

Domain Track Title: Embedded System

Track Total Credits: (4-10-6) – 20 Credits

Courses Division:

1. Microcontroller-Based Embedded System Design (1-3-0)
2. Real-Time Operating system and Porting (1-3-0)
3. Embedded Linux on ARM (1-3-0)
4. AUTOSAR Design using CAN, CAN Analyser, and LIN (1-1-0)
5. Project (0-0-6)

Domain Track Objectives:

- Develop a skilled workforce with the knowledge of the latest trending technologies to meet the Embedded Industry needs.
- To make the student industry-ready with prompt hands-on in the various Real-Time Embedded Systems.

Domain Track Course outcomes:

- Distinguish between the general computing system and an Embedded System.
- Identify and develop various product-based applications based on Embedded Systems.
- Learn to configure and build a customized Linux Kernel.
- Ability to set up and use Cross Development platform.
- Execute the knowledge gained in the Embedded Systems to become an entrepreneur.

Domain Syllabus:

1. Microcontroller Based Embedded System Design (57.5 Hrs)

- 1.1 Introduction to Embedded System
- 1.2 Embedded System Development Life Cycle
- 1.3 Introduction to ARM
- 1.4 AMBA & AHB
- 1.5 Features of ARM7, ARM9, ARM 11, ARM Cortex
- 1.6 Datasheet analysis
- 1.7 GPIO programming – LED, Seven Segment Display, LCD, Matrix keypad, Actuators(Relay, Motors and valves)
- 1.8 System control block-
- 1.9 ADC & DAC -- Sensors(Analog and Digital),

- 1.10 Timer/Counter
- 1.11 Pulse Width Modulation(PWM)
- 1.12 Vectored Interrupt Controller(VIC)
- 1.13 Real Time Clock (RTC)
- 1.14 Watch Dog timer (WDT)
- 1.15 Debugging with JTAG
- 1.16 Inter System Protocols – UART, USART, USB, Bluetooth, BLE, GPS, GSM
- 1.17 Intra System Protocols – I2C, SPI, CAN
- 1.18 Wireless Protocols and its Complete setup – NFC/ RFID, ZigBee, Bluetooth, Wi-Fi, MQTT, LORA,

Real-Time Operating System & Porting (57.5 Hrs)

- 2.1 Real-Time OS
- 2.2 Types of RTOS
- 2.3 GPOS vs RTOS
- 2.4 FreeRTOS
- 2.5 VxWorks
- 2.6 Task Management & its API
- 2.7 Scheduling Algorithms & its API
- 2.8 Inter Task Communication & its API
- 2.9 Resource Management & its API
- 2.10 AWS IoT Core for FreeRTOS
- 2.11 AWS IoT Green Grass setup on Raspberry Pi

3. Embedded Linux for ARM (57.5 Hrs)

- 3.1 Embedded Linux Overview
- 3.2 Linux File System
- 3.3 Types of Kernel
- 3.4 Shell Commands
- 3.5 Shell Scripting
- 3.6 Process Management System calls
- 3.7 Inter-Process Communication System calls

- 3.8 Linux Booting Sequence
- 3.9 Toolchains Configuration and Cross Compilation
- 3.10 Configuring & Installing Bootloaders (U-Boot)
- 3.11 Kernel Configuration and Compilation
- 3.12 Creating Custom Root File system
- 3.13 Remote Debugging Embedded Applications using GDB
- 3.14 Device Drivers- Char drivers
- 3.15 Static Linking & Dynamic Linking
- 3.16 Adding Static Module to the Kernel

4. AUTOSAR Design using CAN, CAN Analyser, and LIN (14 hours)

- 4.1 Introduction to Automotive System
- 4.2 Introduction to AUTOSAR
- 4.3 Details on CAN Protocols
- 4.4 SBUS CAN Analyser
- 4.5 LIN Protocol

Software Handling

1. Keil μ vision 5
2. Arduino IDE
3. ARM GCC Compiler
4. FreeRTOS, Raspbian OS, Ubuntu OS
5. VxSim
6. Proteus Professional 8.9
7. Node-RED

MPU Handling

1. Arduino
2. ESP8266
3. LPC2148/LPC2129
4. Raspberry Pi
5. STM32
6. Beagle Bone Black

List of Projects/ papers/jobs/products to be done in the domain:

(To follow the Gate Process)

1. IoT-based Apparel Tracking System
2. IoT-based Smart Agriculture Monitoring & Controlling
3. Color-based Product Sorting Machine using IoT
4. IoT-based Smart Energy Meter