

25UCST411	EMBEDDED SYSTEMS	Category	L	T	P	R	Credit	Year of Introduction
		PEC	3	0	0	0	3	2025

Preamble: This course teaches the technologies behind embedded computing, focusing on how hardware, software, and firmware work together. It introduces design models and methods to improve system performance. Learners will study how Real-Time Operating Systems use task scheduling and synchronization to meet important timing deadlines.

Prerequisite: 25UCST302 Operating Systems

Course Outcomes: After the completion of the course, the student will be able to:

CO	Course Outcomes	Knowledge Level
CO1	Describe the architecture, fundamental building blocks and the structured design process of embedded systems.	K2 Understand
CO2	Use computational models and trade-off analysis to implement hardware-software co-design solutions for system optimization.	K3 Apply
CO3	Apply criteria for selecting a suitable RTOS for an embedded system based on given functional and non-functional requirements.	K3 Apply
CO4	Apply the principles of distributed control systems and vehicular networking protocols to develop architectural solutions for modern embedded applications and case studies.	K3 Apply

Mapping of course outcomes with program outcomes and program specific outcomes.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	1	-	-	-	-	-	-	-	-	2	3	-
CO2	3	3	2	-	-	-	-	-	-	-	2	3	2
CO3	3	3	3	-	-	-	-	-	-	-	2	2	3
CO4	3	2	2	1	-	1	-	-	-	-	3	2	3
Avg	3	2.25	2.33	1	-	1	-	-	-	-	2.25	2.50	2.67

Mapping of course outcomes with Knowledge and Attitude Profile (WK)

	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
--	-----	-----	-----	-----	-----	-----	-----	-----	-----

CO1	*	*	*	*				*	
CO2	*	*	*	*	*			*	
CO3	*	*	*	*	*			*	
CO4	*	*	*	*	*		*	*	

Wks Mapped with the course

WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
*	*	*	*	*		*	*	

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
100	40	60	2.5 hrs

Continuous Internal Evaluation Pattern:

Attendance	: 5 marks
Continuous Assessment Test	: 20 marks
Continuous Assessment Assignment	: 15 marks

SYLLABUS

Module 1 (10 hrs)

Embedded Systems - Definitions, Embedded Systems vs. General Computing Systems, History, Classification, Application Areas, Purpose.

Building Blocks of a Typical Embedded System - System Core (Microprocessors, Microcontrollers, DSP, ASICs, PLDs), Sensors and Actuators, Communication Interface, Embedded Firmware, Other System Components (Reset and Brown-out Protection Circuits, Oscillator Unit, Real-Time Clock, Watchdog Timer), Printed Circuit Board.

Embedded System Design Process - Requirements, Specification, Architecture Design, Designing Hardware and Software Components, System Integration.

Module 2 (10 hrs)

Computational Models in Embedded Design - Data Flow Graph, Control Data Flow Graph, State Machine Model, Sequential Program Model, Concurrent Process Model, Object-Oriented Model.

Hardware Software Co-Design - Traditional Embedded Development Cycle, History, Advantages of the Co-Design Methodology, The Co-Design Process, Fundamental Issues in Hardware Software Co-Design. Hardware software trade-offs.

Module 3 (10 hrs)

Selection of an RTOS for an Embedded Design - Functional and Non-Functional Requirements.

Embedded Firmware Design and Development - Firmware Design Approaches, Firmware Development Languages. Integration of Embedded Hardware and Firmware.

Embedded Product Development Life Cycle - Objectives, Different Phases, Modelling Techniques - Waterfall Model, Incremental Model, Evolutionary Model, Spiral Model.

Module 4 (12 hrs)

Introduction to Vehicular Networks - CAN bus, Time-triggered Architecture, Flex Ray and LIN, Automotive Ethernet, Vehicle Telematics.

Automotive and Aerospace Systems - Introduction to distributed control systems in modern vehicles and aircraft.

Design Case Studies- Battery Operated Smart Card Reader, Automated Meter Reading System, UAV (Drone) Flight Controller.

Text Books:

1. Shibu, K. V. (2017). *Introduction to embedded systems* (2nd ed.). McGraw Hill Education.
2. Peckol, J. K. (2019). *Embedded systems: A contemporary design tool* (2nd ed.). John Wiley & Sons.
3. Wolf, M. (2016). *Computers as components: Principles of embedded computing system design* (4th ed.). Morgan Kaufmann.

References:

1. Staunstrup, J., & Wolf, W. (2013). *Hardware/software co-design: Principles and practice*. Springer Science & Business Media.
2. Kamal, R. (2011). *Embedded systems: Architecture, programming and design*. Tata McGraw-Hill Education.
3. Gajski, D. D., Abdi, S., Gerstlauer, A., & Schirner, G. (2009). *Embedded system design: Modeling, synthesis and verification*. Springer Science & Business Media.
4. Paret, D. (2007). *Multiplexed networks for embedded systems: CAN, LIN, FlexRay, Safe-by-Wire*. John Wiley & Sons.

END SEMESTER EXAMINATION – MODEL QUESTION PAPER

Course Code	25UCST411
Course Name	EMBEDDED SYSTEMS
Max. Marks: 60	Duration: 2.5 Hours

Part A <i>(Answer all questions. Each question carries 3 marks)</i>				
Q n N o.	Question	Marks	CO	BL
1	Explain the functional role of a Watchdog Timer (WDT) and a Brown-out Protection Circuit in preventing system lockups within unattended industrial applications.	3	1	L2
2	Detail the operational differences between the Requirements Phase and the Specification Phase inside a structured embedded system design process.	3	1	L2
3	Given a simple nested conditional processing block, apply structural modeling principles to convert it into a complete Control Data Flow Graph (CDFG).	3	2	L3
4	Apply hardware-software trade-off analysis to select between an Application-Specific Integrated Circuit (ASIC) and a Field-Programmable Gate Array (FPGA) for a high-volume consumer product.	3	2	L3
5	A medical device requires deterministic execution with a context-switch latency threshold of less than $15\mu\text{s}$. Apply non-functional selection criteria to justify choosing a Real-Time Operating System (RTOS) over a general-purpose OS.	3	3	L3
6	Apply architectural principles to contrast the memory overhead and execution predictability of a Super-Loop firmware layout against an RTOS-based preemptive scheduling kernel.	3	3	L3
7	Apply bit-level protocol logic to trace how the Controller Area Network (CAN) bus uses non-destructive bitwise arbitration to resolve packet collisions when two nodes transmit simultaneously.	3	4	L3
8	In a UAV Flight Controller, apply distributed control system principles to map the flow of real-time sensor data from an IMU to the electronic speed controllers (ESCs).	3	4	L3
Part B <i>(Answer any one full question from each Module. Each carry 9 marks)</i>				
Module I				

Q n N o.	Question	Marks	CO	BL
9 a	Analyse the hardware building blocks of an embedded core ecosystem. Contrast the functional deployment boundaries of Microprocessors, Microcontrollers, and Digital Signal Processors (DSPs) based on execution speed, peripheral integration, and memory topologies.	5	1	L2
9 b	Explain the core differences between an Embedded System and a General-Purpose Computing System across operational criteria : application specificity, real-time determinism, power optimization limits, and structural resource constraints.	4	1	L2
OR				
1 0	Describe the complete step-by-step Embedded System Design Process . Trace this structured layout systematically from initial product requirements and functional specifications through architectural design down to final system integration and hardware-software verification.	9	1	L2
Module II				
Q n N o.	Question	Marks	CO	BL
1 1	An automated industrial mixing plant controls fluid flow using two valves based on concurrent temperature and pressure sensor updates. Apply your understanding of computational models to build a comprehensive Finite State Machine (FSM) model and a parallel Data Flow Graph (DFG) that handles this synchronized process loop.	9	2	L3
OR				
1 2 a	Apply hardware-software co-design principles to contrast the traditional sequential development cycle with a modern concurrent co-design framework. Highlight the time-to-market advantages and	5	2	L3

	system-level validation loops of concurrent engineering.			
1 2 b	An engineer is optimizing an edge-based object detection system. Apply a hardware-software trade-off matrix to resolve resource allocation bounds across three core system variables : mathematical execution speed, hardware unit cost, and overall power consumption.	4	2	L3
Module III				
Q n N o.	Question	Marks	CO	BL
1 3	You are designing a safety-critical cardiac pacemaker system. Apply the structured RTOS selection framework to evaluate and choose a suitable kernel based on an aligned mapping of Functional Requirements (task scheduling, memory management) and Non-Functional Requirements (safety certification, safety margins, footprint bounds).	9	3	L3
OR				
1 4 a	Apply your knowledge of the Embedded Product Development Life Cycle (EDLC) to analyze how system risks are addressed across different development models. Contrast the operational risks of using a rigid Waterfall Model against the adaptive feedback loops of a Spiral Model for complex systems.	5	3	L3
1 4 b	Apply firmware architecture principles to outline the software integration steps needed to merge low-level assembly boot routines with a high-level Embedded C application loop. Detail how hardware-software integration parameters are maintained across memory boundaries.	4	3	L3
Module IV				
Q n N o.	Question	Marks	CO	BL
1 5	A modern luxury vehicle features a distributed electronics platform connecting an engine control unit (ECU), a body control module (BCM), and an active suspension system. Apply automotive networking standards to design an optimized multi-bus topology	9	4	L3

	using CAN, FlexRay, and LIN protocols. Justify your protocol choices based on frame bandwidth, deterministic timing guarantees, and node implementation costs.			
OR				
1 6 a	Apply distributed control primitives to map the architecture of a UAV (Drone) Flight Controller. Diagram the real-time interaction loops between the central processing unit, the sensor telemetry bus (I2C/SPI), and the pulse-width modulation (PWM) actuator lines driving the motors.	5	4	L3
1 6 b	Apply system constraint principles to design the structural power management framework for a Battery-Operated Smart Card Reader. Detail how the application firmware must coordinate ultra-low-power sleep states, wake-up events, and peripheral isolation to maximize battery life.	4	4	L3

