



Module 1

Chapter 1 : Introduction to Embedded Systems

1-1 to 1-11

Syllabus : Definition, Characteristics, Classification, Applications. Design metrics of Embedded system and Challenges in optimization of metrics.

1.1	Characteristics and Design Metrics of Embedded System	1-1
1.1.1	Design Metrics	1-2
1.2	Characteristics, Quality Attributes and Optimization Challenges of Design Metric in Embedded System.....	1-4
1.3	Characteristics of an RTOS.....	1-5
1.4	Real Time Systems : Need for Real-Time Systems.....	1-6
1.5	Real Time System’s Requirements and Issues	1-8
1.5.1	Hard/Soft Real time Systems	1-8
1.6	Challenges in Embedded System Design	1-9
1.7	Exam Pack (Review and University Questions).....	1-11

Module 2

Chapter 2 : Embedded Hardware Elements

2-1 to 2-79

Syllabus : Features of Embedded cores- μ C, ASIC, ASSP, SoC, FPGA, RISC and CISC cores. Types of memories. Case Study: ARM Cortex-M3 Features, Architecture, Programmer’s model, Special Registers, Operating Modes and States, MPU, Memory map and NVIC. Low power - Need and techniques. Case study of Low Power modes in Cortex-M3. Communication Interfaces: Comparative study of Serial communication Interfaces -RS-232, RS-485, SPI, I2C, CAN, USB (v2.0), Bluetooth, Zig-Bee. (Frame formats of above protocols are not expected). Selection Criteria of Sensors and Actuators

2.1	Embedded Core.....	2-1
2.1.1	Microprocessor and Microcontroller.....	2-1
2.1.2	Digital Signal Processors (DSP) and Application Specific Processor (ASP).....	2-3
2.1.3	System On Chip (SOC).....	2-5
2.1.4	IP Core	2-6
2.1.5	Low Power Features	2-7
2.1.6	Idle Mode.....	2-7
2.1.7	Termination / Exit from Idle Mode	2-8
2.1.8	Power Down or Sleep Mode.....	2-9
2.1.9	Termination from Power Down Mode	2-9
2.2	Reduced Instruction Set Computer Principles	2-9
2.2.1	RISC versus CISC.....	2-10
2.2.2	RISC Properties	2-10



2.2.3	Register Window	2-10
2.2.4	Miscellaneous Features or Advantages of RISC Systems	2-11
2.2.5	Risc Shortcomings.....	2-13
2.2.6	ON-Chip Register File versus Cache Evaluation.....	2-14
2.2.7	FPGA and CPLD.....	2-14
2.3	Memory	2-15
2.3.1	Data Memories : SRAM and DRAM.....	2-15
2.3.2	Program Memories	2-17
2.4	Sensors/Input Devices (Resistive, Optical, Position, Thermal, Acceleration/ Gravity, Touch-Screen)	2-17
2.5	Actuators/Output Devices (Solenoid Valves, Relay/ Switch, Opto-couplers, 16×2 Alphanumeric Display).....	2-17
2.6	Communication Interface.....	2-18
2.7	Embedded Firmware (Boot-loader, RTOS, Drivers, Application Programs)	2-18
2.8	PCB and Passive Components.....	2-18
2.9	Safety, Reliability, Environmental, EM-Radiation Issues and Ethical Practice	2-18
2.10	Power Supply Considerations in Embedded Systems.....	2-19
2.10.1	Power-Supply and Oscillators.....	2-19
2.10.2	Power Supply	2-19
2.10.3	Oscillators	2-19
2.10.4	Battery Technology and Solar Operation.....	2-20
2.11	PWM Generation	2-20
2.12	Supervisory Circuits.....	2-21
2.12.1	Reset Circuit.....	2-21
2.13	Communication Interfaces.....	2-22
2.13.1	Introduction of Communication Interfaces	2-22
2.13.2	Types of Communication Systems	2-22
2.13.3	Serial Transmission Formats	2-23
2.13.3(A)	Asynchronous Data Transfer.....	2-23
2.13.3(B)	Synchronous Data Transfer	2-24
2.14	Serial Peripheral Interface (SPI) Bus.....	2-24
2.15	SCI (RS 232, RS 485)	2-26
2.15.1	RS 232 Standard.....	2-26
2.15.2	RS 485.....	2-28
2.15.2(A)	RS - 232 VS RS - 485	2-30



2.16	I ² C	2-31
2.16.1	Interfacing I2C EEPROM with LPC2148	2-31
2.17	Controller Area Network (CAN) Protocol.....	2-31
2.17.1	Features of CAN Standard.....	2-32
2.17.2	CAN Message Format.....	2-32
2.18	Field-bus (Profibus).....	2-33
2.18.1	Application Layer	2-33
2.18.2	Data Link Layer	2-33
2.19	USB (v2.0).....	2-34
2.19.1	Features of USB	2-34
2.19.2	USB Transfers.....	2-36
2.19.3	USB Commands	2-37
2.20	Bluetooth	2-39
2.21	Zig-Bee	2-39
2.21.1	Interfacing ZIGBEE.....	2-39
2.22	Introduction to Wireless Sensor Networks	2-41
2.23	Case Study: ARM Cortex-M3.....	2-42
2.24	Comparison of ARM-v7-A (CortexA8), ARM-v7-R (CortexR4), ARM-v7-M (Cortex-M3)	2-43
2.24.1	Cortex-A Series	2-43
2.24.2	Cortex-R Series	2-43
2.24.3	Cortex-M Series	2-44
2.24.4	† Introduction	2-44
2.24.4(A)	Features of LPC1768.....	2-45
2.24.4(B)	† Applications	2-48
2.24.4(C)	† Block Diagram of LPC1768.....	2-48
2.24.4(D)	† Functional Description.....	2-60
2.24.4(D).1	Architectural Overview.....	2-60
2.24.4(D).2	ARM Cortex-M3 Processor	2-60
2.24.4(D).3	Memory Map.....	2-60
2.24.5	† Architectural Overview	2-61
2.24.6	† ARM Cortex-M3 Processor.....	2-62
2.24.7	Special Registers in Cortex-M3.....	2-62



2.25	Memory Access and Addressing Modes	2-62
2.25.1	Addressing Modes for Data Processing Operands (i.e. op1)	2-63
2.25.2	Addressing Modes for Memory Access Operands.....	2-64
2.26	Exceptions/Interrupts	2-66
2.27	Processor Modes	2-68
2.28	The ARM Programmers Model.....	2-70
2.29	Program Status Registers	2-72
2.30	Barrel Shifter.....	2-76
2.31	Data Types.....	2-77
2.32	Exam Pack (Review and University Questions).....	2-77

Module 3

Chapter 3 : Embedded Software

3-1 to 3-62

Syllabus : Program Modelling concepts: DFG, CDFG, FSM. Real-time Operating system: Need of RTOS in Embedded system software and comparison with GPOS. Task, Task states, Multi-tasking, Task scheduling, and algorithms-Preemptive SJF, Round-Robin, Priority, Rate Monotonic Scheduling, Earliest Deadline First Inter-process communication: Message queues, Mailbox, Event timers. Task synchronization: Need, Issues- Deadlock, Race condition, live Lock, Solutions using Mutex, Semaphores. Shared Data problem, Priority inversion

3.1	Foreground/Background Systems	3-1
3.2	Program Modeling Concepts	3-1
3.2.1	Program Modeling Concepts : DFG, State Machine Programming Models (FSM, Petri-net), Modeling for Multi-Processor Systems, UML	3-1
3.2.2	Data Flow Graph (DFG) Model.....	3-2
3.2.3	State Machine Model.....	3-2
3.2.4	Sequential Program Model.....	3-3
3.2.5	Concurrent Process Model	3-4
3.2.6	Petri Nets Model.....	3-6
3.2.7	Unified Modelling	3-6
3.2.7(A)	Activity Diagram	3-7
3.2.7(B)	Class Diagram.....	3-7
3.3	Embedded C-programming Concepts	3-8
3.4	Assembly Language Based Embedded Software Development.....	3-8
3.4.1	Source Code to Object Code Translation	3-9
3.4.2	Advantage of Assembly Language Based Development.....	3-10



3.4.3	Disadvantages of Assembly Language Based Development.....	3-11
3.5	High Level Language Based Embedded Software Development.....	3-11
3.5.1	Comparison between Object Oriented and Procedural Language.....	3-12
3.5.2	Advantages of High Level Based Embedded Software Development.....	3-13
3.6	Need of an RTOS in Embedded System	3-13
3.7	RTOS Vs GPOS.....	3-15
3.8	The Real Time Kernel.....	3-16
3.8.1	The Scheduler and its Policies	3-16
3.8.2	Services	3-18
3.9	Characteristics of an RTOS.....	3-20
3.10	Real World Issues.....	3-21
3.11	Concepts of Tasks, Process and Threads	3-21
3.11.1	Types of Tasks.....	3-22
3.11.2	Thread	3-24
3.12	Multitasking.....	3-25
3.12.1	Types of Multitasking	3-25
3.13	Task Scheduling.....	3-26
3.13.1	Non-preemptive Scheduling.....	3-27
3.13.1(A)	First - Come - First - Served (FCFS) / FIFO Scheduling.....	3-27
3.13.1(B)	Last - Come - First - Served (LCFS) / LIFO Scheduling	3-29
3.13.1(C)	Shortest Job First (SJF) Scheduling.....	3-30
3.13.2	Priority Based Scheduling	3-32
3.13.3	Preemptive Scheduling	3-34
3.13.3(A)	Round Robin (RR) Scheduling	3-35
3.13.3(B)	Priority Based Scheduling	3-35
3.13.4	Classification of Real-Time Task scheduling Algorithms.....	3-37
3.13.4(A)	Clock-Driven Scheduling	3-38
3.13.4(B)	Event-Driven Scheduling.....	3-39
3.13.4(C)	Hybrid Scheduling.....	3-39
3.14	Task Management.....	3-42
3.15	Inter-Process Communication (IPC) and Synchronization.....	3-43
3.15.1	Queue.....	3-43
3.15.2	Circular Queue	3-44
3.15.3	List.....	3-44



3.15.4	Message Queues	3-45
3.15.5	Mail Boxes.....	3-46
3.15.6	Pipes	3-46
3.15.7	Shared Memory	3-47
3.15.8	Event or Signal Function	3-48
3.15.9	Remote Procedure Call (RPC) and Sockets	3-48
3.15.10	Task Synchronization	3-48
3.15.11	Priority Inversion	3-51
3.15.11(A)	Solutions to Priority Inversion	3-52
3.15.12	Semaphore Operations	3-53
3.15.13	Shared Data Problem and Techniques to Overcome	3-53
3.15.14	Dead Locks.....	3-57
3.16	Need of an RTOS in Embedded System	3-58
3.17	Exam Pack (Review and University Questions).....	3-59

Module 4

Chapter 4 : Introduction to FreeRTOS

4-1 to 4-8

Syllabus : FreeRTOS Task Management features, Resource Management features, Task Synchronization features, Event Management features, Calculation of CPU Utilization of an RTOS, Interrupt Management features, Time Management features.

4.1	Introduction to FreeRTOS	4-1
4.2	RTOS Management Features	4-2
4.3	Resource Management Features	4-2
4.4	Task Synchronization Features.....	4-3
4.4.1	FreeRTOS Recursive Mutexes	4-4
4.5	Event Management Features	4-4
4.5.1	Event Bits (Event Flags)	4-4
4.5.2	Event Groups	4-4
4.5.3	Event Group and Event Bits Data Types.....	4-5
4.5.4	Challenges in Implementing Event Groups.....	4-5
4.6	Calculation of CPU Utilization of an RTOS.....	4-6
4.7	Interrupt Management Features	4-6
4.8	Time Management Features.....	4-7
4.9	Exam Pack (Review Questions)	4-8

**Module 5****Chapter 5 : Testing and Debugging Methodology****5-1 to 5-28**

Syllabus : Testing & Debugging : Hardware testing tools, Boundary-scan/JTAG interface concepts, Emulator. Software Testing tools, Simulator, Debugger. White-Box and Black-Box testing.

5.1	GNU Debugger (gdb).....	5-1
5.1.1	GDB : The GNU Debugger.....	5-2
5.1.1(A)	Background.....	5-2
5.1.1(B)	Introduction to the GNU Debugger.....	5-2
5.1.1(C)	GDB Commands.....	5-3
5.1.1(D)	GDB Session : A basic example.....	5-6
5.1.1(E)	GDB in Embedded Environment.....	5-9
5.1.1(F)	Debugging a Core Dump.....	5-10
5.1.2	Data Display Debugger.....	5-12
5.1.3	cbrowser/cscope.....	5-13
5.1.4	Tracing and Profiling Tools.....	5-13
5.1.4(A)	strace.....	5-13
5.1.4(B)	ltrace.....	5-16
5.1.4(C)	ps.....	5-17
5.1.4(D)	Top.....	5-19
5.1.4(E)	mtrace.....	5-20
5.1.4(F)	dmalloc.....	5-20
5.1.4(G)	Kernel Oops.....	5-20
5.2	Testing and Debugging.....	5-21
5.2.1	Boundary-Scan/JTAG Interface Concepts.....	5-21
5.3	Comparison between Computation Models, Languages and Implementation.....	5-22
5.3.1	Program Validation and Testing.....	5-23
5.3.2	White - Box Testing.....	5-23
5.3.3	Black - Box Testing.....	5-25
5.4	Debugging Methodologies.....	5-25
5.4.1	Firmware (Embedded Software Debugging).....	5-25
5.4.1(A)	Simulator Based Firmware Debugging.....	5-25
5.4.1(B)	In-Circuit (Hardware) Emulators Based Firmware Debugging.....	5-26
5.4.1(C)	Monitor Program Based Firmware Debugging.....	5-26
5.4.1(D)	On Chip Firmware Debugging.....	5-27



5.4.2	Hardware Debugging.....	5-27
5.4.2(A)	Multimeter.....	5-28
5.4.2(B)	Oscilloscope and Logic Analyser.....	5-28
5.5	Exam Pack (Review and University Questions).....	5-28

Module 6

Chapter 6 : System Integration (Case Studies)

6-1 to 6-17

Syllabus : Embedded Product Design Life-Cycle (EDLC)- Waterfall Model. Hardware-Software Co-design. Case studies for Automatic Chocolate Vending Machine, Washing Machine, Smart Card, highlighting : (i) Specification requirements (choice of components), (ii) Hardware architecture (iii) Software architecture.

6.1	Embedded Product Design Life-Cycle (EDLC).....	6-1
6.1.1	Fundamentals of Design and Development.....	6-1
6.1.2	EDLC Phases	6-1
6.2	Hardware-Software Co-design.....	6-9
6.2.1	Modelling of Embedded System Development Life Cycle.....	6-9
6.3	Soft Real-time.....	6-11
6.3.1	Automated Vending Machine	6-11
6.3.2	Requirement Analysis	6-11
6.3.3	Hardware Block Diagram.....	6-11
6.3.4	System Model (FSM / UML).....	6-12
6.3.5	Software Architecture (Modules, Drivers).....	6-13
6.3.6	Component/Hardware Selection.....	6-13
6.3.7	Vegetable / Chocolate Vending	6-14
6.4	Waching Machine Control	6-14
6.4.1	Requirement Analysis	6-14
6.4.2	Hardware Block Diagram.....	6-14
6.4.3	System Model (FSM / UML).....	6-15
6.4.4	Software Architecture (Modules, Drivers).....	6-15
6.4.5	Component/Hardware Selection.....	6-15
6.5	Smart Card	6-16
6.5.1	Software Architecture (Modules, Drivers).....	6-17
6.6	Exam Pack (University and Review Questions).....	6-17