Prelude to the Programme

The growth of electronics is visible in our day today life and it has come to a state where it has become even impossible to imagine a world without gadgets as we have all got addicted to the same in one way or other, especially to our mobile phones or iPods, the most frequently used ones. And of course these gadgets are none other than embedded systems for electronics engineers. In simple words, Embedded System means hardware plus software with a specific task. For electronics aspirants this domain is a very colorful and interesting one with lot of challenges and opportunities. This practical oriented course facilitates all the embedded/electronics aspirants to gain expertise in various aspects of embedded system design.

Objectives:

For Professionals:

For professionals this course would be an easy and fast way to upgrade and meet the current industrial standards and techniques adopted in latest Embedded Technology oriented projects. They can join specifically for some modules in accordance to their professional requirements as per the demand of their company. Since C E I T R is basically an organization which is into research and development, professionals can even get their project prototypes implemented in our labs with the aid of well experienced resources backed with the power of highly sophisticated and well equipped lab environment.
For Fresh Engineering Graduates:

This course helps to bridge the gap between industry and their academics. As engineering college education is mostly theory oriented, students don’t get exposed to real time industrial projects. This helps them to understand how industry works in the real world scenario. The course participant gets hands on experience on various hardware/software tools which would help them to execute their work in a smooth and professional manner. The course participant gains exposure in various Embedded Systems Debugging techniques which is highly demanded in the industries to facilitate project execution in a very fast and precise manner without any last minute frustrations.

Outcomes:

1.) Course participants after the course will become real “Embedded Developers” since they would achieve that level of confidence as they gain strong exposure to various Embedded System Design and Development techniques.

2.) Course participant develops a hopeful patience in the debugging and testing process as the current real time project execution expects that quality from each developer/tester. Basically a successful developer is a good debugger.

3.) Six months advanced skill development programme is equivalent to one year professional work experience and the course participant becomes eligible to enter an Embedded Company with a designation and pay package that suitably rewards their practical exposure gained from the course.
Brochure: Advanced Skill Development Programme in Real Time Embedded System Design, DSP, Robotics & VLSI

Modules

1. Operating System Basics (Linux OS) and Networking Fundamentals
2. C Language in Linux Environment (The GNU Compiler Collection)
3. Microcontrollers 8051 and PIC ALP (on SBCs)
4. Embedded C Programming (KEIL C and MPLAB)
5. Advanced Microcontrollers (ARM uC Boards from Philips and ATMEL)
6. DSP (TMS 320 and C55x series DSPs with Code Composer Studio and MATLAB)
7. Embedded Linux and Device Drivers
8. RTOS (VxWorks and RTLinux)
9. VLSI (VHDL and FPGAs)
10. Embedded Product Design and TQM (Software Engineering Processes, CM Tools, PM)
11. Project on Robotics

Module 1 (2 weeks)

Operating System Basics (Linux OS) and Networking Fundamentals

This module deals with basic concepts of OS and Networking. Nowadays all are familiar with some operating systems such as Windows or Linux which are commonly being used in desktop PCs or laptops. OS is the most important program that runs on a computer. Every general-purpose computer must have an operating system to run other programs. Operating system performs basic tasks such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives, printers etc. This module gives an overall concept about OS working and its principles. Knowledge about networking of computers/ECUs (Electronic Control Units) is very essential now whether it be wired or wireless. Embedded applications in wireless domains are in great demand now. To get an exposure into the networking concepts, we have included topics such as TCP/IP, IP addressing, MAC address, etc. in this module.
Syllabus in Detail

**General Operating System Concepts**

- Operating System Structure
- Processes
- Threads
- CPU Scheduling
- Process Synchronization
- Deadlocks
- Memory Management
- Virtual Memory

**Networking Fundamentals**

- Basics of Networking
- Classification of Networks
- TCP/IP Protocols
- IP Addressing
- Classes of IP Addresses
- Subnet Masking
- Establishing Network connections between nodes
- Pinging Process
- Wireless Networking
Module 2 (3 weeks)

C Language in Linux Environment (The GNU Compiler Collection)

Extensive usage of C language is demanded from an Embedded System Design Engineer. Most of the Engineers are expected to work using Embedded C. C is very compatible with the Assembly Language and prevents code bloat which usually C++ has. Usually Embedded developers are expected to be quite familiar working with Linux Environment using C Language for applications related to Embedded Device Driver Development etc. So a good study phase in C language and Linux OS really caters to the Embedded Product Development scenario. The course gives great importance to learning of C programming as it facilitates the learning of Embedded C programming in a systematic fashion.

Syllabus in Detail

Introduction to Linux

- Introduction to Linux OS
- Frequently used Linux Commands
- VI Editor
- Introduction to GCC Compiler
- C Programming with GCC
- Debugging Techniques

C Language

- Introduction to C-programming
- Control Structures
- Iteration
Brochure: Advanced Skill Development Programme in
Real Time Embedded System Design, DSP, Robotics & VLSI

- Preprocessor
- Functions
- Data types and Storage Classes
- Arrays
- String
- Pointers
- Structures
- Files
- Linked List

Module 3 (2 weeks)

Microcontrollers 8051 and PIC ALP

Microcontroller 8051 is one of the simplest microcontrollers that can introduced to a course participant who is interested in learning various Microcontroller architectures and its assembly language programming. Through a thorough study phase of 8051, the Microcontroller Architecture, Peripherals and the Assembly Language Programming get introduced and migration from one microcontroller to another becomes an easy process based on comparative studies. PIC is Peripheral Interface Controller and it is mostly used in Robotic Applications. The course is designed in such a fashion that the course participant finds it easy to migrate from 8051 to PIC Microcontroller.

Syllabus in Detail

Microcontroller 8051
- Introduction to Microcontrollers
- Difference between Microprocessors and Microcontrollers
Introduction to 8051
Architecture
Pin Description
Memory Organization
Ports, Registers
8051 Addressing Modes
Instruction Sets
I/O Port Programming
Port Functions
Bit Manipulation
Timer Programming
Introduction to Serial Communication
Interrupts Programming

PIC
Introduction to PIC Microcontrollers
Introduction to RISC
Instruction sets
Introduction to MPLAB IDE
Instruction Familiarization Using MPLAB IDE
Configuration of I/O Ports and Its Registers
General Introduction about the Data EEPROM and FLASH Program Memory
Timer Programming in PIC Controllers
Serial Port for Communication
Capture / Compare / PWM Mode Programming
Practical applications of Capture, Compare & PWM
SPI Protocol for Communication
I2C Communication Protocol
Module 4 (2 weeks)

Embedded C

What is the difference between ALP and Embedded C? This can explained by telling that ALP is like our local language/mother tongue (specific for each microcontroller) while Embedded C is like our Universal Language (English). We know that C is English like language. So C can be used for programming any microcontroller without knowing the ALP of that specific microcontroller and a superficial knowledge about the microcontroller architecture is enough to get into its embedded C programming. The difference between C and Embedded C is that we use the microcontroller architectural knowledge, some specific keywords and memory specifiers to do the microcontroller programming. Attaining knowledge in Embedded C programming facilitates faster implementations on any controller with minimal study phase.

Syllabus in Detail

- Introduction to KEIL IDE
- Programming in C for 8051 Microcontrollers
- I/O Port Programming
- Port Functions
- Bit Manipulation
- Timer Programming
- Serial Communication
- Interrupts Programming
- Embedded C Programming for PIC (C Compiler for PIC Microcontrollers)
Brochure: Advanced Skill Development Programme in
Real Time Embedded System Design, DSP, Robotics & VLSI

- Interfacing 7 Segment Displays
- Interfacing an LCD
- Interfacing a Keyboard
- Interfacing a Stepper Motor

Module 5 (2 weeks)

Advanced Microcontrollers (ARM)

The 32-bit Microcontrollers with ‘C’ language support and multifunction peripherals are being used in most of the medium and high end Embedded applications such as Mobile computing, Motion control, Wireless communication and even in Signal processing etc. The processors provide high end computing power as well as an extensive array of peripherals such as USB, LAN support, UART, Modem support, LCD/Graphics interface, General purpose I/O, I2C, CAN etc.

The 32-bit ARM processors are used in high end computation requirements such as Multimedia, Digital Camera and other high end applications. ARM processors are very popular in Industries and are well known for its low power consumption. This module focuses on the architecture of the 32-bit ARM9 Microcontroller. The Assembly Language Programming as well as ‘C’ Language Programming of the controller is also dealt in detail.

Syllabus in Detail

- A Brief History of ARM
- Differences between RISC and CISC Architectures
- Introduction to ARM7TDMI Architecture
- Operating Modes of ARM7TDMI
Brochure: Advanced Skill Development Programme in
Real Time Embedded System Design, DSP, Robotics & VLSI

- Introduction to ARM Instruction Set
- Features of ARM Instructions
- Block Transfer Instructions - ldr, Str, Idrmia, Strmia
- Exception Handling in ARM
- Switching to THUMB MODE
- Introduction to THUMB Instruction set
- Introduction to KEIL IDE for ARM Microcontrollers
- Introduction to PHILIPS ARM Controllers
- Configuring Ports for Simple I/O Operations
- Timer Programming
- Serial Port Programming
- A to D Converter Programming
- Interrupt Programming
- PWM Programming

Module 6 (2 weeks)

Digital Signal Processing and Digital Signal Processors

Digital signal processing techniques are so powerful that sometimes it is extremely difficult for analogue signal processing to achieve the same or closer performance. This module introduces the digital signal processing fundamentals like convolution, DFT, FFT, Spectrum analysis etc. ‘C’ language will be used as the language for code development.

MATLAB is a powerful tool for design and analysis of digital signal processing algorithms. The capabilities of MATLAB tool will also be explored during this course.
The TMS320C6000 (C62XX/C67XX) DSP from Texas Instruments (TI) is the highest performance fixed and floating point DSP processors available from TI. The ADSP-21065L is a general purpose, programmable 32-bit DSP that allows user to program with equal efficiency in both fixed point and floating-point arithmetic.

The participants of this module will learn the architecture and programming of TMS320C6000. The hardware and complete implementation of selectable algorithms are dealt in detail. Code Composer Studio IDE from Texas Instruments shall be used as the programming and debugging tool.

**Syllabus in Detail**

- Introduction to DSP Fundamentals
- FFT
- Filter Design
- Introduction to MATLAB
- DSP Programming using MATLAB
- Introduction to DSP processors
- Architecture and Programming of ‘C6000 DSPs
- Peripherals of C6000 DSPs
- Introduction to Code Composer Studio (CCS)
- Programming with CCS IDE
Module 7 (2 weeks)

Embedded Linux and Device Drivers

The objective of the course is to provide the students with an understanding of the aspects of the Real-time systems and Real-time Operating Systems and to provide an understanding of the techniques essential for the design and implementation of real-time embedded systems. Usually for high end application we need to port different operating systems such as linux etc. to the board on which the application program is supposed to be working. Device drivers are driver softwares or softwares that will drive some hardware units. Writing device drivers for various hardwares is an interesting and challenging field in the Embedded industry.

Syllabus in Detail

- Embedded Software – Real time Vs. Non Real time
- Introduction to Real-time systems and Embedded Real-time Systems
- Discussion of popular OS like Linux and Windows
- Discussion on Embedded applications and Real-time applications
- System architecture of a basic RTOS
- Internals of Linux OS
- Overview of popular RTOSs
- Scheduling - RTOS kernel scheduler
- Inter task communication and Synchronization
- Inter Process Communication
- Considerations for real-time programming
- Embedded and Real time software development process
- Software modeling

CEITR, III Floor, Ammashree Chambers, No. 967, ST Bed Entrance,

4th Block Koramangala, Bangalore – 560034, Ph: 080 32568330
Brochure: Advanced Skill Development Programme in
Real Time Embedded System Design, DSP, Robotics & VLSI

- Assembling, Linking, Compiling, Locators
- Development tools – Emulator, Simulator
- Tasks, Semaphores, Memory management
- Debugging techniques
- Hardware- Software Integration
- Embedded Linux Applications
- Linux Device Drivers

Module 8 (2 weeks)

Real Time Operating Systems (VxWorks and RTLinux)

This module provides course participants an understanding of the aspects of Real-time systems, Real-time Operating Systems and an overview of the techniques essential for the design and implementation of real-time embedded systems using the most widely used VxWorks and RTLinux real time operating systems. The difference between a general purpose operating system and a real time operating system is that the “real-time” factor gets highlighted in any RTOS. The occurrence of events should be at the right time/real time or else the system would be a failure. In short, a real time operating system caters to the real time requirements.

Syllabus in Detail

- Introduction to RTOS
- Multitasking & Scheduling
- Inter Process Communication and Semaphores

C E I T R, III Floor, Ammashree Chambers, No. 967, ST Bed Entrance, 4th Block Koramangala, Bangalore – 560034, Ph: 080 32568330
Brochure: Advanced Skill Development Programme in

Real Time Embedded System Design, DSP, Robotics & VLSI

- Real time Multitasking & Scheduling with VxWorks
- Intertask Communication with VxWorks
- Tornado IDE Demo
- Shell, Browser, Debugger & Wind View
- VxWorks Boot Image
- Interrupts, Exception Handling and Timers
- I/O management
- Device Drivers
- Memory management
- Board Support Packages
- RTLinux

Module 9 (2 weeks)

VLSI (VHDL and FPGAs)

Programmable Logic Devices (PLDs) have become an inevitable part of the embedded systems today. To facilitate PLD based embedded system design, various HDLs (Hardware Description Languages) are used. The most commonly used Hardware Description Languages (HDLs) in the industry are VHDL and Verilog. VHDL means Very high speed integrated circuits Hardware Description Language and it is used for designing Integrated Circuits (ICs). This module also covers the architecture of different FPGAs (Field Programmable Gate Arrays).
Brochure: Advanced Skill Development Programme in Real Time Embedded System Design, DSP, Robotics & VLSI

**Syllabus in Detail**

- Introduction to PLDs, CPLDs
- VHDL
- VHDL Concepts, Types & Operators
- Sequential & Concurrent Statements
- VHDL Simulation
- Various FPGA Architectures
- Synthesis & Implementation on FPGAs

**Module 10 (2 weeks)**

**Embedded Product Design and TQM**

The objective of this module is to let the course participant know how industry really works. This module gives exposure to various product design aspects, project management aspects, quality principles that are expected to be followed in the industrial environment etc. Total Quality Management is the major topic discussed throughout the companies as it is a matter of existence. Without quality no product is stable. Quality should be continuously monitored throughout each of the product development phases such as project design, implementation, unit testing, unit integration testing and final installation. Support and maintenance is another field of interest that comes along with the project management. This module also covers various tools such as Gantt Charts, Control Charts and various configuration management tools which are inevitable with respect to project management.
Syllabus in Detail

- Total Quality Management (TQM)
- Product Development Process
- Software Development Life Cycle
- Project Management and Tools for PM
- Documentation
- Team Work and Communication
- Case Study – A complete Embedded Product Design

Module 11 (3 weeks)

Project on Robotics

This module facilitates course participants to do a project which will make use of all the technologies that they have got acquainted with while advancing through various modules in the course. They can choose a creative project or a project from the list of project that CEITR offers and start executing by themselves. Projects will be basically focused on Robotics where there will be microcontroller programming together with wireless communication technologies.