Marwadi University
Faculty of Technology
Department of Information and Communication Technology

Subject Code: 01CT0403
Subject Name: Microcontroller and Interfacing
B. Tech. Year – II (Semester IV)

Objective:

This course introduces the architecture, assembly language and C language programming of ATmega32 AVR family microcontroller. It gives a hands-on training of interfacing external sensors andactuators with microcontroller. The course objective is to introduce the basic concepts of small and medium scale embedded system design using microcontroller and to develop assembly and C language programming skills for real time applications of Microcontroller.

Credits Earned: 04 Credits

Course Outcomes: After completion of this course, student will be able to:

1. Acquire basic knowledge of microcontroller and utilize real time software and hardware for embedded systems using AVR Atmega-32 microcontroller.
2. Understand architecture of Atmega-32, its pin configuration, data-types, instruction set, addressing modes and advance communication protocols like SPI, I2C etc.
3. Develop assembly and C language programs for ADC, EEPROM, PWM and Timer by applying various instructions like data transfer, ALU, Branch, subroutine etc.
4. Analyze I/O peripherals like LCD, Keyboard, Relay, Sensor, Motor etc. by interfacing it with AVR microcontroller.
5. Evaluate minor microcontroller-based projects that solves real world problems.

Pre-requisite of course:

Basics of Digital Logic Design, Microprocessor architecture and basics of C programming

Teaching and Examination Scheme:

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial / Practical Marks		Total Marks
				E	I		V	T	
Theory	Tutorial	Practical		ESE	IA	CSE	Viva	Term Work	
03	00	02	04	50	30	20	25	25	150

Contents:

Unit	Topics	Hours
1	**Introduction to microcontroller** Microprocessor and Microcontroller difference, RISC and CISC programmer's model, Criterial for selecting microcontroller	04
2	**Introduction to AVR microcontroller** Overview of AVR family, AVR Microcontroller architecture, status register, Special function registers, RAM, ROM & EEPROM space, On-Chip peripherals, ATmega32 pin configuration & function of each pin, Fuse bits of AVR.	07
3	**AVR assembly language programming** AVR data types and assembler directives, addressing modes of AVR, Data transfer, Arithmetic, Logic and Compare, Rotate and Shift, Branch and Call instructions, AVR studio setup for assembly language programming, AVR I/O Port Programming, Time delay loop, Look-up table, Bit addressability, MACROs, Intel HEX file.	08
4	**AVR programming in C** AVR Data types, AVR I/O port programming, Timer programming, Input capture and Wave Generator, PWM programming External Interrupt programming, ADC programming, EEPROM programming.	08
5	**Serial communication protocols** UART protocol, I2C protocol, SPI protocol, Serial Port programming using polling and interrupt, I2C Programming, SPI Programming	07
6	**Peripheral interfacing** LCD and Keyboard Interfacing, Relay interfacing, Stepper and DC Motor control, DS1307 RTC Interfacing, LM35 Temperature sensor interfacing, MAX7219 display controller interfacing,	08
	Total Hours	42

Suggested Text books / Reference books:

1. The AVR Microcontroller and Embedded Systems Using Assembly and C, By Muhammad AliMazidi, Sarmad Naimi and Sepehr Naimi, Pearson Education.
2. Programming and Customizing the AVR Microcontroller, By Dhananjay Gadre, McGraw HillEducation
3. AVR ATmega32 data sheet

Suggested Theory distribution:

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves asguidelines for teachers and students to achieve effective teaching-learning process.

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyze	Evaluate	Create
25%	20%	30%	5%	5%	15%

Suggested List of Experiments:

Minimum 12 experiments to be performed during the semester

1. Installation of AVR STUDIO and familiarization of ATMega32 AVR Development Board.
2. Hands-on experimentation of ATMega32 GPIO programming in Assembly and C.
3. Hands-on experimentation of ATMega32 Timer to generate accurate delay using polling inAssembly and C.
4. Hands-on experimentation of ATMega32 Timer to generate accurate delay using Interrupt inAssembly and C.
5. Hands-on experimentation of ATMega32 Timer to generate waveforms in Assembly and C
6. Hands-on experimentation of Seven Segment Display interfacing with ATMega32 in Assembly andC.
7. Hands-on experimentation of 16x2 LCD interfacing with ATMega32 in Assembly and C.
8. Hands-on experimentation of ATMega32 UART programming in Assembly and C.
9. Hands-on experimentation of 4x4 matrix keyboard interfacing with ATMega32 in Assembly and C
10. Hands-on experimentation of ATMega32 on-chip ADC for interfacing analog sensors inC.
11. Hands-on experimentation of DC motor interfacing and speed/direction control with ATMega32in C.

12. Hands-on experimentation of Stepper motor interfacing with ATMega32 in C.
13. Hands-on experimentation of DS1307 RTC Interfacing with ATMega32 in C using I2C protocol.
14. Hands-on experimentation of MAX7219 LED matrix driver Interfacing with ATMega32 in C using SPI protocol.
15. Design Frequency Counter which displays frequency of unknown pulse on 16x2 LCD using ATMega32 on-chip Timer.
16. Design Pulse period meter which displays ON-time of unknown pulse on 16x2 LCD using ATMega32 on-chip Timer
17. Design Bluetooth controlled 2-ch variable frequency square wave generator using ATMega32 UART and on-chip Timer.
18. Design 4 Channel Data Logger which measures Voltage between 0-5V on 4 ADC Channels of ATMega32 and transmit it to Host PC at every 1 second where it stored in excel sheet with timestamp for future analysis.

Supplementary Resources:

1. http://nptel.ac.in/courses/106108100/7
2. http://nptel.ac.in/courses/Webcourse- contents/IIT-KANPUR/microcontrollers/micro/ui/TOC.htm
3. https://swayam.gov.in/course/4446-microprocessors-and-microcontrollers
4. https://www.coursera.org/courses?languages=en&query=microcontroller
5. http://www.study-hub.com/avr-microcontroller-programming.html