Embedded Systems Design

Course Focus

Realizing the growth of embedded systems in day-to-day life and the need for trained manpower in this promising area, SVJK has launched a Embedded Systems Design (ESD) for Engineers in computers, electronics and IT. Embedded Systems is a unique field, where engineers need to have sound knowledge in hardware and software design. Keeping this aspect in view, SVJK has designed the course giving equal emphasis to hardware and software, enabling engineers to face challenges in the design and development of state of the art embedded systems. The latest curriculum includes a module on wireless embedded systems design, specifically focusing on wireless technologies and Wireless Sensor Networks (WSN).

Course Structure

Embedded Systems Design (ESD) is a 22 weeks full-time course consisting of 7 modules, an industry Relevant project and a seminar. Daily sessions comprise two hours of lecture followed by four hours of lab work.

ESD Course Syllabus

Introduction to Embedded Systems Design
- Trends in Embedded Systems
- Challenges and Design Issues in Embedded Systems
- Assemblers, Compilers, Linkers, Loaders, Debuggers
- Embedded In-Circuit Emulators and JTAG
- Profilers and Test Coverage Tools
- Build Tools for Embedded Systems
- Configuring and Building GNU Cross-Tool chain

Programming Concepts and Data Communication
- C Programming
- OOPS Concepts and C++ Programming
- Data Structures
- Scripting Languages
- Overview of Networking and Packet Switching Concepts
- OSI Reference Model and TCP/IP Protocol Suite
- LAN Protocol Suite
- Application Layer Protocols
- Socket Programming

Real-Time Operating Systems (RTOS)
- Introduction to OS
- Process Management and Inter Process Communication

- Memory management, I/O subsystem, File System Organization
- POSIX Thread Programming
- POSIX Semaphores, Mutexes, Conditional Variables, Barriers, Message Queues
- Debugging and Testing Multi-Threaded Applications
- Introduction to Real-Time / Embedded Operating Systems
- Performance Metrics of RTOS
- Real Time Scheduling, Task Specifications, Schedulability Analysis
- Real Time Linux Internals
- Configuring and Compiling Real Time Linux
- Programming in Real Time Linux

Embedded Systems Programming
- Porting RTOS and Embedded Operating Systems
- Introduction to Boot loaders and Board Support Packages
- Embedded File Systems
- Building RTOS / EOS Image for Target Hardware
- Time, Space and Power aware Programming
- Embedded Linux Kernel Internals
- Embedded Linux Device Drivers

8/16/32 bit Microcontrollers and Interfacing
- Introduction to 8-bit Microcontrollers
- RISC / CISC and Harvard / Princeton Architectures
- Embedded Memory, Timers / Counters, UART, SPI, PWM, WDT
- Input Capture, Output Compare Modes, I2C, CAN
- LED, Switches, ADC, DAC, LCD, RTC
- Emerging Bus Standards (USB, PCI)
- Programming in Assembly and Embedded C
- Introduction to 16 / 32-bit Processors
- ARM Architecture and Organization
- ARM/THUMB Programming Model
- ARM/THUMB Instruction Set
- ARM Exception Handling
- ARM/THUMB Assembly and C Programming (GNU Tools)
- ARM/THUMB Interworking
- ARM Peripheral Programming
- Cortex-M3 Architecture and Programming
- Overview of Multi-Core Embedded Systems
- Overview of FPGA

Digital Signal Processing
- Fundamentals of Digital Signal Processing
- DFT, IDFT, FFT, Convolution
- FIR and IIR Filter Design
- Algorithm implementation using DSP
- Digital Signal Processor Architecture
- DSP based software development tools
- DSP based embedded system design process
- DSP applications
- Introduction to Codec's