Embedded Systems (GE)

Credits: Theory-03 **Theory Lectures:45**

Course Learning Objectives

The course is designed to make students familiar with principles, features, classification, architectures, and design issues involved in embedded system. The selection criteria for choosing microcontroller based on system requirement in embedded systems is also discussed. Interrupts and interfacing concepts are included. A balance between hardware and software exposure is maintained.

Course Learning Outcomes

At the end of this course, students will be able to:

CO1 Learn the fundamental concepts related to embedded systems and architecture of microcontrollers.
CO2 Familiarize with Instruction Set and assembly language.
CO3 Understand the interrupts and interfacing concepts for common applications like general I/O, Timer and Counter.
CO4 Demonstrate knowledge of the development tools for a microcontroller and write assembly language code according to specifications and task.

Prerequisites: Basic knowledge of digital circuits, idea about microprocessors

L-T-P: 3-0-1

Syllabus Contents

Unit – 1 **(11 Lectures)**

Introduction to Embedded system: Embedded systems features, Categories of Embedded System: Standalone, Real Time, Networked, Mobile devices, Embedded system architecture Hardware architecture, processors, memory EPROM, EEPROM, FLASH, SRAM DRAM FRAM (qualitative idea of read write access times and board space), Ready to use Embedded design platforms (Arduino, Raspberry Pi), Use of IoT and machine learning in Embedded Systems.

Unit 2 **(11 Lectures)**

AVR Microcontroller: Overview of Harvard architecture and Von Neumann architecture, RISC and CISC microcontrollers. Introduction to ATMega32 AVR RISC microcontroller, Criteria for choosing a microcontroller, architecture overview, Status Register, General Purpose Register file, reset sources (Power-on, Brownout detector & Watchdog Timer)

Unit –3 **(11 Lectures)**

Instruction Set: Addressing Modes, Data Transfer Instructions, Arithmetic and Logic Instructions, Shift and Rotate instructions, Branch Instructions, Bit manipulation and Bit-test Instructions, MCU Control Instructions.
Assembly language.

Unit – 4 **(12 Lectures)**

Peripheral Interfacing: Input/Output Ports, configuring I/O ports, reading and writing data to I/O ports, Introduction to Interrupts, interrupt vector address and priority, ISR, External Interrupts, Introduction to Timers, Timer 0 modes of operation, Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART).

References

1. AVR Microcontroller and Embedded Systems: Using Assembly and C by Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, PHI ,2013
2. Embedded system Design - Frank Vahid and Tony Givargis, John Wiley, 2002
3. Programming and Customizing the AVR Microcontroller by D. V.Gadre, McGraw-Hill ,2000
4. Atmel AVR Microcontroller Primer: Programming and Interfacing by Steven F. Barrett, Daniel J. Pack, Morgan & Claypool Publishers ,2012
5. AVR Microcontroller Datasheet.

Embedded Systems Lab

SOFTWARE REQUIRED: AVR STUDIO/similar IDE

Credits:01 **Practical Lectures:30**

Course Learning Outcomes

At the end of this course, students will be able to:

CO1 Program microcontroller for common applications like general I/O Port, data transfer, counter.

CO2 Use various peripherals on the microcontroller to implement systems, interrupts driven I/O and modes of timer/ counter.

CO3 Prepare the technical report on the experiments carried.

Syllabus Contents

1. Write a program to flash LED at an observable rate.
2. Write a program to flash LED at an increasing rate till it appears to be always ON.
3. Write a program to generate random numbers using LFSR(Linear feedback Shift Register).
4. Write a program to get data from Port A and Port B and perform arithmetic and logical operations. Display the result on output device.
5. Write a program to read the status of pin 0 of port B and if it is High then send FFH on port D else AAH on port D.
6. Write a program to generate a pulse with different duty cycles on bit 0 of Port D using timer.
7. Write a program to implement a decimal counter and display it on an output device.
8. Write a program to generate square wave on Port D using timer simultaneously transfer the contents from Port B to Port C