p>To reinforce the material covered in lectures, tutorial and lab sessions will provide opportunities for students to engage in hands-on problem-solving, including programming, testing, debugging, and performance modelling exercises. These sessions will help students apply theoretical knowledge to real-world scenarios and develop critical technical skills.</p> <p>Quizzes, tests, and/or exams will be conducted to assess students' understanding of the material, maintain their engagement with the course content, and identify areas requiring further clarification.</p> <p>Finally, a project work component will challenge students to design, implement, and optimise embedded solutions, with a focus on advanced topics such as real-time operating systems (RTOS), performance modelling, AI etc.. This comprehensive combination of activities will equip students with both the knowledge and practical skills necessary for success in embedded systems design and development.</p>

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. Quizzes/Tests	10%	✓	✓			
	2. Project	20%			✓	✓	✓
	3. Final Exam	70%	✓	✓	✓	✓	✓
	Total	100 %					
<p>Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:</p> <p>Quizzes and tests are designed to assess students' understanding of the fundamental concepts of embedded systems. They are particularly suitable for evaluating the ability to identify the unique characteristics and constraints of embedded systems (ILO a) and the theoretical knowledge of software development, debugging, and optimization techniques (ILO b). This ensures that students grasp the foundational knowledge required for more advanced tasks.</p> <p>The project focuses on the practical application of skills and creative problem-solving. It is an opportunity for students to work with advanced embedded platforms to demonstrate hands-on expertise (ILO c). It also allows them to deploy user space tools/models/libraries while addressing real-world constraints (ILO d) and integrate hardware and software to design innovative solutions for real-world problems in areas like IoT, robotics, and AI (ILO e). This method is highly appropriate for assessing students' ability to apply their knowledge in practical, real-world scenarios.</p> <p>The final exam provides a comprehensive evaluation of all the intended learning outcomes. It ensures that students not only have a deep theoretical understanding of embedded systems (ILO a, ILO b) but can also articulate their knowledge of embedded platforms (ILO c), tool/model/library deployment (ILO d), and innovative solution design (ILO e). The exam's weight reflects its role in assessing the breadth and depth of students' understanding across the subject.</p>							
Student Study Effort Expected	Class contact:						
	▪ Lecture						26 Hrs.
	▪ Tutorial/Lab						13 Hrs.
	Other student study effort:						
	▪ After-Class Reading						39 Hrs.
	▪ Exercise and Project						39 Hrs.
	Total student study effort						117 Hrs.

**Reading List and References**

1. Doboli, A., & Currie, E. (2010). Introduction to Mixed-Signal Embedded Design. Springer.
2. Crnkovic, I., & Larsson, M. (2002). Building Reliable Component-Based Software Systems. Boston, MA: Artech House.
3. Lavagno, L., Martin, G., & Selic, B. (Eds.). (2003). UML for Real: Design of Embedded Real-Time Systems. Boston, MA: Kluwer.
4. Ernst, R. (1998). "Codesign of Embedded Systems: Status and Trends." IEEE Design and Test of Computers, 14(3), 45–54.
5. Edwards, S. A. (2000). Languages for Digital Embedded Systems. Boston, MA: Kluwer. (Retained for foundational programming concepts but needs supplementation.)
6. Laplante, P. A., & Ovaska, S. J. (2011). Real-Time Systems Design and Analysis: Tools for the Practitioner (4th ed.). Wiley.
7. Wilmshurst, T. (2019). Designing Embedded Systems with PIC Microcontrollers: Principles and Applications (3rd ed.). Newnes.
8. Oshana, R., & Welch, G. F. (2016). DSP for Embedded and Real-Time Systems (2nd ed.). Elsevier.