

## Enhance your skills for Employment.

CITL-Techvarsity training delivers true value. It's Simply the fastest way to build expertise and efficiency as embedded hardware and software professional.

Why you choose CITL-Tech Varsity.

### \* 60:40 Ratio theory&Practicals, more than 20 Minor Projects & 3 Major Projects

\*Our syllabus is professionally designed in consultation with major recruiting companies where we cover **Basic** as well as **Advance** aspects of Embedded Systems , programing and RTOS.

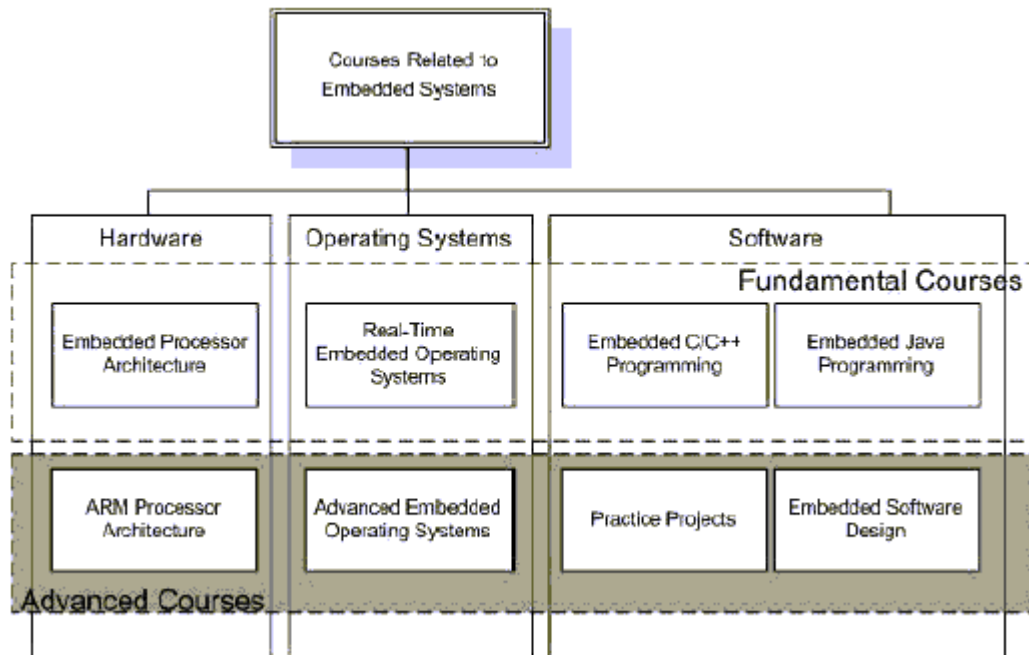
\* Each day of our training is well planned to provide you with **Theoretical** as well as **Practical** knowledge of the module.

\*Each day will come up with **New Practical & Projects** which makes the training Interesting and exciting.

No other training provider can match CITL-Techvarsity for the breadth and quality of the curricula, the variety of training options, and instructor expertise.

**Objective is to improve the Employabilty for candiadates** and candidates will have 100% chances to be in the best of the industries.

Overview of complete training on embedded system development.



# **DIPLOMA IN EMBEDDED SYSTEM DESIGN .**

## **Introduction To Embedded Systems**

**1 DAY**

- Basics of Embedded System Design
- Design methodology and various modules in the system
- Real-Time Embedded Systems
- Embedded System design challenges, embedded controllers etc.
- Development environment
- Current market trends & Applications, job opportunities.

## **Programming in C**

**(10 Days)**

- Introduction to C Programming - Evolution of language, Structure of a C program, The C compilation process, difference between compiler and cross-compiler.
- Data types and Operators - C built in data types and Modifiers, Precedence & Associativity – Arithmetic, logical, relational, bitwise, and ternary operator, Promotion & Typecasting, Qualifiers – const and volatile, Storage classes – auto, register, static, extern.
- Control Flow - Logical expressions and operations, Decision Making, nesting, branching statements, iterators.
- Functions and pointers - declaration, definition, call by value and call by reference, static, extern, inline, recursive functions. Command line arguments. Pointers – Null, wild, dangling, generic.
- Library functions - malloc, calloc, realloc, free, issues.
- Array(integer) - declaration, initialization(sized and unsized), passing array to function using call by reference, 1D, 2D, 3D pointer to 1D, 2D, 3D array, array of pointers.
- Array(char) - declaration, initialization, passing array to function using call by reference, 1D, 2D array of pointers to 2D chars. Stings – standard library functions – strlen, strcpy, strcmp, strcmp, strcmp etc.
- Preprocessor - preprocessor directives – file inclusion, macros, conditional compilation, and miscellaneous directives. Compilation steps – preprocessor, compiler, assembler, linker and loader.
- Structure, union, enum – structure variable, pointers to structures, nesting of structures using structure variable and structure pointers, self- referential structure, applications. Unions - memory map, uses. Enum - uses. Typedef – for data type, arrays, structs, union, enum, function pointers.
- Input and Output - console i\o - formatted and unformatted. File i\o - txt files and binary files.
- Miscellaneous topics - function pointers, difference between #define and Typedef, function returning pointers.

## **Advanced C Programming**

**(6 Days)**

- Stacks - Concepts, issues like over flow and under flow and uses.
- Queues – Concepts, types – ordinary, circular, double ended, priority. Issues and uses.
- Linked list – Concepts, types – linear, circular, doubly, circular doubly and uses.
- Trees – Concepts, types – trees, binary trees, binary search tree and uses.
- Searching – binary, linear , linear search for ordered table,
- Sorting – bubble, selection, insertion, simple merge, modified simple merge, merge sort, quick sort, and tree sort.

## **Embedded System Design and Programming using 8051 MCS-51 family: - 6 days**

Introduction to Embedded system (Microcontroller/ Microprocessors)

Working with 8051 clones (AT89s8252, AT89s52etc) and their special features.

The MCS-51 architecture and PIN detail

Tools and techniques for Required embedded system designing.

Basics of C language & Programming in embedded c language.

Software programming and Hardware interfacing of following External and on chip devices:

LED(using PORTS as Output)

Matrix keyboard

Relay (To control Switching of AC devices e. g. Bulb, Fan, Heater etc.)

Stepper Motor (Speed and Direction Control)

DC motor (Speed and Direction Control)

LCD Display System

Single Chanel Analog to Digital Convertor(ADC0804)

Multichannel Analog to Digital Convertor (ADC809)

Digital to Analog converters

I2C based Serial RTC (DS1307)

I2C based Serial EEPROM

Serial communication b/ w micro-controller and PC & b/ w Two micro-controller

## **Embedded System Design and Programming using ARM (32 bit) Microcontroller-10 days.**

- Introduction to ARM
- ARM7-TDMI Architecture & Programmer's model
- ARM & THUMB Instruction Set Architecture's
- ARM7 Assembly programming using Keil MDK tool-chain
- LPC2128/2129 - ARM7 based MCU Architecture & Programming using Embedded C
- Interfacing and Programming with LPC2378(GPIO - L.E.D., Switches, Matrix Keypad & LCD, Timer, RTC, ADC, UART)
- In-Circuit debugging using Keil Ulink2 JTAG debugger
- Boot sequence & System initialization
- Exception handling
- Advanced simulation , Code profiling & Optimization techniques
- Protocols (Serial,SPI, I2C and CAN) .
- Overview of ARM Cortex Cores
- Projects on ARM controllers,case studies.

## **RTOS-Course Outline:**

**2 days.**

### **Part 1: Real-Time Operating Systems – an introduction.**

- Basic features of real-time operating systems
- Tasks and tasking
- Scheduling – concepts and implementation
- Control of shared resources – mutual exclusion
- Resource contention and deadlocks
- Intertask communication

- Memory management

**Part 2: Practical aspects of realtime operating systems.**

- OS structures from Nanokernels through Microkernels to full RTOSs
  - Process, Memory, Filesystem, Device and Memory management aspects
  - Performance and safety features
  - Real-Time Posix issues
  - Development support
  - Real-Time benchmarking
- Vx-works or Ucos-C