EMBEDDED SYSTEMS

VII Semester: **ECE**

Course Code	Category	Hours / Week			Credits	Maximum Marks		
AEC016	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	1	-	4	30	70	100
Contact Classes: 45	**Tutorial Classes: 15**	**Practical Classes: Nil**				**Total Classes: 60**		

OBJECTIVES:

The course should enable the students to:

I. Imbibe knowledge about the basic functions, structure, concepts and applications of Embedded Systems.
II. Understand Real time operating system concepts.
III. Analyze different tools for development of embedded software.
IV. Be acquainted the architecture of advanced processors.

COURSE OUTCOMES (COs):

CO 1: Understand the basic concepts of embedded system and various applications and characteristics, formalisms for system design of embedded system design

CO 2: Discuss the concepts of C and develop the C programming examples with Keil IDE, and understand the concepts of interfacing modules using embedded C.

CO 3: Understand the fundamentals of RTOS and its programming and task communication, Task synchronization with its issues and techniques.

CO 4: Develop an examples using embedded software and understand the debugging techniques

CO 5: Discuss the concepts of advanced processors like ARM and SHARC and protocols of I2C and CAN bus.

COURSE LEARNING OUTCOMES (CLOs):

1. Understand basic concept of embedded systems.
2. Analyze the applications in various domains of embedded system.
3. Develop the embedded system and Design process and tools with examples.
4. Understand characteristics and quality attributes of embedded systems, formalisms for system design.
5. Understand the basic programming of c and its looping structure.
6. Analyze the embedded C programming in Keil IDE, and compiling and building the hardware.
7. Understand different concepts of display and keyboard interfacing using embedded C.
8. Understand different concepts of serial communication using embedded C and user interfacing.
9. Remember the basics of operating system and its commands.
10. Understand and analyze the RTOS concepts for firmware development.
11. Remember how to choose an RTOS, task scheduling, semaphores and queues, hard real-time scheduling considerations.
12. Understand the task communication, its programming and Task synchronization with its issues and techniques.
13. Develop host and target machines for linking to embedded software.
14. Develop debugging techniques for testing on host machine with examples.
15. Remember the advanced processors such as ARM and SHARC.
16. Understand the bus protocols such as I2C and CAN bus.
17. Design an application based on advanced technological changes.

Unit-I **EMBEDDED COMPUTING** **Classes: 08**

Definition of embedded system, embedded systems vs. general computing systems, history of embedded systems, complex systems and microprocessor, classification, major application areas, the embedded system design process, characteristics and quality attributes of embedded systems, formalisms for system design, design examples

Unit-II **INTRODUCTION TO EMBEDDED C AND APPLICATIONS** **Classes: 10**

C looping structures, register allocation, function calls, pointer aliasing, structure arrangement, bit fields, unaligned data and endianness, inline functions and inline assembly, portability issues; Embedded systems programming in C, binding and running embedded C program in Keil IDE, dissecting the program, building the hardware; Basic techniques for reading and writing from I/O port pins, switch bounce; Applications: Switch bounce, LED interfacing, interfacing with keyboards, displays, D/A and A/D conversions, multiple interrupts, serial data communication using embedded C interfacing.

Unit-III **RTOS FUNDAMENTALS AND PROGRAMMING** **Classes: 10**

Operating system basics, types of operating systems, tasks and task states, process and threads, multiprocessing and multitasking, how to choose an RTOS ,task scheduling, semaphores and queues, hard real-time scheduling considerations, saving memory and power.

Task communication: Shared memory, message passing, remote procedure call and sockets; Task synchronization: Task communication synchronization issues, task synchronization techniques, device drivers.

Unit-IV **EMBEDDED SOFTWARE DEVELOPMENT TOOLS** **Classes: 09**

Host and target machines, linker/locators for embedded software, getting embedded software into the target system; Debugging techniques: Testing on host machine, using laboratory tools, an example system.

Unit-V **INTRODUCTION TO ADVANCED PROCESSORS** **Classes: 08**

Introduction to advanced architectures: ARM and SHARC, processor and memory organization and instruction level parallelism; Networked embedded systems: Bus protocols, I2C bus and CAN bus; Internet-EnAnalyzed systems, design example-Elevator controller.

Text Books:

1. Shibu K.V, “Introduction to Embedded Systems”, Tata McGraw Hill Education Private Limited, 2nd Edition, 2009.
2. Raj Kamal, “Embedded Systems: Architecture, Programming and Design”, Tata McGraw-Hill Education, 2nd Edition, 2011.
3. Andrew Sloss, Dominic Symes, Wright, “ARM System Developer's Guide Designing and Optimizing System Software”, 1st Edition, 2004.

Reference Books:

1. Wayne Wolf, “Computers as Components, Principles of Embedded Computing Systems Design”, Elsevier, 2nd Edition, 2009.
2. Dr. K. V. K. K. Prasad, “Embedded / Real-Time Systems: Concepts, Design & Programming”, dramatic publishers, 1st Edition, 2003.
3. Frank Vahid, Tony Givargis, “Embedded System Design”, John Wiley & Sons, 3 rd Edition, 2006.
4. Lyla B Das, “Embedded Systems”, Pearson Education, 1st Edition, 2012.
5. David E. Simon, “An Embedded Software Primer”, Addison-Wesley, 1st Edition, 1999.
6. Michael J. Pont, “Embedded C”, Pearson Education, 2nd Edition, 2008.

Web References:

1. http://www.igniteengineers.com
2. http://www.ocw.nthu.edu.tw
3. http://www.uotechnology.edu.iq
4. http://www.nptel.com

E-Text Books:

1. https://www.jntubook.com/embedded systems-textbook
2. http://tradownload.com/results/neamen-embedded-systems-.html
3. http://www.everythingvtu.wordpress.com