Title: Microprocessors and Interfacing

Code Number: EE3103

Credit Hours: 3 (3+1)

Prerequisites: MD1102 Computer Hardware Engineering, EE2102 Digital Logic Design

Semester: 6th

Course Objectives

The course will enable students to:

1. Describe microcontroller architectures and basic operations emphasizing the differences and advantages of various microcontroller families.
2. Demonstrate expertise in programming microcontroller timers and serial communication protocols to achieve precise timing control and effective real-world interfacing solutions.
3. Design and implement microcontrollers circuits to demonstrate proficiency in both assembly language and C programming, integrating hardware components effectively to implement control algorithms, interfacing, and real-time system applications.
4. Express understanding of concepts of microcontrollers, methodologies, and problem-solving strategies.

Contents

Unit 1: The Microcontrollers:

1. Microcontrollers and embedded processors
2. Overview of the microcontroller families

Unit 2: Assembly Language Programming:

1. Inside the microcontroller
2. Introduction to Assembly programming
3. Assembling and running microcontroller program
4. The program counter and ROM space in the 8051
5. Data types and directives
6. Flag bits and the PSW register
7. Register banks and stack

Unit 3: Jump, Loop, And Call Instructions:

1. Loop and jump instructions
2. Call instructions
3. Time delay for various chips

Unit 4: I/O Port Programming:

1. I/O programming
2. I/O bit manipulation programming

Unit 5: Addressing Modes:

1. Immediate and register addressing modes
2. Accessing memory using various addressing modes
3. Bit addresses for I/O and RAM

Unit 6: Arithmetic & Logic Instructions:

1. Arithmetic instructions
2. Signed number concepts and arithmetic operations
3. Logic and compare instructions

4. Rotate instruction and data serialization
5. BCD, ASCII, and other application programs

Unit 7: Programming In C:

1. Data types and time delay in C
2. I/O programming in C
3. Logic operations in C
4. Data conversion programs in C
5. Accessing code ROM space in C
6. Data serialization using C

Unit 8: 8051 Hardware Connection:

1. Pin description of the microcontroller
2. Explaining the Intel hex file

Unit 9: Timer Programming:

1. Programming microcontroller timers
2. Counter programming
3. Programming timers 0 and I in C

Unit 10: Serial Port Programming:

1. Basics of serial communication
2. Microcontroller connection to RS232
3. Microcontroller serial port programming
4. Programming the second serial port

Unit 11: Interrupts Programming:

1. Interrupts
2. Programming timer interrupts
3. Programming external hardware interrupts
4. Programming the serial communication interrupt
5. Interrupt priority

Unit 12: LCD And Keyboard Interfacing:

1. LCD interfacing
2. Keyboard interfacing

Unit 13: ADC, DAC, And Sensor Interfacing:

1. Parallel and serial ADC
2. DAC interfacing
3. Sensor interfacing and signal conditioning

Unit 14: Interfacing To External Memory:

1. Semiconductor memory
2. Memory address decoding
3. Microcontroller interfacing with external ROM
4. Data memory space

Unit 15: DS12887 RTC Interfacing and Programming:

1. DS12887 RTC interfacing
2. OSI2887 RTC programming in C
3. Alarm, SQW, and IRQ features of the DS 12887 chip

Unit 16: Motor Control: Relay, PWM, DC And Stepper Motors:

1. Relays and optoisolators
2. Stepper motor interfacing
3. DC motor interfacing and PWM

Lab Work Outline:

In this lab, students will engage in hands-on activities focusing on microcontroller programming and hardware interfacing on software and hardware tools. They will begin with foundational exercises in assembly language programming, learning to write and execute programs that control LEDs and interface with basic input/output devices. Progressing to higher complexity, students will advance to C programming, implementing algorithms for motor control using PWM signals and sensor interfacing for data acquisition. The lab will culminate in projects that integrate various peripherals such as LCD displays, keyboards, and analog-to-digital converters (ADCs), emphasizing real-world applications of embedded systems in controlling and monitoring physical devices. Through these practical exercises, students will develop essential skills in firmware development, hardware integration, and troubleshooting in embedded systems.

Teaching-Learning Strategies:

The pedagogical approach to this course relies on face-to-face teaching in a university classroom environment. The lectures are delivered using multimedia support and on whiteboard. Students are engaged and encouraged to solve real world problems using computer-aided tools.

Assignments/Types and Number with calendar:

A minimum of four assignments to be submitted before the written exams for each term.

Assessment and Examinations:

Sr. No.	Elements	Weightage	Details
1.	Midterm Assessment	35%	It takes place at the mid-point of the semester.
2.	Sessional Assessment	25%	It is continuous assessment. It includes classroom participation, attendance, assignments and presentations, homework, attitude and behavior, hands-on-activities, short tests, quizzes etc.
3.	Final Assessment	40%	It takes place at the end of the semester. It is mostly in the form of a test, but owing to the nature of the course the teacher may assess their students based on term paper, research proposal development, field work and report writing etc.

Recommended Books:

1. David Calcutt, 8051 Microcontrollers An Applications-Based Introduction, Oxford (2004)
2. Muhammad Ali Mazidi, Janice G. Mazidi, Rolin D. McKinlay, The 8051 Microcontroller and Embedded Systems, Prentice Hall;2nd edition (2005).
3. Muhammad Ali Mazidi, Sarmad Naimi, and Sepehr Naimi, The AVR Microcontroller and Embedded Systems: Using Assembly and C published by Pearson Custom Electronics Technology
4. Muhammad Ali Mazidi, Rolin D. McKinlay, and Danny Causey, PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC, Pearson.