EEE.7-3.1 Microcontrollers - Embedded Systems

COURSE CONTENT

Introduction to Embedded Systems – Basic Concepts

- Generic computer – Embedded System
- Hardware/software co-design
- Basic Features of Embedded Systems
- Embedded Systems Applications
- Basic Parts of Embedded Systems
- Embedded systems as Cyber-Physical Systems
- Limitations on Embedded Systems Applications
- Design and Operation Parameters

The Microcontroller as an Embedded System Platform

- Basic concepts and principles of operation – Functional diagrams
- Control, Data and Address buses
- Memory systems
- Microcontrollers' architecture (von Neuman – Harvard)
- Instruction set (CISC, RISC, VLIW)
- Pipelining
- Programing model (Accumulator Based - General Purpose Registers)
- Basic microcontrollers' interface circuits

AVR microcontrollers' families

- Features of 8-bit AVR microcontrollers
- Memory types (data memory: SRAM - EEPROM, program memory: FLASH)
- Introduction to AVR instruction set (Op-Code, commands' execution time)
- AVR Assembly Program outline
- Introduction to interrupt vectors
- Programming in assembly for AVR
- Peripherals of AVR microcontrollers
- Timing circuits
- Parallel I/O ports

Software Architecture for Embedded Systems

- Data Input/Output
- Operators
- Loops
- Bits Management
- Logical operators by binary digit
- Binary digit control

- Shift operators by binary digit
- Binary digit fields
- Peripheral devices I/O commands

Programming microcontrollers in higher-level language

- General Program Outline
- Data Types
- Functions
- Variables - Constants
- Bit Variables
- Casting
- Pointers
- EEPROM access
- Structures
- Definitions – Macros
- Incorporation of Symbolic Language Programs
- I/O Registers Access
- I/O Registers Access at Bit Level
- Library Functions
- Integration of LCD and 7 Segment displays
- Serial communication
- Communication through I2C protocol
- Communication through SPI protocol
- Use of LCD
- ADC– Control, use and Applications
- Real-time programming – External Interrupts
- Use of timer/counters, DC Motors
- Connection and control of graphical LCD screens
- Use of Pulse Wave Modulation

Hands on Laboratory Exercises

The laboratory training incorporates 13 laboratory exercises focused on the main subjects of theoretical teaching. The exercises will cover the following fields:

- Introduction to AVR microcontrollers
- STK600 development board and Arduino
- Familiarization with the Microchip Studio programming environment
- Assembly instruction set – Timing
- Assembly Program outline – Directives
- Two-way I/O ports
- Program outline in C language for microcontrollers (I/O Ports)

- Use of alphanumeric LCD display
- Serial communication (USART, I2C)
- ADC applications
- Timers/Counters (PWM using timers)
- Real-Time Programming - Interrupts

LEARNING OUTCOMES

This course aims to introduce students to the modern field of technology of embedded devices and their wide applications. At the same time, the aim is to familiarize students with the concept and use of microcontrollers in the context of the integrated device, which is characterized by the integration of hardware and software in a special-purpose system. Therefore, an additional goal of the course is to introduce students to the world of microcontrollers and their capabilities. The embedded implementation technologies that will be presented will be based on microcontrollers.
One of the microcontroller family used today in open-source platforms is the AVR microcontrollers. This is the main reason why this family of microcontrollers was chosen to study such systems. Thus, first the general architecture of the central processing unit is presented, and then some of the main peripheral units that it integrates are analyzed. For a full understanding of their function, examples of programming in Assembly symbolic language as well as compilation in machine language (op-code) are presented. In addition, the use of the upper-level programming language (C) for microcontrollers and the analysis of compilation in machine language is introduced. In particular, the basic programming structures and the use of serial communication and LCD devices are examined through the C language. In parallel, the operation of the interrupts and the integrated ADC devices are analyzed. An additional goal is to introduce students to the concept of real-time programming.

Upon successful completion of the course, the student will be able to:

- Understand the technology of embedded systems and their manufacturing processes.
- Analyze circuits with microcontrollers.
- Program AVR microcontrollers in symbolic language (Assembly).
- Program AVR microcontrollers in a high-level language.
- Program at a low level and use the basic peripherals of microcontrollers (ADC, UART, LCD, Timers / Counters)
- Deal effectively with problems that require real-time scheduling using interrupts.
- Design and develop simple integrated systems with microcontrollers.