Devi Ahilya University, Indore, India Institute of Engineering & Technology				III Year B.E. (Electronics and Instrumentation Engg.)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
5EIRC3 MICROCONTROLLERS	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
	3	**1**	**1**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Course Objective:

The course is designed

1. To learn the architecture of AVR.
2. To demonstrate the working of AVR microcontroller
3. To understand basics of interfacing and controlling of peripheral devices through AVR microcontroller.
4. To impart knowledge of programming in assembly as well as in embedded C of AVR microcontroller.
5. To introduce proteus tool to use to design real time applications.
6. Exposure to design and implement real time applications (prototype) using AVR microcontroller

Prerequisites:

Knowledge of Digital Logic Design, Microprocessor architecture, logical ability and programming skills.

COURSE CONTENTS

Unit –I

Introduction to Microcontroller**:** Microcontrollers and Embedded processors, Microcontroller survey, Overview of AVR family, AVR Microcontroller architecture, Register, status register, ROM space and other hardware modules, ATmega32 pin configuration & function of each pin.

Unit-II

AVR Assembly Language Programming: Addressing modes of AVR, Different instructions, assembly language programs, I/O Port Programming, Time delay loop, BCD, ASCII conversion Program, Look-up table, Bit addressability, Accessing EEPROM

Unit-III

AVR Programming in C and Timers: Data types, I/O programming, logic operations, Intel HEX file, Timer programming in assembly and C, Input capture and Wave Generator, PWM programming.

Unit-IV

Interrupt & Serial Port Programming: Interrupt environment, Interrupt programming and applications, Serial Port programming and application

Unit-V

Peripheral Interfacing: LCD and Keyboard Interfacing, ADC, DAC and sensor interfacing, Relay, Opto-isolator and Stepper Motor Interfacing, DC motor control, SPI protocol and Display interfacing, I2C Protocol and RTC interfacing.

Course Outcome:

Students earned credits will develop ability to

CO.No.	CO	PO
CO1	Understand the architecture and function of each pin of AVR 8-bit Microcontroller.	PO-1
CO2	Write, debug and simulate assembly as well as embedded C language programs.	PO-1, PO-2, PO-3
CO3	Understand various features like Timer operation, Interrupt Serial Communication etc . Interface I/O peripheral devices with microcontroller.	PO-2, PO-3, PO-4
CO4	Apply knowledge to design and simulate the real time applications using proteus tools.	PO-3, PO-5
CO5	Develop circuits/PCB to provide solution for the real time application.	PO-3, PO-6, PO-9, PO-10

CO-PO Relationship

CO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12
CO1	2											
CO2	2	3	3									
CO3		2	2	3								
CO4			3		3							
CO5			2			3			3	3		

BOOKS RECOMMENDED:

[1] The AVR Microcontroller and Embedded Systems Using Assembly and C, By Muhammad Ali Mazidi, Sarmad Naimi and Sepehr Naimi, Pearson Education.

[2] Programming and Customizing the AVR Microcontroller, By Dhananjay Gadre, McGraw Hill Education.

[3] AVR ATmega32 data sheet.

List of Practical Assignments:

1. BASIC PROGRAMS IN ASSEMBLY LANGUAGE :-

I. Write a program to Toggle all bits of PORTB by sending value $55 and $AA continuously. Put delay between each issuing of data to PORTB.

II. Write a program when an LED is connected to each pin of PORTD. Turn on each LED from pin D0 to D7.

III. Write a program to create a square wave of 50% duty cycle on bit 0 to PORTC. IV. Write a program when a Switch is connected to pin PB2. a) IF SW=0, send the letter 'N' to PORTD. b) IF SW=1, send the letter 'Y' to PORTD. V. Write a program when a Switch is connected to pin PB0 and an LED to pin PB7. To get status of SW and send it to the LED.

2. AVR PROGRAMMING in C:-

I. Write a program to toggle all bits of PORTB 200 times.

II. Write a program to toggle all bit PORTB continuously with 100ms delay XTAL=8MHz.

III. WAP to read pins 1 and 0 of PORTB, send ASCII code to PORTD as per following status of Pin1 Pin0. Institute of Engineering & Technology, Devi Ahilya University, Indore, (M.P.), India. (Scheme Effective from July 2017) Pin1 Pin0 0 0 send '0' to port D 0 1 send '1' to port D 1 0 send '2' to Port D 1 1 send '3' to Port D

IV. Write a program to send value 44H serially one bit at a time via PORTC pin3 (LSB FIRST).

V. Write a program to store 'G' into location 0x005F of EEPROM.

3. CODE CONVERSION (IN ASSEMBLY AND C):-

I. Write a program to perform checksum byte calculation for data.

II. Write a program to calculate the checksum byte for given data.

III. Write a program to convert hexadecimal number FDH to decimal and display digit on PORTB, PORTC, PORTD.

IV. Write a program to convert Packed BCD to ASCII.

V. Write a program to convert ASCII to packed BCD.

4. TIMER PROGRAMMING (IN ASSEMBLY AND C):-

I. Write a program to toggle all bits of PORTB using some delay. Use CTC mode. Assume XTAL=8MHz.

II. Write a program using of prescaler 64 to generate delay of 1920 µs. Assume XTAL=8MHz.

III. Toggle only the PORTB.4, Use Timer1, normal mode, no prescaler to create the delay of 2ms.

IV. Use TOV0 flag to extend Timer0 to a 16-bit counter and display on PORTC and PORTD.

5. INTERRUPT AND SERIAL PORT PROGRAMMING (IN ASSEMBLY AND C):-

I. Using Timer1, toggle pin PORTB.5 every second, while at the same time transfer data from PORTC to PORTD.

II. Write a program to receive bytes of data serially and put on PORTB. Set baud rate =9600 & use 1 stop bit.

III. Write a program to toggle PORTC.3, whenever INT0 goes low.

IV. Write a program to transmit the letter 'G' serially at 9600baud, continuously. Do this task with interrupt and without Interrupt.

6. LCD, KEYBOARD AND OTHER PERIPHERAL INTERCACING AND PROGRAMMING (IN ASSEMBLY AND C)

I. Write "HELLO" on the LCD using port A of Atmega 32.

II. Interface Keyboard with Atmega 32 and write a program to send ASCII code to Port D for any key pressed.

III. Interface DAC 0808 with Atmega 32 and generate a triangular wave.

IV. Interface LM 34 with Atmega 32 and write a program to read the sensor continuously and display the reading on Port D.

V. Interface Stepper