EMBEDDED SYSTEMS

ENEX 302

Lecture : 3 **Year : III**
Tutorial : 1 **Part : I**
Practical : 3/2

Course Objectives:
The objective of this course is to provide a comprehensive understanding of the design and development of embedded hardware and firmware, their integration and management of embedded system development process focusing on RTOS, microcontroller and VHDL.

1 Introduction (2 hours)

1.1 Definition of embedded system
1.2 Embedded system vs. general purpose computing system
1.3 Characteristics, classification and purposes of embedded system
1.4 Major application areas of embedded systems

2 Core of Embedded System (6 hours)

2.1 Elements and core of embedded system
  2.1.1 General purpose and domain specific processors: Microprocessor, microcontroller, digital signal processor (DSP)
  2.1.2 Application specific integrated circuits (ASICS)
  2.1.3 Programmable logic devices (PLDS)
  2.1.4 Commercial off-the-shelf components (COTS)
2.2 Sensors and actuators: LED, 7 segment display, optocoupler, stepper motor, relay, Piezo buzzer, push button switch, keyboard, PPI
2.3 Communication interface: I2C, SPI bus, UART, 1 – wire interface, parallel interface, IEEE 1394 (Firewire), Wi-Fi, Zigbee, Bluetooth
2.4 Embedded firmware, real-time clock (RTC) and watchdog timer
2.5 Quality attributes of embedded systems
  2.5.1 Operational quality attributes: Response, throughput, reliability, maintainability, security and safety
  2.5.2 Non-operational quality attributes: Testability and debug-ability, evolvability, portability, time to prototype and market and per unit cost and NRE cost

3 Hardware Design Issues (6 hours)

3.1 Transistors and logic gates: Logic gates implementation using CMOS
3.2 Review of combinational logic

3.3 Review if sequential logic
3.4 Design of custom single/dual purpose processor design
3.5 Optimization of custom single/dual purpose processor design

4 Designing Embedded System with Microcontroller (10 hours)

4.1 Microprocessor versus microcontroller
4.2 Factors for selecting a microcontroller (Overview of 8051/AVR/PIC/ARM cortex microcontroller)
4.3 Pin description of 8051
4.4 Designing with 8051
  4.4.1 8051 architecture
  4.4.2 Memory organization: Program and data memory, external program and data memory interfacing
  4.4.3 Registers
  4.4.4 Interrupt and interrupt systems
4.5 Timer units
4.6 Addressing modes and instruction set of 8051
4.7 Assembly programming

5 Embedded System Development Environment (4 hours)

5.1 Integrated development environment (IDE)
5.2 Keil µVision 3/4 IDE for 8051
5.3 An overview of ides for embedded system development
5.4 Files generate on cross – compilation: List file (.lst), preprocessor output file, object file (.obj), map file (.map), hex file (.hex)
5.5 Simulators, emulators and debugging

6 RTOS Based Embedded System Design (10 hours)

6.1 OS basics
  6.1.1 Definition and primary functions
  6.1.2 Kernel and its services: Process management, primary and secondary management, file system management, I/O devices management, protection, interrupt handler
  6.1.3 Kernel space and user space: Monolithic and micro kernels
6.2 Types of OS: General purpose and real time OS
6.3 RT kernel and its basic functions
  6.3.1 Task / process management
  6.3.2 Task / process scheduling
  6.3.3 Task / process synchronization
  6.3.4 Error / exception handling
  6.3.5 Memory management
  6.3.6 Interrupt handling
  6.3.7 Time management

6.4 Hard real time and soft real time
6.5 Process: Structure of process, process state and transition and PCB
6.6 Threads
  6.6.1 Concept of multithreading
  6.6.2 Thread standards: POSIX threads, Win32 threads, java threads
  6.6.3 Thread preemption: User level thread, kernel/system level thread, many – to – one model, one – to – one model, many – to – many model
  6.6.4 Thread vs. process
6.7 Multiprocessing and multitasking: preemptive, non – preemptive and cooperative
6.8 Task scheduling
  6.8.1 Factors for selecting a scheduling criterion
  6.8.2 Non – preemptive scheduling: FCFS/FIFO, LCFS/LIFO, SJF, priority based
  6.8.3 Preemptive scheduling: SRTF, RR, priority based
6.9 Deadlock
  6.9.1 Conditions for deadlock
  6.9.2 Deadlock handling: Ignore deadlocks, detect and recover, avoid deadlocks, prevent deadlocks, livelock and starvation
6.10 How to choose an RTOS
  6.10.1 Functional requirements
  6.10.2 Non – functional requirements

7 VHDL Coding and Logic Synthesis (4 hours)

7.1 Introduction, features, application, design flow and code structure
7.2 VHDL modeling styles: Behavioral model, dataflow model and structural model
7.3 Lexical elements: Library and packages, identifiers, keywords, numbers, character, string, data objects, data types, operator, data type conversion
7.4 Dataflow model: Concurrent statements
7.5 Behavioral modeling: Sequential statements
7.6 Structural modeling
7.7 Finite state machine (FSM) design using VHDL: Coding of counter, register, sequence detector and custom single/dual purpose processor

8 IC Technology (3 hours)

8.1 Introduction
8.2 Full-custom (VLSI) IC technology
8.3 Semi-custom (ASIC) IC technology
8.4 Programmable logic devices (PLD) IC technology

Tutorial **(15 hours)**

1. Design of custom single/dual purpose processor
2. Programming examples in 8051 microcontroller
3. Numerical examples based on different scheduling algorithms
4. Programming examples in VHDL

Practical **(22.5 hours)**

1. Programming with data transfer, arithmetic and logical instruction in 8051
2. Programming with branching and bit manipulation operations in 8051
3. Programming with timers for delay and wave pulse generation in 8051
4. Coding with VHDL for combinational and sequential logic in dataflow model
5. Coding with VHDL for combinational and sequential logic in behavioral model
6. Coding with VHDL for combinational and sequential logic in structural model

The simulator like Xilinx ISE design suite, keil, proteus and other tools can be used for programming in 8051 using assembly and C level and VHDL coding.

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Marks distribution*
1	2	2
2	6	8
3	6	8
4	10	14
5	4	6
6	10	12
7	4	6
8	3	4
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Shibu, K. V. (2012). Introduction to embedded systems. Tata McGraw Hill Education.
2. Vahid, F., Givargis, T. (2011). Embedded system design: A unified hardware/software introduction. Wiley India.
3. Lee, W. F. (2000). VHDL coding and logic synthesis with Synopsis. Academic Press.
4. Neupane, M., Shrestha, S. (2020). Embedded system design principles and practice (1st ed.). Pratibha Pustak Sadan and Stationery.