

DETAIL SYLLABUS COPY

Course: Embedded Systems
Course Code: ES125

Duration: Four months
Theory: 125hrs (minimum)
Practicals: 175hrs (minimum)

Module 1: Software Languages

Part 1: C Language	Part 2: Embedded C
<p>Basics:</p> <ol style="list-style-type: none"> 1. Introduction 2. Basic Features 3. Decision Control in C 4. Flow Control <p>Exploring C:</p> <ol style="list-style-type: none"> 1. Pointers 2. Functions and Macros 3. Arrays & Strings 4. Pre-Processor 5. C Compilation System. 6. Structures and Unions 7. File Handling <p>Data Structures:</p> <ol style="list-style-type: none"> 1. Stacks 2. Queues 3. Linked Lists <p>C Advanced Concepts:</p> <ol style="list-style-type: none"> 1. Input/Output Library in C 2. Interfacing Serial Port 3. Interfacing Parallel Port 	<p>Embedded Programming Using C:</p> <ol style="list-style-type: none"> 1. System Programming vs. Application Programming. 2. Why C for Embedded Programming? 3. Review of C language with embedded perspective. <p>Programming Time/ Memory - Critical Systems:</p> <ol style="list-style-type: none"> 1. Bitwise Operators. 2. Pointer Arithmetic 3. Bit fields 4. Mixing Assembly and C <p>Understanding Embedded Memory Limitations:</p> <ol style="list-style-type: none"> 1. Memory Alignment with Structures 2. Memory Management in C. 3. Memory-Map of Applications <p>Embedded Programming standards:</p> <ol style="list-style-type: none"> 1. Optimization Techniques. 2. Testing and Debugging 3. MISRA C guidelines.
Part 3: Object Oriented Programming with C++	
<p>What is OOPS</p> <ul style="list-style-type: none"> Object-Oriented Terminology Abstraction & Encapsulation Inheritance Polymorphism <p>Graduating from C to C++</p> <p>Functions and Variables</p> <ul style="list-style-type: none"> Declaration Definition & Scope Arrays and Strings in C++ Qualifiers <p>Classes in C++</p> <ul style="list-style-type: none"> Defining Classes Instantiating Member & Friend Functions 	<p>Inheritance</p> <ul style="list-style-type: none"> Defining Base and Derived Classes Constructor and Destructor Calls <p>Polymorphism</p> <ul style="list-style-type: none"> Function Overloading Operator Overloading <p>Storage Management</p> <ul style="list-style-type: none"> Memory Allocation Dynamic Allocation: new and delete <p>Input and Output in C++ Programs</p> <p>Exceptions</p> <ul style="list-style-type: none"> Inheritance and Exceptions Hierarchies Inside an Exception Handler <p>Templates</p> <ul style="list-style-type: none"> Standard Template Library Containers



Module 2: Linux for Embedded Development

Part 1: Introduction To Linux And Its Command Set	Part 2: Shell Programming/ Scripts & Linux Internals.
<p>Getting Started:</p> <ol style="list-style-type: none"> 1. Operating System Concepts 2. Introduction To Linux 3. Why Linux in Embedded systems? 4. Linux Terminals & Shell 5. Linux File System 6. Concept of Process in Linux <p>Linux Shell Command Set:</p> <ol style="list-style-type: none"> 1. Navigating File Systems 2. Handling Files 3. Regular Expressions 4. Process Commands 5. VI editor. 	<p>Shell Programming/ Scripts</p> <ol style="list-style-type: none"> 1. Bourne shell Overview 2. User, Shell, And Read-Only Shell Variables 3. Positional Parameters 4. Control Constructs. <p>Linux Internals:</p> <ol style="list-style-type: none"> 1. Linux Kernel Structure 2. System Calls 3. File Sub-System 4. Process Sub-System. 5. Linux Signals 6. Clocks & Timers 7. Memory Management.
Part 3: Real-Time OS Concepts	Part 4: Non-POSIX And POSIX Programming
<p>Basics:</p> <ol style="list-style-type: none"> 1. Introduction to RTOS 2. Comparison of General OS & RTOS 3. Importance of Interrupts in Real-time Applications 4. Concept of Shared Data Bug <p>Software Styles in Embedded Systems:</p> <ol style="list-style-type: none"> 1. Round-Robin 2. R-R with Interrupts 3. Function Queue Scheduling 4. Real-time Scheduling <p>Exploring Real-Time OS:</p> <ol style="list-style-type: none"> 1. Preemptive & Non-Preemptive OS 2. RTOS Services 3. Design Using RTOS 4. Development Tools 	<p>Primitive IPCs:</p> <ol style="list-style-type: none"> 1. Pipes 2. FIFOs <p>System V IPCs:</p> <ol style="list-style-type: none"> 1. Message Queues 2. Semaphores 3. Shared Memory <p>POSIX IPCs: (IEEE Std 1003.1)</p> <ol style="list-style-type: none"> 1. Why POSIX? 2. POSIX Message Queues 3. POSIX Semaphores 4. POSIX Shared Memory 5. Asynchronous Input / Output

Part 5: POSIX Threads Programming

Basics:

1. Introduction to POSIX Threads.
2. Advantage of Threads in Real-Time.
3. Multi-Threaded vs. Multi-Processing

Thread Programming:

1. Thread Functions (APIs)
2. Thread Synchronization
3. Threads Attributes
4. Conditional Variables
5. Mutexes.
6. Signal Handling in Threads.

Part 6: Network Programming

Socket Programming:

1. What Is A Socket?
2. Daemon Processes
3. Creating a socket
4. Socket Options
5. The select() function
6. The connect() function
7. IP addresses for connect()
8. Domain Name Service
9. The bind() function
10. The listen() function
11. The accept() systems call
12. Receiving Data
13. Sending Data
14. Multiple Server Session

Part 7: Linux Utilities

Make:

1. Introduction to Make
2. Advantage of using Makefiles
3. Creating Makefiles for projects
4. Make Dependencies
5. Dependencies Graphs
6. Make Implicit rules
7. Variables and functions
8. Wildcards and PHONY targets

GNU Debugger:

1. Need for a debugger
2. Introduction to GDB
3. Features of GDB
4. Breakpoints And Watchpoints
5. Stepping and Resuming
6. Advanced GDB

Valgrind:

1. An Overview of Valgrind
2. Using the Valgrind core
3. Reporting & Suppressing errors
4. Command-line Options for Valgrind core

Part 8: Device Drivers

An Introduction To Device Drivers:

1. The Role of the Device Driver
2. Classes of Devices and Modules
3. Kernel Modules Versus Applications
4. Compiling and Loading Modules

Char Drivers:

Major and Minor Numbers
The file_operations Structure
Char Device Registration & Un-registration
open and release operations
read and write operations
Advanced Char Driver Operations

Advanced Topics:

Debugging Techniques
Concurrency and Race Conditions
Allocating Memory
Interrupt Handling
Communicating with Hardware



Module 3: Microcontrollers & Development Tools

Part 1: Hardware Basics	Part 2: MCS-51
<p>Hardware Basics</p> <ol style="list-style-type: none"> 1. Analog Devices 2. Digital Devices 3. Micro Controller architectural features 4. Advanced memory options 5. Power management & low voltage 6. I/O <ul style="list-style-type: none"> • RS-232 / RS-485 • SPI bus / I²C bus / CAN • ADC / DAC • Comparator • PWM • Timers • RTC • MTD • JTAG Interface <p>Interrupts :</p> <ol style="list-style-type: none"> 1. Polling 2. Hardware & software Interrupts 3. Maskable & Non-Maskable Interrupts 4. Vectored & Non- Vectored Interrupts 5. Interrupts Arbitration & Priority 	<p>An Introduction To Microcontrollers(8051)</p> <ul style="list-style-type: none"> Types Of Memory Special Function Registers (Sfrs) Basic Registers Addressing Modes Program Flow Instruction Set, Timing, And Low-Level Information Timers Serial Communication Interrupts 8052 Assembly Language 16-Bit Mathematics With The 8051 8052 Microcontroller Pin Functions <p>Kiel</p> <ul style="list-style-type: none"> Introduction Features Development Tools Creating Applications Testing Programs µVision2 Debug Functions Sample Programs Using On-Chip Peripherals
Part 3: ARM7TDMI-S	
<p>ARM Embedded Systems</p> <ul style="list-style-type: none"> • RISC Design Philosophy • ARM Design Philosophy • ARM core , extensions and families <p>LPC2106/2105/2104</p> <ol style="list-style-type: none"> 1. Introduction <ul style="list-style-type: none"> • Architectural Overview • ARM7TDMI-S Processor 2. System Control Block <ul style="list-style-type: none"> • External interrupts & memory mapping • PLL, Power Control , VPB Divider 3. Memory Accelerator Module <ul style="list-style-type: none"> • MAM Operating Modes • MAM Configuration 4. Vectored Interrupt Controller <ul style="list-style-type: none"> • VIC Registers • Interrupt Sources 	<ol style="list-style-type: none"> 5. Pin Connect Block 6. GPIO 7. UART0/UART1 8. I2C Interface 9. SPI Interface 10. Timer 0/Timer1 11. PWM 12. Real Time clock <ul style="list-style-type: none"> • Real time clock interrupts 13. WatchDog Timer 14. GNU Toolkit <ul style="list-style-type: none"> GCC Obj Dump Obj Copy LPC_2000_flash utility 15. C Development



Module 4: Real-Time Operating System

Part 1: Micro-c OS	Part 2: Micro-c Linux
<p>µc-os:</p> <ul style="list-style-type: none">Introduction:Os Initializing And Creating TasksSemaphoresMessage QueuesEventsTimersInterrupt HandlingDebugger	<p>µc-Linux:</p> <ul style="list-style-type: none">Introduction:Setting up the development environmentUnderstanding the build toolsBuilding the micro-c Linux KernelUnderstanding the Kernel build processThe rom file systemCustomizing the romfsPorting Micro-c Linux with application program into ARM

