

EMBEDDED SYSTEMS

Training Overview:

Our Embedded course aims at imparting technical skills to the students right from the basics to advanced level, such that, by the end of the Program the student is developed as the finished product, ready to join the industry.

Pre-Requisites : B.E/B.Tech, M.Tech, M.sc (electronics branches)

Participant Fee : On Request

Training Name : EMBEDDED SYSTEMS

Description : An embedded system is a programmed controlling and operating system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.

Weekend Batch : Available

Training Time : 2 hrs per day (MON - FRI)

Duration : 6 Months

Venue : No: 96 G.S.T Road Guduvanchery – 603202

Training Content:

Practical C

- Why C in Embedded
- ANSI Standard
- Fundamentals of C
- Datatypes and Constants
- Simple & Formatted I/O
- Memory Usage
- Operators & Expressions
- Flow Control
- Loops

Functions

- Role of Functions

- Pass by value / reference
- Returning values from Functions
- Recursive Functions
- Call Back Functions
- Implications on Stack
- Library Vs User defined function
- Passing variable number of arguments

Arrays

- Defining, initializing and using arrays
- Multi Dimensional Arrays
- Arrays of Characters and Strings
- Arrays and Pointers
- Passing arrays to functions
- String handling with and without library functions

Storage Classes

- Scope and Life
- Automatic, Static, External, Register
- Memory(CPU / RAM)

Structures & Unions

- What structures are for
- Declaration, initialization
- Accessing like objects
- Nested Structures
- Array of Structures
- Passing structures through functions
- Allocation of memory and holes
- Structure Comparison
- Structure bit operation
- Typedef for portability
- Unions
- Overlapping members

Enumerated data types

- Enum, Indexing, enum Vs #define

Bit Operations

- AND (&), OR (|), XOR (^)
- Compliment (~)
- Left-Shift (<<), Right Shift (>>)
- Masking, Setting, Clearing and Testing of Bit / Bits

Pointers

- The purpose of pointers

- Defining pointers
- The & and * operators
- Pointer Assignment
- Pointer Arithmetic
- Multiple indirections
- Advanced pointer types
- Generic and Null Pointer
- Function Pointers
- Pointers to Arrays and Strings
- Array of Pointers
- Pointers to Structure and Union
- Pointers to Dynamic memory
- Far, Near and Huge Pointers
- Pointer Type Casting

Dynamic Memory Allocation

- Malloc(), Calloc(), Realloc(), Free()
- Farmalloc(), Farcalloc()

File Handling Concepts

- Concept of a FILE data type
- Inode, FILE structure
- File pointer
- Character handling routines
- Formatted Data Routines
- Raw data Routines
- Random Access to FILE

Command line Arguments

- Argc, argv
- Variable Inputs to the main

Compiler in Practical

- Preprocessor Directives
- Compiler, Assembler, Linker
- Conditional Compilation
- Multiple File Compilation
- Code Optimization techniques
- Volatile, #pragma

Data Structures

- Linear & non-linear
- Homogeneous & non-homogeneous
- Static & Dynamic
- Single, Double & Circular Linked Lists
- Stacks & Queues
- Binary Trees

Sorting and Searching Techniques

- Insertion, Selection, Bubble, Merge, Quick, Heap

Concepts and Real-time Exposure

- Development Tools and Environment
- Make Utility
- Industry Coding Standards
- Object / Executable File Format
- Debugger

Mini Project 1

Linux Internals

Introduction

- Kernel Architecture
- Application
- Shell and Services
- System Calls
- Error Handling

Library

- Linker and Loader
- Static Dynamic Library

Process Management

- Process Control Block
- Process Creation and Exit
- Process Scheduling Policies
- Process Limits
- Process Priorities
- Foreground & Background Processes
- Race Condition
- Synchronization
- Copy-on-write
- Process time values
- Daemon Process

Interrupts

- Process Interrupt
- Raise of Signal
- Catching signal
- Signal action

File Management

- Files and File Attributes

- File Descriptor
- File I/O
- Duplicating File & File Descriptor
- File Control operations
- File types
- Protection
- Inode

Inter Process Communication & Synchronization

- Pipe
- Fifo
- Message Queue
- Shared Memory
- Client – Server properties
- Semaphore

Threads

- Creation
- Termination
- Synchronization
- Attributes

Memory Management

- Paging
- Reentrancy
- Segmentation
- Virtual Memory
- Memory Protection
- Memory Sharing

Shell Script

- Types of Shell
- Shell Variables
- Control Statements
- Looping
- Command Line Arguments

Mini Project 2

Networking and TCP/IP Applications

- Network Structure
- Classifications and Topologies
- Switching and Routing
- Gateway, repeater, Hub, Bridge
- OSI & TCP/IP Protocol Layers
- Physical & Logical Addresses
- ARP & RARP
- Internet Protocol

- Routing Protocol and IP Datagrams
- Error and Control Messages (ICMP) UDP
- Transfer Control Protocol
- TCP Networking Applications
- (FTP, TFTP, TELNET, DNS, DHCP, SNMP, POP3, IMAP, SNMP)

Socket Programming

- Overview
- Concurrent Processing
- Programming Interface
- Socket Interface
- Client / Server Design
- Concurrent Connection-Oriented Servers
- Socket Calls for TCP and UDP
- Single Process
- Concurrent Servers
- Remote Procedure Call
- Implementation of TFTP / SMTP

Mini Project 3

Microcontroller Intel 8051

Introduction

- Microprocessor vs Microcontroller
- CISC vs RISC

Overview of Architecture of 8051

- Processor Core and Functional Block Diagram
- Description of memory organization
- Overview of ALL SFR's and their basic functionality

Low-level Programming Concepts

- Addressing Modes
- Instruction Set and Assembly Language (ALP)
- Developing, Building and Debugging ALP's

Middle Level Programming Concepts

- Cross Compiler
- Embedded C Implementation, prog. * Debugging
- Differences from ANSI-C
- Memory Models
- Library reference
- Use of #pragma directive
- Functions, Parameter passing and return types

On-Chip Peripherals

- Ports: Input/output

- Timers & Counters
- Interrupts, UART

External Interfaces

- LEDS
- Switches (Momentary type, Toggle type)
- Seven Segment Display: (Normal mode, BCD mode,
- Internal Multiplexing & External Multiplexing)
- LCD (4bit, 8bit, Busy Flag, Custom Character Generation)
- Keypad Matrix

Protocols

- I2C (EEPROM), SPI (EEPROM)

Keil's RTX51 Tiny / Pumpkin's Salvo

- Overview
- Specifications
- Single-Tasking Programs
- Multi-Tasking Programs
- RTX51 Tiny Programs
- Theory of Operation
- Timer Tick Interrupt
- Task Management & Scheduler Events
- Round-Robin & Co-operative Task Switching
- Idle Task
- Stack Management
- Function Reference
- Porting on to H/W
- Implementation Examples

Selective Discussion during Project Development

- A/D & D/A Converter
- Stepper motor, DC Motor
- I2C Protocol (RTC:800583,DS1307 ADC:PCF8591, DS1621)
- SPI Protocols (ADC:MCP3001)
- IR Communications (Phillips RC5 Protocol)
- ZIGBEE, GSM, GPS, USB, MMC & SD
- Ethernet MAC, CAN Protocol

Mini Project 4

Real Time Operating System

RTOS RT-LINUX

- RT- Linux
- Different types of Operating systems
- RTOS basics – Linux as Real Time
- RTOS Introduction (Hard Real Time, Soft Real time)

- Latency in Linux, Priority Inheritance
- Linux 2.6 features for realtime
- 2.6 Kernel Compilation
- RT LINUX patching
- Linux RTPREEMPT Patches
- Configuring the Kernel with RT-PATCH
- Implementation of Real Time application
- Linux real-time API
- Measuring and comparing scheduling latency in standard Linux and in RT-Linux with the latest RT patches
- Porting RT-Linux on ARM and application development

Object Oriented Programming with C++

- Overview
- Characteristics
- Function Overloading
- Scope Resolution Operator
- Classes in C++
- Access Specifiers
- Constructor, Destructor
- Static members, Functions
- Friend Classes, Friend Functions
- Operator Overloading
- Data Conversions
- Inheritance, Polymorphism
- Exception Handling, Templates
- Input and Output Streams

ARM

- Introduction
- Core Features
- Version History
- Data Flow Model
- Registers
- CPU Modes
- Memory Organization
- Interrupts
- Pipelining
- ARM Assembly Language Programming
- Addressing Modes
- ARM 7 Instruction Set (20/80% -rule of assembly language)
- Usage of Keil IDE
- Demonstrating ARM ISA
- Demonstrating THUMB ISA
- ARM Embedded C language Implementation
- Exposure to an ARM7 CPU Core Based Microcontroller
- LPC2114-ARM7 Based Microcontroller from Philips Semiconductors
- On-Chip System Peripherals
- Bus Structure (AMBA)
- Memory Map

- Phase Locked Loop
- VPB Driver
- Pin Connect Block
- On-ChipUser Peripherals
- General Purpose I/O : Demo using switch & LED
- Vectored Interrupt Controller (VIC)
- External Interrupts : Demos