

## IV Semester

MICROCONTROLLER AND EMBEDDED SYSTEMS			
Course Code	21CS43	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40 T + 20 P	Total Marks	100
Credits	04	Exam Hours	03
<b>Course Learning Objectives:</b>			
CLO 1: Understand the fundamentals of ARM-based systems, including programming modules with registers and the CPSR.			
CLO 2: Use the various instructions to program the ARM controller.			
CLO 3: Program various embedded components using the embedded C program.			
CLO 4: Identify various components, their purpose, and their application to the embedded system's applicability.			
CLO 5: Understand the embedded system's real-time operating system and its application in IoT.			
<b>Teaching-Learning Process (General Instructions)</b>			
These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.			
<ol style="list-style-type: none"><li>1. The lecturer method (L) does not mean only the traditional lecture method, but different types of teaching methods may be adopted to develop the outcomes.</li><li>2. Show video/animation films to explain the functioning of various concepts.</li><li>3. Encourage collaborative (group learning) learning in the class.</li><li>4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.</li><li>5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.</li><li>6. Topics will be introduced in multiple representations.</li><li>7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.</li><li>8. Discuss how every concept can be applied to the real world, and when that's possible, it helps improve the students' understanding.</li></ol>			
<b>Module-1</b>			
Microprocessors versus Microcontrollers, ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software.			
<b>ARM Processor Fundamentals:</b> Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table, Core Extensions			
<b>Textbook 1: Chapter 1 - 1.1 to 1.4, Chapter 2 - 2.1 to 2.5</b>			
<b>Laboratory Component:</b>			
1. Using Keil software, observe the various registers, dump, CPSR, with a simple ALP programme.			
<b>Teaching-Learning Process</b>	<ol style="list-style-type: none"><li>1. Demonstration of registers, memory access, and CPSR in a programme module.</li><li>2. For concepts, numerical, and discussion, use chalk and a whiteboard, as well as a PowerPoint presentation.</li></ol>		
<b>Module-2</b>			
<b>Introduction to the ARM Instruction Set:</b> Data Processing Instructions , Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants			
<b>C Compilers and Optimization :</b> Basic C Data Types, C Looping Structures, Register Allocation, Function Calls, Pointer Aliasing,			

<b>Textbook 1: Chapter 3: Sections 3.1 to 3.6 (Excluding 3.5.2), Chapter 5</b>	
<b>Laboratory Component:</b>	
<ol style="list-style-type: none"> <li>Write a program to find the sum of the first 10 integer numbers.</li> <li>Write a program to find the factorial of a number.</li> <li>Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM.</li> <li>Write a program to find the square of a number (1 to 10) using a look-up table.</li> <li>Write a program to find the largest or smallest number in an array of 32 numbers.</li> </ol>	
<b>Teaching-Learning Process</b>	<ol style="list-style-type: none"> <li>Demonstration of sample code using Keil software.</li> <li>Laboratory Demonstration</li> </ol>
<b>Module-3</b>	
<b>C Compilers and Optimization :</b> Structure Arrangement, Bit-fields, Unaligned Data and Endianness, Division, Floating Point, Inline Functions and Inline Assembly, Portability Issues.	
<b>ARM programming using Assembly language:</b> Writing Assembly code, Profiling and cycle counting, instruction scheduling, Register Allocation, Conditional Execution, Looping Constructs	
<b>Textbook 1: Chapter-5,6</b>	
<b>Laboratory Component:</b>	
<ol style="list-style-type: none"> <li>Write a program to arrange a series of 32 bit numbers in ascending/descending order.</li> <li>Write a program to count the number of ones and zeros in two consecutive memory locations.</li> <li>Display "Hello World" message using Internal UART.</li> </ol>	
<b>Teaching-Learning Process</b>	<ol style="list-style-type: none"> <li>Demonstration of sample code using Keil software.</li> <li>Chalk and Board for numerical</li> </ol>
<b>Module-4</b>	
<b>Embedded System Components:</b> Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, purpose of embedded systems.	
Core of an Embedded System including all types of processor/controller, Memory, Sensors, Actuators, LED, 7 segment LED display, stepper motor, Keyboard, Push button switch, Communication Interface (onboard and external types), Embedded firmware, Other system components.	
<b>Textbook 2: Chapter 1 (Sections 1.2 to 1.6), Chapter 2 (Sections 2.1 to 2.6)</b>	
<b>Laboratory Component:</b>	
<ol style="list-style-type: none"> <li>Interface and Control a DC Motor.</li> <li>Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.</li> <li>Determine Digital output for a given Analog input using Internal ADC of ARM controller.</li> <li>Interface a DAC and generate Triangular and Square waveforms.</li> <li>Interface a 4x4 keyboard and display the key code on an LCD.</li> <li>Demonstrate the use of an external interrupt to toggle an LED On/Off.</li> <li>Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between.</li> </ol>	
<b>Teaching-Learning Process</b>	<ol style="list-style-type: none"> <li>Demonstration of sample code for various embedded components using keil.</li> <li>Chalk and Board for numerical and discussion</li> </ol>
<b>Module-5</b>	
<b>RTOS and IDE for Embedded System Design:</b> Operating System basics, Types of operating systems, Task, process and threads (Only POSIX Threads with an example program), Thread preemption, Multiprocessing and Multitasking, Task Communication (without any program), Task synchronization issues – Racing and Deadlock, Concept of Binary and counting semaphores (Mutex example without any	



program), How to choose an RTOS, Integration and testing of Embedded hardware and firmware, Embedded system Development Environment – Block diagram (excluding Keil), Disassembler/decompiler, simulator, emulator and debugging techniques, target hardware debugging, boundary scan.

**Textbook 2: Chapter-10 (Sections 10.1, 10.2, 10.3, 10.4 , 10.7, 10.8.1.1, 10.8.1.2, 10.8.2.2, 10.10 only), Chapter 12, Chapter-13 ( block diagram before 13.1, 13.3, 13.4, 13.5, 13.6 only)**

**Laboratory Component:**

1. Demonstration of IoT applications by using Arduino and Raspberry Pi

**Teaching-Learning Process**

1. Chalk and Board for numerical and discussion
2. Significance of real time operating system[RTOS] using raspberry pi

**Course outcome (Course Skill Set)**

At the end of the course, the student will be able to:

- CO 1. Explain C-Compilers and optimization
- CO 2. Describe the ARM microcontroller's architectural features and program module.
- CO 3. Apply the knowledge gained from programming on ARM to different applications.
- CO 4. Program the basic hardware components and their application selection method.
- CO 5. Demonstrate the need for a real-time operating system for embedded system applications.

**Assessment Details (both CIE and SEE)**

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

**Continuous Internal Evaluation:**

Three Unit Tests each of **20 Marks (duration 01 hour)**

1. First test at the end of 5<sup>th</sup> week of the semester
2. Second test at the end of the 10<sup>th</sup> week of the semester
3. Third test at the end of the 15<sup>th</sup> week of the semester

Two assignments each of **10 Marks**

4. First assignment at the end of 4<sup>th</sup> week of the semester
5. Second assignment at the end of 9<sup>th</sup> week of the semester

Practical Sessions need to be assessed by appropriate rubrics and viva-voce method. This will contribute to **20 marks**.

- Rubrics for each Experiment taken average for all Lab components – 15 Marks.
- Viva-Voce– 5 Marks (more emphasized on demonstration topics)

The sum of three tests, two assignments, and practical sessions will be out of 100 marks and will be **scaled down to 50 marks**

(to have a less stressed CIE, the portion of the syllabus should not be common /repeated for any of the methods of the CIE. Each method of CIE should have a different syllabus portion of the course).

**CIE methods /question paper has to be designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.**

**Semester End Examination:**

Theory SEE will be conducted by University as per the scheduled timetable, with common question

papers for the subject (**duration 03 hours**)

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.

The students have to answer 5 full questions, selecting one full question from each module. Marks scored out of 100 shall be reduced proportionally to 50 marks

#### Suggested Learning Resources:

##### Textbooks


1. Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan Kaufman publishers, 2008.
2. Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education, Private Limited, 2<sup>nd</sup> Edition.

##### Reference Books

1. Raghunandan. G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication, 2019
2. The Insider's Guide to the ARM7 Based Microcontrollers, Hitex Ltd., 1st edition, 2005.
3. Steve Furber, ARM System-on-Chip Architecture, Second Edition, Pearson, 2015.
4. Raj Kamal, Embedded System, Tata McGraw-Hill Publishers, 2nd Edition, 2008.

#### Weblinks and Video Lectures (e-Resources):

#### Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

  
**H.O.D**  
**Dept. of Computer Science and Design**  
**Anna's Institute of Engg. & Technology**  
**Moodelidre - 574 225**