NM 100

MCET
An Autonomous Institution
Since 2011

Dr. MAHALINGAM
COLLEGE OF ENGINEERING AND TECHNOLOGY
Udumalai Road, Pollachi, Coimbatore District - 642003
Established in 1998 ♦ Approved by AICTE ♦ Affiliated to Anna University
(A DIVISION OF NIA EDUCATIONAL INSTITUTIONS)

NAAC A++ GRADE
Cycle 3 (2023-2030)
The Highest Grade

Curriculum and Syllabi

Semesters I to IV

Regulations 2024

Programme: M.E Embedded System Technologies
Curriculum and Syllabi: Semesters –
Approved by Academic Council on -

Action	**Responsibility**	**Signature of Authorized Signatory**
Designed and Developed by	BoS Electrical and Electronics Engineering	
Compiled by	Office of the Controller of Examinations	
Approved by	Principal	

Dr. Mahalingam College of Engineering and Technology, Pollachi – 642003.
(An autonomous institution approved by AICTE and affiliated to Anna University)

Department of Electrical & Electronics Engineering

Vision

We develop globally competent Electrical and Electronics Engineer to solve real time problems of the industry and society and conduct research for the application of knowledge to the society

Mission:

In order to foster growth and empowerment, we commit ourselves to

- Develop electrical and electronics engineers of high caliber to meet the expectations of industries through effective teaching-learning process
- Improve career opportunities in core areas of electrical and electronics engineering.
- Inculcate leadership qualities with ethical and social responsibilities

Dr. Mahalingam College of Engineering and Technology, Pollachi – 642003.

(An autonomous institution approved by AICTE and affiliated to Anna University)

Programme: M.E. Embedded System Technologies

Programme Educational Objectives (PEOs) - Regulation 2024

After 2 to 3 years of completion of the programme the graduates will be able to:

PEO1. Develop solutions to real world problems in the frontier areas of Embedded System Technologies.

PEO2. Adapt to the latest trends in technology through self-learning and to pursue research to meet out the demands in industries and Academia.

PEO3. Exhibit leadership skills and enhance their abilities through lifelong learning.

Programme Outcomes (POs) - Regulations 2024

On successful completion of the programme the graduates will be able to:

PO1 Carry out research /investigation and development work independently in solving practical problems

PO2 Write and present a substantial technical report/document

PO3 Demonstrate a degree of mastery over the area of Embedded System Technologies.

PO4 Practice professional ethics in multidisciplinary environment with a desire for life-long learning

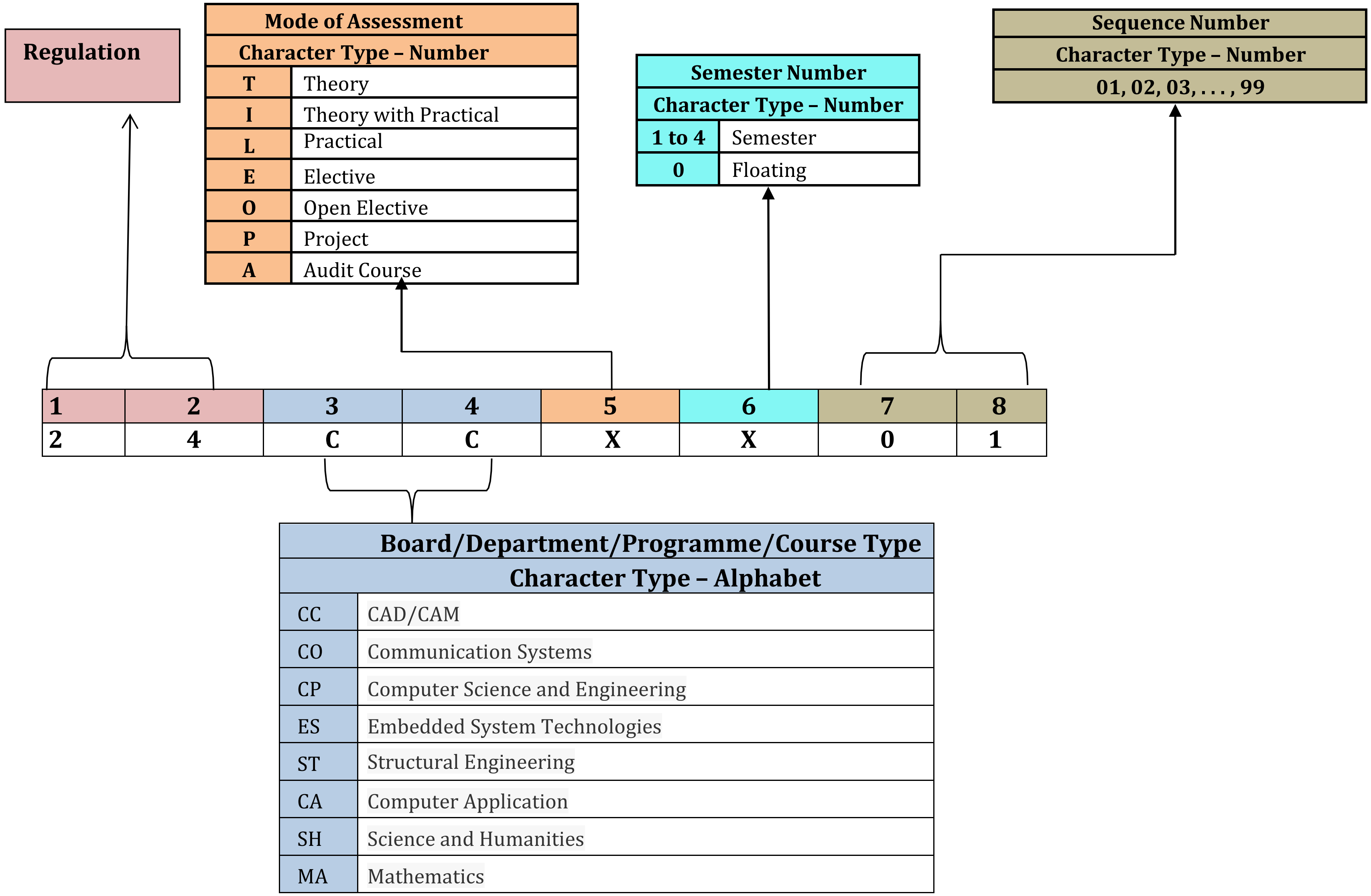
Programme Specific Outcomes (PSOs) - Regulations 2024

On successful completion of the programme the graduates will be able to:

PSO 1. Evaluate the design and provide optimal solutions to problem areas in Embedded System Technologies.

PSO2. Develop electronic systems using modern engineering hardware and software tools

Dr. Mahalingam College of Engineering and Technology, Pollachi
2024 Regulations - Course Code Generation Procedure for PG Courses



Regulation

Mode of Assessment	
Character Type – Number	
T	Theory
I	Theory with Practical
L	Practical
E	Elective
O	Open Elective
P	Project
A	Audit Course

Semester Number	
Character Type – Number	
1 to 4	Semester
0	Floating

Sequence Number
Character Type – Number
01, 02, 03, . . . , 99

1	2	3	4	5	6	7	8
2	4	C	C	X	X	0	1

Board/Department/Programme/Course Type	
Character Type – Alphabet	
CC	CAD/CAM
CO	Communication Systems
CP	Computer Science and Engineering
ES	Embedded System Technologies
ST	Structural Engineering
CA	Computer Application
SH	Science and Humanities
MA	Mathematics

M.E EMBEDDED SYSTEM TECHNOLOGIES

2024 Regulations

Semester I

Course Code	Course Title	Hours/Week			Credits	Marks	Common to Programmes
		L	T	P			
24EST101	Embedded Controllers and Applications	3	0	0	3	100	-
24EST102	Real-Time Embedded Systems	3	0	0	3	100	-
24EST103	FPGA based System Design	3	0	0	3	100	-
24ESEXXX	Professional Elective – I	3	0	0	3	100	-
24CCT101	Research Methodology and IPR	3	0	0	3	100	All
24ESL101	Embedded System Design Laboratory	0	0	4	2	100	-
24ESL102	Object Computing and Data Structures Laboratory	0	0	4	2	100	
24SHA101	English for Research Paper Writing(Audit Course)	2	0	0	-	100	All
Total		**17**	**0**	**8**	**19**	**800**	**-**

Semester II

Course Code	Course Title	Hours/Week			Credits	Marks	Common to Programmes
		L	T	P			
24EST201	Real-Time Operating Systems	3	0	0	3	100	-
24EST202	Linux Architecture and Device Drivers	3	0	0	3	100	-
24EST203	Embedded System Networks	3	0	0	3	100	-
24ESEXXX	Professional Elective II	3	0	0	3	100	-
24ESEXXX	Professional Elective – III	3	0	0	3	100	-
24ESL201	Real-Time Systems Laboratory	0	0	4	2	100	-
24ESL202	Research Paper seminar	0	0	2	1	100	-
24SHA201	Teaching and Learning in Engineering	0	0	4	-	100	All
Total		**15**	**0**	**10**	**18**	**800**	**-**

Semester III

Course Code	Course Title	Hours/Week			Credits	Marks	Common to Programmes
		L	T	P			
24ESEXXX	Professional Elective – IV	3	0	0	3	100	-
24ESEXXX	Professional Elective – V	3	0	0	3	100	-
24ESOXXX	Open Elective/Online Course	3	0	0	3	100	-
24ESP301	Project – I	0	0	20	10	200	-
Total		**9**	**0**	**20**	**19**	**500**	**-**

Semester IV

Course Code	Course Title	Hours/Week			Credits	Marks	Common to Programmes
		L	T	P			
24ESP401	Project – II	0	0	32	16	400	-
Total		**0**	**0**	**32**	**16**	**400**	**-**

Total Credits: 72

Professional Electives

Course Code	Course Title	Hours/Week			Credits	Marks
		L	T	P		
24ESE001	Multi-core Embedded Systems	3	0	0	3	100
24ESE002	Advanced Embedded Controllers	3	0	0	3	100
24ESE003	Automotive Embedded Systems	3	0	0	3	100
24ESE004	Automotive Software Architecture	3	0	0	3	100
24ESE005	Internet of Things	3	0	0	3	100
24ESE006	Python Programming	3	0	0	3	100
24ESE007	Artificial Intelligence	3	0	0	3	100
24ESE008	Industrial Networking and Standards	3	0	0	3	100
24ESE009	Cryptography and Network Security	3	0	0	3	100
24ESE010	Advanced Digital Signal Processing	3	0	0	3	100
24ESE011	Optimization Techniques	3	0	0	3	100
24ESE012	Intelligent Controllers	3	0	0	3	100
24ESE013	Machine Learning	3	0	0	3	100
24ESE014	Advanced Digital Systems Design	3	0	0	3	100
24ESE015	Industrial Drives for Automation	3	0	0	3	100

Semester I

Course Code: 24EST101		Course Title: Embedded Controllers and Applications	
Course Category: FC		Course Level: Mastery	
L:T:P(Hours/Week) 3:0:0	Credits: 3	Total Contact Hours:	Max Marks:100

Course Objectives:

To provide a comprehensive understanding of microcontroller architectures, programming, and real-world interfacing techniques, with a focus on the 8 bit microcontroller and ARM7 processors.

Module I **22 Hours**

8051: 8051 microcontroller – Architecture – Instruction sets – Addressing modes – I/O ports – Timer/Counter – Serial Communication – Interrupts – Assembly language programming.

ARM7: Architecture overview - RISC processor design: ARM Architecture – Programming Model, Pipelined data path design - Pipeline Hazards, Addressing Modes, -Processor modes – Data types – Registers – Program status registers – Floating Point data processing, Interrupts & Exception Handling-– Simple programs.

Module II **23 Hours**

ARM Programming: ARM Instruction Set – Thumb Instruction Set - DSP Extensions, Mixed C and Assembly programming, AMBA bus system Peripherals, SoC design using ARM core, Debug support, Memory system design- Cache Memory, Memory Management unit – Virtual Memory. ARM advanced CPU cores, Applications development using Keil IDE.

REAL WORLD INTERFACING: Master Synchronous Serial Port ((MSSP) structure - Detail study of UART, SPI, I2C, ADC and Comparators, serial port - ADC using I2C. - RTC using I2C. – Design of data acquisition System - frequency counter with display on LCD - Digital Multimeter - DC motor control using PWM with signal.

Course Outcomes	Cognitive Level
At the end of this course, students will be able to:	
CO1: Describe the architecture and instruction sets of 8 bit microcontrollers and ARM7 processors, including their addressing modes and hardware	Understand
CO2 Develop and debug assembly language programs for the 8 bit microcontroller and ARM7 processor using appropriate programming models and instruction sets.	Apply
CO3: Design and implement systems using ARM cores, including mixed C and assembly programming, memory management, and SoC design with the AMBA bus system.	Apply
CO4: Implement real-world interfacing techniques using UART, SPI, I2C, ADC, and PWM, and develop applications like data acquisition systems, frequency counters, and motor control systems.	Apply

Reference Book(s):

1. M.A. Mazidi, J.G. Mazidi and R.D. McKinlay, ‘The 8051 microcontroller and embedded systems’, Prentice Hall India, 2nd Edition, New Delhi, 2013.
2. William Hohl and Christopher Hinds ‘ARM Assembly Language Fundamentals and Techniques, CRC Press, second edition, 2015
3. Andrew Sloss Dominic Symes Chris Wright, ‘ARM System Developer's Guide, Designing and Optimizing System Software’,1st Edition, 2004
4. David A. Patterson and John L. Hennessy, “Computer Organization and Design – The Hardware/Software Interface”, ARM Edition, Morgan Kaufmann Publisher, 2010
5. Steve Furber, “ARM System-on-Chip Architecture”, Pearson India, 2015.

Course Code: 24EST102		**Course Title: Real-Time Embedded Systems**	
Course Category: PCC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits:3**	**Total Contact Hours: 60**	**Max Marks:100**

Course Objectives:

To equip students with the skills and knowledge required for embedded software development, real-time operating systems, and inter-task synchronization and communication, with hands-on experience in programming and tool utilization.

Module I **22 Hours**

PROGRAMMING LANGUAGE AND TOOLS FOR EMBEDDED SOFTWARE DEVELOPMENT: Fundamentals of Embedded Systems – Embedded Software Development Process: Programming Languages - Embedded C Building Blocks – Mixing of Assembly and C – Pre-processor - Compiler – Assembler - Linker and Loader - Cross Platform Development -Compiler Optimization Techniques – Executable File Formats-Concept of Make Utility - Super Loop based Design Approach

REAL-TIME OPERATING SYSTEMS: Basic Terminologies of Real-Time Embedded Systems – Concepts of OS-based Software Development – Real-Time Operating Systems: Definition, Characteristics and Structure – Task Management: Definition, Classification, Structure, States, and Scheduling – Concept of Pseudo Multitasking and True Multitasking

Module II **23 Hours**

INTER-TASK SYNCHRONIZATION AND COMMUNICATION: Critical Sections – Atomic Operation – Concept of Reentrancy – Semaphores – Event Flag Registers - Inter-task Communication Methods: Shared Memory Technique, Mailbox, Message Queues, and Pipes – Common Design Problems: Premature Task Deletion, CPU Starvation, Deadlocks, and Unbounded Priority Inversion

INTERRUPT MANAGEMENT, I/O SUBSYSTEMS AND MEMORY MANAGEMENT: Exceptions and Interrupts – Processing of Exceptions and Interrupts – I/O Sub-systems – Memory Management – Dynamic Memory Allocation and Fixed-size Memory Allocation in Embedded Systems - Application Modularization for Concurrency: Outside-In Approach – UML Diagrams- Design Examples

Course Outcomes	**Cognitive**
At the end of this course, students will be able to:	
CO1: Apply the principles of software development processes to embedded systems, including the usage of assembly code, embedded C, and various development tools.	Apply
CO2: Implement real-time operating systems with task management and scheduling techniques.	Apply
CO3: Utilize inter-task synchronization and communication methods to solve design problems.	Apply
CO4: Develop embedded systems with efficient interrupt management, I/O subsystems, and memory management strategies.	Apply

Reference Book(s):

1. Qing Li, “Real-Time Concepts for Embedded Systems”, CMP Books, 2003.
2. Insup Lee, Joseph Leung, and Sang Son, “Handbook of Real-Time Systems”, Chapman and Hall, 2008.
3. David E. Simon, “An Embedded Software Primer”, Pearson, 2002.
4. Albert Cheng, “Real-Time Systems: Scheduling, Analysis and Verification”, Wiley Interscience, 2002.
5. Bernd Bruegge, Allen Dutoit, “Object-oriented Software Engineering – Using UML, Patterns and Java”, Prentice Hall, USA, 2010

Course Code: 24EST103		Course Title: FPGA Based System Design	
Course Category: PCC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits:3**	**Total Contact Hours: 60**	**Max Marks:100**

Course Objectives:

To provide students with a solid understanding of digital logic circuits, programmable logic devices, and FPGA-based embedded system design, with practical experience in Verilog programming and system implementation.

Module I **22 Hours**

Review of Digital Logic Circuits: Designing combinational circuit using multiplexer, decoder – Finite State Machines – Mealy Machine- Moore Machine – State Diagram – State table - Design of state machines using Algorithmic State Machines (ASM) chart as a design tool. System Design using PLDs: Basic concepts – Programming technologies - Programmable Logic Element (PLE) - Programmable Array Logic (PLA) - Programmable Array Logic (PAL) – Programmable Logic Architectures – 16L8 – 16R4 – 22V10–Design of combinational and sequential circuits using PLDs.

VERILOG: Signals, Identifier , Net and variable types, Operators, Gate instantiations, Modules and ports, data flow, gate level, Behavioural level ,Switch level and state machine modelling , Concurrent and procedural statements, UDP, sub circuit parameters, function and task, timing and delays - test benches-- design of combinational and sequential circuits using Verilog.

Module II **23 Hours**

CPLD and FIELD PROGRAMMABLE GATE ARRAYS: Complex PLDs (CPLDs) –Xilinx cool runner architecture. Types of FPGA - Xilinx XC4000 series - Logic Cell Array (LCA) – Configurable Logic Blocks (CLB) - Input/output Blocks (IOB) - Programmable Interconnection Point(PIP)Implementing Functions in FPGAs Dedicated Memory in FPGAs – Dedicated Multipliers in FPGAs - Mapping, Placement, and Routing - Verilog based design flow for FPGA.

EMBEDDED SYSTEM DESIGN: FPGA based Embedded Processor - Design Re-use Using On-chip Bus Interface – Creating a Customized Microcontroller - Robot Axis Position Control - FPGA-based Signal Interfacing and Conditioning – Motor Control Using FPGA

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Design the combinational and sequential circuits using digital logic and state machine concepts.	Apply
CO2: Develop the circuits using Programmable Logic Devices (PLDs) such as PLE, PLA, and PAL.	Apply
CO3: Model the combinational and sequential circuits in Verilog at various abstraction levels.	Apply
CO4: Implement the FPGA-based embedded systems, including CPLD architecture, FPGA design flow, and application-specific designs.	Apply

Reference Book(s):

1. Charles H Roth and Lizy Kurian John “Digital Systems Design Using VHDL,” Cengage Learning,2013.
2. Bhaskar J., “ A Verilog Primer”, Prentice Hall of India learing,2012.
3. Samir Palnitkar“ Verilog HDL : A Guide to Digital Design and Synthesis” Pearson Education Asia 2014. .
4. MichealD.Ciletti, “Advance Digital Design with the verilog HDL”, Prentice Hall of India learing,2012.
5. Wayne Wolf,” FPGA - Based System Design” Prentice Hall, New Jersey, 2012.
6. Ming-Bo Lin, ―Digital System Designs and Practices: Using Verilog HDL and FPGAs‖ , Willey Indian Edition, 2012.
7. Rahul Dubey, "Introduction to Embedded System Design Using Field Programmable Gate Arrays" Springer- Verlag London Limited, 2009.

Course Code: 24CCT101		**Course Title: Research Methodology and IPR**	
Course Category: RMC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

The course is intended to impart knowledge on various research types and problem identification. It emphasizes selecting experiment designs, data types, tools, and data analysis processes. Additionally, it covers interpreting and presenting results effectively. The syllabus also delves into intellectual property rights, including types and procedures, and provides hands-on experience in executing patent filing and licensing.

Module I **22 Hours**

Overview of Research Methodology

Research methodology – definition, mathematical tools for analysis, Types of research, exploratory research, conclusive research, modeling research, algorithmic research, Research process. Data collection methods- Primary data – observation method, personal interview, telephonic interview, mail survey, questionnaire design. Secondary data- internal sources of data, external sources of data.

Attitude measurements, Scales and Sampling methods

Scales – measurement, Types of scale – Thurstone's Case V scale model, Osgood's Semantic Differential scale, Likert scale, Q- sort scale. Sampling methods- Probability sampling methods – simple random sampling with replacement, simple random sampling without replacement, stratified sampling, cluster sampling. Non- probability sampling method – convenience sampling, judgment sampling, quota sampling.

Module II **23 Hours**

Hypotheses testing

Hypotheses testing – Testing of hypotheses concerning means (one mean and difference between two means -one tailed and two tailed tests)

Report Writing and Presentation

Report writing- Types of report, guidelines to review report, typing instructions, oral presentation

Patenting

Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO 1. Describe the overview of research methodology	Understand
CO 2. Explain the attitude measurements, scales and sampling methods	Understand
CO 3. Apply hypotheses testing in research problem	Apply
CO 4. Elucidate the research report writing and presentation effectively	Understand
CO 5: Apply patent and copyright for their innovative works	Apply

Reference Book(s):

1. Panneerselvam, R., Research Methodology, Prentice-Hall of India, New Delhi, 2004.
2. Kumar, Ranjit, , “Research Methodology: A Step by Step Guide for beginners”, London Sage: Publications, 2005.
3. Halbert, “Resisting Intellectual Property”, Taylor & Francis Publications ,2007.
4. Robert P. Merges, Peter S. Menell, Mark A. Lemley, “ Intellectual Property in New Technological Age”, Clause 8 Publishing, 2016.
5. T. Ramappa, “Intellectual Property Rights Under WTO”, S. Chand Publications, 2008.

Course Code: 24ESL101		**Course Title: Embedded System Design Laboratory**	
Course Category: PCC		**Course Level: Practice**	
L:T:P(Hours/Week) 0:0:4	**Credits:2**	**Total Contact Hours: 60**	**Max Marks:100**

Course Objectives:

To equip students with comprehensive knowledge and practical skills in embedded systems, covering microcontroller architecture, programming, interfacing, and FPGA-based application development.

List of Exercises

List of experiment using 8 bit microcontroller

1. Lamp control using Timer/Counter using
2. Direction and Speed control of DC motor
3. USART - Transmission and Reception of a byte using on chip serial port
4. Controlling PWM period with analogue input (POT)

List of experiment using LPC2148

5. LED and Switch interfacing
6. Keypad and LCD interfacing
7. Waveform generation using DAC
8. Read the temperature sensor value using ADC
9. Direction and Speed control of Stepper motor
10. Displaying the Date and Time using RTC
11. USART - Transmission and Reception of a byte using on chip serial port
12. Data Transmission and Reception using I2C
13. Data Transmission and Reception using SPI

List of experiment using STM32 microcontroller and FPGA board

14. CAN Interfacing using Loopback mode using STM32 microcontroller
15. Design and implementation of simple experiment using FPGA Board

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Demonstrate proficiency in programming on-chip peripherals of 8 bit microcontroller and ARM7 microcontrollers for diverse embedded applications.	Apply
CO2: Develop skills in interfacing sensors and actuators with microcontrollers and effectively managing interrupts and low power modes.	Apply
CO3: Design and implement simple embedded applications using FPGA platforms, applying theoretical knowledge to practical scenarios.	Apply

Reference Book(s):

1. Muhammad Ali Mazidi, RolinD. Mckinlay, Danny Causery, “PIC Microcontroller and Embedded systems using assembly and C PIC18”,Pearson international edition,2008
2. Andrew Sloss Dominic Symes Chris Wright, ‘ARM System Developer's Guide, Designing and Optimizing System Software’,1st Edition, 2004

   .

Course Code: 24ESL102		**Course Title: Data Structures Laboratory**	
Course Category: PCC		**Course Level: Practice**	
L:T:P(Hours/Week) 0:0:4	**Credits: 2**	**Total Contact Hours: 60**	**Max Marks:100**

Course Objectives:

This course aims to provide a solid foundation in data structures, enabling students to implement basic to advanced programming concepts and data manipulation technique

List of Exercises

1. Programs using arrays.
2. Implementation of various sorting algorithms.
3. Implementation of Stacks using array.
4. Application of Stack
5. Implementation of queue using array.
6. Implementation of recursion for problem solving
7. Implementation of Linked Lists: Singly linked and doubly linked
8. Travel salesman Problem
9. 8 Queens Problem
10. Automatic vending machine

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Comprehend various data structures including arrays, stacks, queues, and linked lists, and their applications.	Apply
CO2: Apply data structures and sorting algorithms to solve challenges in the actual world to strengthen problem-solving abilities.	Apply

Reference Book(s):

1. R.C.T. Lee, S.S Tseng R.C Chang and Y.T. Tsai, “Introduction to the design and Analysis of algorithms A strategic Approach”, Tata McGraw Hill, 2012.
2. Data Structures Laboratory Manual.

Course Code: 24SHA101		**Course Title: English for Research Paper Writing (common to all PG Programmes)**	
Course Category: Audit Course		**Course Level: Introductory**	
L:T:P(Hours/Week) 2:0:0	**Credits: -**	**Total Contact Hours: 30**	**Max Marks:100**

Course Objectives:

The course is intended to enhance the language skills concerning research paper writing and to explain the crucial role of technology in enhancing the quality and credibility of research.

Module I **15 Hours**

Foundations of Academic English in Research: Academic English - Key Language Aspects - Clarity and Precision - Objectivity - Formal Tone - Integrating References.

Effective Writing Style for Research Papers: Word Order - Sentences and Paragraphs - Link Words for Cohesion - Avoiding Redundancy / Repetition - Breaking up long sentences - Paraphrasing Skills.

Advanced Reading and Research Vocabulary Development: Critical Reading Strategies - Analysing Research Articles - Identifying Arguments - Evaluating Findings - Formulaic Expressions - Academic Phrase Bank - Discipline-Specific Vocabulary - Commonly Misused Words.

Module II **15 Hours**

Presentation Language Skills: Written vs. Spoken English - Dynamic Vocabulary for Presentations -Expressive Language for Audience Engagement- Language for Clear and Impactful Slides - Adapting Language Style to Different Audiences.

Grammar Refinement for Research Writing: Advanced Punctuation Usage- Proper Use of Modifiers - Avoiding Ambiguous Pronoun References - Verb Tense Consistency - Conditional Sentences.

Technology and Language for Research: Technology and Role of AI in Research Writing - Citations and References - Plagiarism and Ethical Considerations - Tools and Awareness - Fair Practices.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Enhance their English Language Skills concerning research paper writing.	Understand
CO2: Develop a comprehensive set of linguistic skills essential for academic research.	Apply
CO3: Produce well-structured research papers using a variety of research and presentation technologies.	Apply

Reference Book(s):

R1: Craswell, G. 2004. Writing for Academic Success. Sage Publications. Springer, New York.

R2: Wallwork, Adrian. 2015. English for Academic Research: Grammar, Usage and Style

R3: Swales, J. &amp; C. Feak. 2012. Academic Writing for Graduate Students: Essential Skills and Tasks. Michigan University Press.

Web References:

1. https://tiramisutes.github.io/images/PDF/English+for+Writing+Research+Papers.pdf
2. https://libguides.usc.edu/writingguide/grammar
3. https://onlinecourses.swayam2.ac.in/ntr24_ed15/preview

SEMESTER II

Course Code: 24EST201		**Course Title: Real-Time Operating Systems**	
Course Category: PCC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours:**	**Max Marks:100**

Course Objectives:

To equip students with the knowledge and skills to design and manage real-time systems, focusing on uniprocessor and multiprocessor scheduling, resource access control, and real-time kernel design.

Module I **22 Hours**

REAL-TIME SYSTEMS: Basic Terminologies - Limits of Current Real-Time Systems - Desirable Features of Real-Time Systems – Factors affecting Predictability – Types of Task Constraints: Timing Constraints, Precedence Constraints, and Resource Constraints - Classification of Scheduling Algorithms - Metrics for Performance Evaluation - Scheduling Anomalies.

UNIPROCESSOR SCHEDULING ALGORITHMS: Periodic Tasks Scheduling: Cyclic Schedulers, EDF, RMA, and DMA -Aperiodic Task Scheduling: Jackson’s Algorithm, Horn’s Algorithm, Bartley’s Algorithm, Scheduling of Aperiodic Tasks with Precedence Constraints – Hybrid Task Set Scheduling: Foreground and Background Scheduling, Polling Server, Deferrable Server, Priority Exchange Server, Sporadic Server, and Slack Stealing

Module II **23 Hours**

Resource Access Control Protocols: Problems involved in Resource Sharing: Priority Inversion and Deadlock – Deadlock Detection and Avoidance Algorithm - Non-Pre-emptive Protocol – Highest Locker Priority Protocol – Priority Inheritance Protocol – Priority Ceiling Protocol– Comparison of Resource Access Control Protocols – Handling Task Dependencies.

Multiprocessor Scheduling and Real-Time Kernel Design Issues: Multiprocessor Task Partitioning and Scheduling Algorithms - Structure of a typical Real-Time Kernel - Data Structures –List Management – Kernel Primitives - Standards for Real-Time Operating Systems.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Explain the basic terminologies and constraints of real-time systems, and classify scheduling algorithms.	Understand
CO2: Analyze uniprocessor scheduling algorithms for periodic, aperiodic, and hybrid task sets.	Apply
CO3: Evaluate and implement resource access control protocols to address issues like priority inversion and deadlock.	Apply
CO4: Design multiprocessor scheduling algorithms and understand the structure and standards of real-time kernels.	Apply

Reference Book(s):

1. Giorgio C. Buttazzo, “Hard Real-Time Computing Systems”, Springer, New York, 2011.
2. Jean J. Labrosse, “ µC/OS-III, The Real-Time Kernel”, Micrium Press, 2009.
3. Jane W. Liu, “Real-Time Systems”, Pearson, New Delhi, 2006.

Course Code: 24EST202		Course Title: Linux Architecture and Device Drivers	
Course Category: PCC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a thorough understanding of Linux operating system architecture, file systems, and device drivers, along with practical skills in Linux system programming and device driver development.

Module I **22 Hours**

Basic Architecture: Evolution of Linux OS – Main characteristics of Linux – Typical Linux distributions – Linux directory structure – User and super/root users – access rights – Home directory – Vi editior - Commands – Overview of shell and GUI.

Linux Kernal Architecture: Layer diagram of OS - Hardware Abstraction Layer (HAL) – Memory manager – scheduler – file system – I/O subsystem – Networking subsystem – IPC – user space.

Module II **23 Hours**

Linux File System: Layers of Linux file system – structure of inode – process file system – The Ext2 File system –System programming concepts – API & ABIs – C library and compiler.

Device Driver: System start up (Booting) Methods - PC I/O architecture – classification of Linux devices: character and block devices – port I/O – PCI and ISA bus – polling, interrupt, and waiting queue – Device Files - Device driver Registration – Device driver initialization – I/O operation - typical Linux driver – dynamic and static drivers - kernel modules – Linking and unlinking of modules.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Describe the evolution, characteristics and architecture of the Linux operating system.	Understand
CO2: Utilize the Linux directory structure, user permissions and basic commands effectively.	Apply
CO3: Explain the layers and structure of the Linux file system and utilize system programming concepts	Understand
CO4: Develop and manage Linux device drivers, including driver registration, initialization and I/O operations	Apply

Reference Book(s):

1. Michael beck, Harald bohme, Mirko dziadzka, Ulrich Kunitz &quot;Linux Kernel Programming&quot;, Pearson Education, Reprint 2009.
2. Raghavan P., Amol Lad, Sriram Neelakandan “Embedded Linux System Design and Development”, Tailor &amp; Francis Group, reprint 2019.
3. Daniel P.Bovet, Marco Cesati “Understanding the Linux kernel”, Shroff publishers &amp; distributors Pvt Ltd, 2019.
4. Robert Love “LINUX System Programming” Shroff publishers &amp; distributors Pvt Ltd,2013.
5. Tim jones M. “GNU/Linux Application Programming”, Wiley Dreamtech India Pvt. Ltd, New Delhi, 2008.

Course Code: 24EST203		Course Title: Embedded System Networks	
Course Category: PCC		Course Level: Mastery	
L:T:P(Hours/Week) 3:0:0	Credits: 3	Total Contact Hours: 45	Max Marks:100

Course Objectives:

To provide the comprehensive knowledge of communication protocols used in embedded systems, focusing on CAN, LIN, MOST, FlexRay, USB, and TCP/IP, along with their applications and development tools.

Module I **22 Hours**

The CAN BUS: Introduction – Concepts of Bus Access and Arbitration – Error Processing and Management – Definition of the CAN Protocol ISO 11898-1 – Error Properties, Detection and Processing – Framing - The Rest of the Frame - CAN 2.0B

The CAN PHYSICAL LAYER: Introduction – Signal Propagation – Bit Synchronisation – Network Speed and Range – High Speed CAN – Low Speed CAN – CAN Components – Event-Triggered and Time-Triggered Protocols

Module II **23 Hours**

LIN, MOST and FLEXRAY: LIN: Introduction - Basic Concept of the LIN 2.0 Protocol - Conformity of LIN – **MOST**: The MOST (Media Oriented Systems Transport) Bus – General - MOST concept – **Flexray**: Genesis of FlexRay - FlexRay Consortium - Aim of FlexRay - Physical Time - Local Time - Channels, Cycles, Segments and Slots - Channels and Cycles – Segments -Communication Frames - Symbol Window Segment - Network Idle Time Segment.

USB AND TCP/IP FOR EMBEDDED SYSTEMS: Introduction – Types of USB Transfers: Control Transfer – Bulk Transfer – Interrupt Transfer – Isochronous Transfer – Introduction to the Enumeration Process – Introduction to USB Development Tools.

TCP/IP for Embedded Systems: Introduction – Embedded SMTP Client – Embedded SMTP Server

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Understand the basic concepts of CAN protocol, including bus access, arbitration, error processing, and framing.	Understand
CO2: Analyze the physical layer aspects of CAN, LIN, MOST, and FlexRay, including signal propagation, network speed, and protocols.	Apply
CO3: Utilize the application layers and development tools for CAN, and understand the basics of USB transfers and the enumeration process.	Understand
CO4: Implement TCP/IP protocols for embedded systems, including SMTP client and server functionalities.	Apply

Reference Book(s):

1. Dominique Paret, “Multiplexed Networks for Embedded Systems”, Wiley, 2007
2. Dominique Paret, “Flexray and Its Applications”, A John Wiley & Sons, Ltd., Publication Wiley, 2012.
3. John Hyde, “USB Design by Example”, Intel University Press, 2001
4. Jan, Axelson, “USB Complete”, Lake View Research, 2005
5. Edward Insam, “TCP/IP Embedded Internet Applications”, Elsevier, 2003
6. Tim Jones, “TCP/IP Application Layer Protocols for Embedded Systems”, Charles River Media, 2002

Course Code: 24ESL201		Course Title: Real-Time Systems Laboratory	
Course Category: PCC		**Course Level: Practice**	
L:T:P(Hours/Week) 0:0:4	**Credits: 2**	**Total Contact Hours: 60**	**Max Marks:100**

Course Objectives:

This course aims to provide students with practical experience in embedded systems development and real-time operating systems (RTOS), focusing on task management, resource management, inter-task communication, interrupt handling, memory management, and performance evaluation of scheduling algorithms.

List of Exercises

1. Creating a Makefile for an Embedded Application
2. Basic Task Management using Free RTOS
3. Resource Management with Semaphores
4. Inter-task Communication with Queues
5. Interrupt Management in Free RTOS
6. Memory Management with Dynamic Allocation
7. Task Synchronization using Event Groups
8. Performance Evaluation of Single-core Scheduling
9. Multi-core Task Scheduling with Free RTOS
10. Performance Comparison of Single-core vs. Multi-core Scheduling

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Develop competence in managing interrupts and memory resources within real-time systems, crucial for ensuring system responsiveness and stability.	Apply
CO2: Develop the ability to assess the efficiency of both single- and multi-core scheduling algorithms and make well-informed decisions on real-time system architecture.	Apply

Reference Book(s):

1.Giorgio C. Buttazzo, “Hard Real-Time Computing Systems”, Springer, New York, 2011.

2.Jean J. Labrosse, “ µC/OS-III, The Real-Time Kernel”, Micrium Press, 2009.

3.Jane W. Liu, “Real-Time Systems”, Pearson, New Delhi, 2006.

Course Code: 24SHA201		**Course Title: Teaching and Learning in Engineering (common to all PG Programmes)**	
Course Category: Audit Course		**Course Level: Introductory**	
L:T:P(Hours/Week) 2:0:0	**Credits: -**	**Total Contact Hours: 30**	**Max Marks:100**

Course Objectives:

The course is intended to impart knowledge on an outcome-based approach, employing active learning methods in lecture/practical/tutorial sessions. Assessments will be conducted using rubrics, focusing on higher-order thinking skills.

Module I **15 Hours**

Outcome Based Approach

Outcome based Education- Need & Approach- Washington accord- Graduate attributes- Learning outcomes –Blooms Taxonomy.

Active Learning Methods

Design and Delivery plan for lectures/practical/tutorial sessions-Need for Active learning methods-Active learning strategies- Benefits of Active learning Methods.

Module II **15 Hours**

Assessments

Assessments- types of assessments-need for rubrics, Types of rubrics- Assessment using rubrics.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO 1: Use outcome based approach in teaching courses in engineering Programmes.	Apply
CO 2: Conduct lecture/practical/tutorial sessions using active learning methods.	Apply
CO 3: Conduct higher order assessments by using rubrics.	Apply

Reference Book(s):

1. William G. Spady and Francis Aldrine A. Uy (2014). Outcome-Based Education: Critical Issues and Answers, ISBN: 978-971-0167-41-8, Maxcor Publishing House, Inc.
2. Dr. William G. Spady, Wajid Hussain, Joan Largo, Dr. Francis Uy (2018). Beyond Outcomes Accreditation: Exploring the Power of 'Real' OBE Practices.
3. Richard M. Felder, Rebecca Brent (2016), Teaching and Learning STEM: A Practical Guide, John Wiley & Sons Inc.

PROFESSIONAL ELECTIVES

Course Code: 24ESE001		**Course Title: Multi-Core Embedded Systems**	
Course Category: : PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide students with a comprehensive understanding of multi-core architectures, program optimization techniques, operating system issues, and power management strategies in embedded systems

Module I **22 Hours**

Multi-Core Architectures: Introduction to parallel computers: Instruction Level Parallelism (ILP) vs. Thread Level Parallelism (TLP); performance issues: brief introduction to cache-hierarchy and communication latency. Shared memory multiprocessors: general architecture and the problem of cache- coherence; synchronization primitives: atomic primitives; locks: tickets, array; barriers: central and tree; performance implications in shared memory programs.

Program Optimization in Multi-Core Processors: overview of parallelism, shared memory programming; introduction to OpenMP; data flow analysis, pointer analysis, alias analysis, data dependence analysis, solving data dependence equations (integer linear programming problem); loop optimizations; memory hierarchy issues in code optimization

Module II **23 Hours**

Operating System Issues for Multiprocessing: scheduling techniques: usual OS scheduling techniques, threads, distributed- scheduler, multiprocessor scheduling, gang scheduling; communication between processes, message boxes, shared memory; sharing issues and synchronization, sharing memory and other structures.

Power Management in Embedded Multi-Cores : Power Management Techniques: Clock Gating, Power Gating, Dynamic Voltage Scaling, Dynamic Frequency Scaling, Smart Caching, Scheduling

Course Outcomes	Cognitive Level
At the end of this course, students will be able to:	
CO1: Illustrate the fundamentals of multi-core architectures, including the distinction between Instruction Level Parallelism (ILP) and Thread	Understand
CO2: Apply synchronization primitives and optimization techniques in shared memory multiprocessors, including atomic primitives, locks, barriers, and loop optimizations	Apply
CO3: Build proficiency in shared memory programming and parallelism using OpenMP, and conduct data flow analysis, pointer analysis,	Apply
CO4: Evaluate power management techniques in embedded multi-core processors	Apply

Reference Book(s):

1. Georgious Kornoras, “Multi-core Embedded Systems”, CRC Press, 2010.
2. David E. Culler, Jaswinder Pai Singh, Anoop Gupta. Parallel Computer Architecture: A Hardware/Software Approach”, Elsevier India, First Edition, 2003.
3. Steven S. Muchnick. Advanced Compiler Design and Implementation”, Elsevier, First Edition, 2007.
4. Peter S. Pacheco, “An Introduction to Parallel Programming”, Morgan-Kauffman/ Elsevier, 2011.
5. Andrew S. Tanenbaum. Distributed Operating Systems. Pearson Education, 2002.

Course Code: 24ESE002		**Course Title: Advanced Embedded Controllers**	
Course Category: : PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding of mixed-signal processor architecture and ARM Cortex-Mx architecture, including their hardware components, peripherals, and programming environment

Module I **22 Hours**

Architecture of Mixed Signal Processor: Introduction to 16-bit Mixed Signal Controller-Important aspects of Mixed Signal Controller's Hardware – CPU – Functional Block Diagram - Memory Mapping – Clock System - Addressing Modes - Register Mode – Indexed Mode – Introduction to functions – Interrupts - Low Power Modes - Development Environment - Programming and Debugging

Peripherals of Mixed Signal Processor : Parallel ports - Digital Inputs/ Outputs – Timers - Watchdog Timer Capture/Compare module –Generation of Periodic Signal – Generation of PWM Signal - Operation of the ADC Peripheral (ADC10) - Internal Temperature Sensor – Serial Communication Protocols

Module II **23 Hours**

Architecture of ARM Cortex Mx: ARM Cortex-Mx Processor Core overview - Programmers Model - Memory Model - Exception and Fault Handling - Power Management - Instruction Set Summary - CMSIS Functions - Hardware-Software Synchronization - Interrupt Synchronization - Multithreading - Register Map - System Timer - Nested Vectored Interrupt Controller - Floating Point Unit (FPU)-Optional Memory Protection Unit.

Peripherals of ARM Cortex – Mx Controller: Cortex-Mx Peripherals - Parallel I/O Ports - Timer Interfacing - Pulse Width Modulation - Frequency Measurement - Binary Actuators - Integral Control of a DC Motor – DAC - ADC -Serial Communication Protocols

Course Outcomes	Cognitive Level
At the end of this course, students will be able to:	
CO1: Analyze the architecture of mixed-signal processors, including important hardware aspects, CPU functionality, memory mapping, addressing modes, interrupts, and low-power modes.	Apply
CO2|: Choose the peripherals of mixed-signal processors and serial communication protocols.	Apply
CO3: Understand the architecture of ARM Cortex-Mx processors and hardware-software synchronization.	Understand
CO4: Examine the peripherals of ARM Cortex-Mx controllers.	Apply

Reference Book(s):

1. Steven F.Barret, Daniel J Pack, “Microcontroller Programming and Interfacing: Texas Instruments MSP430” , Morgan & Claypool Publishers, 2011.
2. John H. Davies, “MSP430 Microcontroller Basics”, First Edition, Newnes Publication , ISBN: 978-93-80501-85-7, 2010.
3. C.P.Ravikumar. “MSP430 Microcontroller in Embedded System Project”, First Edition, Elite Publishing House Private Ltd, Dec , ISBN:978-81-88901-46-3, 2011
4. J. W. Valvano, “Embedded Systems: Introduction to ARM Cortex -M Microcontrollers”, Fourth edition, Volume 1, ISBN: 978- 1477508992, 2013
5. J. W. Valvano, “Embedded Systems: Real-Time Interfacing ARM Cortex – Microcontrollers”, Fourth edition, Volume 2, ISBN: 978-1477508992, 2014

Course Code: 24ESE003		**Course Title: Automotive Embedded Systems**	
Course Category: : PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To explore the intricacies of automotive electronic systems, covering domains like powertrain, body, chassis, infotronics, safety, security, networking, and diagnostics.

Module I **22 Hours**

Introduction: Current trends in modern automobiles – Drive by wire Systems -Vehicle functional domains and their requirements - Components of an Automobile Electronic system and their functions: Sensors, Actuators, Control Units and Software structure of Control units.

Power Train, Body and Chassis Domain: Power Train Domain: Gasoline engine management -Body Electronics: Vehicle power supply controllers – Lighting technology– Adaptive lighting system – Automatic wiper system – Door control modules - Vehicle to vehicle communication - Chassis Domain: Antilock Braking System (ABS) – Electronic Stability Program (ESP)

Module II **23 Hours**

Automotive Infotronics and Safety & Security Systems : Automotive Vision System - Advanced Driver Assistant Systems (ADAS) – Multimedia systems- Intelligent Automotive Systems: Navigation Systems – Adaptive Cruise Control (ACC) - Active and Passive safety- Airbag System – Seat belt tightening system - Electronic Brake Force Distribution (EBD) - Lane Departure Warning System - Anti-theft technologies – Electronic Immobilizers – Remote Keyless entry.

Automotive Networking and Diagnostics : Cross-system functions - Bus systems: Requirements, classification and applications –– Introduction – Diagnostics Theory – On-Board Diagnostics – Off-board diagnostics – Diagnostics Link Connector – Vehicle Condition Monitoring - Diagnostic Interfaces – examples of networked vehicles.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Analyse the components and functions of modern automobile electronic systems, including sensors, actuators, and control units.	Analyse
CO2: Evaluate the functionality and requirements of different vehicle domains such as powertrain, body, and chassis.	Apply
CO3: Implement advanced automotive technologies like ADAS, infotronics, and safety systems, including airbag systems and lane departure warning.	Apply
CO4: Examine automotive networking and diagnostics, including bus systems, on-board and off-board diagnostics, and vehicle condition monitoring	Apply

Reference Book(s):

1. Nicolas Navet and Francoise Simonot-Lion, “Automotive Embedded Systems Handbook”, CRC Press, USA, 2008.
2. Robert Bosch,” Bosch Automotive Electrics and Automotive Electronics: Systems and Components, Networking and Hybrid Drive (Bosch Professional Automotive Information)”, 5th Edition, Springer Vieweg, 2013.
3. LjuboVlacic, Michel Parent &FurnioHarshima, “Intelligent Vehicle Technologies: Theory and Applications”, ButterworthHeinemann publications, 2001.
4. Robert Bosch, “Automotive Hand Book”, SAE (5TH Edition),2000.
5. William Ribbens “Understanding Automotive Electronics- An Engineering Perspective”, 8th Edition, Butterworth-Heinamann, 2017.

Course Code: 24ESE004		**Course Title: Automotive Software Architecture**	
Course Category: PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) **3:0:0**	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding of automotive software architectures, development methodologies, standards, and safety requirements, focusing on AUTOSAR and functional safety.

Module I **22 Hours**

Automotive Software Architectures: Introduction - History of Software in the Automotive Industry- Software Architectures: Views and Documentation - Common View on Architecture in General - Architectural Views - Architectural Styles - Describing the Architectures - Current Trends in Automotive Software Architectures.

Automotive Software Development: V-Model of Automotive Software Development – Requirements – Variant Management - Testing Strategies - Construction Database and Its Role in Automotive Software Engineering - Design of Automotive Software: Simulink Modelling - Simulink Compared to SySML/UML - MISRA

Module II **23 Hours**

AUTOSAR: AUTOSAR Reference Architecture - AUTOSAR Development Methodology – AUTOSAR Meta-Model - AUTOSAR ECU Middleware - AUTOSAR Evolution – ARTOP – MICROSAR - Functional Safety of Automotive Software: Management and Support for Functional Safety - Concept and System Development - Planning of Software Development - Software Safety Requirements - Software Architectural Design - Software Unit Design.

Software Standard: Functional Safety Standards – ISO-26262 - IEC 62304 and ISO 14971- DESIGN VALIDATION: Markov Models - The Fault Tree - Software Failure Rates - Coding Guidelines - Code Coverage Metrics - Static Analysis - Implementation and Testing - Software Integration and Testing - Verification of Software Safety Requirements - Integration, Testing, Validation, Assessment.

Course Outcomes	Cognitive Level
At the end of this course, students will be able to:	
CO1:Analyze automotive software architectures, including views, documentation, styles, and current trends	Analyze
CO2: Apply the V-Model of automotive software development, including requirements management, variant management, and testing strategies.	Apply
CO3: Utilize AUTOSAR reference architecture and development methodology for ECU middleware and functional safety.	Apply
CO4:Evaluate functional safety standards such as ISO-26262, IEC 62304, and ISO 14971, and implement validation techniques including Markov models, fault trees, coding guidelines, and static analysis.	Analyze

Reference Book(s):

1. Miroslaw Staron “Automotive Software Architectures An Introduction”, Springer International Publishing, 2017.
2. Chris Hobbs, “Embedded Software Development for Safety-Critical Systems”, Taylor & Francis Group, LLC. 2016.
3. Pradeep Oak and Renu Rajani , “Software Testing – Effective Methods, Tools and Techniques”, Tata McGraw Hill Publications, 2004.
4. Stephen L. Montgomery , “MISRA C: Guidelines for the Use of the C Language in Critical Systems “, Motor Industry Research Association, 2013.
5. Justyna Zander, Ina Schieferdecker, Pieter J. Mosterman, “Model-Based Testing for Embedded Systems” CRC press Taylor & Francis Group, 2012.

Course Code: 24ESE005		**Course Title: Internet of Things**	
Course Category: PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) **3:0:0**	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding of Internet of Things (IoT) fundamentals, protocols, cloud computing, security issues, and applications, addressing both consumer and industrial aspects.

Module I **22 Hours**

Fundamentals of IoT: Introduction to Internet of Things (IoT) – Machine to Machine (M2M) – Functional Characteristics – Recent Trends in the Adoption of IoT – Societal Benefits of IoT – Consumer IoT v_s Industrial Functional Components of a typical IoT System: Sensors, Actuators, Embedded Computation Units, Communication Interfaces , Software Development

IoT Protocols: Physical and Data Link Layer Protocols: RFID: NFC, FFC, ZigBEE, Bluetooth Low Energy, Z-Wave, Wi-Fi, LoRA - Network Layer Protocols: IPv4, IPv6, TCP & UDP, 6LoWPAN - Application Layer Protocols: COAP, MQTT.

Module II **23 Hours**

Cloud Computing: NIST Visual Model – Essential Characteristics –Components of Cloud Computing - Service Models – Deployment Models – Service Management and Security – Examples – Basics of Fog Computing

IoT Security and Applications: IEEE 802.11 Wireless Networks Attacks: Basic Types, RFID Security – Security Issues in ZigBEE: Bluetooth Security: Threats to Bluetooth Devices and Networks - IoT Applications: Health Care, Connected Vehicles, Smart Grid, Smart Home, and Smart City

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Analyze the basics of IoT including machine to machine communication, functional characteristics, and recent trends, understanding the components of typical IoT systems.	Analyze
CO2: Evaluate IoT protocols at different layers including physical, data link, network, and application layers, covering RFID, NFC, ZigBee, Bluetooth Low Energy, Wi-Fi, LoRA, IPv4, IPv6, TCP, UDP, COAP, and MQTT.	Analyze
CO3: Illustrate the cloud computing principles including NIST visual model, essential characteristics, service models, deployment models, service management, and security, along with fog computing basics.	Understand
CO4: Examine IoT security issues in IoT applications	Apply

Reference Book(s):

1. Adrian McEwen and Hakim Cassimally, “Designing the Internet of Things”, John Wiley and Sons Ltd, UK, 2014.
2. Olivier Hersent, David Boswarthick and Omar Elloumi, “The Internet of Things: Key Applications and Protocols”, John Wiley and Sons Ltd., UK 2012.
3. Dieter Uckelmann, Mark Harrison, Florian Michahelles, “Architecting the Internet of Things”, Springer, New York, 2011.
4. Johnny Cache, Joshua Wright and Vincent Liu, “Hacking Exposed Wireless: Wireless Security Secrets and Solutions”, Tata McGraw Hill, New Delhi, 2010.
5. HimanshuDwivedi, Chris Clark and David Thiel, “Mobile Application Security”, Tata McGraw Hill, Nw Delhi, 2010.
6. Vijay Madisetti, ArshdeepBahga, “Internet of Things (A Hands-on Approach), Universities Press, 2015.

Course Code: 24ESE006		Course Title: Python Programming	
Course Category: : PEC		Course Level: Mastery	
L:T:P(Hours/Week) 3:0:0	Credits: 3	Total Contact Hours: 45	Max Marks:100

Course Objectives:

To provide a comprehensive understanding of Python programming language fundamentals, including basic features, object-oriented programming, I/O handling, error handling, and various applications such as network programming, database access, GUI development, and web application development.

Module I **22 Hours**

Basics and Object Oriented Features: Variables - Keywords - Data Types – Lists- Tuples - Sets - Dictionaries – Operators – Control Statements – Loops – Functions - Lambda – Modules - Standard Functions - Classes and Objects – Instance Methods – Special Methods - Class Variables – Inheritance – Polymorphism

I/O Handling and Error Handling: File Handling- Access Modes – Read, Write, Create and Delete File –Exception Handling: Run Time Errors - Exception Model - Exception Hierarchy - Handling Multiple Exceptions - Handling I/O Exceptions - Regular Expressions

Module II **23 Hours**

Applications Using Python: Network programming-Database Access- Creating simple Graphical User Interfaces - Sending e-mail using SMTP Library-Multithreading-CGI Programming - Extensions- Web application development: opening an URL-creating a simple web page- Overview of webapp2 and Flask – Programming Raspberry Pi and Arduino using Python

Micro PYTHON: Introduction-Supporting Boards- GPIO- Digital and Analog Inputs and Outputs-PWM-Interrupts- Installation and Programming- Web Servers-Sensors and Modules-MQTT- IoT case study

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Analyze the basics of Python programming language including variables, data types, operators, control statements, functions, modules, classes, objects, inheritance, polymorphism, and exception handling.	Analyze
CO2: Illustrate I/O handling concepts in Python including file handling, access modes, and exception handling for runtime errors and I/O exceptions, along with regular expressions.	Understand
CO3: Design various applications using Python programming and Raspberry Pi and Arduino.	Apply
CO4: Examine MicroPython supporting boards and its applications	Apply

Reference Book(s):

1. Wesley J Chun, “Core Python Applications Programming”, Prentice Hall, 2012.
2. Mark Summerfield. “Programming in Python 3: A Complete introduction to the Python Language”, AddisonWesley Professional, 2009.
3. Sumit Gupta “Building Web Applications with Python and Neo4j”,Packt publishers,2015
4. Ron DuPlain, “ Instant Flask Web Development “,Packt publishers ,Second edition ,2013.
5. Nicholas H. Tollervey, “Programming with MicroPython: Embedded Programming with Microcontrollers and Python“, O’Reilly Publications, 2018.

Course Code: 24ESE007		**Course Title: Artificial Intelligence**	
Course Category: : PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding of artificial intelligence (AI) principles, covering intelligent agents, problem-solving algorithms, knowledge representation and reasoning, uncertain knowledge handling, reasoning under uncertainty, and decision-making techniques.

Module I **22 Hours**

Introduction: Overview, foundations, scope, problems and approaches of AI, Intelligent Agents and environments, Structure of Agents, Problem Solving Agents, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions, Search in complex Environments

Knowledge Representation And Reasoning: Ontologies, foundations of knowledge representation and reasoning, Logical Agents, Proportional Logic, First Order Logic, Inference in first order logic, Rule based systems, Knowledge representation, Automated planning.

Module II **23 Hours**

Uncertain Knowledge and Reasoning: Quantifying Uncertainty, Basic probability notation, Naïve bayes models, Probabilistic reasoning, Exact and Approximate Inference in Bayesian Network, Causal Networks, Time and Uncertainty, Hidden Markov models, Kalman filter, Probabilistic programming.

Decision Making: Basis of Utility Theory, Utility Functions, Decision Networks, Sequential Decision Problems, Properties of Multi agent Environments, Game theory, Making Collective Decisions, Bargaining, Sample applications: Industrial automation and Business Intelligence.

Course Outcomes	Cognitive Level
At the end of this course, students will be able to:	
CO1: Analyse the foundations, scope, and approaches of AI, including the structure of intelligent agents, problem-solving agents, and various search algorithms for problem-solving.	Analyse
CO2: Illustrate the knowledge representation and reasoning techniques such as ontologies, logical agents, propositional logic, first-order rule-based systems, and automated planning.	Understand
CO3: Examine decision-making principles including utility theory, utility functions, decision networks, sequential decision problems, properties of multi-agent environments, game theory, collective decision-making, bargaining, and their applications in industrial automation and business intelligence.	Apply

Reference Book(s):

1. Deepak Khemani, “A First Course in Artificial Intelligence”, McGraw Hill Education (India) Private Limited, New Delhi, 2014
2. Stuart Russsel and Peter Norvig, “Artificial Intelligence – A Modern Approach”, Pearson Education, New Delhi, 2020
3. Padhy N P, “Artificial Intelligence and Intelligent Systems”, Oxford University Press, New Delhi, 2005
4. Nils J Nilsson, “Artificial Intelligence – A New Synthesis:, Morgan Kaufmann, New Delhi, 2007
5. George F Luger, “Artificial Intelligence – Structures and Strategies for Complex Problem Solving”, Pearson Education, New Delhi, 2009

Course Code: 24ESE008		**Course Title: Industrial Networking and Standards**	
Course Category: : PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) **3:0:0**	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding of serial interface standards, communication protocols such as HART, Modbus, AS-i, DeviceNet, PROFIBUS, Foundation Fieldbus, SERCOS III, and Industrial Ethernet, focusing on their structures, applications, and

Module I **22 Hours**

Serial Interface Standards: Modern Instrumentation and Control Systems – Open Systems Interconnection Model –EIA-232 Interface Standard – Major Elements of EIA-232 – Half-Duplex and Full-Duplex operation of EIA-232 Interface –Overview of EIA-422 and EIA-423 Interface Standards- EIA-485 Interface Standard – Comparison of Serial Interface Standards–Noise problems in serial communication and troubleshooting.

HART and MODBUS Protocol: HART PROTOCOL over 4-20 mA Signal Base – Wireless HART Protocol - MODBUS: Modbus Protocol Structure: Data types, Transmission modes, Messaging Structure–Modbus Function Codes- Fault Handling Mechanisms of Modbus Protocol – Applications of Modbus Protocol

Module II **23 Hours**

Field Area Networking Protocols: Actuator Sensor Interface - Structure of AS-i slave ICs, AS-i messages, AS-I modulation technique, Troubleshooting-Device Net: Physical Layer Topology – Device Taps – Data link Layer: Frame Format – Medium Access – Fragmentation- Process Field Bus (PROFIBUS) –Profisafe – ProfiDrive - Foundation Fieldbus: Physical Layer and Wiring Rules – Datalink Layer– Application Layer – Error Detection and Diagnostics – SERCOS III – Control NET

Industrial Ethernet: Overview – Ethernet Hardware Basics – Ethernet Protocol and Addressing – Introduction to10 Mbps, 100 Mbps and Gigabit Ethernet – Real-time Ethernet for Automation Applications – Time-triggered Ethernet – Security in Industrial Communications – EtherCAT

Course Outcomes	Cognitive Level
At the end of this course, students will be able to:	
CO1: Analyse serial interface standards including EIA-232, EIA-422, EIA-423, and EIA-485, and compare their features and noise problems.	Analyse
CO2: Illustrate the structure and operation of communication protocols like HART and Modbus, including wireless HART and fault handling mechanisms.	Understand
CO3: Model the architecture and features of field area networking protocols such as AS-i, DeviceNet, PROFIBUS, Foundation Fieldbus, and SERCOS III.	Apply
CO4: Evaluate Industrial Ethernet technologies including Ethernet hardware basics, protocol, addressing, real-time Ethernet, time-triggered Ethernet, and security aspects like EtherCAT.	Apply

Reference Book(s):

1. John Park, Steve Mackey, and Edwin Wright, “Data Communications for Instrumentation and Control”, Elsevier, 2003
2. Perry Marshall and John Rinaldi, “Industrial Ethernet”, The Instrumentation, Systems and Automation Society, 2005
3. Richard Zurawski, “Industrial Communications Technology Handbook”, CRC Press, 2005

Course Code: 24ESE009		Course Title: Cryptography and Network Security	
Course Category: : PEC		Course Level: Mastery	
L:T:P(Hours/Week) 3:0:0	Credits: 3	Total Contact Hours: 45	Max Marks:100

Course Objectives:

To provide a comprehensive understanding of security fundamentals, public key cryptography, hash functions, network security principles, and standards, addressing authentication, encryption, intrusion detection, and secure communication protocols.

Module I **22 Hours**

Security Basics: The OSI Security Architectures-Conventional Encryption – Classical Techniques and Modern Techniques-Modes of operation - DES, AES, Key Distribution.

Public key Cryptography and Hash Functions: Number Theory Concepts – Prime numbers- Modular Arithmetic – Fermat & Euler Theorem – Euclid Algorithm – RSA Algorithm – Diffie Hellman Key Exchange Elliptic Curve Cryptography – Hashing techniques- SHA-HMAC – Digital Signatures- DSS, Digital Signature Algorithm.

Module II **23 Hours**

Network Security and Standards: Intruders and Intrusion – Viruses and Worms – OS Security – Firewalls – Design Principles – Packet Filtering – Application gateways – Trusted systems - Security Standards: IEEE, RSA and ISO standards Blueprint for Security – Design of Security Architecture.

Network Issues: Authentication Applications – Kerberos –Electronic Mail Security –PGP – IP Security –Architecture- Web Security- SSL – TLS – SET.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Analyse security basics including OSI security architectures, encryption techniques like DES and AES, and key distribution mechanisms.	Analyse
CO2: Evaluate public key cryptography concepts such as RSA algorithm, Diffie Hellman key exchange, elliptic curve cryptography, and hash functions like SHA-HMAC.	Apply
CO3: Organize network security issues including intruders, viruses, worms, OS security, and firewalls, along with security standards and design principles	Apply
CO4: Examine network security protocols like Kerberos, PGP, IP security, and web security protocols SSL and TLS, understanding their architectures and applications.	Apply

Reference Book(s):

1. William Stallings, “Network Security Essentials, Applications and Standards”, Dorling Kindersley I P. Ltd, Delhi, 2008.
2. William Stallings, “Cryptography and Network Security - Principles and Practice”, Pearson Education, Delhi, 2007.
3. Behrouz A Forouzan, “Cryptography and Network Security”, Tata McGraw Hill Ltd, New Delhi, 2008.
4. Wenbo Mao, “Modern Cryptography: Theory and Practice”, Prentice Hall, New Delhi, 2003.
5. AtulKahate , “ Cryptography and Network Security” Tata McGraw Hill Ltd, New Delhi, 2008.
6. David R Mirza, “Hack Proofing your Network”, Dream Tech (SYNGRESS) Publication, New Delhi, 2002.
7. Richard E. Smith, “Internet Cryptography”, Addison – Wesley, 2004.

Course Code: 24ESE010		**Course Title: Advanced Digital Signal Processing**	
Course Category: PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding on multirate digital signal processing techniques, including decimation, interpolation, spectral representations, and the design and analysis of filter banks.

Module I **22 Hours**

Multirate DSP: Sampling–Spectral representation: DFT and FFT–Review of Digital filters-Decimation and Interpolation by an integer and rational factors– Multistaging– Decimation and Interpolation with poly phase filters – Realizations – Applications multirate signal processing.

Filter Banks: Analysis and Synthesis of Filter Banks– Quadrature Mirror Filter (QMF) banks– Filter bank with perfect reconstruction– 2-Channel and M-channel– Paraunitary filter banks– Biorthogonal and Linear phase filter banks– Tree and parallel structured filter banks Transmultiplexer filter banks– Multi resolution analysis – Subband coding and its applications

Module II **23 Hours**

Wavelet Transform: Short-Time Fourier Transform – limitations - time-frequency scaling-Heisenberg's uncertainty – Continuous Wavelet Transform – Discrete Wavelet Transform – Haar, Daubechy's wavelets – Multi Resolution Analysis of audio signal.

Adaptive Filters: FIR adaptive filters – adaptive filters based on steepest descent method – LMS algorithm – Variants of LMS algorithm – adaptive channel equalization – adaptive echo cancellation – RLS adaptive algorithm

Course Outcomes	Cognitive Level
At the end of this course, students will be able to:	
CO1 : Design and implement decimation and interpolation systems, and understand their impact on signal spectral properties using DFT and FFT.	Apply
CO2 : Analyze and synthesize filter banks, including designing quadrature mirror filter banks and ensuring perfect reconstruction in multi-channel systems.	Analyze
CO3 : Perform continuous and discrete wavelet transforms and apply multi-resolution analysis techniques to audio signals, utilizing wavelets like Haar and Daubechies.	Apply
CO4 : Develop and implement adaptive filters using the steepest descent method, LMS, and RLS algorithms, and apply these filters in applications such as channel equalization and echo cancellation.	Apply
CO5 : Integrate and apply multirate DSP techniques, filter banks, wavelet transforms, and adaptive filters to solve practical problems in signal processing, enhancing their readiness for industry and research roles.	Apply

Reference Book(s):

1. Vikram Gadre and Aditya S Abhyankar, ” Multi resolution and Multirate Signal Processing: Introduction, Principles and Applications” , McGraw Hill Education, 2017.
2. N.J.Fliege, “Multirate Digital Signal Processing” John wiley & sons Ltd., Reprinted with correction, 2000.
3. Vaidyanathan P P, “Multirate Systems and Filter Banks”, Pearson Education, 2011.
4. Stephane Mallat, “A Wavelet Tour of Signal Processing”, Elsevier, Academic Press, Third Edition, December 2008.
5. Rao, R.M and A.S.Bopardikar, ”Wavelet Transforms: Introduction to Theory and Applications, Addison Wesley, Reprint 2003.
6. Simon Haykin, ” Adaptive Filter Theory”, Pearson Education, Fourth Edition, 2008.

Course Code: 24ESE011		**Course Title: OPTIMIZATION TECHNIQUES**	
Course Category: : PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide students with a solid understanding of the fundamental concepts and methods of linear and non-linear programming, including the formulation and solving of optimization problems, proficiency in linear programming techniques, methods for non-linear optimization, and principles of dynamic programming in multistage decision processes.

Module I **22 Hours**

Linear Programming: Statement of Optimization problems, Graphical method, Simplex method, Revised simplex method, Two phase simplex method, Duality in linear programming, Sensitivity analysis.

Non-Linear Programming (Unconstrained Optimization): Direct search methods: Univariate method, Pattern search method, Simplex method, Descent methods: Steepest Descent method, Conjugate gradient method, Quasi Newton method.

Module II **23 Hours**

Non-Linear Programming (Constrained Optimization): Direct methods: The Complex method, Zoutendijk’s Method of Feasible Directions, Rosen’s Gradient Projection Method , Indirect methods: Transformation Techniques, Basic Approach of the Penalty Function Method, Interior Penalty Function Method, Exterior Penalty Function Method.

Dynamic Programming: Multistage decision process, Suboptimization and Principle of Optimality, Computational procedure, Final value problem to initial value problem, Linear Programming as a Case of Dynamic Programming, Continuous dynamic programming.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Formulate and solve linear programming problems using the graphical method and simplex-based techniques.	Analyze
CO2: Apply direct search and descent methods to solve unconstrained non-linear programming problems.	Apply
CO3: Implement complex method and feasible directions approach for constrained non-linear programming.	Apply
CO4: Utilize penalty function methods for solving non-linear programming problems with constraints.	Apply
CO5: Analyze and solve dynamic programming problems using principles of sub optimization and computational procedures.	Analyze

Reference Book(s):

1. Hamdy A Taha, “Operations Research: An Introduction”, Pearson Education, New Delhi, 2012.
2. Singaresu S Rao, “Engineering Optimization: Theory and Practice”, New Age International, New Delhi, 2011.
3. David.G.Luenberger, Yinyu Ye, "Linear and Nonlinear Programming", Springer, Newyork, 2015.
4. Gupta C B, “Optimization Techniques in Operations Research”, I K International, New Delhi, 2012.
5. Sharma J K, “Operations Research: Theory and Applications”, Macmillan Company, New Delhi, 2013.

Course Code: 24ESE012		Course Title: Intelligent Controllers	
Course Category: PEC		Course Level: Mastery	
L:T:P(Hours/Week) 3:0:0	Credits: 3	Total Contact Hours: 45	Max Marks:100

Course Objectives:

To provide a comprehensive understanding of the principles and applications of artificial neural networks, fuzzy logic systems, genetic algorithms, hybrid control schemes, and practical applications through control toolboxes and case studies.

Module I **22 Hours**

Overview, Modelling and Control of Artificial Neural Network (ANN): Review of fundamentals - Biological neuron, Artificial neuron, Activation function, Single Layer Perceptron – Limitations – Multi Layer Perceptron – Back propagation algorithm (BPA) - Generation of training data - optimal architecture – Model validation- Control of non linear system using ANN- Direct and Indirect neuro control schemes- Adaptive neuro controller – Case study - Familiarization of Neural Network Control Tool Box.

Overview, Modelling and Control Of Fuzzy Logic: Fuzzy set theory – Fuzzy sets – Operation on Fuzzy sets - Scalar cardinality, fuzzy cardinality, union and intersection, complement (yager and sugeno), equilibrium points, aggregation, projection, composition, fuzzy relation – Fuzzy membership functions. Modeling of nonlinear systems using fuzzy models(Mamdani and Sugeno) –TSK model - Fuzzy Logic controller – Fuzzification – Knowledge base – Decision making logic – Defuzzification- Adaptive fuzz y systems-Case study-Familiarization of Fuzzy Logic Tool Box.

Module II **23 Hours**

Genetic Algorithm: Basic concept of Genetic algorithm and detail algorithmic steps, adjustment of free parameters. Solution of typical control problems using genetic algorithm. Concept on some other search techniques like Tabu search, Ant-colony search and Particle Swarm Optimization.

Hybrid Control Schemes: Fuzzification and rule base using ANN–Neuro fuzzy systems-ANFIS–Optimization of membership function and rule base using Genetic Algorithm and Particle Swarm Optimization - Case study– Familiarization of ANFIS Tool Box.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1 : Outline the basic architectures of NN and Fuzzy sets	Understand
CO2 : Design and implement ANN architectures, algorithms and know their limitations.	Apply
CO3 : Identify and work with different operations on the fuzzy sets.	Apply
CO4 : Develop ANN and fuzzy logic based models and control schemes for non-linear systems.	Apply
CO5 : Outline and explore hybrid control schemes and PSO	Understand

Reference Book(s):

1. Sivanandam S N, and Deepa S. N., “Principles of Soft Computing”, Wiley India (P) Ltd.,New Delhi,2nd Edition, June 2011.
2. Sivanandam S N, Sumathi S., and Deepa S. N., “Introduction to Neural Networks using Matlab 6.0”, Tata McGrawHill Publications, New Delhi, 20th reprint 2014.
3. LaureneFausett, “Fundamentals of Neural Networks”, Pearson Education India, New Delhi, 2004.
4. Timothy Ross, “Fuzzy Logic with Engineering Applications”, McGraw Hill, Singapore, 3rd Edition, 2010.
5. David E Goldberg, “Genetic Algorithms in Search, Optimisation and Machine Learning”, Pearson Education, New Delhi, 2004.

Course Code: 24ESE013		Course Title: Machine Learning	
Course Category: PEC		Course Level: Mastery	
L:T:P(Hours/Week) 3:0:0	Credits: 3	Total Contact Hours: 45	Max Marks:100

Course Objectives:

To provide students with a solid understanding of the fundamental objectives and concepts of machine learning, data preprocessing techniques, supervised and unsupervised learning methods, clustering techniques, and Bayesian learning methods.

Module I **22 Hours**

Introduction to Machine Learning: Objectives of machine learning – Human learning/ Machine learning – Types of Machine learning:- Supervised Learning – Unsupervised learning – Reinforcement Learning – Evolutionary Learning - Regression – Classification – The Machine Learning Process:- Data Collection and Preparation – Feature Selection – Algorithm Choice – Parameter and Model Selection – Training – Evaluation.

Data Pre-processing: Data quality – Data preprocessing: - Data Cleaning:– Handling missing data and noisy data – ata integration:- Redundancy and correlation analysis – Data Reduction:- Dimensionality reduction (Linear Discriminant Analysis – Principal Components Analysis – Factor Analysis –Independent Components Analysis) – Numerosity Reduction - Data Compression - Data Normalization and Data Discretization.

Module II **23 Hours**

Supervised Learning : Linearly separable and nonlinearly separable populations – Multi Layer Perceptron – Back propagation Learning Algorithm – Radial Basis Function Network – Support Vector Machines: - Kernels – Risk and Loss Functions - Support Vector Machine Algorithm – Multi Class Classification – Support Vector Regression.

Clustering and Unsupervised Learning: Introduction – Clustering:- Partitioning Methods:- K-means algorithm - Hierarchical clustering – Fuzzy Clustering – Clustering High-Dimensional Data:- Problems – Challenges – Subspace Clustering – Biclustering - Self Organizing Map (SOM) - SOM algorithm.

Bayesian Learning: Probability based clustering – The Expectation Maximization Algorithm – Bayesian Classification – Bayesian Networks – Learning Bayesian Networks – Hidden Markov Models.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1 : Outline the basic theory underlying machine learning.	Understand
CO2 : Outline the range of machine learning algorithms along with their strengths and weaknesses.	Understand
CO3 : Formulate machine learning problems corresponding to different applications	Create
CO4 : Apply machine learning algorithms to solve problems of moderate complexity.	Apply
CO5 : Analyze current research papers and understand the issues raised by current research.	Analyze

Reference Book(s):

1. Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press, 2011.
2. Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, 2011
3. Jiawei Han, MichelineKamber, Jian Pei, Data Mining: Concepts and Techniques: Concepts and Techniques, Elsevier, 2011.
4. Ferdinand van der Heijden, Robert Duin, Dick de Ridder, David M. J. Tax, Classification,Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB, John Wiley & Sons, 2005.

Course Code: 24ESE014		**Course Title: Advanced Digital Systems Design**	
Course Category: PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding of the fundamentals and advanced concepts in sequential system design, asynchronous circuits, fault identification in digital switching circuits, and the design and modeling of programmable devices, enhancing their employability skills through practical exercises and discussions.

Module I **22 Hours**

Sequential Circuit Design: Analysis of Clocked Synchronous Sequential Networks (CSSN) Modelling of CSSN – State Stable Assignment and Reduction – Design of CSSN – ASM Chart – ASM Realization.

Asynchronous Sequential Circuit Design : Analysis of Asynchronous Sequential Circuit (ASC) – Flow Table Reduction – Races in ASC – State – Design of ASC – Static and Dynamic Hazards – Essential Hazards – Designing Vending Machine Controller

Module II **23 Hours**

Fault Diagnosis and Testability Algorithms : Fault Table Method – Path Sensitization Method – Boolean Difference Method – Kohavi Algorithm – Tolerance Techniques-Built-in Self Test.

Synchronous Design Using Programmable Devices :

Architecture of EPLD, Programmable Electrically Erasable Logic - Realize combinational, Arithmetic, Sequential Circuit with Programmable Array Logic; Architecture and application of Field Programmable Logic Sequence.

Architectures and Programming Programmable Logic Devices : FPGA Fundamentals– SRAM based FPGA architecture – Advanced FPGA features – FPGA selection and Design decisions - Xilinx Spartan and Virtex family.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1 : Select synchronous switching logics, with clocked circuits design remainder theorem, to solve problems.	Apply
CO2 : Select asynchronous switching logics, with clocked circuits design	Apply
CO3 : Apply the testing algorithms and fault diagnostic techniques for digital systems	Apply
CO4 : Design of computation logics of processors using IEEE standard Software Emulator on reconfigurable device like FPGAs	Analyze
CO5 : Utilize Employability and entrepreneurship capacity due to knowledge up gradation on digital circuits design, testing and programming of reconfigurable digital logic processors.	Apply

Reference Book(s):

1. Donald G. Givone, "Digital principles and Design", Tata McGraw Hill 2002.
2. By R. C. Cofer, Benjamin F. Harding, "Rapid System Prototyping with FPGAs: Accelerating the Design Process" , Elsevier, 2006.
3. Charles H. Roth Jr., "Digital Systems design using VHDL", Cengage Learning, 2010.
4. Mark Zwolinski, "Digital System Design with VHDL", Pearson Education, 2004
5. Parag K Lala, "Digital System design using PLD", BS Publications, 2003
6. John M Yarbrough, "Digital Logic applications and Design", Thomson Learning,2001
7. Nripendra N Biswas, "Logic Design Theory", Prentice Hall of India, 2001
8. Charles H. Roth Jr., "Fundamentals of Logic design", Thomson Learning, 2004.
9. John V.Oldfeild,Richard C.Dorf,"Field Programmable Gate Arrays",Wiley India Edition,2008

Course Code: 24ESE015		**Course Title: Industrial Drives for Automation**	
Course Category: : PEC		**Course Level: Mastery**	
L:T:P(Hours/Week) 3:0:0	**Credits: 3**	**Total Contact Hours: 45**	**Max Marks:100**

Course Objectives:

To provide a comprehensive understanding of electric drives, focusing on induction motor drives, vector control techniques, special drives like BLDC and PMSM, and configurations of I/O control in AC drives

Module I **22 Hours**

Dynamics of Electric Drives: Fundamental torque equation- multi-quadrant operation-nature and classification of load torques- modes of operation. Induction Motor Drives: Construction-Principle – performance characteristics – stator voltage control, frequency control, v/f control, rotor resistance control, static rotor resistance control, slip power recovery: Static Krammer drive, Static Scherbius drive.

Vector Control of Induction Motor Drives: Introduction to Park's and Clarke's transformation- Principle of vector control-Direct vector control-indirect vector control- stator flux oriented vector control- rotor flux oriented vector control– sensorless control- Direct torque control.

Module II **23 Hours**

Special Drives: BLDC-principle, controllers; PMSM-principle-PMSM flux density distribution-Controller– SynRM – principle magnetic flux density and operating point- converter VA requirements.

Configurations of I/O Control: AC drive Hardware Blocks – Control Blocks – Automatic Motor Adaptation – Parameterization of Drives (Local and Remote).Digital input and output-Analog input and Output control- word access- motion control- sequential logic control(SLC)-parameterization of different communication protocol: RS 485 – MODBUS – PROFIBUS.

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1: Analyze the dynamics of electric drives, including fundamental torque equations, multi-quadrant operation, and classification of load torques, with a focus on induction motor drives.	Analyze
CO2:Apply various control techniques for induction motor drives	Apply
CO3: Make use of the the special drives and configurations of i/o control for a suitable application.	Apply

Reference Book(s):

1. Gopal K Dubey, "Fundamentals of Electric Drives", Narosa Publishing House, New Delhi, 2005
2. Bimal K Bose, "Power Electronics and Variable Frequency Drives - Technology and Application", Wiley, 2010.
3. Peter Vas, "Vector Control of AC Machines", Oxford University Press, 1990
4. John Park, Steve Mackey and Edwin Wright, "Data Communications for Instrumentation and Control", Elsevier 2003
5. Ned Mohan, “Advanced Electric Drives: Analysis, Control and Modeling using Simulink”, John Wiley and Sons Ltd, 2001.