COEP Technological University Pune

(A Unitary Public University of Govt. of Maharashtra)

NEP 2020 Compliant

Proposed Curriculum Structure

M. Tech.
Electronics – Embedded System and Computing

(Effective from: A.Y. 2024-25)

PG Program [M. Tech. Electronics – Embedded System and Computing]
Proposed Curriculum Structure
W. e. f AY 2024-25

List of Abbreviations

Abbreviation	Title	No of courses	Credits	% of Credits
PSMC	Program Specific Mathematics Course	1	4	5.88 %
PSBC	Program Specific Bridge Course	1	3	4.41 %
PCC	Program Core Course	5	15	22.06 %
PEC	Program Specific Elective Course	3	9	13.24 %
LC	Laboratory Course	5	8	11.76 %
VSEC	Vocational and Skill Enhancement Course	2	18	26.47 %
OE	Open Elective	1	3	4.41 %
SLC	Self-Learning Course	2	6	8.82 %
AEC	Ability Enhancement Course	1	1	1.47 %
MLC	Mandatory Learning Course	2	--	--
CCA	Co-curricular & Extracurricular Activities	1	1	1.47 %
	Total	**25**	**68**	**100%**

PG Program [M. Tech. Electronics – Embedded System and Computing]
Proposed Curriculum Structure

Semester I

Sr. No.	Course Category	Course Code	Course Name	Teaching Scheme				Credits
				L	T	P	S	
1.	PSMC	PSMC-01	Statistics, Probability, Graph and Field Theory	3	1	--	1	4
2.	PSBC	PSBC-01	Software Tools for Embedded system and Edge computing	3	--	2	2	4
3.	PCC	PCC-01	IoT Architecture and Computing	3	--	--	--	3
4.	PCC	PCC-02	Processors and Controllers: Architecture and application programming	3	--	--	--	3
5.	LC	LC-01	IoT Architecture and Computing Lab	--	--	3	2	2
6.	LC	LC-02	Processor and Microcontrollers Programming Lab	--	--	3	2	2
7.	AEC	AEC-01	Seminar	--	--	2	2	1
8.	PEC	PEC-01	Program Specific Elective –I a) RTL Simulation and Synthesis b) Advanced Digital Signal and Image Processing c) Hardware and Software Co-Design (Advanced Digital Design) d) IoT sensors-actuators and Communication protocols e) Automotive Embedded Product Development *	3	--	--	1	3
9.	MLC	MLC-01	Research Methodology and Intellectual Property Rights	--	--	--	2	--
10.	MLC	MLC-02	Effective Technical Communication Skills	--	--	--	1	--
			Total	**15**	**01**	**10**	**13**	**22**

- *Note: '*' Industry based Elective Courses to be offered for selective students.*

PG Program [M. Tech. Electronics -Embedded System and Computing]
Proposed Curriculum Structure

Semester II

Sr. No.	Course Category	Course Code	Course Name	Teaching Scheme				Credits
				L	T	P	S	
1.	OE	OE-01	Open Elective Networked Embedded System Design	3	--	--	1	3
2.	PCC	PCC-03	Embedded system Security	3	--	--	--	3
3.	PCC	PCC-04	Embedded Operating system	3	--	--	--	3
4.	PCC	PCC-05	Data analytics on Edge Computing	2	1	--	--	3
5.	LC	LC-03	Embedded Security Lab	--	--	2	2	1
6.	LC	LC-04	Embedded OS Lab	--	--	2	2	1
7.	LC	LC-05	Data analytics on Edge computing Lab	--	--	2	2	1
8.	PEC	PEC-02	Program Specific Elective –II a) AD-CMOS b) Cloud Computing c) AI-ML d) Automotive Embedded Hardware Development *	3	--	--	1	3
9	PEC	PEC-03	Program Specific Elective –III a) Advanced VLSI architecture b) SCADA systems Applications c) Wireless Sensor Network d) Automotive Embedded Software Development *	3	--	--	1	3
10.	CCA	CCA-01	Liberal Learning Course	--	--	2	2	1
			Total	**17**	**01**	**08**	**11**	**22**

- *Note: '*' Industry based Elective Courses to be offered for selective students.*
- Exit option to qualify for **PG Diploma in Embedded System and Computing.**
  - Eight weeks domain specific industrial internship in the month of June-July after successfully completing first year of the program.

PG Program [M. Tech.– Embedded System and Computing]

Proposed Curriculum Structure

Semester-III

Sr. No.	Course Category	Course Code	Course Name	Teaching Scheme				Credits
				L	T	P	S	
1.	VSEC	VSEC-01	Dissertation Phase – I	--	--	18	12	9
2.	SLC	SLC-01	Massive Open Online Course –I	3	--	--	3	3
			Total	**3**	**--**	**18**	**15**	**12**

Semester-IV

Sr. No.	Course Category	Course Code	Course Name	Teaching Scheme				Credits
				L	T	P	S	
1.	VSEC	VSEC-02	Dissertation Phase – II	--	--	18	12	9
2.	SLC	SLC-02	Massive Open Online Course –II	3	--	--	3	3
			Total	**3**	**--**	**18**	**15**	**12**

➢ **MOOC Courses Identified:**

- Real Time Embedded Systems
- CMOS Design
- Edge Computing
- Advanced IOT Applications

***

Semester I

(PSMC) [PSMC-01] Statistics , Probability, Graph and Field Theory

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each, End-Sem Exam – 60
Tutorials: 1 hr/week	Credits: 04

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Understand and apply the linear algebra and statistical techniques.
2. Grasp and apply Graph theory for engineering problem solving and CAD toolsdevelopments
3. Culminate operations in groups, rings and field theory towards applications indigital electronic systems.
4. Characterize random variables and its functions with probability distributions andcumulative distributions

Syllabus Contents:

Graph Theory:

Basic concepts of Graph Theory, Digraphs, Paths and Circuits, Reachability and Connectedness, Matrix representation of graphs, Subgraphs & Quotient Graphs, Isomorphic digraphs & Transitive Closure digraph, Euler's Path & Circuit- definitions and examples, Connected graphs and shortest paths: Walks, trails, paths, connected graphs, distance, cut-vertices, cut-edges, blocks, connectivity, weighted graphs, shortest path algorithms, Special classes of graphs: Bipartite graphs, line graphs, Chordal graphs, Planar graphs

Group, Rings & Fields:

Set theory Relations and Functions, Semigroups, Monoids, Subsemigroup, Submonoid, Isomorphism & Homomorphism, Fields, Integral Domain, Ring Homomorphism, Ring homomorphisms and their kernels, Ideals in commutative rings, First Isomorphism Theorem for commutative rings; the example of integers mod n, **Ab**elian groups, Quotient Groups, fields of characteristic p > 0, algebraic field extension, algebraic closure, perfect field, Galois groups of polynomials, Galois groups over rationals,, Field automorphisms. Automorphism group, decomposition of a permutation into a product of disjoint cycles, Homological algebra, polynomials, Algebraic extensions, Galois extensions, Galois group of a polynomial. Galois correspondence, Calculating Galois groups. Cyclotomic extensions, Applications to Cryptography, Basic definitions of cellular automata and symbolic dynamics

Probability and Statistics:

Definitions, conditional probability, Bayes Theorem and independence. - Random Variables: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, moment generating function, Chebyshev inequality. - Special Distributions: Discrete uniform, Binomial, Geometric,
Poisson, Exponential, Gamma, Normal distributions. - Pseudo random sequence generation with

given distribution, Functions of a Random Variable - Joint Distributions: Joint, marginal and conditional distributions, product moments, correlation, independence of random variables, bi-variate normal distribution. - Stochastic Processes: Definition and classification of stochastic processes, Poisson process - Norms, Statistical methods for ranking data

References:

1. Kolman, Bernard, Robert C. Busby, and Sharon Cutler Ross, "Discrete mathematical structure" Prentice-Hall, Inc., 2003.
2. Dummit, David Steven, and Richard M. Foote, "Abstract algebra" Vol. 3. Hoboken: Wiley,2004.
3. McIntosh, Harold V, "One dimensional Cellular Automata", Luniver Press, 2009.
4. Hoekstra, Alfons G., Jiri Kroc, and Peter MA Sloot, eds., "Simulating complex systems by Cellular Automata", Springer, 2010.
5. Steeb, Willi-Hans, "The nonlinear workbook: Chaos, fractals, cellular automata, genetic algorithms, gene expression programming, support vector machine, wavelets, hidden Markov models, fuzzy logic with C++", World Scientific Publishing Company, 2014.
6. Bondy, John Adrian, and Uppaluri Siva Ramachandra Murty, "Graph theory with applications", Vol. 290. London: Macmillan, 1976.
7. Baron, Michael, "Probability and statistics for computer scientists", Chapman and Hall/CRC,2013.
8. Web Resources http://mathworld.wolfram.com/ElementaryCellularAutomaton.html

(PSBC) [PSBC-01] Software Tools for Embedded System and Edge Computing

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1 - 20, Assignments – 30, End-Sem Exam – 50marks
Practical: 2 hrs/week	Credits: 04

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Write an embedded C application of moderate complexity.
2. Develop and analyze algorithms in C++.
3. Differentiate interpreted languages from compiled languages.
4. Acquire programming skills in core Python
5. Develop applications using scripting languages such as Python.

Syllabus Contents:

Embedded 'C' Programming:

- Bitwise operations, Dynamic memory allocation, OS services
- Linked stack and queue, Sparse matrices, Binary tree
- Interrupt handling in C, Code optimization issues
- Writing LCD drives, LED drivers, Drivers for serial port communication

- Embedded Software Development Cycle and Methods (Waterfall, Agile)

Assignments:

1. Develop a simple embedded application using C to be deployed on any hardware platform.
2. Write an LCD/LED/Serial port device driver in C.

Object Oriented Programming with Embedded C++:

- Differences between C and C++, Fundamentals of object oriented programming; OOP vs. Procedure oriented programming.
- OOP concepts: classes, objects, abstraction, polymorphism, inheritance, data binding and encapsulation.
- Basics of C++: features of C++, data types, standard I/O, arrays and strings in C++.
- Classes in C++, instantiation, creating objects and object scope, data abstraction, data encapsulation, constructors and destructors, methods and access modifiers, function and operator overloading
- Inheritance-Base and Derived classes, Inheritance types, Scope Resolution operator; polymorphism and virtual functions, exception handling

Assignments:

1. Implement a C++ program which reverses the words in a given sentence.
2. Write a complex application in C++ (e.g. CRC implementation) using Object Orientated designprinciples.

Scripting Languages:

- Overview of Scripting Languages – Python, Overview, History and Features
- Python: Variables, Operators, Decision Making, Loops, Strings, Lists, Tuples, Functions, Modules, Packages, Exceptions, Classes/Objects, Regular Expressions, CGI Programming, Database Access, Networking, Sending Email, Multithreading, XML Processing, GUI programming.

Assignments:

1. Write a program for Encryption and Decryption of messages in Python.
2. Write a Python program to read, write and append contents to the text and binary file
3. Write a simple Python script that serves a simple HTTP Response and a simple HTML Page.
4. Develop GUI for an Expression Calculator using 'tk'.

References:

1. Michael J. Pont , "Embedded C", Pearson Education, 2nd Edition, 2008
2. Martin C. Brown," Python: The Complete Reference", McGraw-Hill Education, 2001
3. A. Michael Berman, "Data structures via C++", Oxford University Press, 2002
4. Robert Sedgewick, "Algorithms in C++", Addison Wesley Publishing Company, 1999
5. Abraham Silberschatz, Peter B, Greg Gagne, "Operating System Concepts", John Willey &Sons, 2005
6. E. Balaguruswami, "Object-Oriented Programming With C++", McGraw-Hill Education, 7thEdition, 2017

PCC [PCC-01] IoT Architecture and Computing

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs./week	
Credits-3	Test I - 20 Marks
	Test II - 20 Marks
	End Sem Exam - 60 marks

Course Outcomes:

At the end of the course, students will demonstrate the ability to

1. To understand basic concepts and communication protocols of IoT
2. To understand types of technologies that can be utilized to implement IoT solutions
3. Develop IoT solutions based on popular embedded platforms to address real life problems

Syllabus Contents

Unit -1

IoT Overview: Internet of Things Framework, Architecture, Technology, Machine 2 Machine Communication (M2M)

Unit -2

Sensors, Actuators, Embedded platforms, Wireless Sensor Networks, RFID

Unit-3

Embedded Devices: Platforms, computing, Operating Systems-TinyOS, Contiki OS

Communication Protocols: IPV6, 6LOWPAN, MQTT, COAP, AMPQ, Zigbee Smart energy 2.0, Lora WAN

Unit -4

Cloud Computing, Fog computing, Big Data, NoSQL Database, Security and Privacy issues in IoT

IoT and Artificial Intelligence, Machine Learning , Blockchain, 5G Technology IoT

Unit -5

Case Study: Smart grid, Smart agriculture, Smart healthcare, Smart transportation, Smart City challenges

Textbooks:

- A Bahaga, V. Madisetti, “Internet of Things- Hands on approach”, 1st Edition, VPT publisher, 2014.
- Raj Kamal, “ Internet of Things-Architecture and Design Principles, McGraw Hill

Reference Book:

- Buyya, Rajkumar, and Amir Vahid Dastjerdi, eds. “Internet of Things: Principles and paradigms”, Elsevier, 2016.
- Hassan, Qusay F., ed. “Internet of things A to Z: technologies and applications”, John Wiley & Sons, 2018.
- Samuel Greenguard, “Internet of things”, MIT Press.

LC (LC-01] IoT Architecture and Computing Lab

Teaching Scheme	**Exam Scheme**
Practical: 2hrs/week	Term work-50 marks
	Oral-50 marks
	Credits: 2

1. Develop programming ability using C/C++/VB/Python.
2. Explore to the interconnection and integration of the real time parameters and the smart application

1) To study Arduino IDE and different types of Arduino Board.

2) Write a program to calculate the distance using ultrasonic sensor.

3) Write a program to measure room-temperature , humidity and heat index.

4) To determine the gas level/smoke using MQ2 Gas sensor

5) Write a program to detect motion using PIR sensor.

6) Write a program to detect change in displacement using accelerometer.

7) To detect the presence with the RFID Tag and Reader.

8) To study GSM/GPRS Module interfacing with Arduino.

9) To study Thingspeak cloud.

10) To design a Graphical user interface using VB/Python

(PCC) [PCC-02] Processors and Controllers: Architecture and application programming

Teaching Scheme	Examination Scheme
Lectures: 3 hrs/week	T1, T2 – 20 marks each, End-Sem Exam – 60 Credits: 03

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Employ ARM Cortex Mx processor core architectural features and its programmer's model in developing applications
2. Design and Develop code having sections with different processing priority using nested vectored interrupt controller features supported by ARM Cortex Mx processor core
3. Apply the signal processing capability native to ARM Cortex Mx processor in developing small applications.
4. Identify and utilize on-chip/onboard peripherals on microcontroller evaluation board and their corresponding memory mapped addresses
5. Knowhow of advanced edge computing and security features supported by recently introduced ARM Cortex Mx processors.

Syllabus Contents:

Unit 1: ARM Cortex-M3/M4 processor

Applications, Programming model - Registers, Operation modes, Exceptions and Interrupts, Reset Sequence, Introduction to the Tiva microcontrollers, Tiva block diagram, System Clocks and Control, Hibernation Module on Tiva

Unit 2: Overview of Instruction set and Memory System

Instruction Set, Unified Assembler Language, Memory Maps, Memory Access Attributes, Permissions, Bit-Band Operations, Unaligned and Exclusive Transfers. Coretx-M4 specific instructions, Barrel shifter, Pipeline, Bus Interfaces

Unit 3: Nested vectored Interrupt Controller

Exceptions, Types, Priority, Vector Tables, Interrupt Inputs and Pending behavior, Fault Exceptions, Supervisor and Pendable Service Call, Nested Vectored Interrupt Controller, Basic Configuration, SYSTICK Timer, Interrupt Sequences, Exits, Tail Chaining, Interrupt Latency

Unit 4: Floating Point Unit and DSP Support

Floating point operations, Floating point unit (FPU), Registers, Lazy stacking, DSP on a Micro-controller, Architecture of traditional DSP processor, DSP instructions – fractional arithmetic, SIMD data, load store instructions, arithmetic instructions, optimization strategies, DSP applications

Unit 5: TM4C123x micro-controller:

Internal memory, GPIOs, Timers, ADC, UART and other serial interfaces, PWM, RTC, WDT

Unit 6: Introduction to ARM Cortex M-55/85 processor:

Architectural features supporting enhanced, energy-efficient digital signal processing (DSP) and machine learning (ML) performance, Enhanced software security with TrustZone and PACBTI extension.

References:

1. Joseph Yiu, "The Definitive Guide to ARM Cortex M3 and M4 Processor",Newnes, Third Edition, 2013
2. Cem Unsalan and Huseyin Deniz Gurhan Mehmet ErkjingYucel, "Embedded System Design with ARM Cortex – M Microcontrollers", Springer, First Edition, 2022
3. Jonathan W Valvano, "Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers", Createspace publications ISBN: 978-1463590154, 2014.
4. Jon Marsh, Arm Helium Technology M-Profile Vector Extension (MVE) for Arm Cortex-M Processors, ISBN: 978-1-911531-23-4, arm Education Media
5. Technical references and user manuals on www.arm.com, www.ti.com

(LC) [LC -02 Processor and controllers Programming Lab

Teaching Scheme	Examination Scheme
Tutorials: 1 hr/week Practical: 3 hrs/week	Marks- 100 Credits: 02

Laboratory Outcomes:

At the end of the laboratory work, students will demonstrate the ability to:

1. Deal and develop applications utilizing components at different abstraction levels viz. HAL, Middle-ware, Application layer
2. Develop applications using peripheral interfaced to several asynchronous and synchronous communication buses.

List of Experiments:

Experiments to be carried out on Tiva Launch-pads:

1. Blink an LED with software delay, delay generated using the SysTick timer.
2. System clock real time alteration using the PLL modules.
3. Control intensity of an LED using PWM implemented in software and hardware.
4. Control an LED using switch by polling method, by interrupt method and flash the LED once every five switch presses.
5. UART Echo Test.
6. Take analog readings on rotation of rotary potentiometer connected to an ADC channel.
7. Program - Read, Write and Verify - serial EEPROM using I2C interface.
8. Calling C function from assembly program and vice versa.
9. Evaluate the various sleep modes by putting core in sleep and deep sleep modes.
10. System reset using watchdog timer in case something goes wrong.
11. To develop an assembly code and C code to compute Euclidian distance between any two points
12. To develop assembly and C code for implementation of convolution operation
13. To design and implement filters in C to enhance the features of given input sequence/signal

(PEC) [PEC-01] RTL Simulation and Synthesis

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each, End-Sem Exam – 60 Credits: 03

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Understand proven design methodologies based on standard EDA tools
2. Design combinational devices with full set of EDA tools.
3. Design, Analyze and Verify the synchronous digital design on FPGAs
4. Design state machines, datapath controllers , and assorted CPUs with a full set of EDA tools.

Syllabus Contents:

Top down approach, Hardware modelling of combinational and sequential circuits with verilog HDL, writing a test bench

Design of finite state machines (Synchronous and asynchronous), system design using ASM chart, Static Timing analysis, Meta-stability, clock issues, Need and design strategies for Multi-clock Domain designs.

Data path and Control path design, Arithmetic implementation strategies for data path design, Processor Design, Micro-programmed control design, Single cycle MMIPS

Programmable Logic Devices: Fine grained and coarse grained FPGA , Xilinx series

ASIC Design flow- Introduction to ASIC Design Flow, SOC, Floor planning, Placement, Clock tree synthesis, Routing, Physical verification, Power analysis, Design for performance, Low power VLSI design techniques, Technology Challenges

IP and Prototyping.

References:

1. Stephen Brown and ZvonkoVranesic, "Fundamentals of Digital logic with Verilog Design", Mc-GrawHill, 3rd edition
2. Donald D Givone, "Digital principles and Design", Tata Mc-Graw Hill, 2003
3. David Harris and Sarah Harris, "Digital Design and Computer Architecture", Morgan Kaufmann, 2nd edition, 2012
4. Doug Amos, Austin Lesea, Rene Richter, "FPGA based prototyping methodology manual",Xilinx
5. Douglas Smith, "HDL Chip Design: A Practical Guide for Designing, Synthesizing & Simulating ASICs & FPGAs Using VHDL or Verilog", Doone Publications.
6. IEEE standard HDL based on Verilog HDL, published by IEEE.
7. Ben Cohen, "Real Chip design and Verification using Verilog and VHDL", Vhdl Cohen Publishing, 2002

(PEC) [PEC-02] Advanced Digital Design

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each, End-Sem Exam – 60 Credits: 03

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Design Finite state machines and perform timing analysis
2. Model, simulate, verify the synchronous digital design with Verilog HDL
3. Design, Analyze and Verify the synchronous digital design on FPGAs
4. Understand the concept of CPLD, FPGA, SoC and IP

Syllabus Contents:

1. Review of Combinational and Sequential Digital Logic Design

2. Basic Verilog Language Structures (Datatypes, Modules, etc.) – Datatypes: nets, registers, event, bitvectors, arrays, parameters – Modules: ports, hierarchical names

3. Structural and Behavioral Specifications – Basic gates, User-defined primitives, Modeling levels – Synthesizable operations, Continuous assignments (Examples: Adders, ALU)

4. Simulation. Testbenches and debugging.

5. Synthesis flow. Synthesis to Standard cells and FPGA.

6. Procedural Specifications and Designing Single Modules – The always block – Functions and Tasks – Blocking and Non-blocking assignments – Control constructs and their Synthesis – Design examples: Counters, Unsigned Multiplier – Validation: Verification Vectors, Testbench Coding Approaches, Post-synthesis verification

7. Finite State Machine Specifications and Styles – Explicit and Implicit Specification Styles – Example: Booth multiplier – Example: First-in-First-Out buffer (FIFO)

8. Design Reuse – Instantiation of parametrized modules. – Control-point style for design reuse (Examples with FIFO) – Using vendor components (Booth multiplier)

9. Improving Timing, Area, and Power – Delay calculations – Timing design with Flip-flops and Latches – Low-power design issues and Area considerations.

Text Books:

1. Michael D. Celetti: Advanced Digital Design with the Verilog HDL, PHI, 2013

Reference Books: 1. PeterJ. Asheden: Degital Design –An Embedded Systems Approach Using VERILOG, ELSEVIER 2013. 2. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic with Verilog Design, Tata Mc-Graw Hill 2009.

(PEC) [PEC-01] Advanced Digital Signal and Image Processing

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each, End-Sem Exam – 60 Credits:03

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Understand and apply knowledge of various transforms and probability theory insignal and image processing
2. Perform fundamental operations in digital image processing like enhancement,encoding, feature extraction, and segmentation.
3. Analyze, apply and critically evaluate various signal/image processing algorithmsappropriate for practical applications.

Syllabus Contents:

Review of Discrete Time signals and systems:
Characterization in time and Z and Fourier – domain, Fast Fourier Transform algorithms – In-place computations, Butterfly computations, bit reversals.

Digital Filter design:
FIR and IIR filters–Impulse invariance, bilinear transformation

Representation of Signal Processing Algorithms:
Signal flow, Data flow and Dependence Graphs. Iteration Bound, algorithms for computing iteration bound, Pipelining, Parallel processing.

Introduction to Image Processing:
Applications and fields of image processing, Fundamental steps in Digital image processing, Elements of visual perception, Image sensing and acquisition, Basic Concepts in Sampling and Quantization, representing digital images.

Image Enhancement:
Some basic gray level transformations, Histogram Processing, Sharpening Spatial filters, Image Enhancement in the spatial and Frequency domain, Pseudo-colouring

Segmentation:
Some Basic Relationships between pixels, point, Edge based segmentation, Boundary detection, extraction and representation, Threshold based segmentation, Region based segmentation, Texture based segmentation. Morphological operations

Image Compression:
Data redundancies Variable-length coding, Predictive coding, Transform coding, Image compression standards.

Case studies:

VLSI architectures for implementation of Image Processing algorithms

References:

1. J.G. Proakis, Manolakis "Digital Signal Processing", Pearson, 4th Edition
2. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson , 4rd Edition
3. Keshab Parhi, "VLSI Digital Signal Processing Systems – Design and Implementation", WileyIndia

(MLC) [MLC-01] Research Methodology and Intellectual Property Rights

Teaching Scheme	**Examination Scheme**
Lectures: 2 hrs/week	Continuous evaluation: Assignments/Presentations/Quizzes/Tests

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Understand research problem formulation and approaches of investigation ofsolutions for research problems
2. Learn ethical practices to be followed in research and apply research methodologyin case studies and acquire skills required for presentation of research outcomes
3. Discover how IPR is regarded as a source of national wealth and mark of aneconomic leadership in context of global market scenario
4. Summarize that it is an incentive for further research work and investment in R & D,leading to creation of new and better products and generation of economic and social benefits

Syllabus Contents:

Unit 1: [5 Hrs]

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem.
Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, necessary instrumentations.

Unit 2: [5 Hrs]

Effective literature studies approaches, analysis
Use Design of Experiments /Taguchi Method to plan a set of experiments or simulations or build prototype
Analyze your results and draw conclusions or Build Prototype, Test and Redesign

Unit 3: [5 Hrs]

Plagiarism, Research ethics
Effective technical writing, how to write report, Paper.
Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee

Unit 4 : [4 Hrs]

Introduction to the concepts Property and Intellectual Property, Nature and Importance of Intellectual Property Rights, Objectives and Importance of understanding Intellectual Property Rights

Unit 5 : [7 Hrs]

Understanding the types of Intellectual Property Rights: -Patents-Indian Patent Office and its Administration, Administration of Patent System – Patenting under Indian Patent Act, Patent Rights and its Scope, Licensing and transfer of technology, Patent information and database. Provisional and Non Provisional Patent Application and Specification, Plant Patenting, Idea Patenting

Integrated Circuits, Industrial Designs, Trademarks (Registered and unregistered trademarks), Copyrights, Traditional Knowledge, Geographical Indications, Trade Secrets, Case Studies

Unit 6 : [4 Hrs]

New Developments in IPR, Process of Patenting and Development: technological research, innovation, patenting, development

International Scenario: WIPO, TRIPs, Patenting under PCT

References:

1. Aswani Kumar Bansal, “Law of Trademarks in India”
2. B L Wadehr, “Law Relating to Patents, Trademarks, Copyright, Designs and GeographicalIndications”
3. G.V.G Krishnamurthy, “The Law of Trademarks, Copyright, Patents and Design”
4. Satyawrat Ponkse, “The Management of Intellectual Property”
5. S K Roy Chaudhary & H K Saharay, “The Law of Trademarks, Copyright, Patents”
6. T. Ramappa, S. Chand, “Intellectual Property Rights under WTO”
7. Manual of “Patent Office Practice and Procedure”
8. WIPO: “WIPO Guide To Using Patent Information”
9. Halbert , “Resisting Intellectual Property”, Taylor & Francis
10. Mayall , “Industrial Design”, Mc Graw Hill
11. Niebel, “Product Design”, Mc Graw Hill
12. Asimov , “Introduction to Design”, Prentice Hall
13. Robert P. Merges, Peter S. Menell, Mark A. Lemley, “Intellectual Property in New Technological Age”

(MLC) [MLC-02] Effective Technical Communication Skills

Teaching Scheme	**Examination Scheme**
Lectures: 1 hr/week	Marks: 100 (4 Assignments - 25 Marks each) Credits: --

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Produce effective dialogue for business related situations
2. Use listening, speaking, reading and writing skills for communication purposes andattempt tasks by using functional grammar and vocabulary effectively
3. Analyze critically different concepts / principles of communication skills
4. Demonstrate productive skills and have a knack for structured conversations
5. Appreciate, analyze, evaluate business reports and research papers

Syllabus Contents:

Unit 1: Fundamentals of Communication [4 Hrs]

7 Cs of communication, common errors in English, enriching vocabulary, styles and registers

Unit 2: Aural-Oral Communication [4 Hrs]

The art of listening, stress and intonation, group discussion, oral presentation skills

Unit 3: Reading and Writing [4 Hrs]

Types of reading, effective writing, business correspondence, interpretation of technical reportsand research papers

References:

1. Raman Sharma, "Technical Communication", Oxford University Press.
2. Raymond Murphy, "Essential English Grammar" (Elementary & Intermediate) Cambridge University Press
3. Mark Hancock, "English Pronunciation in Use", Cambridge University Press
4. Shirley Taylor, "Model Business Letters, Emails and Other Business Documents", PrenticeHall, Seventh Edition
5. Thomas Huckin, Leslie Olsen, "Technical writing and Professional Communications for Non-native speakers of English", McGraw Hill

(LC) [LC-01] Seminar

Teaching Scheme	**Examination Scheme**
Practical: 2 hrs/week	Marks- 100 Credits: 01

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Identify contemporary topics/concepts pertaining to VLSI and Embedded Systemsand prepare documentation with improved substance.
2. Present the selected topic with superiority demonstrating good communication skills

Guidelines:

Selection of Topic:

- Select a topic relevant to the stream of study with content suitable for M. Tech. level presentation. For selection topics refer internationally reputed journals. The primary referenceshould be published during the last two or three years.
- Some of the journals/publications suitable for reference are: IEEE/the IET/IETE/Springer/Science Direct/ACM journals in the areas of Computing and Processing (Hardware and Software), Circuits-Devices and Systems, Communication - Networking andSecurity, Robotics and Control Systems, Signal Processing and Analysis, Machine Learning, IoT and any other related domain
- Get the topic approved by the seminar guide well in advance.

Preparation of Presentation and Report:

- In slides, list out key point only. You may include figures, charts equations tables etc. but notrunning paragraphs. Font size used should be at least 20.
- Figures should be very clear and possibly drawn by you using suitable software tools. Thereshould be a slide on “Conclusion”.
- A report of the seminar should be prepared which should contain the following.
  - Title of the seminar.
  - Name and other details of presenter and the guide.
  - Abstract of the topic.
  - Contents such as Introduction, Theory to elaborate the concept, Implementation if carriedout by the presenter/or fellow researcher/s, Comparison with other relevant techniques, Conclusion etc.
  - List of references strictly in IEEE format.

Oral Presentation:

- Student needs to orally present the topic for 20 minutes with good voice projection and withmodest pace

Answering Queries:

- Student needs to answer queries raised by the audience and evaluators. This session shall berestricted to 5 minutes. In case of more queries, student is supposed to solve the queries offline

Semester II

(OE) [OE-01] Networked Embedded System Design

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each, End-Sem Exam – 60 Credits-03

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Implement simple sketches on the Arduino boards involving several peripherals
2. Identify, design and implement applications on the Arduino boards producingcustom shields.
3. Deploy low end applications using low and high level languages on microcontrollerplatform.

Syllabus Contents:

Introduction to processors:
Introduction of Microprocessors and Microcontrollers, Introduction of Arduino Microcontrollers.

Introduction to architecture:
Atmega328: Basics and Architecture, Instruction Set

Arduino programming:
Arduino programming basics, Analog/Digital components and its application with Arduino, IDE forArduino.

Other utilities in Arduino:
Timers, Analog comparators and hardware interrupts

Interfacing with peripherals:
Communication buses, Interfacing of I/O devices

Case studies:
Case studies of a few projects using Arduino boards and Shields

References:

1. Brian Evans, “Beginning Arduino Programming”, Springer, 2011
2. Michael J. Pont , “Embedded C”, Pearson Education, 2nd Edition, 2008
3. Raj Kamal, “ Embedded Systems – Architecture: Programming and Design”, TMH
4. Frank Vahid and Tony Givargis, “Embedded System Design”, Wiley

PCC [PCC-03] Embedded Systems Security

Teaching Scheme	Examination Scheme
Lectures: 3 hrs/week	T1, T2 – 20 marks each, End-Sem Exam – 60 Credits: 03

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Recognize vulnerabilities, attacks and need of protection mechanisms forembedded systems
2. Analyze and evaluate software vulnerabilities and attacks on operating systems
3. Identify terms/concepts relevant to embedded cryptography.
4. Develop and deploy solutions for security of embedded software and dataprotection.

Syllabus Contents:

Introduction to Embedded Systems Security: Trends, Policies, Threats

Systems Software Considerations:
Role of the Operating System, Multiple Independent Levels of Security, Microkernel versus Monolith, Core Embedded Operating System Security Requirements, Access Control and Capabilities, Hypervisors and System Virtualization, I/O Virtualization, Remote Management, Assuring Integrity of the TCB

Secure Embedded Software Development:
PHASE—Principles of High-Assurance Software Engineering, Minimal Implementation, Component Architecture, Least Privilege, Secure Development Process, Independent Expert Validation, Case Study: HAWS—High-Assurance Web Server, Model-Driven Design

Embedded Cryptography:
Cryptographic Modes, Block Ciphers, Authenticated Encryption, Public Key Cryptography, Key Agreement, Public Key Authentication, Elliptic Curve Cryptography, Cryptographic Hashes, Message Authentication Codes, Random Number Generation, Key Management for Embedded Systems, Cryptographic Certifications

Data Protection Protocols for Embedded Systems:

Data-in-Motion Protocols, Data-at-Rest Protocols

Emerging Applications: Embedded Network Transactions, Automotive Security, Secure Android, next generation software defined radio.

References:

1. David Kleidermacher and Mike Kleidermacher, “Embedded Systems Security”, Elsevier
2. Gebotys, Catherine H., “Security in Embedded Devices”, Springer
3. Stapko T., “Practical Embedded Security”, Elsevier/Newnes

(PCC) [PCC-04] Embedded Operating System

Teaching Scheme Lectures: 3 hrs/week	**Examination Scheme** T1, T2 – 30 marks each, End-Sem Exam – 40 Credits - 03

Course Outcomes

At the end of the course, students shall demonstrate the ability to

A. Interact with and control the board using simple Linux commands.
B. Perform Linux system administration tasks on the Beagle boards and use a range of Linux commands for file and process management
C. Compare and contrast scripting, hybrid, and compiled programming languages, and their application to the embedded Linux applications
D. Develop shell scripts and C++ code to control a Beagle board GPIOs, PWM pin and read from analog inputs
E. Implement C/C++ code that interfaces to and wraps the functionality of devices attached to the UART, I^{2}C and SPI buses
F. Build circuits that interface to the Beagle board CAN buses and use LinuxSocketCAN to send and receive messages to and from the bus
G. Interface actuators, analog sensors, low-cost display modules to Beagle boards

Syllabus Contents

Module 1: Beagle Board Basics

Beagle hardware, accessories, capes, handling the board, Beagle software, Linux on Beagle board, Communicating with the board, Controlling the board, basic Linux commands, package management, Updating the kernel, Shutdown

Module 2: Exploring Embedded Linux System

Introduction, Booting, Kernel space, User space, Managing Linux system, System administration, Linux commands, Processes

Module 3: Beagle Board Programming

Performance of different languages, setting clock frequency, Scripting languages, C and C++ on Beagle boards, LED flashing program using C and C++, Writing multi-call binary, Interfacing to Linux OS – Glibc and Syscall

Module4: Interfacing to Beagle Board Inputs/Outputs

GPIOs, Introduction, Configuration, Linux device tree, Analog Inputs and Outputs, examples such as light meters, controlling servo motors, Callback functions, POSIX threads, Linux poll

Module 5: Interfacing to Beagle board Buses

Introduction, I2C Hardware, Test circuit, I2C tools, I2C communication in C and C++, SPI Hardware, Testing, Bidirectional SPI communication in C and C++, Multiple Slave devices, UART, examples LED serial driver, application such as GPS, CAN bus, SocketCAN, Linux CAN utils

Module 6: Interfacing with Physical Environment

Interfacing to actuators – DC motors, Stepper motors, Relays, Interfacing to analog sensors, protecting ADC inputs, Signal conditioning, Interfacing local displays, Building C/C++ libraries – shared and static

References

1. Derek Molloy, Exploring Beaglebone Tools and Techniques for building with Embedded Linux, Wiley, Second Edition, 2019
2. W. Richard Stevens and Stephen A. Rago, Advanced Programming in Unix Enviornment, Pearson, Second edition, 2011
3. Robert Love, Linux System Programming, O'Reilly, Second Edition, 2013
4. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci – Dusseau, Operating System: Three Easy Pieces, Version 1.0, 2018

Embedded OS Lab

Teaching Scheme	**Examination Scheme**
Practicals: 2 hrs/week	In semester evaluation: 50 Marks End semester evaluation: 50 Marks

Course Outcomes

At the end of the course, students shall demonstrate the ability to

A) Build and boot the board with custom boot loader, Linux kernel and root file system.

B) Develop applications using user interfaces and device files supported by the drivers managing GPIOs, ADC, PWM and Asynchronous and synchronous communication buses.

List of Experiments:

Experiments to be carried out on Beaglebone boards and suitable open source tool-chain

1) Booting the board with latest headless image available online and interacting with the board using virtual serial and Ethernet over USB buses.
2) Experimenting with the booting process with step by step loading of multistage booloader files, kernel image, DTB files and root file system.
3) Creating and using custom root files system using the Busybox
4) Building the cross tool chain, bootloader, kenel image, DTB files and root file system using the Builroot.
5) Installing, configuring the latest Eclispe IDE for development and debugging of applications.
6) Interfacing and programming onboard and off-board peripherals connected to GPIO lines
7) Programming ADCs and PWM peripherals to realize a small real-word application.
8) Programming UART, I2C and SPI buses to carry out transactions with devices connected.

PCC [PCC-04] Data Analytics on IoT	
Teaching Scheme Lectures: 3 hrs/week	**Examination Scheme** T1, T2 – 20 marks each, End-Sem Exam – 60 Credits: 03

Course objective:

1. Understand the basics of organization of big data

2. Analyze the architectural issues of big data tools.

3. Design and implement various search methods and visualization techniques for big data analytics

Course Content:

Unit-I
Fundamentals of Big-data analytics, Overview & analytics life cycle, Need, Structured and multistructured data analysis, Big- data analytics major components, Analytical models and approaches, Big data challenges.

Unit-II
Designing and building big data applications, Big data architecture, Distributed Computing platforms and Data Storage, Security and Data Privacy, Application Areas, Application Tools and Platforms.

Unit-III

Clustered Hadoop environment, HDFS and data managements using HDFS, Analytics Using Map Reduce and programming, Map Reduce design patterns. Unit-IV Introduction to Modern databases-No SQL, New SQL, No SQLVs RDBMS databases Tradeoffs, Working with MongoDb, Data warehouse system for Hadoop

Unit-V
Introduction to Pig and HIVE- Programming Pig: Engine for executing data flows in parallel on Hadoop, Programming with Hive.

TextBook:

1. Dean Wampler, Jason Rutherglen, Edward Capriolo,“Programming Hive” O'Reilly Media, 1 stEdition, 2012.

2. Sawant, Nitin, Shah, Himanshu, “Big Data Application Architecture Q & A: A Problem-Solution Approach”, Apress, 1st Edition, 2013.

Reference Books
1. David Loshin, “Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph”, Morgan Kaufmann, 1stEdition, 2013.

2. Jonathan R. Owens, Jon Lentz, Brian Femiano, “Hadoop Real, World Solutions Cookbook”, Packt, 2nd Edition, 2016.

(LLC) [] Liberal Learning Course

Teaching Scheme Contact Period: 1 hr/week	**Examination Scheme** T1, T2 – 20 marks each, End-Sem Exam – 60 Credits: 01

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Learn new topics from various disciplines without any structured teaching ortutoring
2. Understand qualitative attributes of a good learner
3. Understand quantitative measurements of learning approaches and learning styles
4. Understand various sources and avenues to harvest/gather information
5. Assess yourself at various stages of learning

Course Features:

- 10 Areas, Sub areas in each
- Voluntary selection
- Areas (Sub areas):

1. Agriculture (Landscaping, Farming, etc.)
2. Business (Management, Entrepreneurship, etc.)

3. Defense (Study about functioning of Armed Forces)
4. Education (Education system, Policies, Importance, etc.)
5. Fine Arts (Painting, Sculpting, Sketching, etc.)
6. Linguistics
7. Medicine and Health (Diseases, Remedies, Nutrition, Dietetics, etc.)
8. Performing Arts (Music, Dance, Instruments, Drama, etc.)
9. Philosophy
10. Social Sciences (History, Political Science, Archaeology, Geography, Civics, Economics, etc.)

Evaluation:

- **T1:** A brief format about your reason for selecting the area, sub area, topic and a list of 5 questions (20 marks)
- **T2:** Identify and meet an expert (in or outside college) in your choice of topic and give a write up about their ideas regarding your topic (video /audio recording of your conversation permitted (20 marks)
- **ESE:** Presentation in the form of PPT, demonstration, performance, charts, etc. in front of everyone involved in your sub area and one external expert (60 marks)

Resources:

- Expert (s), Books, Texts, Newspaper, Magazines, Research Papers, Journal, Discussion withpeers or faculty, Internet, etc.

Semester III and IV

SBC Dissertation Phase – I and II

Teaching Scheme	**Examination Scheme**
--	Marks: For both phase I and II Mid-sem: 30 Marks End-sem: 70 Marks

Course Outcomes:

At the end of the course, students will demonstrate the ability to:

1. Conceive a problem statement either from rigorous literature survey or from therequirements raised by external entity.
2. Design, implement and test the prototype/algorithm in order to solve theconceived problem.
3. Publish the research work in journals/conferences of repute contributing to growthof technology in the domain.

Guidelines:

As per the AICTE directives, the dissertation is a yearlong activity, to be carried out and evaluated intwo phases i.e. Phase – I: July to December and Phase – II: January to June.

The dissertation may be carried out preferably in-house i.e. department's laboratories and centers OR in industry allotted through department's T & P coordinator.

After multiple interactions with guide and based on comprehensive literature survey, the student shall identify the domain and define dissertation objectives. The referred literature should preferably include IEEE/IET/IETE/Springer/Science Direct/ACM journals in the areas of Computing and Processing (Hardware and Software), Circuits-Devices and Systems, Communication- Networking and Security, Robotics and Control Systems, Signal Processing and Analysis, Machine Learning, IoT and any other related domain. In case of Industry sponsored projects, the relevant application notes, white papers, product catalogues should be referred and reported.

Student is expected to detail out specifications, methodology, resources required, critical issues involved in design and implementation and phase wise work distribution, and submit the proposal within a month from the date of registration.

Phase – I deliverables:

A document report comprising of summary of literature survey, detailed objectives, project specifications, paper and/or computer aided design, proof of concept/functionality, part results, A record of continuous progress.

Phase – I evaluation:

A committee comprising of guides of respective specialization shall assess the progress/performance of the student based on report, presentation and Q & A. In case of unsatisfactory performance, committee may recommend repeating the phase-I work.

During phase – II, student is expected to exert on design, development and testing of the proposed work as per the schedule. Accomplished results/contributions/innovations should be published in terms of research papers in reputed journals and reviewed focused conferences OR IP/Patents.

Phase – II deliverables:
A dissertation report as per the specified format, developed system in the form of hardware and/orsoftware, A record of continuous progress.

Phase – II evaluation:
Guide along with appointed external examiner shall assess the progress/performance of the student based on report, presentation and Q & A. In case of unsatisfactory performance, committee may recommend for extension or repeating the work.

***