MTECH SYLLABUS - COMPUTER SCIENCE WITH SPECIALIZATION IN EMBEDDED SYSTEMS

M.Tech. Computer Science with Specialization in Embedded Systems

Semester - I

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSE 3101	Embedded Computer Architecture	C	4	4	0	100
2	CSE 3102	Real-Time Operating Systems	C	4	3	2	100
3	CSE 3103	Software Engineering Principles	C	4	4	0	100
			Total for Semester I	12	11	2	300

Semester - II

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSE 3201	Model Driven Real-time Software	C	4	4	0	100
2	CSE 3202	Programming Embedded Systems	C	4	3	2	100
3		Elective I	E	4	4	0	100
			Total for Semester II	12	11	2	300

Electives

CSE 3203: Networked Embedded Systems

CSE 3204: Software Reliability Engineering

Semester - III

1	CSE 3301	Real-Time Digital Signal Processing	C	4	3	2	100
2	CSE 3302	Real-Time Design Patterns	C	4	4	0	100
3		Elective II	E	4	4	0	100
			Total for Semester III	12	11	2	300

Electives

CSE 3303: Control Systems Engineering

CSE 3304: High Performance Embedded Computing

CSE 3305: Embedded Communications Software

CSE 3306: Massively Parallel Programming

Semester - IV

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSE 3401	Project & Viva Voce	C	18	0	15	300

Semester -V

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSE 3501	Project & Viva Voce	C	18	0	15	300

Total credits for Degree: **72**

CSE3101: EMBEDDED COMPUTER ARCHITECTURE

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

This course provides a detailed examination of the internal structure and operation of embedded computer systems. Embedded system technologies including processors, DSP, memory and software are explained. Performance evaluation and optimization techniques are also discussed

Course Objectives

To Know the characteristics of embedded systems dependent on application domain
To Know the mechanisms behind the state-of-the-art architectures for embedded systems
To Evaluate the features of the various architectures
To Select the embedded architecture which is most suitable for different applications
To Study performance evaluation and optimization methods

Profile

1. Definition of embedded system - Constraints on embedded vs. standalone systems - Concept of real-time design - Time scales for real-time systems - Embedded system design processes- Applications - Embedded Systems design process – Formalisms for System Design - Design example

2. Overview of computer architecture – Embedded Processors - Case studies of INTEL, MOTOROLA, RISC and DSP – Embedded Computing Platform – CPU Bus – Memory systems: Bus Snooping – Basic Peripherals: PC Bus, UART – Interrupts and exceptions - Component interfacing – Designing with microprocessors – Development and Debugging

3. Program design and Analysis – Introduction to Program design – Models of programs – Assembly and Linking – Basic compilation techniques – Optimization: Time, Energy, Power, Size

4. Processes and Operating systems – Multiple tasks and multiple processes – Context switching – Scheduling Policies – Interprocess communication- Device drivers – Embedded Operating systems – POSIX standard

5. Evaluating system performance: Correctness - Speed - Profiling system performance - Performance optimization: Hand-optimization - Optimizing compilers -Pareto principle

REFERNCES

1. Wayne Wolf “Computers as Components: Principles of Embedded Computer Systems Design” Morgan Kaufmann; 1st edition (September 2000) ISBN-13: 978-1558605411
2. Steve Heath “Embedded Systems Design” Newnes (2002) ISBN-13: 978-0750655460
3. Tammy Noergaard “Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers” Newnes (February 24, 2005) ISBN-13: 978-0750677929
4. Raj Kamal “Embedded Systems: Architecture, Programming and Design” McGraw-Hill Education (India); 2nd Edition edition (March 9, 2009) ISBN-13: 978-0070151253
5. K V K K Prasad “Embedded/Realtime Systems: Concepts, Design and Programming”

Dreamtech Press (2004) ISBN-13:9788177224610,978-8177224610

CSE3102: REAL TIME OPERATING SYSTEMS

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

This course provides an introduction to the theory, structure and practice of real-time operating systems. Particular emphasis is placed on issues related to embedded applications. The course covers basic concepts, practical issues and commercial implementations of modern operating systems.

Course Objectives

To Understand the fundamental concepts of real-time operating systems
To Appreciate the use of multitasking techniques in real-time systems
To Understand the features and structures of practical implementations
To Appreciate how application areas impact on RTOS facilities

Profile

1. Real-Time System concepts: Shared Resources – Multitasking – kernels – Schedulers – task priorities – mutual exclusion – deadlock – inter task communication – kernel objects

2. Timer and timer services: real time clocks – Timer ISRs – Soft timers – I/O subsystem – Memory management techniques – Concurrency – synchronization and communication

3. File system, device and memory management aspects – Performance and safety features – Real-Time POSIX issues

4. Structured Design for Real-time Systems - Designing for Multitasking; UML for Real-time Systems - System Integrity - Selecting Languages for RTS Development - C, Ada & Java - Cross Development Techniques

5. Case studies of RealTime Embedded operating systems – Montavista Linux, QNX, VxWorks, WindowsCE – eCos

REFERNCES

1. Qing Li and Caroline Yao “Real-Time Concepts for Embedded Systems” CMP Books; 1st edition (July 2003) ISBN-13: 978-1578201242
2. Rob Williams “Real-Time Systems Development” BUTTERWORTH HEINEMANN(2005) ISBN-13: 978-0-7506-6471-4

CSE3103: SOFTWARE ENGINEERING PRINCIPLES

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, as well as maintenance of software, and the study of these approaches. This course is concerned with all aspects of software production; developed programs and associated documentation.

Course Objectives

To familiarize principles, concepts and practice of software engineering.
To familiarize the methods and processes of constructing the different types of software Systems
To Apply techniques and tools of software engineering within the context of systematic construction of quality software

Profile

1. Software life cycle – Models – Requirements engineering – Requirement analysis – Requirement specifications - Formal specification – Algebraic specification – Model based specification – Diagramming techniques – Data Flow Diagrams (DFD) – Michael Jackson diagrams – ER diagrams – Decision tables – Decision trees

2. System engineering – Analysis, concepts and principles – Analysis modeling – Design concepts and principles – Cohesion – Coupling – Abstraction – Partitioning – Architectural design – Interface design – Component level design

3. Software analysis and design – Structured analysis and design methodology – Transform analysis – Transaction analysis – Design optimization – Case studies

4. Computer aided software engineering – CASE classification – CASE life cycle – CASE workbenches – Programming workbenches – Analysis and design workbenches – Testing workbenches – Meta CASE workbenches

5. Advanced topics – Clean room software engineering – Web engineering – Reengineering – Client/Server software engineering

REFERNCES

1. Ian Sommerville “Software Engineering” AW 8th Ed (2008)
2. Roger S. Pressman “Software Engineering – A practitioners approach” McGraw Hill 7thEd (2009)
3. James Martin “Structured techniques for computing” PH 1st Ed (1985)
4. Pankaj Jalote “An Integrated Approach to Software Engineering” Springer 3rd Ed (2005)

CSE3201: MODEL DRIVEN REAL-TIME SOFTWARE

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description

Model-Driven Architecture (MDA) is the concept of using models developed using a modelling language (UML) to generate real application. This course emphasises modelling of real-time and reactive systems.

Course Objectives

To give a comprehensive overview of MDA and how it relates to OMG standards
Detailed study of real time UML and modelling using UML 2.0

Profile

1. MDA life cycle - Benefits - Building Blocks - Framework - Models - Transformations - OMG

Standards - tools - Development Processes - Application of MDA - Case Study

2. Transformations features - Traceability - Metamodeling - layers of OMG - Transformations definition language -MOF –OCL- Role of UML in MDA

3. Platform independent modelling- structural modelling-behaviour modelling-class behaviour and interactions- architectural modelling - case study.

4. ROPES Process - MDD -ROPES Spiral - Scheduling model -based projects -UML 2.0 - Structural aspects - Dynamic aspects- Schedulability - Performance & Time

5. Requirement Analysis of Real-Rime System- Object domain analysis - Object Behavior - Definition - Mechanistic design -Observer Pattern -Proxy Pattern - Operational state transition- System performance matrix

REFERNCES

1. Bruce Powel Douglass "Real-Time UML"Addison-Wesley Third Edition (2004)
2. Anneke Kleppe, Jos Warmer and Wim Bast "MDA Explained" Addison-Wesley (2003)

CSE3202: PROGRAMMING EMBEDDED SYSTEMS

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description

This course will develop the programming skills in the embedded systems field. Emphasis is given to device driver and application development. Programming for mobile devices also included

Course Objectives

To Familiarize programming methods and tools
To Write efficient programs in C
To Develop Device Drivers for embedded systems
To Program mobile devices

Profile

1. Embedded Microcontrollers - PIC microcontroller - Programming PIC - Application development using 8,16 and 32 bit PIC

2. C for embedded systems - data types and variables - structure and operations - luibraires - Compiling, linking and locating - debugging - code optimization and testing - GNU development tools

3. Device drivers - Types - Developing a device driver - analysing a device driver - coding, configuring and testing device driver - Case studies of development tool chains

4. Real-time Java platform - java closures - memory access - Real-time threads - async events - example programs

5. Programming mobile devices - Development process - Memory management - Applications -

Dynamic linking - Concurrency - resource management - Case studies of mobile platforms like Symbian, Android, Linux

REFERNCES

1. Tim Wilmshurst "Designing Embedded Systems with PIC Microcontrollers: Principles and Applications" Newnes (2006) ISBN-13: 978-0750667555
2. Kirk Zurell "C Programming for Embedded Systems" CMP Books (2000) ISBN-13: 978-1929629046
3. Tim Burke "Writing Device Drivers, Tutorial and Reference" Butterworth-Heinemann (1995) ISBN-13: 978-1555581411
4. Peter C Dibble "Real-Time Java Platform Programming" Prentice Hall (2002) ISBN-13: 978-0130282613

CSE3203: NETWORKED EMBEDDED SYSTEMS

Core/Elective: **Elective** Semester: **2** Credits: **4**

Course Description

Continuing advances in integrated circuit technology have enabled the integration of computation and communication capabilities into devices which monitor or control physical processes. This course is an introduction to fundamental concepts of networked embedded systems and wireless sensor networks.

Course Objectives

To understand the composition of embedded systems that may function as the connection of the Internet to the physical world
To study communication protocols for ubiquitous computing systems specifically power-aware communication protocols for networks of small devices
To gain knowledge to construct innovative-networked systems

Profile

1. Embedded Network systems – Representation of signals – Signal propagation – sensor principles – source detection and identification

2. Networking – Network position and synchronization – Energy management – Data management – Articulation, Mobility and infrastructure - Node architecture – Node data integrity – Experimental systems design – Design principles for Embedded Network systems

3. Wireless Sensor Networks – Architectures for WSN – Data aggregation - Routing concepts - Operating Systems for WSN – Sensor Programming

4. Automotive networked embedded systems – trends – time triggered communication – CAN for embedded systems – LIN standard – System software for automotive applications

5. Networked Embedded systems in Industrial Automation – Fieldbus Systems – Real-Time Ethernet – WLAN and WPAN for industrial environments

REFERNCES

1. Gregory J. Pottie and William J. Kaiser "Principles of Embedded Networked Systems Design"

Cambridge University Press (2005) ISBN-13: 978-0521840125
2. Richard Zurawski "Embedded Systems Handbook, Second Edition: Networked Embedded Systems", CRC (2009) ISBN-13: 978-1439807613

CSE3204: SOFTWARE RELIABILITY ENGINEERING

Core/Elective: **Elective** Semester: **2** Credits: **4**

Course Description

Reliability is the ability of a device or system to perform a required function under stated condition for a specified period of time. Reliability engineering is performed throughout the life cycle of a system including development, production and operation

Course Objectives

To Study the state of the art technologies, methodologies and tools used to assess and ensure the reliability of software system
To evaluate the reliability of embedded systems through hardware based and software based technologies

Profile

1. Reliability Engineering measures -SMTF - reliability function for common distribution - Maintainability -availability -software verification and validation

2. Relablity Modeling - Halsted's software metric -cylomatic complexity metric -error seeding models-failure rate models-curve fitting models-Reliability Growth Models-Markov Structure Models

3. NHPP software reliability models –Fault tolerant Software - Recovery Block Scheme - N-version Programming-Self checking Duplex Scheme -Relability with Enhanced Proportional Hazard Jelinski-Moranda (EPJM) Model

4. Embedded Systems Reliability Evaluation-Fault Injection Techniques-Fault Category- fault coverage estimation - FARM Model -Hardware based Fault Injection - Soft Errors on Digital Components-Protection against Soft Errors

5. Pin-Level Hardware Fault Injection - Fault model set- SEEs in FPGAs-Agents-Based Fault Injector for Windows NT-BOND Tool-Xception-MEFISTO-Mutation testing-Dynamic Fault Collapsing

REFERNCES

1. Alfredo Benso andP Prinetto "Fault Injection Techniques And Tools For Embedded Systems Reliability Evaluation" Kluwer Academic Publications (2003)
2. Hoang Pham "Software reliability" Springer (2000)

CSE3301: REAL-TIME DIGITAL SIGNAL PROCESSING

Core/Elective: **Core** Semester: **3** Credits: **4**

Course Description

This course introduces the theory of digital signal processing and the important skill of real-time

DSP design and implementation techniques.

Course Objectives

To study principles, concepts and practice of Digital Signal Processing.
To familiarize Explain the methods and processes of constructing the different types of Filers
To Understand Speech and Channel coding techniques

Profile

1. Elements of Real-Time DSP systems – DSP hardware – DSP System design – DSP Development tools - Digital signals and systems – System concepts – Random variables – Fixed point representations and Quantization effects – overflow and solutions – Quantization of signals

2. Introduction to FIR filters – Design of FIR filters – implementation considerations – Interpolation and Decimation Filters – Design of IIR filters – Frequency analysis and FFT

3. Adaptive filtering – Random Processes – Digital signal generators – Adaptive echo cancellation

4. Speech coding techniques – overview of CODECs – VoIP – Speech enhancement techniques – Spectral substraction – Voice activity detection – Voice enhancement – Audio signal processing

5. Channel coding techniques – Block codes – Convolutional codes – Viterbi decoding and applications – Digital Image Processing – Color spaces – Color balance and correction – Image histogram – Image filtering using Fast convolution

REFERNCES

1. Sen M. Kuo, Bob H. Lee and Wenshun Tian, "Real-Time Digital Signal Processing: Implementations and Applications" Wiley; 2 edition (2006) ISBN-13: 978-0470014950
2. Nasser Kehtarnavaz, "Real-Time Digital Signal Processing" Newnes (2004) ISBN-13: 978-0750678308
3. Paul Embree, "C Algorithms for Real-Time DSP" Prentice Hall (1995) ISBN-13: 978-0133373530

CSE3302: REAL-TIME DESIGN PATTERNS

Core/Elective: **Core** Semester: **3** Credits: **4**

Course Description

The course reveals and examines the different aspects of Embedded/Real-Time systems design. It teaches the essentials of developing such systems solutions using Real-Time design patterns. It explores in-depth each one of the Real-Time\Embedded system components, their functionality and interfaces, placing an emphasis on understanding components integration. Investigating advanced aspects of Object Oriented Programming in the RTOS environment.

Course Objectives

Build better Real-Time software
Use contemporary advanced real-time patterns
Be able to identify and implement many common patterns in low level design

Profile

1. Real-Time OO Challenge - What is a design pattern – Basic structure of Design pattern - Application scope of design pattern Principles of RT OOP - Real-time response - Failures recovery - Distributed architectures - Synchronous and asynchronous communication - race conditions – Timing

2. Real-Time UML - Analysis of Requirements - Modeling of RT systems using UML - Modeling Dynamic behavior - design considerations

3. Subsystem patterns: layered - microkernel - hierarchical control patterns - Architecture patterns: Virtual machine - component based pattern

4. Memory and resource patterns: Memory management - Static allocation - Pool allocation - Fixed size buffer - Smart pointer - Garbage collection - Garbage compactor pattern - Critical section - Priority inheritance - Highest locker - ordered locking pattern

5. Distribution Patterns: Shared Memory - Remote Method Call - Observer - Proxy - Broker - Safety and Reliability: Protected Single channel - Redundancy patterns - Watch dog - Safety executive pattern

REFERNCES

1. Bruce Powell Douglass "Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems" AW (2002) ISBN-13: 978-0201699562
2. Bruce Powell Douglass “Real Time UML Workshop for Embedded Systems” Newnes (2006) ISBN-13: 978-0750679060

CSE3303: CONTROL SYSTEMS ENGINEERING

Core/Elective: **Elective** Semester: **3** Credits: **4**

Course Description

Control System engineering deals with application of Control theory to design system with predictable behaviors. It focuses on the mathematical modeling, analysis and controller design of systems of diverse nature

Course Objectives

Study analysis of discrete time systems.
Design of Control Systems using state space model
Lyapunov’s method for stability analysis of systems

Profile

1. Basic Control Theory - Time Response Analysis - Frequency Domain Analysis - Stability of Linear Systems - Compensating Networks - Design of Lead/Lag Compensator - PID Controller Design and Tuning

2. Discrete Data Control Systems - Types of Discrete Time Systems - Examples of Discrete Data Systems - Sampling Process and Analysis - Data Reconstruction and Hold Circuits - Mathematical Analysis of Discrete Data Systems - z-Transforms - Pulse Transfer Function - Time Response Analysis of Discrete Time Systems - Mapping Properties of Z-Transform -

Stability of Discrete Time Systems

3. State Space Analysis of Systems - State Model for Linear Systems and Nonlinear Systems - State Space Model for Physical Systems - State Space Model from Transfer Functions - Canonical Models - Time Domain Analysis in State Space - Time Invariant State Equation - State Transition Matrix from Cayleigh–Hamilton Theorem - Solution of State Equation in Canonical Forms.

4. Characteristics of Nonlinear Systems - Multiple Equilibrium States and Zones - Jump Phenomenon - Linearisation Techniques - Lyapunov's Method for Stability - Classification of Nonlinearities - Phase Plane Analysis -Delta Method-Pell's Method -Evaluation of Time on Phase Trajectory -Limit Cycles on Phase Plane

5. Describing Function - Definition - Analysis - Describing Function for Ideal Relay -Relay with Dead Zone - Simple Dead Zone - Friction Controlled Backlash - Stability of the Limit Cycles - Closed Loop Frequency Response - Lyapunov's method for nonlinear systems - Application of Lyapunov Function to Estimate Transients - Optimal Control Performance Measures - Quadratic Performance Measures

REFERNCES

1. K. P. Mohandas "Modern Control. Engineering" Sangvine Publications (2006)
2. I.J Nagrath & M Gopal "Control Systems Engineering" New Age International Publications 4th Edition (2005)

CSE3304: HIGH PERFORMANCE EMBEDDED COMPUTING

Core/Elective: **Elective** Semester: **3** Credits: **4**

Course Description

This course introduces the design aspects of high performance embedded systems. Multiprocessor architecture and software structure

Course Objectives

To know principles, concepts and practice of High Performance embedded systems.
To know Processor evaluation techniques
Understand multiprocessor architecture
Understand hardware/software co-design

Profile

1. The landscape of HPEC - Example applications - Design methodologies - Embedded Systems Design flows - Models of computation - Parallelism and computation - Reliable system design - CE architectures

2. Evaluating processors - RISC and DSP processors - Parallel execution mechanisms - Super scalar, SMID and Vector processors- Variable performance CPU architectures - CPU Simulation - Automated CPU Design

3. Code generation and back-end compilation - Memory oriented optimizations - Program

performance analysis - Models of computation and languages

4. Multiprocessor Architectures - Multiprocessor design techniques - Processing elements - Interconnection networks - Memory systems - Physically distributed systems and networks - multiprocessor design methodologies and algorithms

5. Multiprocessor software - RT multiprocessor operating systems - services and middleware for embedded multiprocessors - Hardware/Software co-design - performance analysis - Hardware/Software Co-Synthesis algorithms - Hardware/Software Co-Simulation

REFERNCES

1.Wayne Wolf "High-Performance Embedded Computing: Architectures, Applications, and Methodologies" Morgan Kaufmann(2006) ISBN-13: 978-0123694850
2. Larry L Peterson “Computer Networks: A Systems Approach” Morgan Kaufmann (2007) ISBN-13: 978-0123705488

CSE3305: EMBEDDED COMMUNICATIONS SOFTWARE

Core/Elective: **Elective** Semester: **3** Credits: **4**

Course Description

Communication systems include many devices ranging in complexity from small hand held phones to large, central office switching devices. This course is designed to understand the issues common to communication equipments while developing a communications software strategy.

Course Objectives

To familiarize Software strategy for developing communications equipments.
To Understand context for the software functions of each communication layers
To Understand Hardware and software related issues and solutions

Profile

1. Communication devices - the communications ecosystem - embedded communications software - Software partitioning - module and task decomposition - Partitioning case study

2. Protocol software - debugging protocols - tables and other data structures - table access routines - Buffer and timer management

3. Device management - Router management - Agent to protocol interface - Device to manager communication

4. Common architectures for communication equipment - multi-board architectures - RTOS support for distribution - Failures and Fault tolerence in multiboard systems

5. Product development - Hardware independent development - using COTS board - Development tools - test tools - Case studies on commercial systems

REFERNCES

1. T. Sridhar "Designing Embedded Communications Software" CMP Books (2003) ISBN-13: 978-1578201259
2. Larry L Peterson “Computer Networks: A Systems Approach” Morgan Kaufmann (2007)

ISBN-13: 978-0123705488

CSE3306: MASSIVELY PARALLEL PROGRAMMING

Core/Elective: **Elective** Semester: **3** Credits: **3**

Course Description

It used to be the case that parallel computing was confined to giant supercomputers. But nowadays it is literally everywhere – even in the small mobile handsets that most of us carry around. This course introduces parallel computing with a strong emphasis on programming.

Course Objectives

1. To understand the basics of parallel computing.
2. To develop programming skills required for parallel computing.

Profile

1. Introduction – parallel computing – more speed or parallelism – languages and models – sequential vs parallel – concurrent, parallel, distributed – parallel hardware architecture – modifications to the von Neumann model

2. Evolution of GPU – GPGPU – introduction to data parallelism – CUDA program structure – vector addition kernel – device global memory and data transfer

3. CUDA thread organization – mapping threads to multi-dimensional data – assigning resources to blocks – synchronization and transparent scalability – thread scheduling and latency tolerance

4. Memory access efficiency – CUDA device memory types – performance considerations – global memory bandwidth – instruction mix and thread granularity – floating point considerations

5. Parallel programming patterns – convolution – prefix sum – sparse matrix and vector multiplication – application case studies – strategies for solving problems using parallel programming

REFERNCES

[1] David B. Kirk, Wen-mei W Hwu Programming Massively Parallel Processors, 2nd Edition, Morgan Kaufmann, 2012.
[2] Peter Pacheco, Introduction to Parallel Programming, Morgan Kaufmann, 2011.
[3] Shane Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufmann, 2012.
[4] Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Pro- gramming, Addison-Westley Professional, 2010.