National Institute of Electronics and Information Technology, Calicut

COURSE SYLLABUS

Name of the Group: *STED*
Name of the Course: *PG Program in Embedded System Design & IoT*
Course Code: *ES 700*
Duration: 600 Hours / 26 credits

Course Structure: The ES 700 course has seven modules including project work. The Participants are required to do project work in any one of the modular areas, to be eligible for the issue of a PG Program in Embedded System Design & IoT.

Target Audience: B.E /B. Tech completed or result awaiting students of the following branches: Electronics/ Electronics & Communication/ Electrical/ Electrical and Electronics/ Mechatronics/ Instrumentation/ Biomedical /Computer Science/Information Technology.

The modules are as follows:

Sl. No	**Module Title**	**Duration (Hours)**			**Credit**	
		Theory	**Lab**	**Total**	**Theory**	**Lab**
1	Embedded C and ARM Cortex Microcontrollers	13	52	65	*1*	*2*
2	Embedded RTOS	13	52	65	*1*	*2*
3	Embedded Linux	13	52	65	*1*	*2*
4	Internet of Things	13	52	65	*1*	*2*
5	Scripting Tool & GUI development for Industrial Application	13	52	65	*1*	*2*
6	Industrial Product Design	13	52	65	*1*	*2*
7	Project Work		210	210		*8*
	Total Duration/Credit	***78***	***522***	***600***	***26***	

* These programs are conducted as workshops/internships. Modular admission is also available, those who complete all the modules and the project work (with the remaining fee) are eligible for the PG Program in Embedded System Design & IoT on the production of the Provisional Degree Certificate of B.E./B.Tech.
** An individual modular target audience is provided in the modular information.

Module 1: Embedded C and ARM Cortex Microcontrollers

Microcontrollers Module Duration: 4 Weeks

Objective

This module aims to familiarize the students with embedded concepts, and programming in ‘C’ and ARM Architecture. This module covers the introduction to

Embedded Systems and advanced topics in ‘C’ such as Memory management, Pointers, and Data structures which are of high relevance in Embedded software is considered in depth. This module also covers the Architecture of ARM and application development with ARM Cortex Microcontrollers.

Learning Outcomes

After successful completion of the module, the students shall be able to understand:

- Development of Embedded applications using Embedded C
- Usage of ARM Cortex Microcontrollers with Embedded C Programming for Application Development

Prerequisite: Knowledge of Analog and Digital Electronics Fundamentals, basic C Programming, and basic knowledge about Microprocessors/ Microcontrollers

Course Description

Embedded Concepts, C and Embedded C, Introduction to ARM Cortex Architecture, ARM Cortex M4 Microcontrollers and Peripherals.

Target Audience: B.E /B. Tech completed or result awaiting students of the following branches: Electronics/ Electronics & Communication/ Electrical/ Electrical and Electronics/ Mechatronics/ Instrumentation/ Biomedical /Computer Science/Information Technology or ongoing students with 3rd semester completed.

Course Plan:

Course Code: ES 701 Title: Embedded C and ARM Cortex Microcontrollers
Topics
Embedded Concepts Introduction to Embedded Systems, Application areas and categories of Embedded Systems, Overview of embedded system architecture, Specialties and trends in Embedded Systems, Development and debugging Tools.
‘C’ and Embedded C Programming Introduction to ‘C’ programming, Storage Classes, Data Types, Controlling program flow, Arrays, Functions, Memory Management, Pointers, Arrays and Pointers, Pointer to Functions and advanced topics on Pointers, Structures and Unions, Data Structures, Linked List, Stacks, Queues, Conditional Compilation, Preprocessor directives, File operations, Variable arguments in Functions, Command line arguments, bitwise operations.

Introduction to ARM Cortex Architecture

Introduction to ARM Architecture, Overview of ARM, Overview of Cortex-M Architecture

Cortex M4 Microcontrollers & Peripherals

Cortex M4-based Microcontroller architecture, Memory mapping, ARM Cortex M4 Peripherals – GPIOs, Timers, UARTs, Cortex M4 interrupt handling (NVIC), ARM Cotex-M4 Programming and application development.

Text Books:

1. Embedded/Real-Time Systems Concepts, Design and Programming Black Book, Prasad, KVK.
2. Let us C by YashwantKanetkar.

Reference Books:

1. Embedded Systems Architecture Programming and Design: Raj Kamal, Tata McGraw Hill.
2. ‘Embedded C, Pont, Michael J
3. Embedded Systems an Integrated Approach: Lyla B Das, Pearson
4. C Programming language, Kernighan, Brian W, Ritchie, Dennis M
5. Art of C Programming, JONES, ROBIN,STEWART, IAN
6. ARM System Developer's Guide - Designing and Optimizing System Software by: Andrew N Sloss, Dominic Symes, Chris Wright; 2004, Elseiver.
7. ARM Cortex M4 Reference manual.
8. STM32Ldiscovery datasheets, referene manuals & Application notes.

Module 2: Embedded RTOS

Module Duration: 4 Weeks

Objective

This module aims to generate confidence among students to design and program real-time operating systems on ARM-based platforms.

Learning Outcomes

After successful completion of the module, the students shall be able to understand

- Basic and advanced concepts of RTOS, tasks, and threads
- Task scheduling and memory allocation
- File system and data management
- Parallel programming principles

Target Audience: B.E /B. Tech completed or result awaiting students of the following branches: Electronics/ Electronics & Communication/ Electrical/ Electrical and Electronics/ Mechatronics/ Instrumentation/ Biomedical /Computer Science/Information Technology or ongoing students with 6th semester completed.

Prerequisite: Completion of modules ES 701: Embedded C and ARM Cortex Microcontrollers

Course Description

Introduction Real-Time Systems and RTOS, System architecture of FreeRTOS/RTX, System architecture of Real-Time Linux/ VxWorks.

Course Plan:

Course Code: ES 702 Title: Embedded RTOS
Topics
Introduction Real-Time Systems and RTOS Real-time Vs Non Real-time, Introduction to Real-time systems and Embedded Real-time Systems, Discussion of popular RTOS – FreeRTOS/RTX, Comparison of Embedded RTOSs, Design Goals for Real-time software, Discussion on Embedded Real-time applications, Considerations for real-time programming. **System architecture of FreeRTOS** Introduction to Task Creation and Management Inter Task Communication Mechanisms, Semaphores, Mutex, Message Queues, Interrupts, Development tools, Peripheral Interfacing and porting RTOS on target board.

Text Books:

1. ARM-Based Microcontroller Multitasking Projects Using the FreeRTOS Multitasking Kernel, By Dogan Ibrahim
2. Embedded/Real-Time Systems Concepts, Design and Programming Black Book, Prasad, KVK

Reference Books:

1. Embedded Systems Architecture Programming and Design: Raj Kamal, Tata McGraw Hill
2. Beginning STM32 Developing with FreeRTOS, Libopencm3 and GCC , By Warren Gay Real-time Systems – Jane Liu, PH 2000
3. FreeRTOS Users Guide
4. Real-Time Systems Design and Analysis: An Engineer's Handbook: Laplante, Phillip A
5. Structured Development for Real-Time Systems V3: Implementation Modeling Techniques: Ward, Paul T & Mellor, Stephen J
6. Monitoring and Debugging of Distributed Real-Time Systems: TSAI, Jeffrey J P & Yang, J H
7. Embedded Software Primer: Simon, David E.

Module 3: Embedded Linux

Module Duration: 4 Weeks

Objective

The objective of the course is to provide an understanding of the development of an embedded Linux system on ARM-based platforms.

Learning Outcomes

After successful completion of the module, the students shall be able to understand:

- Embedded Linux operating system architecture
- Linux Internals
- Linux System calls
- Kernel Module Programming

Target Audience: B.E /B. Tech completed or result awaiting students of the following branches: Electronics/ Electronics & Communication/ Electrical/ Electrical and Electronics/ Mechatronics/ Instrumentation/ Biomedical /Computer Science/Information Technology or ongoing students with the 6th semester completed.

Prerequisite: Completion of module ES 701: Embedded C and ARM Cortex Microcontrollers.

Course Description

Introduction to Embedded Systems and Linux, System architecture of Embedded Linux OS, Linux Internal & Linux System Calls, Kernel Module Programming.

Course Plan:

• **Course Code: ES 703 Title: Embedded Linux**
Topics
Introduction to Linux OS Overview of GPOS, Specialty of Linux OS, Linux Directory Structure, Linux basic commands
System architecture of Embedded Linux OS Internals of Linux OS, System Calls, File management, Process Management, Inter Process Communication, Pipe and FIFOs, Shared memory, Sockets, Multithreading and Synchronization - Synchronization mechanisms – Semaphore, Spin locks, Mutex Locks, I/O Management - Serial port programming, Linux Kernel Module Programming, Linux Build Process

Text Books:

1. GNU/LINUX Application Programming, Jones, M Tims
2. Embedded Linux: Hardware, Software, and Interfacing, Hollabaugh, Craig,

Reference Books:

1. Building Embedded Linux Systems: Yaghmour, Karim
2. Embedded Software Primer: Simon, David E.
3. Linux Kernel Internals: Beck, Michael At Al 6.
4. UNIX Network Programming: Steven, Richard 7. Linux: The Complete Reference: Petersen, Richard 8.
5. Linux Device Drivers: Rubini, Alessandro, Corbet, Jonathan
6. Linux Kernel Programming: Algorithms and Structures of version 2.4: Beck, Michael At Al
7. Linux Kernel Development: Love, Robert

Module 4: Internet of Things

Module Duration: 4 Weeks

Objective

This module aims to build Engineers who can specify, design, and program modern connected electronic systems in response to the ever-growing number of connected devices.

Learning Outcomes

After successful completion of the module, the students shall be able to understand:

- Internet of Things (IoT) fundamentals
- IoT and embedded system architectures
- Embedded and Android application programming
- Connectivity and networking technologies
- Cloud computing
- IoT security

Target Audience: B.E /B. Tech completed or result awaiting students of the following branches: Electronics/ Electronics & Communication/ Electrical/ Electrical and Electronics/ Mechatronics/ Instrumentation/ Biomedical /Computer Science/Information Technology or ongoing students with the 6th semester completed.

Prerequisite: Completion of module ES 701: Embedded C and ARM Cortex Microcontrollers.

Course Description

Embedded and IoT Overview, IoT Hardware Platform, IoT Connectivity Solutions, IoT Protocols Overview, Mobile Application Development for IoT, IoT Cloud, IoT Security.

Course Plan:

Course Code: ES 704 Title: Internet of Things (L-T-P): 3-0-1 Credits: 4
Topics
IoT Overview IoT Overview and Layering Concepts in IoT. Overview of IoT hardware platforms, Introduction to various IoT Standards, Interfacing Sensors and actuators.
IoT Connectivity Solutions Wireless connectivity standards for IoT, Bluetooth, BLE, Bluetooth 5.0, ZigBee, Wi-Fi standards. Micropython, Raspberry-Pi
IoT Protocols Overview IoT protocol architecture, IoT application layer protocols, MQTT, CoAP.
Mobile Application Development for IoT Overview of Android Mobile Application development, Software tools for Mobile application development - Introduction to MIT App inventor tool, Android Studio for app development. **IoT Cloud** Cloud basics, open source and commercial cloud for IoT **IoT Security** IoT security overview, Threat modeling, Code signing, Encryption, Wireless security.

Text Books:

1. The Definitive Guide to the ARM Cortex M4 Processors, Joseph Yiu, Newnes.
2. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, Dr. OvidiuVermesan, Dr. Peter Friess, River Publishers
3.

Reference Books:

1. ARM Cortex M4 Cook book, Dr. Mark Fisher
2. Getting Started with Bluetooth Low Energy, Tools and Techniques for Low-Power Networking, By Kevin Townsend, Carles Cufí, Akiba, Robert Davidson
3. Designing Embedded Systems and the Internet of Things (IoT) with the ARM Mbed, By Perry Xiao
4. Internet of Things (A Hands-on-Approach), Vijay Madisetti , ArshdeepBahga
5. ARM Cortex M4 Reference manual.
6. STM32F discovery datasheets, reference manuals & Application notes.

Module 5: Scripting Tool & GUI development for industrial application

Module Duration: 4 Weeks

Objective

The primary objective of this specialization is to develop knowledge about scripting tools such as Python and GUI frameworks like PyQt. Python's simplicity and readability used for scripting and PyQt facilitates the creation of intuitive user interfaces

Learning Outcomes

After successful completion of the module, the students shall be able to understand

- Understand the fundamentals of the Python programming language.
- Able to write basic Python scripts.
- Knowledge about file handling and error handling.
- Understand the principles of object-oriented programming in Python.
- Basic GUI programming using PyQt.
- Familiarity with PyQt widgets and layout management.
- Understanding signals and slots for event handling.
- Proficiency in creating dialogs, and menus, and utilizing advanced features.

Prerequisite: Basic knowledge of any programming language

Course Description

Introduction to Python, Control Flow, and Functions, Data Structures in Python, File Handling and Error Handling, Object-Oriented Programming (OOP) in Python. Introduction to GUI Programming, PyQt Widgets and Layouts, Signals and Slots, Dialogs and Menus.

Course Plan:

Course Code: ES 708

Title: Scripting Tool & GUI development for industrial application

Topics

Introduction to Python

Overview, installation, and setting up a development environment. Basic syntax, variables, and data types.

Control Flow and Functions

Conditional statements (if, elif, else), Loops (for, while), Functions, and parameter passing.

Data Structures in Python

Lists, tuples, dictionaries, String manipulation, Basic input/output.

File Handling and Error Handling

Reading and writing to files, Exception handling.

Object-Oriented Programming (OOP) in Python

Classes and objects, Inheritance, polymorphism, encapsulation, Introduction to modules and packages.

Introduction to GUI Programming

Basics of GUI programming, Introduction to PyQt and its advantages.

PyQt Widgets and Layouts

Common widgets (buttons, labels, textboxes), Layout management (layouts, grids).

Signals and Slots

Understanding the core concept of PyQt for event handling, Connecting signals to slots.

Dialogs and Menus

Creating dialogs and menus, Customizing dialog behavior, GUI Customizations

Text Books:

1. Rapid GUI Programming with Python and Qt by Mark Summerfield
2. Python GUI Programming with Tkinter and PyQt5 by Aleksey Zuev
3. Automate the Boring Stuff with Python by Al Sweigart

Reference Books:

1. PyQt6: Desktop and Mobile App Development with Python" by Gabriel Rodríguez Millán
2. Fluent Python by Luciano Ramalho

Module 6: Industrial Product Design

Module Duration: 4 Weeks

Objective

The objective of this module is to help fresh graduates and practicing engineers to enhance their knowledge and skills of industrial product design covering the various aspects of product development process and design of an Industrial Electronics Product.

Learning Outcomes

After successful completion of the module, the students shall be able to

- Understand the design and development process of an Industrial Electronics Product
- Apply product development process for realization of a product

Target Audience: B.E /B. Tech completed or result awaiting students of the following branches: Electronics/ Electronics & Communication/ Electrical/ Electrical and Electronics/ Mechatronics/ Instrumentation/ Biomedical /Computer Science/Information Technology. Or ongoing students with the 6th semester completed.

Prerequisite: Completion of module ES 701: Embedded C and ARM Cortex Microcontrollers

Course Description

Case Study: Design and Development of Embedded products like Weather Monitoring System, Product Development Process, Detailed design, Circuit Design, Semiconductor Packages, Printed circuit board Design, High Speed PCB Design, Hardware Testing, Software Testing, Debugging, Certification and regulatory requirements.

Course Plan:

Course Code: ES 706: Industrial Product Design
Topics
Product Development Process ● Product Idea & Specification. ● Statement of Work (SOW) ● Concept development & High Level Design.

- Hardware and Software Modular design
- Firmware Flow chart preparation.

Detailed Design

- Enclosure design aspects.
- IP ratings overview.
- Understanding component datasheet.
- Selection of Micro controllers.
- Selection of Discrete Components (passive and active)

Circuit Design

- Interfacing techniques of sensors with Microcontrollers
- Power supply requirements for Electronic circuits
- Low power design technique for portable products
- Thermal management of electronic devices and systems

Semiconductor Packages

- SMD packages, Single chip packages or modules (SCM)
- Commonly used packages and advanced packages
- Materials in packages, Current trends in Packaging

Printed circuit board Design

- Evolution and Classification of Printed Circuit Boards.
- PCB fabrication methodologies design considerations
- Design rules for analog, digital and power applications
- Basics of IPC standards.

High Speed PCB Design

- High speed design overview.
- Signal waveforms, frequency and rise time.
- Signal Integrity and Power Integrity.
- Impedance control of power distribution system
- Track impedance, reflections and terminations
- Differential transmission, crosstalk
- Design guidelines for EMI/EMC reduction in PCB.
- PCB routing topologies & PCB structure

Hardware Testing

- Board bring up methodologies (Visual inspection, Impedance measurement on power rails, Plugging in, measure the power, measure oscillations, signal quality check)
- Functional testing
- Test and measurement equipments (MSO, Power analyzer, Protocol analyzer, Data logging multimeter, Electronic load)

Software Testing

- Design of Test Patterns
- Simulation and Testing of functional modules

Debugging

- Software Debugging tools – Simulators
- Hardware Debugging tools In-circuit emulators, Logic Analyzers etc.
- Debugging Techniques
- Breakpoint
- Memory/Register view and modification

Documentation

- Software and Hardware Design Documentation
- User Manuals

Certification and regulatory requirements

- Federal communications commission (FCC)
- International and non international radiators, Specific absorption rate (SAR),
- Underwriters Laboratories (UL), Conformite Europeenne (CE)
- RoHS, BIS, JSS 55555 Certification, ESD immunity.

Text Books

1. Product Design & Development - Karl T Ulrich & Steven D. Eppinger; McGraw Hill
2. Printed Circuit Boards Design, Fabrication and Testing by Khandpur, Tata McGraw Hill
3. Semiconductor Packaging: Materials interaction and reliability by Andrea Chen & Randy.
4. Complete PCB Design using OrCAD Capture and Layout by Kraig Mitzner & Bob doe.

Reference Books

1. IPC design standards manual.

2. Relevant Data sheets and application notes
3. Complete guide to IP ratings.

Module 7: Project Work

Module Duration: 8 weeks

Objective
The objective of project work is to demonstrate the candidates' skill and knowledge in solving a real work Engineering problem involving Embedded or IoT System Design.

Learning Outcomes
After successful completion of this module, the candidate shall be able to:

- Undertake and indecently complete a real-world Industry problem involving Embedded or IoT System Design using state-of-the-art industry standard tools and practices.

Target Audience: B.E /B. Tech completed or result awaiting students of the following branches: Electronics/ Electronics & Communication/ Electrical/ Electrical and Electronics/ Mechatronics/ Instrumentation/ Biomedical /Computer Science/Information Technology or ongoing students with the 6th semester completed.

Prerequisite: Completion of all the modules relevant to the chosen project problem.

Description
The participants can choose projects involving one among the following: Embedded System/ Product Design with ARM Microcontrollers/ARM SoC's, Embedded OS/RTOS, IoT Application Development, etc…

The Participants have to arrange the components for completion of the project work. However, the participants are allowed to utilize the resources of NIELIT SMART Lab for doing project work with Raspberry-pi, ARM Cortex controllers etc… in online mode.

Certificate will be awarded on successful completion of the project work.

Assessment for each module:

MODULE ASSESSMENT	Marks
MODULE THEORY ASSESSMENT	30 Marks
MODULE PRACTICAL ASSESSMENT	70 Marks
TOTAL ASSESSMENT	100 Marks