INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

ELECTRONICS AND COMMUNICATION ENGINEERING

COURSE DESCRIPTOR

Course Title	**EMBEDDED SYSTEMS**				
Course Code	AEC016				
Program	B.Tech				
Semester	SEVEN				
Course Type	Core				
Regulation	IARE - R16				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	-	3	3	2
Chief Coordinator	Mr. S Lakshmanachari, Assistant Professor.				

I. COURSE OVERVIEW:

Embedded systems course is continuous of the microprocessor and microcontrollers, is intended to designing, implementing and testing of embedded applications. The topics covered are definition of embedded systems, history, classification, and major applications. Introduction to embedded C and interfacing modules, embedded firmware design and development, RTOS, task scheduling, threads, multitasking, task communication, task synchronization. Finally discussing about the introduction of advanced architectures like ARM and SHARC processors.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites
B.Tech	ACS007	IV	Operating Systems
B.Tech	AEC013	VI	Microprocessors and Microcontrollers

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Embedded Systems	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✔	PPT	✔	Chalk & Talk	✔	Assignments	✗	MOOCs
✔	Open Ended Experiments	✔	Seminars	✗	Mini Project	✔	Videos
✔	Others:						

V. EVALUATION METHODOLOGY:

The course will be evaluated for a total of 100 marks, with 30 marks for Continuous Internal Assessment (CIA) and 70 marks for SEE. Out of 30 marks allotted for CIA during the semester, marks are awarded by taking average of two CIA examinations or the marks scored in the make-up examination.

Semester End Examination (SEE): The SEE is conducted for 70 marks of 3 hours duration. The syllabus for the theory courses is divided into five units and each unit carries equal weightage in terms of marks distribution. The question paper pattern is as follows: Two full questions with "either‟ or ‟choice" are drawn from each unit of the syllabus. Each question carries 14 marks. **There could be a maximum of two sub divisions in a question.**

The expected percentage of cognitive level of the questions is broadly based on the criteria given in Table: 1.

Table 1: The expected percentage of cognitive level of questions in SEE.

Percentage of Cognitive Level	Blooms Taxonomy Level
10 %	Remember
50 %	Understand
25 %	Apply
15 %	Analyze
0 %	Evaluate
0 %	Create

Continuous Internal Assessment (CIA):

CIA is conducted for a total of 30 marks (Table 2), with 20 marks for Continuous Internal Examination (CIE), 05 marks for Quiz and 05 marks for Alternative Assessment Tool (Table 3).

Table 2: Assessment pattern for CIA

Component	Theory			Total Marks
Type of Assessment	CIE Exam	Quiz	AAT	
CIA Marks	20	05	05	30

Continuous Internal Examination (CIE):

Two CIE exams shall be conducted at the end of the 8th and 16th week of the semester respectively. The CIE exam is conducted for 20 marks of 2 hours duration consisting of five descriptive type questions out of which four questions have to be answered where, each question carries 5 marks. Marks are awarded by taking average of marks scored in two CIE exams

Quiz –Online Examination:

Two Quiz exams shall be online examination consisting of 25 multiple choice questions and are to be answered by choosing the correct answer from a given set of choices (commonly four). Such a question paper shall be useful in testing of knowledge, skills, application, analysis, evaluation and understanding of the students. Marks shall be awarded considering the average of two quiz examinations for every course.

Alternative Assessment Tool (AAT):

This AAT enables faculty to design own assessment patterns during the CIA. The AAT converts the classroom into an effective learning center. The AAT may include tutorial hours/classes, seminars, assignments, term paper, open ended experiments, METE (Modeling and Experimental Tools in Engineering), five minutes video, MOOCs etc. The AAT chosen for this course is given in table 3**.**

Table 3: Assessment pattern for AAT

5 Minutes Video	**Assignment**	**Tech-talk**	**Seminar**	**Open Ended Experiment**
20%	30%	30%	10%	10%

VI. COURSE OBJECTIVES:

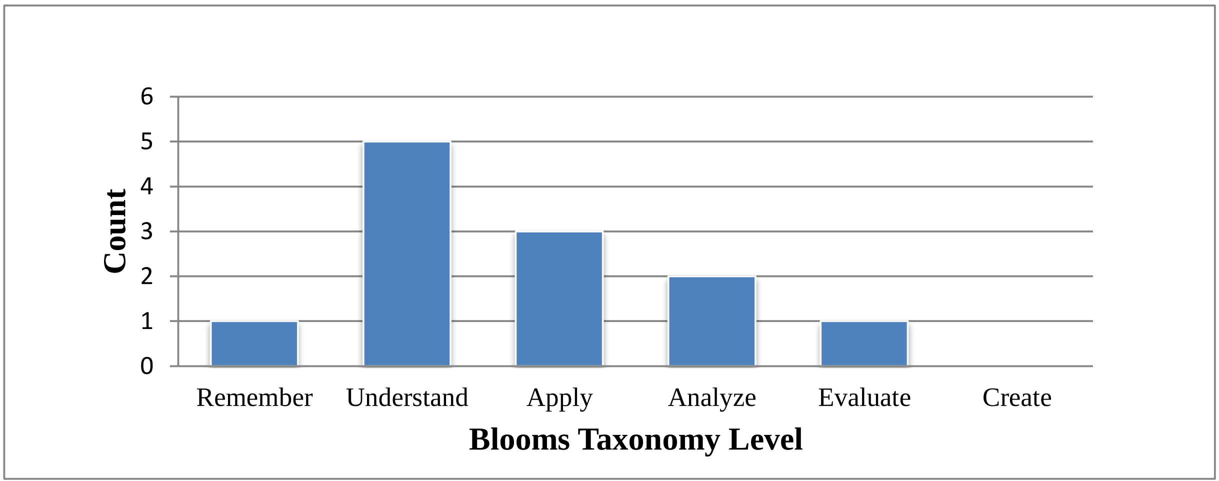
The students will try to learn:	
I	The fundamental concepts of embedded computing, embedded C, RTOS and embedded software development tools for implementing of real time embedded systems.
II	The embedded C is required to develop the software for different applications of the embedded systems.
III	The basics of various development tools are necessary to develop an embedded software.
IV	The architecture and memory organization of advanced general purpose microprocessors and digital signal processors like ARM and SHARC.

VII. COURSE OUTCOMES:

After successful completion of the course, students will be able to:		
	Course Outcomes	**Knowledge Level (Bloom's Taxonomy)**
CO 1	**Summarize** the applications of embedded systems in various domains.	Understand
CO 2	**Analyze** the embedded system design process, characteristics and quality attributes of an embedded system.	Analyze

CO 3	**Apply** the looping structure concept to the programming of embedded C.	Apply
CO 4	**Analyze** the concepts of interfacing modules using embedded C programming.	Analyze
CO 5	**Evaluate** the basic techniques used in interfacing in terms of reading and writing data from I/O port pins.	Evaluate
CO 6	**Develop** the embedded software using the basics and fundamentals of RTOS.	Apply
CO 7	**Demonstrate** the multiprocessing and multitasking in real time operating system to estimate the performance of embedded system.	Understand
CO 8	**Describe** the process of task communication using shared memory and message passing.	Understand
CO 9	**Illustrate** the implementation of real-time operating system using task communication and task synchronization.	Understand
CO 10	**List** the embedded software development tools for getting embedded software into the target system.	Remember
CO 11	**Describe** the concepts of advanced processors in terms of ARM and SHARC processors.	Understand
CO 12	**Explain** the memory organization and instruction level parallelism in advanced processors.	Understand

COURSE KNOWLEDGE COMPETENCY LEVELS



VIII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Strength	Proficiency Assessed by
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	CIE/Quiz/AAT
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	10	CIE/Quiz/AAT
PO 5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools	1	Assignments/ Discussion

Program Outcomes		Strength	Proficiency Assessed by
	including prediction and modeling to complex engineering activities with an understanding of the limitations.		

3 = High; 2 = Medium; 1 = Low

IX. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Strength	Proficiency assessed by
PSO 1	Formulate and Evaluate the applications in the field of Intelligent Embedded and Semiconductor technologies.	1	SEE
PSO 2	Focus on the practical experience of ASIC prototype designs, Virtual Instrumentation and SOC designs.	2	Research papers/ Group discussion/ Short term courses
PSO 3	Build the Embedded hardware design and software programming skills for entry level job positions to meet the requirements of employers.	3	Research papers / Industry exposure

3 = High; 2 = Medium; 1 = Low

X. MAPPING OF EACH CO WITH PO(s), PSO(s):

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO 1	√	√	-	-	-	-	-	-	-	-	-	-	√	-	-
CO 2	√	-	-	-	-	-	-	-	-	-	-	-	-	-	√
CO 3	√	-	-	-	√	-	-	-	-	-	-	-	-	-	-
CO 4	√	√	-	-	-	-	-	-	-	-	-	-	√	-	-
CO 5	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 6	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 7	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 8	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 9	-	√	-	-	-	-	-	-	-	-	-	-	-	√	-
CO 10	√	-	-	-	√	-	-	-	-	-	-	-	-	-	-
CO 11	√	-	-	-	-	-	-	-	-	-	-	-	√	-	-
CO 12	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-

XI. JUSTIFICATIONS FOR CO-PO MAPPING:

Course Outcomes	POs / PSOs	Justification for mapping (Students will be able to)	No. of key competencies
CO 1	**PO 1**	Summarize the applications (**knowledge)** of embedded systems in various domains by applying the principles of **science and engineering fundamentals**.	2
	PO 2	Understand the classification of embedded systems **for complex engineering problems.**	1
	PSO 1	Understand and evaluate the applications in the field of **intelligent embedded systems.**	1
CO 2	**PO 1**	Analyze the embedded system design process through various steps by applying the principles of **mathematics, science to the solutions of complex engineering problems**.	2
	PSO 3	Understand the embedded system design process by using the **embedded software skills**.	1
CO 3	**PO 1**	Apply the looping structures to the programming of embedded C by **analyzing complex engineering problems**.	1
	PO 5	Create the embedded C program and **apply appropriate techniques** to simulate using the tool Keil IDE software.	1
CO 4	**PO 1**	Analyze (Understand) the concepts of interfacing modules through the embedded C programming by applying **engineering fundamentals.**	1
	PO 2	**Identify** the various modules to interface with the microcontroller and **analyze** its interfacing program using embedded C.	2
	PSO 1	Understand how the different interfacing modules are interfacing using the embedded C programming by **evaluating** the applications in the field of **intelligent embedded technologies**.	2
CO 5	**PO 1**	Evaluate the basic techniques used in interfacing in terms of reading and writing data from I/O port pins by applying the principles of **science for engineering problems.**	1
	PO 2	**Identify** the basic technique used in interfacing for **analyzing the complex engineering problems** in reading and writing data from I/O port pins.	2
CO 6	**PO 1**	Develop the embedded software using the basics and fundamentals of real time operating systems (RTOS) by applying **engineering fundamentals.**	1
CO 7	**PO 1**	Demonstrate the multiprocessing and multitasking in real time operating system to estimate the performance of embedded system by using **science and engineering fundamentals**.	1
	PO 2	**Identify and analyze** the differences between the multiprocessing and multitasking in real time operating system.	2
CO 8	**PO 1**	Describe (Knowledge) the process of task communication using shared memory and message passing by applying the **science and engineering fundamentals**.	2
CO 9	**PO 2**	**Analyze** the implementation of real-time operating system using task communication and task synchronization.	1
	PSO2	Illustrate the implementation of real-time operating system using task communication and task synchronization by **focusing on the practical experience of ASIC prototype designs**.	1

Course Outcomes	POs / PSOs	Justification for mapping (Students will be able to)	No. of key competencies
CO 10	PO 1	List the embedded software development tools for getting embedded software into the target system by applying **fundamentals for complex engineering problems**.	1
	PO 5	**Apply the appropriate techniques** to implant the embedded software development tools for getting embedded software into the target system.	1
CO 11	PO 1	Describe the concepts of advanced processors in terms of ARM and SHARC processors by applying the **engineering fundamentals.**	1
	PSO 1	Understand the architectures of the advanced processors ARM and SHARC in the field of **intelligent embedded systems.**	1
CO 12	PO 1	Understand the memory organization and instruction level parallelism in advanced processors by applying **engineering fundamentals.**	1

XII. TOTAL COUNT OF KEY COMPETENCIES FOR CO – (PO, PSO) MAPPING

Course Outcomes	Program Outcomes / No. of Key Competencies Matched												PSOs/ Number of key competencies		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
	3	**10**	**10**	**11**	**1**	**5**	**3**	**3**	**12**	**5**	**12**	**8**	**1**	**3**	**2**
CO 1	2	1	-	-	-	-	-	-	-	-	-	-	1	-	-
CO 2	2	-	-	-	-	-	-	-	-	-	-	-	-	-	1
CO 3	1	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO 4	1	2	-	-	-	-	-	-	-	-	-	-	1	-	-
CO 5	1	2	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 6	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 7	1	2	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 8	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 9	-	1	-	-	-	-	-	-	-	-	-	-	-	1	-
CO 10	1	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO 11	1	-	-	-	-	-	-	-	-	-	-	-	1	-	-
CO 12	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

XIII. PERCENTAGE OF KEY COMPETENCIES FOR CO-(PO/PSO):

Course Outcomes	Program Outcomes / No. of key competencies												PSOs/ No. of key competencies		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
	3	10	10	11	1	5	3	3	12	5	12	12	1	3	2
CO 1	66.7	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
CO 2	66.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0
CO 3	33.3	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	00
CO 4	33.3	20.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
CO 5	33.3	20.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 6	33.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 7	33.3	20.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 8	66.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 9	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	33.3	0.0
CO 10	33.3	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 11	33.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
CO 12	33.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

XIV. COURSE ARTICULATION MATRIX (CO-PO/PSO MAPPING)

COs and POs and COs and PSOs on the scale of 0 to 3, **0** being **no correlation**, **1** being the **low correlation**, **2** being **medium correlation** and **3** being **high correlation**.

0 – **0**≤ C≤ 5%–No correlation

1 – 5 <C≤ 40%– Low/ Slight

2 − 40 % <C< 60% –Moderate

3 – 60% ≤ C< 100% – Substantial /High

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO 1	3	1	-	-	-	-	-	-	-	-	-	-	3	-	-
CO 2	3	-	-	-	-	-	-	-	-	-	-	-	-	-	2
CO 3	1	-	-	-	3	-	-	-	-	-	-	-	-	-	-
CO 4	1	1	-	-	-	-	-	-	-	-	-	-	3	-	-
CO 5	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 6	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

CO 7	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 8	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 9	-	1	-	-	-	-	-	-	-	-	-	-	-	1	-
CO 10	1	-	-	-	3	-	-	-	-	-	-	-	-	-	-
CO 11	1	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO 12	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TOTAL	17	5			6								9	1	2
AVERAGE	**1.41**	**0.41**			**0.5**								**0.75**	**0.08**	**0.16**

XV. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO 1,PO 2	SEE Exams	PO 1,PO 2, PO 5,PO 9	Assignment	PO 1,PO 2, PO 5,PO 9	Seminars	PO 9, PO 10, PO 12
Laboratory Practices	-	Student Viva	-	Mini Project	-	Certification	-
Term Paper	PO 4,PO 5						

XVI. ASSESSMENT METHODOLOGIES - INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✘	Assessment of Mini Projects by Experts		

XVII. SYLLABUS

MODULE-I	**EMBEDDED COMPUTING**
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	
MODULE-II	**INTRODUCTION TO EMBEDDED C AND APPLICATIONS**
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.	
MODULE-III	**RTOS FUNDAMENTALS AND PROGRAMMING**
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.

MODULE-IV	**EMBEDDED SOFTWARE DEVELOPMENT TOOLS**
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.	
MODULE-V	**INTRODUCTION TO ADVANCED PROCESSORS**
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-Enabled 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”, dreamtech publishers, 1st Edition, 2003.
3. Frank Vahid, Tony Givargis, “Embedded System Design”, John Wiley & Sons, 3rd 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.

XVIII. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Lecture No	Topics to be Covered	COs	Reference
1-2	Definition of embedded system vs general computing system history of embedded systems.	CO 1	T1-1.1
3-4	Complex systems and microprocessor, classification, major application areas, the embedded system design process.	CO 2	T1-1.2
5-6	Analyze the Characteristics and quality attributes of embedded systems, formalisms for system design.	CO 2	T1-1.3
7	Design examples.	CO 2	T2-1.4
8-9	Explain about the C looping structures, register allocation, function calls, pointer aliasing.	CO 3	T2-1.5
10-12	Describe the Structure arrangement, bit fields, unaligned data and endianness, inline functions and inline assembly.	CO 3	R2-1.2
13-14	Identify the Portability issues; Embedded systems programming in C, binding and running embedded C program in Keil IDE.	CO 4	T3-1.3
15-16	Dissecting the program, building the hardware; Basic techniques for reading and writing from I/O port pins, switch bounce.	CO 5	T3-2.4

Lecture No	Topics to be Covered	COs	Reference
17-18	List the Applications: Switch bounce, LED interfacing, interfacing with keyboards, displays.	CO 4	T3-2.5
19-20	List the A/D and D/A conversions.	CO 5	T3-2.6
21-22	Multiple interrupts, serial data communication using embedded C interfacing.	CO 5	T3-2.7
23-24	Recall Operating system basics, types of operating systems, tasks and task states.	CO 6	T3-2.8
25-26	Describe the Process and threads, multiprocessing and multitasking, how to choose an RTOS, task scheduling, semaphores.	CO 7	T3-2.9
27-28	Queues, hard real-time scheduling considerations, saving memory and power. Task communication: Shared memory.	CO 8	R2-3.1
29-30	Explain about the Message passing, remote procedure call and sockets; Task synchronization.	CO 9	R2-3.2
31-32	Task communication synchronization issues, task synchronization techniques.	CO 9	R2-3.3
33-34	Explain the Device drivers.	CO 9	R2-3.4
35-36	Explain the Host and target machines.	CO 10	R2-3.5
37-38	Recall the Linker/locators for embedded software.	CO 9	R2-3.6
39-40	Getting embedded software into the target system; Debugging techniques.	CO 10	R3-3.7
41-42	Testing on host machine, using laboratory tools.	CO 10	R3-3.8
43-44	Example programs.	CO 10	R3-4.1
45-46	Example programs.	CO 10	R3-4.2
47-48	Explain about the Introduction to advanced architectures.	CO 11	R3-4.3
49-50	Describe about the ARM and SHARC, processor and memory organization.	CO 12	R3-4.4
51-52	Describe the Instruction level parallelism; Networked embedded systems.	CO 12	R3-4.5
53-54	List the Bus protocols, I2C bus and CAN bus.	CO 11	T2-8.1
55-56	Recognize the Internet-Enabled systems.	CO 10	T2-8.2
57-58	Design example-Elevator controller.	CO 11	T2-8.3
59-60	Example programs.	CO 11	T2-8.4

Prepared by:
Mr. S Lakshmanachari, Assistant Professor **HOD, ECE**