

## CSPE3010 EMBEDDED SYSTEMS (3-0-0)

### Course Objectives:

This course aims to:

1. Provide foundational understanding of embedded system characteristics, architecture, and hardware/software components.
2. Familiarize students with interfacing techniques, sensors, actuators, and communication protocols used in embedded platforms.
3. Introduce real-time system concepts, scheduling strategies, and basic RTOS functionality.
4. Explain the embedded system design flow and hardware–software co-design methodologies.
5. Expose students to modern embedded applications and IoT-based embedded integration.

### Module–I: Embedded System Fundamentals & Hardware Basics (06 Hours)

Characteristics and classification of embedded systems, Embedded hardware components: processor, memory, I/O, timers, ADC/DAC, Embedded software concepts: firmware, drivers, toolchains, bootloader overview.

Basics of ARM architecture: RISC principles, register model, instruction groups, Embedded development workflow: cross-compilation, linking, debugging, IDE/toolchain overview

### Module–II: Interfacing & Communication Protocols (06 Hours)

Digital I/O concepts, timers, interrupts, PWM basics, ADC/DAC principles and interfacing considerations, Communication protocols: UART, SPI, I<sup>2</sup>C, CAN basics, USB fundamentals, Device drivers: polling vs interrupt-driven I/O, basic driver structure.

### Module–III: Real-Time Systems & RTOS Concepts (07 Hours)

Real-time embedded systems: characteristics and requirements, Types of tasks: periodic, aperiodic, sporadic

Real-time scheduling: Rate Monotonic (RM), Earliest Deadline First (EDF), Interrupt latency, priority inversion

RTOS fundamentals: Tasks and priorities, Semaphores, mutex, queues, Context switching.

### Module–IV: Embedded System Design Cycle & Hardware–Software Co-Design (06 Hours)

Embedded system design flow, Processor selection criteria, Embedded design metrics: performance, power, cost, size, reliability. Hardware–software co-design concepts, Hardware–software partitioning fundamentals

Introduction to co-simulation and co-verification, Case studies of embedded system design (automotive, biomedical, consumer devices).

### Module–V: Embedded System Applications & IoT Integration (05 Hours)

Overview of IoT and embedded platforms: ESP32, ESP8266, Raspberry Pi, Embedded networking basics: Wi-Fi, BLE, MQTT, HTTP, System integration: sensors, actuators, microcontroller, communication module.

### Course Outcomes:

After the completion of this course, students will be able to:

CO1 Explain embedded system characteristics, architectures, and development workflows.

CO2 Apply interfacing techniques and communication protocols in embedded design.

- CO3 Analyze real-time requirements and design basic real-time applications using scheduling and RTOS concepts.
- CO4 Demonstrate embedded system design methodologies and hardware–software co-design principles.
- CO5 Integrate microcontrollers, sensors, actuators, and communication modules to develop embedded and IoT-based applications.

**Text Books:**

- 1. Raj Kamal, Embedded System Architecture, Programming and Design, TMH.
- 2. Frank Vahid & Tony Givargis, Embedded Systems Design – A Unified Hardware/Software Introduction, Wiley.
- 3. Santanu Chattopadhyay, Embedded Systems Design, PHI.

**References:**

- 1. Sriram V. Iyer & Pankaj Gupta, Embedded Real-Time System Programming, TMH.
- 2. Wayne Wolf, Computers as Components, Morgan Kaufmann.
- 3. FreeRTOS Official Documentation.