SCHOOL OF ELECTRONICS ENGINEERING

M. Tech – Embedded Systems (MES)

Curriculum and Syllabus

(*2025-2026 admitted students*)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Adaptive Curriculum for Excellence –ACE 2025-26

M. Tech Embedded Systems

Program Educational Objectives

The engineering graduates of the programme will

1. Establish successful careers in industry, research, and academia, leveraging their expertise in advanced Embedded Systems.
2. Identify and analyse societal challenges in Embedded Systems, carry out independent research, and articulate technological solutions effectively.
3. Effectively collaborate, manage, and execute projects using relevant tools and technologies, upholding professionalism and best practices.

Program Outcomes

PO1: An ability to independently carry out research /investigation and development work to solve practical problems.

PO2: An ability to write and present a substantial technical report/document.

PO3: Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program.

PO4: Apply advanced concepts in Embedded System design, addressing real-time constraints using Microcontrollers and FPGA-based platforms.

PO5: Utilize cutting-edge hardware and software technologies to design and develop Embedded System applications in compliance with industry standards.

PO6: Identify and address research gaps using Embedded Systems to develop solutions for socio-economic, environmental, and multidisciplinary challenges.

Curriculum Structure

Program Credit Structure	Credits
University Core Courses	39
Professional Core Courses	24
Professional Elective courses	14
Open Elective Courses	03
Total Graded Credit Requirement	**80**

University Core courses (39 Credits)

S.No	Course Title	L	T	P	C
1	Technical Report Writing	1	0	3	4
2	Qualitative and Quantitative Skills Practice I	3	0	0	3
3	Qualitative and Quantitative Skills Practice II	3	0	0	3
4	Project Work	0	0	0	10
5	Internship I/ Dissertation I	0	0	0	10
6	Internship II/ Dissertation II	0	0	0	10
	Total Credits				**39**

Professional Core Courses (24 Credits)					
S. No	**Name of the Course**	**L**	**T**	**P**	**C**
1	Embedded System Design	3	1	0	4
2	Microcontroller Architecture and Organization	3	0	2	4
3	Embedded Programming	3	0	2	4
4	In-Vehicle Networking	3	1	0	4
5	Real Time Operating System	3	0	2	4
6	System Design using FPGA	3	0	2	4
	Total Credits				**24**

Professional Elective Courses (14 Credits)					
S. No	**Name of the Course**	**L**	**T**	**P**	**C**
1.	Electromagnetic Interference and Compatibility	3	0	0	3
2.	Advanced Digital Image Processing	3	1	0	4
3.	Design and Analysis of Algorithms	3	1	0	4
4.	Hardware Software Codesign	3	0	0	3
5.	Modern Automotive Electronic Systems	3	0	0	3
6.	Intelligent IoT System Design and Architecture	3	1	0	4
7.	Fault Tolerance and Dependable Systems	3	0	0	3
8.	Parallel Processing and Computing	3	1	0	**4**
9.	Edge and Cloud Computing	3	0	0	3
10.	Cyber Physical Systems	3	1	0	4
11.	5G and Next Generation Wireless Systems for Embedded Applications	3	0	0	3
12.	Machine Learning and Deep Learning	3	1	0	4
13.	Wireless and Mobile Communication for Embedded Applications	3	0	0	3
14.	Advanced Computer Architecture	3	0	0	3
15.	System on Chip Design	3	1	0	4

Course Code	Course Title	L	T	P	C
MAEDS501	Embedded System Design	3	1	0	4
Pre-requisite	NIL	Syllabus Version			
		1.0			

Course Objectives

1. Develop skillset to apply the underlying technologies and techniques for embedded real-time solutions using appropriate hardware and software.
2. Introduce advanced Modelling schemes for different embedded use cases.
3. Bring out the step-by-step building process of embedded systems.

Course Outcomes

1. Develop a typical embedded system using different modeling approaches.
2. Select the appropriate processors and memory architecture.
3. Compare various wired and wireless protocols.
4. Apply the concepts of RTOS for developing real-time embedded systems.
5. Identify the need for hardware/software co-design approach for embedded system design.

Module	Title	Hours
Module:1	**Embedded System Modelling**	**12 hours**
Introduction to Embedded System, Embedded system processor, hardware unit, software embedded into a system, Example of an embedded system, Embedded Design life cycle, Layers of Embedded Systems. Embedded System modelling: FSM, UML as Design tool, UML notation, Requirement Analysis and Use case Modelling, Petrinet, Design Examples: UI UX.		
Module:2	**Embedded system: Processor selection and building process**	**12 hours**
Microcontroller architectures: RISC, CISC. Embedded Memory, Strategic selection of processor and memory, Memory Devices and their Characteristics, Cache Memory and Various memory mapping techniques, DMA. Preprocessing, Compiling, Cross Compiling, Linking, Locating, Compiler Driver, Linker Map Files, Linker Scripts and scatter loading, loading on the target, Embedded File System, Debugging methods.		
Module:3	**Component Interfacing & Networks**	**12 hours**
Memory Interfacing, I/O Device Interfacing, Interrupt Controllers, Networks for Embedded systems-USB, UART, SPI, I2C, CAN, Wireless Applications - Bluetooth, Zigbee, Wi-Fi, 6LoWPAN, LoRa, Wireless Networking for IoT devices.		
Module:4	**RTOS based embedded solutions**	**12 hours**
Multitasking: Process and Thread, Introduction to RTOS, Kernel & its Features: polled loop system, interrupt driven system, multirate system. Scheduler, Dispatcher. Context Switching, Inter-process Communication: Shared Memory, Mailbox, Message Queue, Inter Task Synchronization: Semaphore, Mutex, Dead Lock. Security and testing in Embedded Systems: Secure boot, Encryption, Authentication, Various testing methods for embedded system.		
Module:5	**Hardware/Software Co-Design**	**10 hours**
Hardware/Software Partitioning, Co-Design Approaches for System Specification and modeling-CoSynthesis- features comparing Single-processor Architectures & Multi-Processor Architectures.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**60 hours**

Text Books

1.	Marilyn Wolf, Computers as components: Principles of Embedded Computing System Design, 2022, 2nd Edition, The Morgan Kaufmann Series in Computer Architecture and Design.
2.	Raj Kamal, Embedded systems,2020, 4th Edition, Tata McGraw- Hill.

Reference Books

1.	LylaB.Das, Embedded Systems an Integrated Approach, 2013, 1stEdition, Pearson Education.

2.	Shibu KV, Introduction to Embedded Systems,2014,1st Edition, McGraw-Hill Education (India) Private Limited.		
3.	Steve Heath, Embedded Systems Design, 2013, 2nd Edition, EDN Series.		
Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS502	**Microcontroller Architecture and Organization**	**3**	**0**	**2**	**4**
Pre-requisite	None	**Syllabus Version**			
		1.0			

Course Objectives

1.Acquaint students with the fundamental concepts of computer architecture and organization.
2.Familiarize students with ARM and THUMB architectures and their assembly language programming.
3.Provide hands-on knowledge of ARM7 LPC214x microcontroller and its peripheral interfaces.

Course Outcomes

1.Analyze basic computer architecture components and differentiate instruction execution mechanisms across RISC, CISC, and ARM architectures.
2.Develop ARM and THUMB assembly language programs to perform data processing and control operations, and demonstrate comprehension of their instruction sets.
3.Design and implement embedded applications using LPC214x peripherals such as GPIO, timers, PWM, and UART.
4.Apply interfacing techniques for external devices using I2C, SPI, ADC, and DAC, and analyze interrupt handling mechanisms on the ARM7 LPC214x microcontroller.
5.Develop and analyze embedded programs using ARM/THUMB assembly and LPC2148 peripherals to perform arithmetic, logic, control, and interfacing tasks in real-time applications.

Module:1	**Processor Architecture and Organization**	**6 hours**

Basics of Processor Architecture: von Neumann vs Harvard, CPU Organization: ALU, Control Unit, Registers, RAM, ROM, Instruction Cycle and Execution Flow, Register Bank, Bit and Byte addressable location, Architecture of 8051 and its peripherals.

Module:2	**ARM Architecture Basics and its Assembly Language Programming**	**8 hours**

ARM Architecture Overview (ARM7TDMI, ARM9, Cortex), Programmer's Model: Registers, CPSR, SPSR, Addressing Modes, Data Processing Instructions, Branch Instructions, Load-Store Instructions, SWI, CPSR Instructions, Loading Constants, Conditional Execution.

Module:3	**THUMB Instruction Set Architecture and its Programming**	**8 hours**

Need for THUMB Instruction Set, THUMB state: Register Set and State Switching, ARM-THUMB Interworking, Instruction Set: Data Processing, Single Register Load-store Instructions, Multiple Register Load-store Instructions, Branch Instructions, Stack Instructions, software Interrupt Instructions.

Module:4	**ARM7 LPC214x Peripherals -I**	**12 hours**

Overview of LPC214x Architecture and Features, GPIO Programming and I/O Port Access, Timers and Counters: Configuration and Applications, PWM: Modes and Use Cases, UART Programming: Serial Communication Basics, Baud Rate Settings.

Module:5	**ARM7 LPC214x Peripherals-II**	**9 hours**

ADC and DAC, Sensor interfacing and its application, I2C, SPI, Watchdog Timer and Real-Time Clock, Interrupt System in LPC214x: VIC, IRQ, FIQ.

Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Books

1	M. A. Mazidi, S. Naimi, and S. Mazidi, ARM Assembly Language Programming & Architecture, 2016, 2nd Edition, MicroDigitalEd.
2	L. D. Pyeatt, Modern Assembly Language Programming with the ARM Processor, 2016 ,1st Edition,Burlington, MA: Morgan Kaufmann.

Reference Books	
1.	M. A. Mazidi, J. G. Mazidi, and R. D. McKinlay, The 8051 Microcontroller and Embedded Systems: Using Assembly and C, 2014,2nd Edition, Pearson New International Edition, Pearson Education.
2.	Sloss, Andrew N., Dominic Symes, and Chris Wright. ARM System Developer's Guide: Designing and Optimizing System Software, 2004, 1st Edition,Morgan Kaufmann.
3.	P. Cockerell, ARM Assembly Language Programming, 1996, Cambridge, UK: Cambridge University Press.
4.	NXP Semiconductors, LPC214x User Manual, 2006, Rev. 2.1, User manual UM10139.
Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT	

List of Experiments (Indicative)	
Simple Arithmetic Logical and Bitwise Operations Using ARM Assembly	**2 hours**
Branching and Looping in ARM Assembly (Use of conditional and unconditional branches with loops)	**2 hours**
Write and Simulate an ARM Assembly Program for Sorting an Array	**2 hours**
Switching Between ARM and THUMB Modes (Demonstration of state switching and dual-mode programming)	**2 hours**
Arithmetic and Logical Instructions in THUMB Mode	**2 hours**
Branching, Conditional Execution, and Stack Operations in THUMB Mode	**2 hours**
Implement Subroutine Calls and Return Mechanism in THUMB	**2 hours**
LED Blinking Using GPIO in LPC2148	**2 hours**
Interfacing Push Button and Controlling LED via GPIO	**2 hours**
Generating Time Delay Using On-chip Timers	**2 hours**
I2C,SPI Protocol Implementation using LPC2148	**2 hours**
PWM Signal Generation to Control Brightness of an LED or Speed of a Motor	**2 hours**
Serial Communication Using UART (Sending and Receiving Characters)	**2 hours**
Analog to Digital Conversion Using On-chip ADC (Read analog voltage and display corresponding digital value via UART)	**4 hours**
Total Laboratory Hours	**30 hours**
Mode of Evaluation: Continuous Assessment and Final Assessment Test.	

Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

<table>
<tr><td>Course Code</td><td colspan="2">Course Title</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>MAEDS503</td><td colspan="2">Embedded Programming</td><td>3</td><td>0</td><td>2</td><td>4</td></tr>
<tr><td>Pre-requisite</td><td colspan="2">Nil</td><td colspan="4">Syllabus Version</td></tr>
<tr><td></td><td colspan="2"></td><td colspan="4">1.0</td></tr>
<tr><td colspan="7">Course Objectives</td></tr>
<tr><td colspan="7">1.Acquaint students with the fundamental concepts and syntax of the C programming language with essential data structures and their implementation in programming.
2.Familiarize students with SHELL programming and the Linux operating system environment.
3.Enable students to implement and work with device drivers in a Linux environment.</td></tr>
<tr><td colspan="7">Course Outcomes</td></tr>
<tr><td colspan="7">1.Apply the fundamental concepts of C programming and data structures to solve basic computational problems.
2.Apply knowledge of Linux basics in SHELL programming.
3.Analyze the components of embedded Linux systems to understand their roles and interactions.
4.Apply kernel module programming techniques to develop and test basic device driver programs.
5.Design and implement embedded and Linux-based development environments using state-of-the-art hardware and software tools.</td></tr>
<tr><td>Module:1</td><td colspan="2">Core Programming Concepts with C and Data Structures</td><td colspan="4">14 hours</td></tr>
<tr><td colspan="7">Basic concepts of C, Embedded C Vs C, Embedded programming aspects with respect to firmware, Bitwise manipulation, Functions, Recursive Functions, Arrays, pointers, Function Pointers, File Handling, Preprocessor Directives, structures, Union, and Inputs/Outputs, static and dynamic memory allocation, Linked list, Single linked list, Double linked list and Queues.</td></tr>
<tr><td>Module:2</td><td colspan="2">Linux Essentials with Shell Scripting Techniques</td><td colspan="4">9 hours</td></tr>
<tr><td colspan="7">Command prompt, X windows basics, navigating file system, finding files, working with folders, reading files text editing in Linux, Compression and archiving tools, Basic shell commands, File Management, I/O Handling, File Locking.
Processes, giving more than one command at a time, prioritizing and killing processes, Scheduling Commands, pipes and redirection, regular expression, pattern matching, Scripting using for while, if and other commands.</td></tr>
<tr><td>Module:3</td><td colspan="2">Embedded Linux Development Essentials</td><td colspan="4">7 hours</td></tr>
<tr><td colspan="7">Introduction to Embedded Linux - Linux Kernel Basics for Embedded Systems - Build Systems and Cross-Compilation - Root Filesystem Essentials - Device Drivers and Hardware Interfacing - Debugging Techniques.</td></tr>
<tr><td>Module:4</td><td colspan="2">Kernel Module Programming</td><td colspan="4">7 hours</td></tr>
<tr><td colspan="7">Compiling kernel, Configuring Kernel and compilation, Kernel code, browsers,Static linking, dynamic linking of modules, User space, kernel space concepts, writing simple modules – Writing, Make-files for modules.</td></tr>
<tr><td>Module:5</td><td colspan="2">Device Driver concepts</td><td colspan="4">6 hours</td></tr>
<tr><td colspan="7">Driver concepts, Block & character driver distinction, Low level drivers, OS drivers etc, Writing character drivers – GPIO driver, Device major, minor number.</td></tr>
<tr><td>Module:6</td><td colspan="2">Contemporary Issues</td><td colspan="4">2 hours</td></tr>
<tr><td colspan="7"></td></tr>
<tr><td></td><td colspan="2">Total Lecture hours:</td><td colspan="4">45 hours</td></tr>
<tr><td colspan="7">Text Books</td></tr>
<tr><td>1</td><td colspan="6">Neil Mathew, Richard stones, Beginning Linux Programming, 2012,4th Edition, Wrox – Wiley Publishing, USA.</td></tr>
<tr><td>2</td><td colspan="6">Eric Foster Johnson, John C. Welch, Micah Anderson, Beginning shell scripting, 2012, 1st Edition, Wrox – Wiley Publishing, USA.</td></tr>
</table>

Reference Books			
1.	Derek Molloy, Exploring Beagle Bone: Tools and Techniques for Building with Embedded Linux, 2015, 1st Edition, Wiley Publications, USA		
2.	Frank Vasquez, Profile IconChris Simmonds, Mastering Embedded Linux Programming, 2021, 3rd Edition, Packt Publishing, UK.		
Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT			
List of Experiments (Indicative)			
1.Application of functional Pointers			**6 hours**
2. Implementation of Linked List Concepts.			**6 hours**
3. Application of Shell Script Programming.			**6 hours**
4.Build the process for an embedded board: Build a kernel for a Beagle Bone Black (BBB)/Raspberry Pi board and board bring up, kernel module program on an embedded board			**6 hours**
5.Device driver programming –Implementation of Device Driver			**6 hours**
Total Laboratory Hours			**30 hours**
Mode Evaluation: Continuous Assessment and Final Assessment Test.			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS504	**In-Vehicle Networking**	**3**	**1**	**0**	**4**
Pre-requisite		**Syllabus Version**			
		1.0			

Course Objectives

1.Providing students with a working knowledge of In-Vehicle network systems
2.Acquainting aspects of design, development, application and performance issues associated with vehicle networking systems.
3. Illustrating concepts of sensor data capture, storage and exchange of data to access remote services

Course Outcomes

1.Infer the need for data communication, networking, and Automotive Software—AUTOSAR in practical in-vehicle communication systems.
2.Implement CAN and its higher-layer protocols, including CANopen, DeviceNet, TTCAN, and SAE J1939, in automotive applications
3.Differentiate the functionalities of the LIN, MOST, and FlexRay protocols in various automotive applications.
4.Examine the role of automotive Ethernet and TSN technologies in modern vehicle communication systems
5. Analyze the effectiveness of wireless communication technologies, general-purpose networking protocols -TCP/IP, UDP, and V2X in connected vehicle systems.

Module:1	**Fundamentals of In-Vehicle Communication Systems**	**12 hours**

Fundamentals of Data communication System, Components, Performance metrics, Network topologies, OSI & TCP/IP models; Emphasis on the Data Link Layer and its sublayers; Logical Link Control (LLC) : Framing and flow control; Error detection and correction techniques; Medium Access Control (MAC) protocols - Random Access- CSMA, CD versus CA; Line coding and block coding methods; Recap of serial communication protocols, Multiplexing techniques; Need for in-vehicle networking; Role and types of ECUs; Software abstraction through AUTOSAR architecture.

Module:2	**Controller Area Network (CAN) and Higher Layer Protocols**	**14 hours**

CAN and Time-Triggered Communication – Introduction to CAN protocol: features, architecture, and applications in in-vehicle networking; CAN frame structure: base and extended frames, arbitration, bit stuffing, and CRC; CAN physical and data link layer details; error handling and fault confinement mechanisms; baud rate, synchronization, and bus loading concepts; Higher Layer Protocols: CANopen, DeviceNet, and SAE J1939, CAN FD; need for deterministic communication in safety-critical systems; limitations of event-triggered CAN; introduction to Time-Triggered CAN (TTCAN) for real-time and scheduled message transmission; TTCAN communication cycle and global time synchronization; use cases of TTCAN in automotive safety and automation systems.

Module:3	**Automotive Protocols for Body, Safety and Infotainment**	**10 hours**

Overview of LIN (Local Interconnect Network): architecture, master-slave scheduling, frame structure, checksum, and use in body control modules; FlexRay protocol: need for fault tolerance and high data rates, static vs dynamic segments, global time synchronization, dual-channel operation, and configuration; MOST (Media Oriented Systems Transport): Topology, Application Framework, Function Blocks, use in multimedia systems; Timing Master - Data Flow. network security in automotive systems: attack types and security solutions.

Module:4	**Automotive Ethernet**	**12 hours**

Introduction to Ethernet in Automotive Networks: ; frame structure, MAC addressing, Ethernet types-100BASE-T1, 1000BASE-T1 ; Automotive Ethernet standards: BroadR-Reach ; Time-Sensitive Networking (TSN): introduction to IEEE 802.1Qbv, 802.1AS, 802.1Qbu; protocols for automotive Ethernet: Dynamic Service Oriented architecture : Evolution of AUTOSAR for Ethernet stack; SOME/IP, DoIP.

Module:5	Wireless and V2X Communication Technologies	10 hours
Overview of wireless networking in vehicles and its role in infotainment, diagnostics, and safety; General-purpose protocols: TCP, UDP, IP; Short-range wireless technologies: Bluetooth, NFC, Zigbee, UWB; WiFi in automotive and IEEE 802.11 evolution; DSRC and IEEE 802.11p (WAVE); Cellular technologies: GSM, LTE, 5G and their role in telematics, OTA updates, and cybersecurity considerations; V2X communication types: V2V, V2I, V2P, V2D, V2G, V2C, V2X; C-V2X architecture and comparison with DSRC; VANETs vs MANETs; safety vs comfort applications in VANETs.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**60 hours**
Text Books		
1.	D. Paret and H. Rebaine, Autonomous and Connected Vehicles: Network Architectures from Legacy Networks to Automotive Ethernet, 2022, 1^{st} Edition, John Wiley & Sons.	
2.	Mueck, Markus, and Ingolf Karls. Networking vehicles to everything: Evolving automotive solutions. 2018,1^{st} Edition, Walter de Gruyter GmbH & Co KG.	
Reference Books		
1.	D. Paret, Multiplexed Networks for Embedded Systems CAN, LIN, FlexRay, Safe-By-Wire, Jun. 2007,1^{st} Edition, Nashville, TN: John Wiley & Sons.	
2.	B. A. Forouzan, Data Communications and Networking with TCP/IP Protocol Suite, 2021, 6^{th} Edition,McGraw-Hill Education.	
3.	W. Voss, A comprehensible guide to controller area network, 2008, Copperhill Media.	
4.	A. Grzemba, MOST: the automotive multimedia network, Jan. 2012, 1^{st} Edition,ser. Netzwerk. Haar, Germany: Franzis.	
5.	K. Matheus and T. K¨onigseder, Automotive Ethernet, Apr. 2021,3^{rd} Edition, Cambridge University Press.	
Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT		
Recommended by Board of Studies		
Approved by Academic Council	**No.**	**Date**

Course Code	Course Title	L	T	P	C
MAEDS505	**Real Time Operating System**	**3**	**0**	**2**	**4**
Pre-requisite	**Nil**	**Syllabus Version**			
		1.0			

Course Objectives

1.Introduce the fundamental concepts and architecture of Operting systems and RTOS.
2.Acquaint students about Task Management and Familiarize students with various scheduling algorithms.
3.Expose student to inter-process synchronization and communication techniques in real-time environments and memory management in RTOS.

Course Outcomes

1.Illustrate the fundamentals of Operating Systems and Real-Time Operating Systems (RTOS).
2.Apply task management, scheduling techniques, and memory management in RTOS.
3.Apply inter-process synchronization and communication mechanisms to facilitate reliable data exchange in RTOS.
4. Analyze real-time system resource utilization to ensure optimal performance and responsiveness.
5. Design and develop real-time applications by integrating concepts and tools of modern RTOS platforms.

Module	Title	Hours
Module:1	**Essentials of Real-Time Operating Systems**	**9 hours**
Layers of Operating Systems, System Boot up - BIOS & Boot Process, UEFI Specification, Kernel Types, GPOS vs RTOS, RTOS: Characteristics, Types, Architecture, Functions, Performance metrics, Tradeoffs, Use cases in automotive, industrial, and consumer systems.		
Module:2	**Task Management and Task Scheduling**	**9 hours**
Process, Process Control Block, Process Attributes, Mutliprocessing, Task Types, Multitasking, Task/Process states, Threads, POSIX Threads, Mutlithreading, Priority based scheduling: Shortest Job First, Shortest Remaining Job Next, Rate-Monotonic scheduling, Earliest Deadline first scheduling, Linux RT scheduler.		
Module:3	**Inter Process and Task Synchronization**	**9 hours**
IPC, Race conditions and critical sections, Signals, Atomic operations, Semaphore, Mutex, Deadlock, Priority Inversion and Priority inheritance.		
Module:4	**Inter Process Communication – Data Exchange**	**8 hours**
Shared memory, Pipes, Messages Queues and Mailbox, Circular and swinging buffers.		
Module:5	**Memory Management**	**8 hours**
Memory Allocation Strategies in RTOS - Static, Dynamic, virtual memory, Paging, Segmentation, Swapping.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Books

1	Jiacun Wang, Real-Time Embedded Systems, 2017, 1^{st} Edition,Wiley, USA
2	K. Erciyes, Distributed Real-Time Systems: Theory and Practice,2019,1^{st} Edition, Springer,USA.

Reference Books

1.	Ivan CibrarioBertolotti, Politecnico di Torino and Gabriele Manduchi, Real-Time Embedded Systems: Open-Source Operating Systems Perspective, 2017, 1^{st} Edition., CRC Press, USA
2.	Lyla B. Das, Embedded Systems an Integrated Approach, 2012, 1^{s}t Edition., Pearson Education, India.

Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT

List of Experiments (Indicative)			
Working with Pthreads in Linux			**4 hours**
Working with Task Management-Task Creation, deletion, Task Properties			**4 hours**
Implementation of Real time Task Scheduling			**6 hours**
Implementation of Inter task/process synchronization techniques - Signals, Semaphore, Mutex			**6 hours**
Implementation of Inter task/process Communication - Data exchange techniques - Message Queue, Pipes, Timers			**6 hours.**
Working with Memory Management techniques			**4 hours**
Total Laboratory Hours			**30 hours**
Mode Evaluation: Continuous Assessment and Final Assessment Test.			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS506	System Design using FPGA	3	0	2	4
Pre-requisite		Syllabus Version			
		1.0			

Course Objectives

1. Developing a comprehensive understanding of FPGA architecture, design flows, and HDL-based development for implementing digital systems.
2. Design, simulate, and prototype embedded systems using FPGAs, including IP integration and peripheral interfacing.
3. Demonstrate an understanding of hardware-accelerated applications, including matrix multiplication, convolution operations, and neural network inference acceleration using FPGAs.

Course Outcomes

1. Develop and simulate combinational circuits using HDL and test benches.
2. Develop and simulate sequential circuits using HDL and test benches.
3. Build and integrate IP cores for an FPGA based system.
4. Apply a hardware-software co-design approach to accelerate real-world applications.
5. Simulate an IP-based system design using an FPGA.

Module:1	FPGA architecture and Hardware Descriptive languages	9 hours

Historical evolution of programmable logic devices to FPGAs, Internal FPGA architecture: Logic elements,CLBs, LUTs, routing architecture, memory blocks, DSP slices, and transceivers, Comparision of FPGAs with other Hardware Development Platforms-Microcontroller, ASIC, and GPUs. Exploring FPGA Families from Industry Leaders: AMD and Intel, Execution models: Parallel execution in FPGAs vs Processor-centric execution in CPUs and microcontrollers, FPGA design flow: From RTL to bitstream – synthesis, implementation, place and route, and programming,Embedded System Design Cycle, Top Down Design Flow, Bottom Up Design Flow.

Module:2	Modeling of Combinational logic circuits	9 hours

Language constructs in HDL, Operators, data types, attributes, behavioral, dataflow, and structural modelling styles, Functional simulation to validate logic correctness using testbenches,Design examples. Testbenches for simulation and functional verification,Half adder, Full adder, 4-bit/8-bit binary adder, ALU design, Multiplexer and De-multiplexer, Encoder, Decoder, Comparator, Ripple Carry Adder, Carry Look ahead adder, Simple Barrel Shifter

Module:3	Modelling of Sequential logic circuits	9 hours

Flip Flops -Realization of Shift Register -Realization of a Counter -Synchronous and Asynchronous – BCD counter, Mealy and Moore State Machines, Sequence detector, FIFO, Memory Design, Serial Data Receiver, Serial to parallel data converter. RAM design,Wave Form Generator,Traffic Light Controller

Module:4	Interfacing peripherals with FPGA	8 hours

IP based system design, Interfacing FPGA logic with ARM CPU via AXI bridges , IEEE 754 Floating Point IP Cores, Interfacing to LEDs ,Switches, Buzzer, LCD display, 7 segment display, Stepper Motor, ADC and Sensors, SPI interface, I2C Interface.

Module:5	Advanced System Design Concepts	8 hours

Hardware Software Co-Design, Concept of Hardware Software partitioning, Custom IP core design and integration,Phase Locked Loop(PLL), Design of software drivers for custom hardware peripherals, Interfacing custom IPs with Hard Processor System (HPS) in SoC FPGA, Case study: Hardware accelerated matrix multiplication and convolution, Convolutional Neural Network, Partial Reconfiguration flow: Static vs Dynamic regions,Security in FPGA Designs -bitstream encryption and tamper detection.

Module:6	Contemporary Issues	2 hours
	Total Lecture hours:	45 hours

Text Books	
1.	Volnei A. Pedroni , Circuit Design with VHDL, 2020,3^{rd} Edition, Massachusetts Institute of Technology.
2.	Cem Ünsalan, Bora Tar, Digital System Design with FPGA: Implementation Using Verilog and VHDL, 2017,1^{st} Edition, McGraw-Hill Education.
Reference Books	
1.	Steve Kilts, Advanced FPGA Design: Architecture, Implementation, and Optimization, 2007,1^{st} Edition,Wiley-IEEE Press.
2.	Wayne Wolf, FPGA-based System Design,2004,1^{st} Edition Prentice Hall.
3.	Ian Grout, Digital Systems, Design with FPGAs and CPLDs,2011, Newnes.
4.	Mohammad Tehranipoor, Cliff Wang, Introduction to Hardware Security and Trust, 2011, Springer New York, NY.
5.	Ronald R. Sass and Andrew Schmidt, Embedded Systems Design with Platform FPGAs: Principles and Practices, 2010,2^{nd} Edition,Morgan Kaufman Publishers.
Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT	

List of Experiments (Indicative)	
Modelling basic components of combination logic using concurrent style.	**6 hours**
Develop a modular Arithmetic Logic Unit (ALU) by integrating submodules using both structural and concurrent approaches.	**4 hours**
Implement counters, shift registers, and flip-flop-based systems using mixed HDL modeling styles.	**6 hours**
Design and simulate FSM-based applications such as a string detector, traffic light controller, and vending machine controller.	**4 hours**
Utilize SoC and IP integrator tools to implement hardware interfacing for basic I/O peripherals.	**6 hours**
Custom IP Core Design and Integration in SoC FPGA Environment or Create, synthesize, and integrate a user-defined IP block into a Zynq-based embedded system using AXI interfaces.	**4 hours**
Total Laboratory Hours	**30 hours**
Mode Evaluation: Continuous Assessment and Final Assessment Test.	

Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS601	Electromagnetic Interference and Compatibility	3	0	0	3
Pre-requisite	None	Syllabus Version			
		1.0			

Course Objectives

1. Imparting knowledge about EMI environment.
2. Acquaint students with control techniques and design of PCBs for EMC.
3. Providing exposure to EMI standards, Regulations and Measurements.

Course Outcomes

1. Classify different sources of EMI and their coupling mechanisms.
2. Identify various techniques needed to suppress EMI.
3. Ability to design Electromagnetic compatible systems.
4. Illustrate various EMI test and measurement methods.

Module:1	Introduction to Electromagnetic compatibility	7 hours

EMI-EMC, Sources of EMI, Conducted and radiated EMI, EMC Engineering application, Coupling mechanisms, EMC requirements for electronic systems, Non-ideal behavior of components: Wires, resistors, capacitors, inductors, Elemental radiators, Signal spectral analysis.

Module:2	Cabling & Grounding	9 hours

Capacitive coupling, inductive coupling, mutual inductance, effect of shield on capacitive and inductive coupling, Braided shields, Spiral shields, shield terminations, ribbon cables, cross talk.
AC power distribution and safety grounds, Signal grounds, System grounding, ground loops and single ground reference for a circuit.

Module:3	Filtering & Shielding	9 hours

Common Mode Rejection Ratio (CMRR), Cable balance, system balance, balanced loads, Common-mode filters, parasitic effects in filters, power supply decoupling.
Near and far fields, shielding effectiveness, apertures, waveguides, conductive gaskets, windows, coatings, grounding of shields.

Module:4	System design for EMC	9 hours

PCB Design: Component selection, placement, Input and Output cable placement, filtering, power distribution, decoupling, loop area reduction, partitioning.
System design: Enclosures, power line filter placement, interconnection, internal cable routing and connector placement, PCB and subsystem placement, decoupling.

Module:5	EMC standards, Pre-compliance EMC measurements	9 hours

Standards: FCC, CISPR, ANSI, IEC, SAE, MIL.
Antennas, Probes, conducted emission testing, radiated emission testing, conducted immunity and radiated immunity testing, ESD testing.

Module:6	Contemporary Issues	2 hours
	Total Lecture hours:	**45 hours**

Text Books

1.	Paul, Clayton R., Robert C. Scully, and Mark A. Steffka. Introduction to Electromagnetic Compatibility,2022,3rd Edition, John Wiley & Sons.
2.	Kodali, V. Prasad. Engineering Electromagnetic Compatibility: principles, measurements, and technologies.,2010,2nd Edition, Wiely -IEEE Press.

Reference Books

1.	Ott, Henry W. Electromagnetic Compatibility Engineering. 2011,3rd Edition, John Wiley & Sons.
2.	Williams, Tim. EMC for product designers, 2016, 5th Edition, Newnes.

Mode of Evaluation: Quiz, Assignment, CAT and FAT

Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS602	**Advanced Digital Image Processing**	**3**	**1**	**0**	**4**
Pre-requisite	**NIL**	**Syllabus Version**			
		1.0			

Course Objectives

1.Acquaint the mathematical foundation of digital image processing and computer vision.
2.Familiarize the image processing techniques like enhancement, restoration methods, segmentation, object detection and video analysis.
3.Provide knowledge on machine learning and deep learning applications in segmentation, object detection and image classification and deploy them on embedded platforms.

Course Outcomes

1.Use the mathematical concepts to solve the problems in digital image processing.
2.Apply the restoration methods, spatial and frequency domain techniques for image enhancement.
3.Distinguish the classical and deep learning models for segmenting images into meaningful regions.
4.Apply feature extraction techniques to machine learning and deep learning models for image classification and object detection.
5.Apply motion estimation techniques to implement single and multiple object tracking algorithms.

Module:1	Mathematical Foundations for Digital image processing	12 hours

Linear Algebra - Vector Spaces, Norms, Eigenvalues, Eigenvectors; Matrix Decompositions - Singular Value Decomposition(SVD), Principal Component Analysis (PCA);Random variables – Distributions-Gaussian, Uniform, Exponential; Statistical Measures - expectations, variance, covariance, correlation; Bayesian Probability - Bayes' rule and Applications; Fundamental steps in DIP – Elements of visual perception - Image Sampling and Quantization - Basic relationship between pixels. Noise models – Gaussian Noise, salt-and-pepper noise, speckle noise; Fourier Analysis - 2D Fourier Transform and properties - linearity, translation, scaling, rotation; 2D Convolution Theorem, Discrete Cosine Transform (DCT); Introduction to TinyML for embedded hardware.

Module:2	Image Enhancement and Restoration	12 hours

Spatial Domain operations - Point operations, Mask Operations; Image Smoothing – Mean, Median, Gaussian; Image Sharpening - Laplacian, High-boost Filtering, Unsharp Masking, Edge-Modulation; Image Gradients - First-order Derivative Filters-Sobel, Prewitt, Canny ; Second-order Derivative Filters -Laplacian of Gaussian; Frequency Domain Filtering: Low pass and high-pass filters; Color transforms - Histogram equalization, Adaptive histogram equalization- Contrast Limited Adaptive Histogram Equalization(CLAHE); Basic Color adjustment - Gamma correction, S-curve adjustment, White balance correction; Image Restoration - inverse and Wiener filtering; Pre-processing using TinyML with embedded hardware.

Module:3	Classical and Deep Learning Approaches for Image Segmentation	12 hours

Classical segmentation – Thresholding -Global, Adaptive; Edge-based Segmentation-Sobel, Canny, Edge Linking, Boundary Extraction; Region-based Segmentation-watershed Algorithm; Clustering-based Segmentation -K-means clustering; Deep-Learning based Segmentation - Semantic Segmentation-U-Net, DeepLAB Family; Instance Segmentation –Mask Region-based Convolutional Neural Network(R-CNN) , You Only Look At CoefficienTs (YOLACT); Prompt-based Segmentation - Segment Anything Model (SAM); Deploying Lightweight Segmentation Models on edge devices.

Module:4	Classical and Deep Learning Approaches for Image Classification and Object Detection	12 hours

Pipeline of Image Classification: Feature Extraction, Classification, Evaluation;Classical Feature Extraction - Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform(SIFT);

Classical Classifiers – Support Vector Machine(SVM), K-Nearest Neighbors (KNN), Decision Trees, Random Forests; Evaluation Metrics - Precision, Recall, F1 Score; Deep-Learning for classification - VGG16, ResNet, Inception, MobileNet; Introduction to Object Detection; Classical Approaches - Sliding Window,HOG and SVM; Deep Learning-based Object Detection – YOLO, Faster R-CNN. Deploying Object Detection Models -YOLOv5n and MobileNet-SSD on Embedded Platforms.		
Module:5	**Video Analysis**	**10 hours**
Introduction - Motion estimation, Tracking, Stabilization; Motion estimation - Lucas-Kanade Optical Flow, Application: Video stabilization; Object tracking – Single Object Tracking Algorithms: MeanShift, CamShift; Multiple object tracking Algorithms: Kalman filtering, Multi-Object Tracking (MOT), Particle Filters; Tracking Evaluation Metrics - Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Precision,Recall, and ID Switching. Embedded Real-time Object Tracking using OpenCV on embedded hardware with camera.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**60 hours**

Text Books	
1	R.C. Gonzalez and R.E. Woods, Digital Image Processing, 2018, 4th Ed., Pearson.
2	Szeliski, Richard, Computer Vision: Algorithms and Applications, 2023, 2nd edition, Springer.
Reference Books	
1.	Anirudh Koul, Siddha Ganju, Meher Kasam, Practical Deep Learning for Cloud, Mobile, and Edge, 2023, O'Reilly Media, Inc.
2.	S Jayaraman, S Esakkirajan, T Veerakumar, Digital image processing, 2020, 2nd Edition, McGraw Hill.

Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS603	**Design and Analysis of Algorithms**	**3**	**1**	**0**	**4**
Pre-requisite		**Syllabus Version**			
		1.0			

Course Objectives

1. Enabling the students to carry out analysis of various algorithms for mainly time and space complexity.
2. Equipping the students to decide the appropriate data type and data structure for a given problem.
3. Fostering Analytical skills to select the best algorithm to solve a problem by considering various problem characteristics, such as data size, type of operations, etc.

Course Outcomes

1. Develop proficiency in problem solving and programming.
2. Analyse advanced algorithms for mainly time and space complexity.
3. Interpret mathematical models of Cryptographic Algorithms.
4. Analyse the compute complexities of Geometric Algorithms.
5. Analyse Parallel Algorithms and understand the Deep Learning Algorithms.

Module	Title	Hours
Module:1	**Introduction to Algorithmic Analysis**	**12 hours**
Role of Algorithms in computing, Analysis of Algorithms, Asymptotic notation, Euclid's algorithm, Problem, Instance, RAM model, Principles of Algorithm Design, Sorting Algorithm - Insertion Sort & Complexity Analysis, Divide and Conquer Technique, solving recurrences - substitution, Iteration, Recursion tree, Changing variable and Master's Method.		
Module:2	**Advanced Algorithmic Analysis**	**11 hours**
Backtracking; Dynamic programming; Greedy Technique; Branch & Bound; Amortized analysis; Online and offline algorithms; Randomized algorithms, NP Completeness		
Module:3	**Cryptographic Algorithms**	**14 hours**
Historical overview of cryptography; Private-key cryptography and the key-exchange problem; Public-key cryptography; Digital signatures; Security protocols; Applications- Zero knowledge proofs, authentication etc..		
Module:4	**Geometric Algorithms**	**10 hours**
Line segments: properties, intersections; convex hull finding algorithms, Voronoi Diagram, Delaunay Triangulation.		
Module:5	**Deep Learning Algorithms & Parallel Algorithms**	**11 hours**
Recurrent Neural Networks (RNNs), Autoencoders, Generative Adversarial Networks (GANs): Architecture & Training Algorithm, PRAM model; Exclusive versus concurrent reads and writes; Pointer jumping; Brent’s theorem and work efficiency.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**60 hours**

Text Books

1	Anany Levitin, Introduction to the Design and Analysis of Algorithms,2017,3rd edition., Addison Wesley.
2	William Stallings, Cryptography & Network Security,2023, 4th Edition, Prentice Hall.

Reference Books

1.	Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, 2009,3rd edition, McGraw-Hill.
2.	Simon J.D. Prince, Understanding Deep Learning: Building Machine Learning Systems,2023, The MIT Press.
3.	Ellis Horowitz, Fundamentals of Computer Algorithms, 2008,2nd Edition, Universities Press.

4.	M. J. Quinn, Parallel computing – theory and practice, 2017,2nd Edition, McGraw Hill.
5.	Sukumar Ghosh, Distributed Systems: An Algorithmic Approach ,2006,1st edition, Chapman & Hall/CRC Computer & Information Science Series.

Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS604	**Hardware Software Codesign**	**3**	**0**	**0**	**3**
Pre-requisite	None	**Syllabus Version**			
		1.0			

Course Objectives

1. Acquaint students with hardware/software codesign and its role in energy efficiency and system optimization.
2. Familiarize students with modeling techniques and enable software-to-hardware mapping using FSMD and SoC simulation.
3. Provide basic knowledge to design interfaces and implement communication between hardware and software.

Course Outcomes

1. Analyze key principles of hardware/software codesign and evaluate their role in achieving energy efficiency in embedded systems.
2. Apply data flow and control flow modeling techniques, including concurrency analysis, to develop and examine embedded system behavior under design constraints.
3. Design and simulate embedded system components such as custom processors and SoCs using FSMD models and RISC architectures with appropriate simulation tools.
4. Implement and analyze hardware/software interfaces using synchronization techniques, bus communication protocols, and memory-mapped I/O to optimize system performance on FPGA platforms.

Module:1	**Foundations in Hardware/Software Codesign and Partitioning Methodologies**	**10 hours**

Introducing Hardware/Software Codesign, The Quest for Energy Efficiency, Driving Factors in Codesign, Codesign Space, Dualism in Hardware Design and Software Design. Codesign concepts: Hardware/Software Partitioning, Hardware/Software Alternatives, Hardware/Software Trade-offs Partitioning Granularity, Kernighan-Lin Algorithm, Extended Partitioning, Binary Partitioning, GCLP Algorithm.

Module:2	**Modeling Techniques**	**8 hours**

Modeling, Concurrency, Data Flow Graphs, Synchronous Data Flow, Schedule Analysis, ControlFlow Modeling, Timing & Resource Constraints, Transformations, Software Implementation of Data flow :Queues, Actors, Scheduling, Hardware Implementation of Data flow. Control Flow and Data flow Analysis using C Program.

Module:3	**Custom Architectures in Hardware/Software Codesign**	**10 hours**

Finite State Machine with Datapath (FSMD) Modeling, FSMD Design Example, Microprogrammed Architecture: Limitations of FSM, Microprogrammed control, Microinstruction encoding, Datapath, Microprogrammed Machine, Microprogram Pipeline, Microprogramming with Microcontrollers. General Purpose Processors Embedded Cores- RISC Pipeline, Program Organization, Compiler tools, Low Level Program Analysis, Processor Simulation, SoC Concept, Design Principle in SoC Architecture.

Module:4	**Interface Design and Communication Mechanisms**	**8 hours**

Principles of Hardware/Software Communication: Connecting Hardware-Software, Synchronization schemes, Communication constrained and computation constrained, On-Chip Bus Systems, BusTransfer Mechanisms, Multi-Master Bus Systems, Bus Topologies, Microprocessor Interfaces: Memory-Mapped interface, Coprocessor Interfaces, Custom Instruction Interfaces.

Module:5	**Embedded Applications in Hardware/Software Codesign**	**7 hours**

Trivium Algorithm, 8-bit & 32-bit Implementations, Hardware Testbenches, Driver Code, T-box Implementation, Custom Instructions, System Performance Evaluation, CORDIC Algorithm & Implementation, FPGA Prototyping, Hardware Interface.

Module:6	**Contemporary Issues**	**2 hours**

	Total Lecture hours:	**45 hours**

Text Books	
1	Patrick Schaumont, A Practical Introduction to Hardware/Software Codesign, 2017 reprint, Springer.
2	S. Ha and J. Teich, Handbook of Hardware/Software Codesign. 2018, 1st Edition,Cham, Switzerland: Springer.
Reference Books	
1.	J. Staunstrup and W. Wolf, Hardware/Software Co-Design: Principles and Practice.2007, Springer (India) Pvt. Ltd.
2.	G. De Micheli, R. Ernst, W. Wolf, Readings in Hardware/Software Co-Design, 2002, 1st Edition,Morgan Kaufman.
Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT	

Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS605	**Modern Automotive Electronic Systems**	**3**	**0**	**0**	**3**
Pre-requisite	None	**Syllabus Version**			
		1.0			

Course Objectives

1. Preparing Students to develop various Automatic control systems and Instrumentation involved in Automobiles.
2. Enabling Students to explore various Automobile condition measurement and monitoring Mechanisms.
3. Imparting Students with the knowledge of advanced Electronic elements and their functional aspects in Automobiles.

Course Outcomes

1. Illustrate the Engine Management System and various Ignition and Injection Systems.
2. Interpret the various Automotive Control mechanisms.
3. Examine the implementation of suitable Sensors and Driver Assistance Systems for Automobiles.
4. Explore the various architectures for Hybrid Electric Vehicles and Fuel Cells.

Module:1	**Engine Management Systems**	**12 hours**

Introduction - components for engine management system - Open loop and Closed loop Control system Engine cranking and warm up control Acceleration, deceleration and idle speed control. Injection and Ignition systems : Feedback Carburetor system,Throttle body injection and Multi point Fuel injection system.Injection System controls,Advantage of Electronic Ignition systems,Types of Solid State Ignition Systems and their principles of operation Electronic spark timing control, Exhaust emission control Engineering.

Module:2	**Automotive Control Mechanism**	**7 hours**

Electronic Management of Chassis systems, Vehicle motion control, anti lock braking system, Collision avoidance system, Steering control system, Active suspension system,Traction control system.

Module:3	**Sensors for Transportation**	**8 hours**

Basic Sensor arrangement ,Types of Sensors, Oxygen Sensor , Cranking Sensor, Position Sensors, Engine cooling Water Temperature Sensor, Engine Oil Pressure Sensor, Fuel metering, Vehicle Speed Sensor and Detonation Sensor, Smart Sensors. Use Cases – AI / ML in Transportation.

Module:4	**Automotive Driver Assistance Systems**	**10 hours**

Keyless entry system,Lighting design,Brake actuation warning systems,Infotainment,Speed warning systems, Oil Pressure warning system, Engine over heat warning system, Air Pressure warning system, Tyre Pressure Monitoring System,Safety devices-Wind shield wiper and Washer, SAE J3016 ADAS -J2808_202406 Lane Departure Warning, J2802_202110 Blind Spot Monitoring System,VANET,IEEE 802.11p V2X,On Board Diagnostics,AI powered Driver Assistance System.

Module:5	**Hybrid Electric Vehicles**	**6 hours**

History of EVs,Configurations of Electric Vehicles, Performance of Electric Vehicles, Hybrid Electric Vehicles, Concept of Hybrid Electric Drivetrains, Architectures of Hybrid Electric Drivetrains,Fuel Cells, Operation Principles of Fuel Cells, Fuel Cell Technologies, Fuel Supply.

Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Books

1	Tom Denton, Automobile Electrical and Electronic Systems, 2018, 5th Edition, Routledge,UK.
2	William, B. Ribbens, Understanding Automotive Electronics, 2017,8th Edition,Butter Worth Heinemann, United States.

<table>
<tr><td colspan="4">Reference Books</td></tr>
<tr><td>1.</td><td colspan="3">Robert Bosch Gmph, Automotive Hand Book, 2022, 11th Edition, Wiley, United States.</td></tr>
<tr><td>2.</td><td colspan="3">Mehrdad Ehsani, Yimin Gao, Stefano Longo, Kambiz Ebrahimi, Modern Electric, Hybrid Electric, and Fuel Cell Vehicles,2018,3rd Edition, CRC Press,India.</td></tr>
<tr><td>3.</td><td colspan="3">Andrew Lapthorn, Pedram Asef, Sanjeevikumar Padmanaban,Modern Automotive Electrical Systems,2022,1st Edition,Wiley-Scrivener,United States.</td></tr>
<tr><td>4.</td><td colspan="3">Shanzhi Chen,Jinling Hu,Li Zhao,Rui Zhao,Jiayi Fang,Yan Shi,Hui Xu, Cellular Vehicle-to-Everything (C-V2X) (Wireless Networks),2023,1st Edition,Springer,Singapore.</td></tr>
<tr><td colspan="4">Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT</td></tr>
<tr><td colspan="2">Recommended by Board of Studies</td><td colspan="2"></td></tr>
<tr><td colspan="2">Approved by Academic Council</td><td>No.</td><td>Date</td></tr>
</table>

Course Code	Course Title	L	T	P	C
MAEDS606	Intelligent IoT System Design and Architecture	3	1	0	4
Pre-requisite		Syllabus Version			
		1.0			

Course Objectives

1. Familiarize the foundational concepts and design methodologies of intelligent IoT systems.
2. Impart knowledge on edge computing and standardized communication protocols with security models and privacy challenges in distributed IoT systems.
3. Develop skills for applying intelligent data analytics in cloud-edge integrated IoT platforms and to enable students to design IoT solutions across real-world use cases with a focus on contemporary trends.

Course Outcomes

1. Interpret the core IoT architectural components and reference models.
2. Apply edge computing paradigms and protocols in real-time IoT environments.
3. Illustrate secure communication and data privacy methods in IoT.
4. Use machine learning and analytics tools to build intelligent IoT applications.
5. Implement domain-specific IoT solutions and adapt to emerging technologies.

Module	Title	Hours
Module:1	**Foundations of IoT Systems**	**12 hours**

Evolution of IoT - characteristics - enabling technologies-Planning for an IoT solution - use case development - architecture reference model-Functional blocks of IoT - communication - security model - SOA - event-driven architecture- IoT-based digital twins in system modelling-Semantic web integration in IoT systems.

Module	Title	Hours
Module:2	**Edge and Communication Architectures**	**12 hours**

Edge computing - nodes - gateway - node to edge interfaces-IoT edge architecture and hardware - Communication models - 6LoWPAN - IPv4/IPv6 - MQTT - CoAP - LoRaWAN – RPL-5G integration with edge IoT devices.

Module	Title	Hours
Module:3	**IoT Security, Privacy, and Trust**	**12 hours**

IoT risks - trust models - network access control-Data confidentiality - user authentication/authorization-Blockchain for IoT - secure API practices-Zero Trust Architecture in IoT Security-Federated identity management for distributed IoT-Quantum-resistant cryptographic techniques for IoT.

Module	Title	Hours
Module:4	**Smart Data Analytics for IoT**	**12 hours**

Data generation - pre-processing - handling imbalanced data-Machine learning - deep learning for IoT - predictive analytics-Cloud-based analytics - EDA - big data frameworks - cloud platforms-AutoML for sensor data classification-DataOps for intelligent IoT pipeline automation- TinyML - Edge-to-cloud AI model deployment.

Module	Title	Hours
Module:5	**IoT Applications**	**10 hours**

Smart Healthcare - remote diagnostics, wearables integration, Health Data Analytics and Insights; Precision agriculture - autonomous irrigation, drone surveillance; Supply chain asset tracking; AI-enabled smart cities - traffic prediction, pollution control.

Module	Title	Hours
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**60 hours**

Text Books

1.	Dey, Hassanien, Bhatt, Ashour and Satapathy, Internet of Things and Big Data Analytics towards Next-Generation Intelligence, 2018,1st Edition, Springer.
2.	Arshdeep Bahga, Vijay Madisetti, Internet of Things – A hands-on approach, 2015,1st Edition, Universities Press.

Reference Books	
1.	Adrian McEwen & Hakim Cassimally, Designing the Internet of Things, 2013,1st Edition, Wiley.
2.	Ovidiu Vermesan, Peter Friess, Internet of Things: Converging Technologies for Smart Environments,2013,1st Edition, River Publishers.
3.	Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things – Key Applications and Protocols,2012,2nd Edition, Wiley.
4.	Rajkumar Buyya, Amir Vahid Dastjerdi, Internet of Things: Principles and Paradigms, 2016,1st Edition, Morgan Kaufmann.
5.	Pethuru Raj, Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, 2017,1st Edition, CRC Press.
6.	John R. Vacca, Cloud Computing Security: Foundations and Challenges, 2016,1st Edition, CRC Press.

Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS607	**Fault Tolerance and Dependable Systems**	**3**	**0**	**0**	**3**
Pre-requisite	None	**Syllabus Version**			
		1.0			

Course Objectives

1. Equip students to understand the concepts of fault tolerance and dependability in embedded systems, including fault origins, quantitative measure, and theoretical approaches to improving system reliability.
2. Provide students with the knowledge and skills to model and evaluate the dependability of embedded systems using standardized tools and techniques.
3. Develop in students a comprehensive understanding of hardware and information redundancy techniques and their application in designing fault-tolerant systems.

Course Outcomes

1. Implement key dependability concepts to systematically identify system threats and apply quantitative methods for assessing and improving reliability.
2. Model and analyze system dependability using RBD, FTA, Markov Processes, and FMEA; leverage tools like PyRBD and OpenMarkov for practical dependability analysis.
3. Apply hardware & information redundancy techniques to design fault-tolerant systems compliance with IEC 61508 standards.
4. Implement error detection, correction, and advanced encoding techniques for reliable transmission, ensuring compliance with international safety standards like IEC 61508.

Module:1	**Fundamentals of Dependability**	**8 hours**

Dependability attributes - Reliability, Availbility, Safety, Maintainability, Security; Dependability Threats - Faults, Errors, Failures; Origins of Faults - Specification faults, Design faults, Fabrication faults, External faults; Classification of faults - Duration-based categorization, Predictability-based categorization; Dependability Measures - Failure rate, MTTF, MTTR, MTBF, Fault coverage, Avaialbility Calculation.

Module:2	**Dependability Evaluation Techniques**	**9 hours**

Dependability Modeling Techniques - Reliability Block Diagrams (RBD), Fault Tree Analysis (FTA), Markov Processes, Failure Modes and Effects Analysis (FMEA); Software Tools –OpenMarkov, PyRBD.

Module:3	**Fault Tolerance Design – Hardware**	**9 hours**

Hardware Redundancy Classification – Passive Redundancy - TMR, NMR; Active Redundancy - Duplication with Comparison, Standby Redundancy - Cold Standby, Hot Standby; Pair-And-A-Spare; Hybrid Redundancy - Self-Purging Redundancy, N-Modular Redundancy with Spares; Industry Standards - IEC 61508 Compliance.

Module:4	**Fault Tolerance Design - Information**	**8 hours**

Fundamental Notion - code, encoding, information rate, decoding, hamming distance, code distance; Error Detection and Correction Codes - Parity Codes, Checksums, Cyclic Redundancy Check (CRC), Hamming Codes, BCH and Reed-Solomon Codes.

Module:5	**Fault Tolerance in Interconnects and Distributed Systems**	**9 hours**

Interconnection Network Topologies: mesh, torus, hypercube, and Clos networks; Fault Models in Interconnection Networks – Link Faults, Switch Faults, Fault Tolerance Strategies - Redundant Pathways, Dynamic Reconfiguration - Fault Tolerance in Distributed Systems - Common failure models: crash, omission, timing, and Byzantine failures; Fault Tolerance Mechanisms in Distributed Systems - Consensus algorithms, Checkpointing and rollback recovery methods.

Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Books			
1.	Elena Dubrova, Fault-Tolerant Design, 2013, Springer, Sweden.		
2.	Israel Koren & C. Mani Krishna, Fault-Tolerant Systems, 2nd Edition, 2020, Morgan Kaufmann, Cambridge, MA.		
Reference Books			
1.	Raynal, M. Fault-tolerant message-passing distributed systems: An algorithmic approach, 2018, Springer.		
Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS608	**Parallel Processing and Computing**	**3**	**1**	**0**	**4**
Pre-requisite		**Syllabus Version**			
		1.0			

Course Objectives

1. Equipping students with a comprehensive understanding of parallel computing architectures, models, and the scope and design principles of parallelism.
2. Developing analytical skills to model and evaluate the performance of parallel algorithms in shared memory and distributed memory architectures.
3. Fostering practical skills in GPU programming with CUDA and utilizing GPU-accelerated libraries for high-performance computing applications.

Course Outcomes

1. Develop fundamental skills on parallel architectures, and theoretical frameworks of parallel processing systems and their performance analysis.
2. Design efficient parallel algorithms using shared memory programming models and OpenMP directives.
3. Analyse the performance of distributed applications using Message Passing Interface (MPI) to optimize communication patterns in parallel systems.
4. Simulate kernel-based parallel programming concepts using CUDA to accelerate computationally intensive tasks.
5. Investigate GPU-accelerated libraries for complex computational problems and their performance in high-performance computing applications.

Module:1	Introduction to Parallel Processing	8 hours

Parallel processing – Concepts and Terminology- Parallel Computer Memory Architectures [UMA, NUMA, distributed memory] - Parallel Programming Models - Designing Parallel Programs- Performance Analysis - law, - Theoretical foundations: Amdahl's law, Gustafson's law, speedup, efficiency, and scaling.

Module:2	Shared Memory Programming	13 hours

Shared memory architecture and programming paradigms - Processes and Threads - Scope of Variables and visibility in multithreaded applications – OpenMP directives and programming model -Data parallelism and work sharing constructs - Reduction operations and Clauses – Loop scheduling strategies and performance considerations - – Cache Architectures Cache coherence and False Sharing – Thread Safety and synchronization mechanisms - Parallel algorithm design patterns – Case Studies: Bubble sort, Odd-even transposition sort, Matrix operations.

Module:3	Message Passing Interface	13 hours

Distributed memory programming models - MPI architecture and programming model, Process Management and point-to-point communication, Blocking and non-blocking communication mechanisms, Collective Communication and Optimization, Derived datatypes and communicators, synchronous-asynchronous- send-receive, Implementation and analysis of algorithms for: Gather, Scatter, Broadcast, Reduction and prefix operations, All-to-all communication patterns, Hybrid programming with MPI and OpenMP, Performance analysis and optimization of MPI programs, Case studies: Parallel sorting, N-body simulation, Linear system solvers.

Module:4	Kernel-Based Parallel Programming	13 hours

GPU architecture and computing paradigm - - Introduction to CUDA programming model and Execution model, Data Parallelism and Threads hierarchy - Memory Allocation and Data Movement API- -, Thread scheduling and synchronization - Control flow and divergence management - CUDA memory hierarchy and optimization Parallel patterns implementation: Array and Matrix Multiplication, 1D Stencil, Reduction Operations, Scan Operations - Tiled Matrix Multiplication-Integration with machine learning frameworks: TensorFlow, PyTorch - Case studies and performance analysis.

<table>
<tr><td>Module:5</td><td colspan="3">GPU-Accelerated Libraries</td><td>11 hours</td></tr>
<tr><td colspan="5">Overview of GPU-accelerated computing libraries - Accelerated linear algebra (BLAS) library - Accelerated random number generation (RAND) - Deep learning acceleration - Sparse matrix operations - Fast Fourier Transforms - Parallel algorithms library - Multi-GPU programming and scaling - Mixed precision computing - Profiling and debugging tools for GPU applications - - - Case studies: Deep learning, scientific computing, big data analytics.</td></tr>
<tr><td>Module:6</td><td colspan="3">Contemporary Issues</td><td>2 hours</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td></td><td colspan="3">Total Lecture hours:</td><td>60 hours</td></tr>
<tr><td colspan="5">Text Books</td></tr>
<tr><td>1</td><td colspan="4">AnantaGrama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, 2011, 2nd Edition, Addison Wesley Professional, UK.</td></tr>
<tr><td>2</td><td colspan="4">Pacheco, Peter. An Introduction to Parallel programming, 2021, 2nd Edition, Morgan Kaufmann Publishers, USA</td></tr>
<tr><td colspan="5">Reference Books</td></tr>
<tr><td>1.</td><td colspan="4">Gabriele Jost and Ruud van der Pas, Portable Shared Memory Parallel Programming, 2007, The MIT Press, USA.</td></tr>
<tr><td>2.</td><td colspan="4">William Gropp and Ewing Lusk, Portable Parallel Programming with the Message Passing Interface, 2009, MIT Press, USA.</td></tr>
<tr><td>3.</td><td colspan="4">David B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, 2016, 3rd Edition, Morgan Kaufmann Publishers, USA.</td></tr>
<tr><td>4.</td><td colspan="4">CUDA cuBLAS Release 12.8, NVIDIA Corporation, 2025</td></tr>
<tr><td>5.</td><td colspan="4">CUDA cuTENSOR Release 2.2.0, NVIDIA Corporation, 2025</td></tr>
<tr><td>6.</td><td colspan="4">CUDA cuDNN Release 2.2.2, NVIDIA Corporation, 2025</td></tr>
<tr><td colspan="5">Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT</td></tr>
<tr><td colspan="2">Recommended by Board of Studies</td><td colspan="3"></td></tr>
<tr><td colspan="2">Approved by Academic Council</td><td>No.</td><td>Date</td><td></td></tr>
</table>

Course Code	Course Title	L	T	P	C
MAEDS609	**Edge and Cloud Computing**	**3**	**0**	**0**	**3**
Pre-requisite	**None**	**Syllabus Version**			
		1.0			

Course Objectives

1. Acquaint students with the fundamental principles, architectural models, and enabling technologies of edge and cloud computing for embedded systems.
2. Familiarize learners with cloud services, virtualization techniques, cloud storage solutions, security mechanisms, and disaster recovery strategies.
3. Provide knowledge of edge and cloud platforms along with their services for a wide range of applications.

Course Outcomes

1. Apply the fundamental concepts, architecture and protocols in the edge computing.
2. Illustrate the key technologies of cloud computing, deployment models, virtualization, containerization, and multi-tenant technologies.
3. Develop cloud-based systems by using compute, storage, database, application, and analytics services.
4. Identify edge-to-cloud platforms and security techniques for real-world embedded systems deployments.

Module:1	**Edge Computing Fundamentals**	**8 hours**
Elements of Parallel and Distributed Computing, Introduction to Fog, Edge and Cloud Computing, Edge Computing Architecture, Edge Device management, Communication Protocols at the Edge – RFID, BLE and LPWAN, Edge Machine Learning and Applications, Edge Vs. Cloud.		
Module:2	**Cloud Enabling Technologies**	**9 hours**
Cloud Characteristics, Cloud Delivery Models, Cloud Deployment Models- Public, Private, Hybrid and Multi-cloud environments. Multi-tenant technology, Service APIs, Virtualization- Hypervisor, Containerization– Kubernetes, Dockers, Container Images, Multi-Container Types.		
Module:3	**Cloud Architecture and services**	**9 hours**
NIST Cloud Computing Reference Architecture, Compute Services, Storage Services, Database Services, Application Services, Analytics Services, Cloud Storage Devices, Resource replication, Load Balancer, Failover System, Dynamic Scalability, Elasticity, Disaster Recovery, Virtual Private Cloud, Cost management and optimization.		
Module:4	**Edge and Cloud Security Mechanisms**	**10 hours**
Data security at the Edge – Device ID Registry and Multi-Factor Authentication ,Encryption At-Rest and In-Transit, Public Key Infrastructure, Single Sign-On (SSO), Firewall, Virtual Private Network, Biometric Scanner, Identity and Access Management (IAM), Network Intrusion Monitor, Data Loss Prevention, Trusted Platform Module, Data Backup and Recovery, Activity Log Monitor, SLA Monitor, User specific Access Control, Zero Trust Security Model.		
Module:5	**Edge to Cloud Platforms and Applications**	**7 hours**
Cloud provisioning for Embedded Edge Devices, Edge and Cloud Environments – Google Cloud, Microsoft Azure, Amazon Web Services, IBM Cloud, Cloud service migration, Edge AI Applications of Embedded Systems, Energy Efficiency, Market based Cloud Management, Federated Cloud.		
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Books

1	Erl, Thomas, Eric Barceló Monroy, Zaigham Mahmood, and Ricardo Puttini, Cloud Computing: Concepts, Technology, Security & Architecture,2024, 2nd Edition, Pearson Education, UK.
2	Robert Oshana, Essentials of Edge Computing, 2021, 1st Edition, NXP, Netherlands.

Reference Books	
1.	Lambert Spaanenburg and Hendrik Spaanenburg, Cloud Connectivity and Embedded Sensory Systems, 2011, 1st Edition, Springer, New York.
2.	Rajkumar Buyya and Satish Narayana Srirama, Fog and Edge Computing Principles and Paradigms, 2019, 2nd Edition, Wiley, USA.
3.	Sudeep Pasricha and Muhammad Shafique, Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing: Hardware Architectures,2024, 1st Edition, Springer Nature, Switzerland.
4.	Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, Distributed and Cloud Computing From Parallel Processing to the Internet of Things, 2020,3rd Edition, Elsevier, USA.

Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT

Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS610	**Cyber-Physical Systems**	**3**	**1**	**0**	**4**
Pre-requisite	None	**Syllabus Version**			
		1.0			

Course Objectives

1. Familiarizing with the fundamentals of modelling Cyber-Physical Systems (CPS).
2. Providing an overview of design automation and verification problems with systems perspective for designing, monitoring, and managing large scale infrastructure.
3. Exposing practical applications of modelling and verification to address real-world problems through Cyber-Physical Systems.

Course Outcomes

1.Infer the need and purpose of the different components of CPS.
2. Implement secured and trustworthy prototypes for simple Cyber Physical Systems for real world scenarios.
3.Analyze the performance of CPS systems using mathematical modelling in various fields with parametric measures.
4.Evaluate the design of a CPS that delivers high performance in terms of security, privacy, and data integrity.
5.Investigate intelligent AI components within cyber-physical systems for real-world applications.

Module:1	**Overview of Cyber-Physical Systems (CPS) and its Models**	**12 hours**

CPS: Motivation, overview, requirements, features, Real-time examples.
Design drivers and quality attributes of CPS. Attributes of high confidence CPS, Interface between Physical and Cyber World: Industry 4.0, Digital Twin, AutoSAR, IIOT implications, Building Automation. Basic principles of design and validation of CPS, Relationship between embedded systems and CPS, Design Process- Modeling, Design, Analysis, Continuous Dynamics - Discrete Dynamics. Hybrid Systems - State Machines - Concurrent Models of Computation.

Module:2	**CPS-Implementation, Analysis and Verification**	**12 hours**

CPS Implementation: Sensors and Actuators. Embedded Processors - Memory Architecture - Input and Output, CPS case study- design, RTOS for CPS, Real-time scheduling.
CPS Analysis and Verification: Invariants and Temporal Logic - Equivalence and Refinement. Evaluate the performance. Reachability analysis - Quantitative analysis. CPS Case Study-Analysis, and Verification.

Module:3	**CPS-Networking and Protocol**	**12 hours**

CPS Communications, CPS Network – Wireless Hart, CAN, Automotive Ethernet, Scheduling Real Time CPS tasks, security requirements, intrusion detection, CPS vs WSN, CPS vs IoT, scalability and interoperability, Networking Technologies, Wireless Sensor Networks, Transportation, IPv6-Connected Internet of Things, Machine-to-Machine Communications, PROFINET, Mobile Cloud Computing.

Module:4	**CPS- Security and Trust**	**12 hours**

CPS Security Challenges, risk, Vulnerabilities: hardware, software, and network. Cyberattacks – Quantifying Security & Risk, Trustworthy Operational Readiness –Security Technologies, Data Integrity & Privacy. Security Control and Solutions. Intrusion Detection Systems (IDS), Blockchain for CPS Security, Datagram Transport Layer Security (DTLS), AI-Driven CPS Security: Threat Detection, Intrusion Prevention, Secure Communication. Network security.

Module:5	**AI driven CPS**	**10 hours**

AI architecture for CPS, Perception Layer, Processing Layer: Edge computing for low-latency AI inference, Cloud-based AI models for large-scale data processing; Decision-Making Layer: - Machine learning algorithms for predictive analytics, Reinforcement learning for autonomous control, case studies: Vehicular CPS, Smart Cities, Drones. Smart Manufacturing, Healthcare, Predictive Maintenance.

Module:6	Contemporary Issues		2 hours
		Total Lecture hours:	60 hours
Text Books			
1	Rajeev Alur, Principles of Cyber-Physical Systems,2023,1st Edition, MIT Press.		
2	Edward Ashford Lee and Sanjit Arunkumar Seshia, Introduction to Embedded Systems- A Cyber-Physical Systems Approach, 2019,2nd Edition, PHI Learning PVT LTD.		
Reference Books			
1.	Nikolas Flourentzou, Stella Hadjistassou, and Irina Ciornei, Cyber-Physical Systems Modelling and Simulation, 2022, Riga, RTU Press.		
2.	Ashish Kumar Luhach, Atilla Elçi, Artificial Intelligence Paradigms for Smart Cyber-Physical Systems,2020,1st Edition, IGI Global.		
3.	A Ragunathan Rajkumar, Dionisio de Niz and Mark Klein, Cyber-Physical Systems, 2017,1st Edition, Pearson Education.		
4.	Alexander Romanovsky, Fuyuki Ishikawa (eds,) –Trustworthy Cyber-Physical Systems Engineering,2016, Chapman and Hall CRC.		
5.	Houbing Song, Danda B Rawat, Sabina Jeschke, and Christian Brecher, Cyber-Physical Systems: Foundations, Principles and Applications (Intelligent Data-Centric Systems: Sensor Collected Intelligence), 2016, Academic Press.		
Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS611	**5G and Next Generation Wireless Systems for Embedded Applications**	**3**	**0**	**0**	**3**
Pre-requisite		**Syllabus Version**			
		1.0			

Course Objectives

1. Familiarize the standardization and evolution of 5th Generation wireless networks.
2. Analyze the terminologies and concepts relating to 5G NR use cases and applications.
3. Provide insights into future communication technologies

Course Outcomes

1. Use the 3GPP standards, specifications, key concepts, terminologies of 5G NR for M2M communications.
2. Employ the 5G Interface Architecture for embedded devices.
3. Implement different 5G use cases and intelligent applications.
4. Interpret the Next Generation networks for embedded system design.

Module	Title	Hours
Module:1	**Overview of 5G Technologies**	**9 hours**

Evolution of Wireless Technologies (1G to 4G), Need for 5G- Spectrum for 5G – 5G Standardization: 3GPP and IMT2020. IMT-2020 Requirements for embedded and IoT applications. Comparison of Channel Access methods, 5G Network Terminology and concepts – 5G Call flow, Bandwidth Parts (BWP), 5G deployment - Options, Challenges in embedded environment.

Module	Title	Hours
Module:2	**M2M Communications**	**7 hours**

LTE evolution for M2M, 5G for M2M communication, 5G Physical Channels, Low-latency Radio-interface Perspectives for Small-cell 5G Networks, Massive Internet of Things, Narrow Band IoT.

Module	Title	Hours
Module:3	**5G NR Architecture for Embedded devices**	**11 hours**

Overall system architecture for embedded devices – Open RAN, Radio Protocol Architecture, QoS handling, User Plane Protocols-Radio Link Control - Medium-Access Control. Network Slicing. Massive MIMO (mMIMO), mmwave and Beamforming Concepts. Advanced Multiple-access and MIMO Techniques – Orthogonal Time Frequency Space and Non-Orthogonal Multiple Access in Embedded Signal Processors.

Module	Title	Hours
Module:4	**5G NR Use Cases and Intelligent applications**	**9 hours**

5G NR Service Classes Overview: Enhanced Mobile Broadband (eMBB), Massive Machine-Type Communications (m-MTC), Ultra Reliable Low Latency Communications (URLLC), Application of NFV and SDN to 5G Infrastructure. 5G Use Cases: Smart Healthcare - remote diagnostics, wearables integration; Precision agriculture - autonomous irrigation, drone surveillance; AI-enabled smart cities - traffic prediction, pollution control.

Module	Title	Hours
Module:5	**Next Generation Communication Systems**	**7 hours**

6G Technology trends- NR beyond 52.6 GHz, IAB enhancements, NR – Broadcast and Multicast General enhancements, 6G Spectrum; 6G Key Research Areas - Reconfigurable intelligent surfaces, Tera Hertz Communications, Quantum Communication.

Module	Title	Hours
Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Books

1	Erik Dahlman, Stefan Parkvall, Johan Skold,5G NR: The Next Generation Wireless Access Technology, 2018, 1st Edition, Academic Press.
2	Paulo Sergio Rufino Henrique and Ramjee Prasad, 6G: The Road to the Future Wireless Technologies 2030, 2021, 1st Edition, River Publishers.

Reference Books

1.	Robert W. Heath Jr., Angel Lozano, Foundations of MIMO Communication,2019, 1st Edition Cambridge University Press.

2.	Long Zhao, Hui Zhao, Kan Zheng, Wei Xiang, Massive MIMO in 5G Networks: Selected Applications, 2018,1st Edition, Springer.
3.	Jonathan Rodriguez, Fundamentals 5G Mobile Networks,2015,1st Edition, John Wiley & Sons.
4.	R. Vannithamby and S. Talwar, Towards 5G: Applications, Requirements and Candidate Technologies, 2017,1st Edition, John Willey & Sons.
5.	Saad Z. Asif, 5G Mobile Communications Concepts and Technologies, 2019,1st Edition CRC Press.

Mode of Evaluation: Quiz, Assignment, Design Project, CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS612	**Machine Learning and Deep Learning**	**3**	**1**	**0**	**4**
Pre-requisite	**None**	**Syllabus Version**			
		1.0			

Course Objectives

1.Familiarizing students with the fundamental concepts of machine learning and neural networks.
2.Enabling the students to acquire knowledge about pattern recognition.
3.Motivating the students to apply deep learning algorithms for solving real life problems.

Course Outcomes

1.Evaluate different paradigms of machine learning to determine their suitability for embedded systems.
2.Interpret the architecture and training mechanisms of neural network models suitable for embedded deployment.
3.Apply feature selection, dimensionality reduction, and classification techniques for efficient pattern recognition.
4.Analyze the design and performance of deep learning models including CNNs, RNNs, and autoencoders in embedded systems.
5.Demonstrate the use of generative models and deployment frameworks in real-world embedded machine learning applications.

Module:1	**Foundations of Machine Learning in Embedded System**	**12 hours**

Various paradigms of learning problems, Forms of Learning: Supervised, Semi-supervised, and Unsupervised algorithms, Reinforcement Learning, Machine Learning Terminologies and Model Evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1-Score, the curse of dimensionality, training, testing, validation, cross-validation, overfitting, underfitting, early stopping, regularization, bias and variance, Introduction to TinyML and Edge AI, Challenges in real-time ML deployment :latency, memory, and power.

Module:2	**Neural Networks and Model Training**	**12 hours**

Differences between Biological and Artificial Neural Networks - General Architecture, Multi-layer neural network, Linear Separability, Hebb Net, Perceptron, Adaline, Standard Back propagation, Training Algorithms for Pattern Association - Hebb rule and Delta rule, Hetero associative, Auto associative, Kohonen Self Organising Maps, Learning Vector Quantization, Gradient descent, Boltzmann Machine Learning, Model training vs inference on embedded systems, Lightweight Neural Network deployment strategies -quantization, and pruning.

Module:3	**Advanced Machine Learning Techniques**	**12 hours**

Feature Engineering, Dimensionality Reduction, Classifiers: K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Decision Trees, Naïve Bayes, Binary Classification, Multi-class Classification, Clustering, Ensemble learning, Meta Learning, Foundation to Quantum Machine Learning, Low-power classification methods for Embedded controllers.

Module:4	**Deep Learning Architectures**	**12 hours**

Convolutional Neural Networks: Convolution layers, Pooling layers, Fully connected layers, Advanced CNN models: Alexnet, VGGnet, ResNet, Google net, Handling overfitting in CNN: drop out, transfer learning, data augmentation, Recurrent Neural Networks and Autoencoders, State, Structure of RNN Cell, LSTM and GRU, Time Distributed Layers, Autoencoders: Convolutional Autoencoders, Denoising Autoencoders, Variational Autoencoders, Lightweight DL architectures for embedded systems: SqueezeNet, MobileNetV3, and Tiny-YOLO.

Module:5	**Generative Models, Deployment and Real-World Applications**	**10 hours**

Generative Adversarial Networks (GANs): The Discriminator, Generator, Deep Convolutional GANs (DCGANs), Introduction to BERT Transformer, Real-world applications in NLP, Computer Vision, Healthcare, and Finance, Fundamentals of Federated Learning, Federated Learning in IoT-based embedded networks, Deployment frameworks: TensorFlow Lite, PyTorch Mobile.

Module:6	**Contemporary Issues**	**2 hours**
	Total Lecture hours:	**60 hours**

Text Books	
1	ShaiShalev-Shwartz and Shai Ben-David, Understanding Machine Learning, 2017,1st Edition, Cambridge University Press.
2	Ian Good fellow, Yoshua Bengio and Aaron Courville, Deep Learning, 2016, MIT Press, ISBN: 9780262035613.
Reference Books	
1.	J. S. R. Jang, C. T. Sun, E. Mizutani, Neuro Fuzzy and Soft Computing - A Computational Approach to Learning and Machine Intelligence, 2012,1st Edition, PHI learning.
2.	Trevor Hastie, Robert Tibshirani and Jerome Friedman, The Elements of Statistical Learning, 2009,2nd Edition, Springer Series.
3.	P. Warden and D. Situnayake , TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, 2020, CA: O'Reilly Media Publishers.
4.	Christopher M. Bishop, Pattern Recognition and Machine Learning, 2016, Springer, ISBN: 978-0-387-31073-2.
5.	Christopher M. Bishop and Hugh Bishop, Deep Learning: Foundations and Concepts, 2023, Springer Nature.

Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	

Course Code	Course Title	L	T	P	C
MAEDS613	**Wireless and Mobile Communication for Embedded Applications**	**3**	**0**	**0**	**3**
Pre-requisite	**Nil**	**Syllabus Version**			
		1.0			

Course Objectives

1. Familiarize the fundamental concepts of cellular mobile communication systems, their evolution, and techniques to improve coverage and capacity.
2. Evaluate mobile radio propagation models, explore advanced multiple access techniques, and examine MIMO systems that are crucial for next-generation wireless communications.
3. Explore various wireless technologies and their suitability for enhancing embedded system Functionality.

Course Outcomes

1. Analyse the evolution of cellular mobile communication systems from 1G to 5G and cellular concepts for system performance improvement.
2. Apply knowledge of mobile radio propagation models and fading effects to assess their impact on wireless communication channels.
3. Design advanced multiple access techniques, MIMO systems, and intelligent cell concepts for next-generation wireless networks and in-building communication systems.
4. Analyze various wireless technologies to determine their suitability for different embedded applications, considering factors such as efficiency, compatibility, and performance constraints.

Module:1	Cellular Mobile Evolution and Concepts	9 hours

Cellular Mobile Communication Evolution- 1G to 5G Evolution, Wireless Standards. Cellular concept- Frequency reuse – Channel assignment strategies – Handoff strategies – Interference & system capacity – Trunking & Grade of service – Improving coverage and capacity in cellular system.

Module:2	Mobile Radio Propagation Models	9 hours

Basic Propagation mechanism - Free Space Propagation Model – Ground Reflection -Two Ray model – Outdoor Propagation Models: Okumura Model, Hata Model Indoor Propagation Model: Attenuation Factor Model. Parameters of mobile multipath channels – Types of small scale fading – Fading effects due to Multipath time delay spread and Doppler spread

Module:3	Advance Multiple access and MIMO techniques	9 hours

MIMO (Multiple Input-Multiple Output) – Concept of Spatial domain multiple access- Moving to 5G Cellular with Large-scale Antenna Arrays, Antenna-array Architectures for 5G Cellular - MIMO Capacity. Massive MIMO (mMIMO) and Beamforming Concepts. Advanced Multiple access and MIMO Techniques – NOMA, successive inference cancellation for NOMA

Module:4	Intelligent Cell Concept and Applications for next generation wireless mobile communications	9 hours

Intelligent Cell: Power-Delivery Intelligent Cells - Processing-Gain Intelligent Cells - Intelligent Microcell Systems. In-Building Communication: Natural In-Building Radio Environment - In-Building Communication System - In-Building System Configuration. Machine Learning for Wireless Resource Management: Predictive Resource Allocation - AI-based Channel Modeling - Deep Learning for Signal Processing. Intelligent Network for Wireless Communications: Advanced Intelligent Network (AIN) - AIN Elements - AIN Interfaces - Machine Learning Algorithms for Network Optimization and Energy Efficiency.

Module:5	Wireless Technologies for Embedded Applications	7 hours

Short Range: PANs- Bluetooth-Classic and Low Energy - BLE, Zigbee, Z-Wave and NFC (Near Field Communication). Medium Range: LANs and Wi-Fi -WLAN. Long Range – WAN-LPWANs-LoRaWAN, NB-IoT and LTE-M (Long Term Evolution for Machines), Sigfox and 5G NR, Design and simulation of Wireless networks.

Module:6	Contemporary Issues	2 hours

			Total Lecture hours:	45 hours
Text Books				
1.	Randy L. Haupt, Wireless Communications Systems: An Introduction, January 2020, 1st Edition, Wiley-IEEE Press.			
2.	T.S. Rappaport, Wireless Communication -Principle and Practice, 2012,2nd Edition, Pearson Education Asia Ltd., Prentice Hall.			
Reference Books				
1.	R. Vannithamby and S. Talwar, Towards 5G: Applications, Requirements and Candidate Technologies, 2017, 1st Edition, John Willey & Sons.			
2.	Schiller, Mobile Communications, 2008,2nd Edition, Pearson Education Asia Ltd.			
3.	Saad Z. Asif, 5G Mobile Communications Concepts and Technologies, 2019, 1 st Edition, CRC Press.			
4.	Yonina C. Eldar, Andrea Goldsmith, Deniz Gündüz and H. Vincent Poor, Machine Learning and Wireless Communications, 2022, 1st Edition, Cambridge university press.			
5.	W.C.Y. Lee, Wireless and Cellular Telecommunications, 2006,3rd Edition, McGraw Hill			
6.	Robert W. Heath Jr., Angel Lozano, Foundations of MIMO Communication, 2019, 1st Edition, Cambridge University Press.			
Mode of Evaluation: Quiz, Assignment, Design Project,CAT and FAT				
Recommended by Board of Studies				
Approved by Academic Council	**No.**	**Date**		