



Software Programming Course

CURRICULA 2022 - 2023

Obiectiv:

Cursul pune bazele programarii in limbajul C a sistemelor Embedded cu microcontrollere, cu aplicatii in numeroase domenii, inclusiv in automotive, industria aerospaciala, telecomunicatii, medicina, o gama variata de sisteme electronice si media cu control inteligent.

Cursul este adaptat la cerintele actuale ale Continental Automotive Systems.

Beneficii pentru cursanti

Programare C:

- Vor avea o imagine de ansamblu despre ce înseamnă programarea in general
- Vor intelege logica si rolul unui algoritm
- Vor intelege care sunt etapele dezvoltarii unei aplicatii software
- Vor intelege structura si sintaxa limbajului de programare C
- Vor intelege si folosi notiuni abstracte de programare (variabile, constante, tipuri de date, functii predefinite si definite de utilizator, array-uri etc)
- Vor intelege si trata exceptiile si conditiile de eroare

Sisteme Embedded:

- Vor cunoaste functionalitatile si arhitectura unui microcontroller
- Vor putea configura si implementa comunicarea intre microcontroller si diverse periferice / senzori
- Vor intelege particularitatile fiecarui tip de interfata si de ce este folosita
- Vor putea solutiona problemele tipice care apar in proiectarea unei aplicatii bazate pe microcontrollere
- Vor putea intelege specificatiile hardware si software pentru un sistem embedded



Continut curs

1. LIMBAJUL DE PROGRAMARE C (64 ore):

Introduction / Getting Started

- Introduction to programming languages
- Introduction to C
- Introduction to development
- Installing an Integrated Development Environment (IDE)
- Compiling your first program
- Build configurations
- A few common C problems

C Basics

- Structure of a program
- Comments
- A first look at variables, initialization, and assignment
- A first look at cout, cin, and endl
- A first look at functions and return values
- A first look at function parameters and arguments
- Why functions are useful, and how to use them effectively
- Keywords and naming identifiers
- A first look at local scope
- A first look at operators
- Whitespace and basic formatting
- Forward declarations and definitions
- Programs with multiple files
- Naming conflicts and the std namespace
- Header files
- A first look at the preprocessor
- Header guards
- How to design your first programs
- Debugging your program (stepping and breakpoints)
- Debugging your program (watching variables and the call stack)

Variables and Fundamental Data Types

- Fundamental variable definition, initialization, and assignment
- Void
- Variable sizes and the sizeof operator
- Integers
- Fixed-width integers
- Floating point numbers
- Boolean values and an introduction to if statements



- Chars
- Literals
- Const, constexpr, and symbolic constants
- Chapter 2 comprehensive quiz
- Operators
- Operator precedence and associativity
- Arithmetic operators
- Increment/decrement operators, and side effects
- Sizeof, comma, and conditional operators
- Relational operators (comparisons)
- Logical operators
- Converting between binary and decimal
- Bitwise operators
- Bit flags and bit masks

Variable Scope and More Types

- Blocks (compound statements)
- Local variables, scope, and duration
- Global variables and linkage
- Why global variables are evil
- Static duration variables
- Scope, duration, and linkage summary
- Namespaces
- Using statements
- Implicit type conversion (coercion)
- Explicit type conversion (casting)
- An introduction to std::string
- Enumerated types
- Enum classes
- Typedefs and type aliases
- Structs
- The auto keyword

Control Flow

- Control flow introduction
- If statements
- Switch statements
- Goto statements
- While statements
- Do while statements
- For statements
- Break and continue
- Random number generation



Arrays, Strings, Pointers, and References

- Arrays and loops
- Sorting an array using selection sort
- Multidimensional arrays
- Introduction to pointers
- Null pointers
- Pointers and arrays
- Pointer arithmetic and array indexing
- For each loops
- Void pointers

Functions

- Function parameters and arguments
- Passing arguments by value
- Passing arguments by reference
- Passing arguments by address
- Returning values by value, reference, and address
- Inline functions
- Function overloading
- Default parameters
- Function Pointers
- The stack and the heap
- Recursion
- Handling errors, cerr and exit
- Assert and static_assert
- Command line arguments
- Input and output (I/O)
- Input and output (I/O) streams
- Input with istream
- Output with ostream and ios
- Stream classes for strings
- Stream states and input validation
- Basic file I/O
- Random file I/O



2. EMBEDDED SYSTEMS

Introduction to embedded systems

Introduction to microprocessor and microcontroller

What is a microprocessor

What is a microcontroller

- Difference between microprocessor and microcontroller
- Different type of microcontroller
- Importance of microcontroller in embedded system

Introduction to Arduino board and platform

- An overview of open hardware
- Arduino board description

Introduction to software and tool chain

- Software installation
- Starting with arduino IDE
- Programming basics of Arduino
- Configuring I/O and controlling I/O
- Setting mode of pin as input and output
- Driving pin high and low
- Led interfacing
- Reading input pin data
- Led control with button

Segment interfacing

- Introduction to 7segment
- Interfacing a 7segment to Arduino
- Interfacing Multi segment Displays

Memory drivers

Multiplexing

- Introduction to multiplexing
- 7 segment multiplexing
- Introduction of 74595 serial buffer IC
- Led multiplexing
- 16x2 lcd inter with Arduino
- Introduction to 16x2 LCD
- LCD programming with Arduino
- Scrolling message Display



- Writing order change
- Custom character on Arduino

DC motor interfacing with Arduino

- Introduction to DC motor
- DC motor Driver L293D
- DC motor Programming

Stepper motor interfacing with Arduino

- Introduction to stepper motor
- Stepper motor Driver ULN2803
- Programming stepper motor

Servo motor interfacing

- Introduction to servo motor
- Servo motor programming

ADC

- Introduction to ADC
- ADC programming
- Voltmeter sensor using ADC
- Temperature sensor using ADC
- Analog Sensors (Temperature, movement, light etc)

Serial Communication

- Introduction to UART
- Programming UART
- PC based LCD e-notice board
- PC based controlled robot

Interrupt programming

- Introduction to interrupt
- Interrupt programming

PWM with Arduino

- Introduction to PWM
- Generating different duty cycle of PWM

Tone generation

- How a tone is generated
- Programming different tones with different amount of delay with Arduino

Relays



- Type of relays.
- Working of a magnetic relay.
- Controlling Electrical appliances with electromagnetic relays.

Introducere in protocoale de comunicare in automotive

- CAN protocol : basics , frames , utility and importance.
 - CAN Network
 - CAN Node
 - CAN Controller
 - CAN Transceiver
 - CAN Bus
 - CAN Bus Levels
 - CAN Bus Logic
 - CAN Frame
 - Communication Principle
- LIN protocol : basics , frames , utility and importance.
 - MASTER AND SLAVE
 - LIN APPLICATIONS
 - LIN FRAME
- FLEXRAY protocol : basics , frames , utility and importance.
 - COMPARISON WITH CAN
 - FLEXRAY NEW FEATURE
 - COMMUNICATION CYCLE
 - SEGMENTS
 - FLEXRAY FRAME
 - DEMONSTRATIONS
 - PHYSICAL BUS ARCHITECTURE
 - SYNCHRONIZATION
- Vectopr tools, simularea comunicatiei inter-ECU (in automotive)