Embedded Systems Using C

Course Description

The course "Embedded Systems Using C" provides participants with a comprehensive understanding of embedded systems development using the C programming language. Participants will learn the fundamental concepts, techniques, and best practices for developing efficient and reliable embedded systems applications. Through hands-on exercises and practical examples, participants will gain the necessary skills to design and implement embedded systems using C.

Audience

This course is suitable for:

- Software developers interested in entering the field of embedded systems programming.
- Electrical and electronics engineers seeking to enhance their knowledge of embedded systems development.
- Students or professionals looking to transition into embedded systems programming.

Pre-requisite Knowledge/Skills

Participants should have a basic understanding of programming concepts and familiarity with the C programming language. Some knowledge of electronics and microcontroller architecture would be beneficial but not mandatory.

Course Objectives

By the end of this course, participants will be able to:

- Understand the characteristics of embedded systems and their programming requirements.
- Develop embedded systems applications using the C programming language.
- Employ best practices for efficient and reliable embedded systems programming.
- Debug and troubleshoot embedded systems applications effectively.
- Design and implement software architectures suitable for embedded systems.
- Utilize scheduling techniques to optimize system performance.
- Develop drivers and interfaces for peripheral devices in an embedded system.

Course Outline

The course comprises 32-hours of theory and labs. It's divided into different modules.

Module 1: Introduction to C

- Characteristics of Embedded Systems

- C Language Overview
- Structure of a C Program
- Identifiers
- Name Spaces and Scope
- Compilation & Linking
- MCU Boot Process
- C Best Practices for Embedded Systems

Module 2: Variables, Types, and Debugging

- MCU Architecture
- Program Execution
- Variables
- Representing Numbers
- Types
- Casting
- Debugging Embedded Systems

Module 3: Operators and Hardware Manipulation

- Understanding Register Maps
- Operators
- Bit Manipulation
- Modulus and Shifting
- Memory Addressing
- Sizeof
- Ternary Operator
- Precedence Rules
- Best Practices for Embedded Systems

Module 4: Basic Program Flow Control

- Software Design Cycle
- Software Architecture
- UML
- Flowcharts
- Round Robin Scheduling
- Statements
- For and While Loops
- If and Switch statements

- Infinite Loops
- Best Practices for Embedded Systems

Module 5: Advanced Flow Control

- Introduction to Real-time Concepts
- Interrupt Basics
- Interrupt Vector Tables
- Nesting and Priorities
- Software Interrupts
- Volatile keyword
- Shared Data Problems and Solutions
- RMA Analysis
- Interrupts Best Practice

Module 6: Advanced Types, Constants, and Expressions

- Enumerations
- Derived Types
- Literals
- Expressions and Evaluation
- State Machines
- State Charts
- Software Architecture Concepts

Module 7: Arrays and Pointer Basics

- Arrays
- Multidimensional Arrays
- Strings
- String Conversion
- Pointer Types
- Pointers and Arrays
- Pointers Operations
- Best Practices for Embedded Systems

Module 8: More Pointers and Strings

- Pointers to Pointers
- Pointers to Constants

- Constant Pointers
- String Libraries
- Manipulating Memory
- Best Practices for Embedded Systems

Module 9: Functions

- Syntax
- Variable Scope
- Recursion
- Inline Functions
- Software Metrics
- Static Code Analysis
- Testing Techniques
- Best Practices for Embedded Systems

Module 10: Structures and Unions

- Overview of Structures
- Unions
- Driver Design
- Defining APIs
- Driver Models
- GPIO Driver Example

Module 11: Scheduling Techniques

- Arrays of Pointers to Functions
- Function Queue Scheduling
- Cooperative Scheduling
- Scheduler Design
- Energy Profiling
- Low Power Software Design

Module 12: Declarations

- Syntax
- Storage Class Specifiers
- Global Variables
- Type Qualifiers

- Linkage Identifiers
- Best Practice for Embedded Systems

Module 13: Preprocessor

- #define
- Macros
- Precedence
- Conditional Compilation
- Warnings
- #pragma
- Predefined Macros