III - Detailed programme by subject

1. Detailed program by subject of the semester S1

Semester:1
Teaching unit: UEF 1.1.1
Subject 1: Advanced Digital Systems
VHS: 45h00 (Lecture: 3h00)
Credits: 5
Coefficient: 3

Teaching objectives:

To be competent in designing complex digital systems using VHDL and their Implementation in FPGAs.

Recommended prior knowledge:

Digital Systems Design with VHDL I and II

Computer Architecture.

Material content:

- **Review of logic design fundamentals** **(01 Week)**
- **Structural design modelling** **(01 Week)**
- **Design, optimization and synthesis of FSM using VHDL** **(02 Weeks)**
- **Advanced topics in VHDL** **(02 Weeks)**
- **Advanced LPMs** **(02 Weeks)**
- **Register files in VHDL** **(01 Week)**
- **Design & synthesis of datapath controllers with VHDL** **(02 Weeks)**
- **Programmable logic devices** **(02 Weeks)**
- **FPGA Architecture and technologies** **(02 Weeks)**

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

References:

1. Charles Roth, "*Digital Systems Design Using VHDL*", 1998.
2. Ion Grout, "*Digital Systems Design with FPGAs*", 2008.

Semester:1
Teaching unit: UEF 1.1.1
Subject2: Computer-Controlled Systems
VHS: 45h00 (Lecture: 3h00)
Credits: 5
Coefficient: 3

Teaching objectives:

Students will get hands-on the field of digital control system with I/O interface which is widely applied in industry, this course addresses the practical aspects of the implementation of common digital controllers on digital computers as well as the study of the used Platforms and Tools.

Recommended prior knowledge:

Programming I, Computer Architecture, Microprocessor Systems design
Linear control Systems, Process Control and Instrumentation.

Material content:

1. **Introduction to Digital Control Systems** **(02 Weeks)**
   - Overview of control systems
   - Basics of digital control and advantages
   - Comparison of analog and digital control systems
   - Discretization of continuous-time systems
2. **Digital Controller Design** **(02 Weeks)**
   - Transfer functions and block diagrams in discrete-time
   - Z-transform and its applications
   - Design of digital controllers (PID, state-space, etc.)
   - Stability analysis in the discrete domain
3. **Implementation Platforms and Tools** **(03 Weeks)**
   - Introduction to microcontrollers and digital signal processors (DSP)
   - Programming languages for controller implementation (C, Assembly)
   - Overview of control development environments (MATLAB/Simulink, LabVIEW)
   - Building simple control systems using simulation tools.
4. **Digital Controller Tuning Techniques** **(02 Weeks)**
   - Tuning methods for digital controllers
   - Ziegler-Nichols tuning rules for digital systems
   - Frequency domain tuning techniques
   - Practical examples and case studies

- Implementing and tuning PID controllers.
- Addressing challenges such as derivative noise and integral windup.

5. Hardware-in-the-Loop (HIL) Simulation (03 Weeks)

- Introduction to HIL simulation
- Connecting real controllers to simulation environments
- Advantages and applications of HIL simulation
- I/O Interfaces (ADC, DAC, Delays, Sensors, and Actuator Interfacing)
    - Interfacing with sensors and actuators.
    - Understanding Analog-to-Digital (ADC) and Digital-to-Analog (DAC) conversion.

6. Practical Implementation Challenges (03 Weeks)

- Anti-aliasing and reconstruction filters
- Quantization effects in digital controllers
- Signal conditioning and sensor interfacing
- State-Feedback and Observer Feedback Implementations
    - Implementing state-feedback and observer feedback controllers.
    - Addressing practical challenges in state-space control.

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. J. Ledin, "*Embedded Control Systems in C/C++: An Introduction for Software Developers Using MATLAB*", CMP books, 2003.
2. T. Wescott, "*Applied Control Theory for Embedded Systems*", Newnes, 2006.
3. K.J. Åström and B. Wittenmark, "*Computer-Controlled Systems: Theory and Design*", Courier Corporation, 2011.

Semester:1
Teaching unit: UEF 1.1.2
Subject 1: Advanced Programming
VHS: 45h00 (Lecture: 3h00, TD: 00h)
Credits: 5
Coefficient: 3

Teaching objectives:

This course will focus on the study in some depth of the programming language Java. At the end of the semester the student will grasp the powerful features of Java, such as encapsulation, inheritance, interface, polymorphism, generics, collections etc.

Recommended prior knowledge:

Programming I.

Material content:

- **Main Features of Object Oriented Programming Languages.** **(02 Weeks)**
- **An Overview of Java.** **(02 Weeks)**
- **Introduction to Classes.** **(02 Weeks)**
- **In-depth Study of Classes. Inheritance.** **(02 Weeks)**
- **Packages and Interfaces.** **(01 Week)**
- **Exceptions and Multi-Threadings.** **(02 Weeks)**
- **I/O and Applet Interfaces.** **(02 Weeks)**
- **Generics.** **(01 Week)**
- **The API Library** **(01 Week)**

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references :

1. H. Shildt. "*Java, a beginner's guide*". McGraw Hill, 2005
2. H. Shildt. "*Java cookbook*". Mc Graw Hill 2008
3. H. Deitel. "*Java for programmers*". Pearson Education 2012.

Semester: 1
Teaching unit: UEF 1.1.2
Material 2: Applied Optimization
VHS:45h00 (Lecture: 1h30, TD: 1h30)
Credits: 5
Coefficient: 2

Teaching objectives:

Many problems in science, technology, economy etc. can be modeled by mathematical formalisms whose exact solutions are either unknown or are computationally expensive. In these cases, a set of techniques, known collectively as Numerical Methods, produce efficiently approximate solutions to exact solutions.
Optimization is a very powerful and versatile instrument which could potentially be applied to any engineering discipline. Engineers may intend to maximize or minimize some factors in their problems and they are constantly searching for optimal solutions. Several general approaches for optimization are available; the most important general one is based on numerical methods. The aim of this course is to introduce students to numerical methods and optimization, to provide the basic concepts and the implementation of some methods and algorithms.

Recommended prior knowledge:

Basic courses in calculus and computer programming

Material content:

Chapter 1: Solutions of equations of one variable. **(01 Week)**
- Secant Method, Newton-Raphson Method.

Chapter 2: Solution to a system of equations. **(1.5 Weeks)**
- Gauss Elimination, Gauss-Siedel Methods.

Chapter 3: Numerical Integration. **(01 Week)**
- Quadrature Rules, Romberg Integration.

Chapter 4: Interpolation and Polynomial Approximation **(01 Week)**
- Lagrange Polynomials, Spline Interpolation.

Chapter 5: Numerical Solution of Initial-Value Problems **(1.5 Weeks)**
- Runge-Kutta's Methods.

Chapter 6: Introduction to Optimizations: **(03 Weeks)**
- Formulation and modeling with examples.
- Basic concepts: Objective function, Decision variables, Constraints.
- Local and Global Optimality.
- Types of optimization problems: Definitions and application examples(Linear, nonlinear, constrained, unconstrained, convex, non-convex, single objective, multi-objective, deterministic, stochastic, continuous, and discrete).

Chapter 7: Linear Programming **(1.5 Weeks)**

- Standard form of LP.
- LPs examples.
- Simplex Method.
- Application.

Chapter 8: Metaheuristics **(2.5 Weeks)**

- Heuristic vs Stochasticapproach.
- Exploration and exploitation.
- Metaheuristics Classifications.
- Metaheuristic Algorithms:Genetic Algorithm, Deferential Evolution, Particle Swarm Optimization, Simulated Annealing.
- Parameter tuning of Metaheuristics.
- Application.

Chapter 9: Multi objective optimization. **(01 Week)**

- Pareto Optimality.
- Weighted Sum Method.
- Utility Method.
- Application.

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. H.M. Antia. "*Numerical Method for Scientists and Engineers*". McGraw Hill, 1995.
2. W. Dos Passos. "*Numerical Methods, Algorithms and Tools*". Taylor and Francis Group, 2010.
3. Yang, X. S. "*Engineering optimization: an introduction with metaheuristic applications*". John Wiley & Sons, (2010).

Semester:1
Teaching unit: UEM1.1.1
Subject 1: Advanced Digital Systems Lab
VHS: 45h00 (TP: 3h00)
Credits: 2
Coefficient: 1

Teaching objectives:

Get hands on experiments on advanced topics in digital systems using VHDL and the DE2 board.

Recommended prior knowledge:

Digital Systems Design using VHDL I and II
Computer Architecture.

Material content:

- **Design and implementation of Structural model application**
- **Implementation of FSMs**
- **Implementation of Datapath controllers**
- **Implementation of an LSI controller (such as UART, PIO, PPI, …)**
- **Implementation of a digital systems using Megafunctions**

Method of evaluation:

Continuous Assessment : 100%

Bibliographical references:

The Altera DE2 Development Board.

Semester:1
Teaching unit: UEM1.1.1
Subject 2: Computer-Controlled Systems Lab
VHS: 45h00 (TP: 3h00)
Credits: 2
Coefficient: 1

Teaching objectives:

This laboratory experiment addresses the practical aspects of the implementation of common digital controllers (PID, state and observer-feedback) and I/O Interfaces (ADC, DAC, Delays, Sensors, and Actuator Interfacing) as well as Software Prototyping used.

Recommended prior knowledge:

Programming I, Computer Architecture, Microprocessor Systems design
Linear control Systems, Process Control and Instrumentation.

Material content:

Lab 1: Software Prototyping (Simulink, LabVIEW)

- Introduction to Simulink and LabVIEW for controller prototyping.
- Building simple control systems using simulation tools.

Lab 2: PID Controller Implementations (Derivative Noise, Integral Windup)

- Implementing and tuning PID controllers.
- Addressing challenges such as derivative noise and integral windup.

Lab 3: I/O Interfaces (ADC, DAC, Delays, Sensors, and Actuator Interfacing)

- Interfacing with sensors and actuators.
- Understanding Analog-to-Digital (ADC) and Digital-to-Analog (DAC) conversion.

Lab 4: State-Feedback and Observer Feedback Implementations

- Implementing state-feedback and observer feedback controllers.
- Addressing practical challenges in state-space control.

Final Project: Digital Controller Application

Method of evaluation:

Continuous Assessment : 100%

Bibliographical references:

1. J. Ledin, "*Embedded Control Systems in C/C++: An Introduction for Software Developers Using MATLAB*", CMP books, 2003.
2. T. Wescott, "*Applied Control Theory for Embedded Systems*", Newnes, 2006.

Semester:1
Teaching unit: UEM 1.1.2
Subject 1: Advanced Programming Lab
VHS: 37h30 (TP: 2h30)
Credits: 2
Coefficient: 1

Teaching objectives:

Implementation of the different features (encapsulation, inheritance, polymorphism, multithreading etc.) of the java programming language.

Recommended prior knowledge:

A first course in a programming language like C.

Material content:

- **Writing, Compiling and executing a Java Program.**
- **Working with Classes, Objects and Methods.**
- **More on Classes, Objects and Methods.**
- **Working with Inheritance, Interfaces and Packages.**
- **Working with the Class String.**
- **Working with Threads.**
- **Working with the Interface Collection.**

Method of evaluation:

Continuous Assessment : 100%

Bibliographical references:

1. H. Shildt. "*Java, a beginner's guide*". McGraw Hill, 2005.
2. H. Shildt. "*Java cookbook*". McGraw Hill 2008.
3. H. Deitel. "*Java for programmers*". Pearson Education 2012.

Semester:1
Teaching unit: UEM 1.1.2
Topic 2: Applied Optimization Lab
VHS: 37h30 (TP: 2h30)
Credits: 2
Coefficient: 1

Teaching objectives:

This is a companion laboratory course for the Numerical methods and Optimization course; the objective is to implement and codify the various methods and algorithms to solve some engineering problems.

Recommended prior knowledge:

A basic course in programming

Material content:

Lab 1: Implementation of Secant, and Newton-Raphson Methods. **(01 Week)**
Lab 2: Implementation of Gauss Elimination, Gauss-Siedel Methods. **(01 Week)**
Lab 3: Implementation of Quadrature Rules, Romberg Integration methods. **(01 Week)**
Lab 4: Implementation of Lagrange Polynomials, Spline Interpolation methods. **(01 Week)**
Lab 5: Implementation of Runge-Kutta's Methods. **(01 Week)**
Lab 6: Implementation of Simplex method to solve LP problem. **(01 Week)**
Lab 7: Implementation of Metaheuristic optimization Algorithms. **(01 Week)**
Lab 8: Parameter tuning of Metaheuristics. **(01 Week)**
Lab 9: Solving application problemsusing Metaheuristics. **(02 Weeks)**
Lab 10: Multi-objective optimization. **(02 Weeks)**

Evaluation method:

Continuous Assessment: 100%

Bibliographical references:

Laboratory handouts available before each Lab session.

Semester: 1
Teaching unit: UED 1.1
Subject 1: Elective Course 1
VHS: 22:30 (lecture: 1h30)
Credits: 1
Coefficient: 1

Semester: 1
Teaching unit: UED 1.1
Subject 2: Elective Course 2
VHS: 22:30 (lecture: 1h30)
Credits: 1
Coefficient: 1

Remark:

Members of "***Computer Engineering*** " curriculum committee are requested to select the courses, "**Elective Course1"** and "**Elective Course2"** from the list proposed in this document. Selection of the courses depends on the need and priority.

2. Detailed program by subject of semester S2

Semester: 2
Teaching unit: UEF 1.2.1
Subject 1: Data Structures and Algorithms
VHS: 67h30 (Lecture: 3h00, TD: 1h30)
Credits: 5
Coefficient: 3

Teaching objectives:

The aim of this course is make the student understand how to organize computer data in such a way that it can be accessed and processed efficiently.

Recommended prior knowledge:

A first course in programming
A first course on calculus
Familiarity with some counting techniques

Material content:

- **Design and Analysis of Algorithms** **(2 Weeks)**
- **The Abstract Data Type List** **(2 Weeks)**
- **The Abstract Data Type Tree** **(2 Weeks)**
- **Directed Graphs** **(3 Weeks)**
- **Undirected Graphs** **(3 Weeks)**
- **Sorting** (simple sorting algorithms, merge sort, heap sort, bin sort). **(3 Weeks)**

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. Aho, Hopcroft, Ullman, "*Data Structures and Algorithms*", Addison-Wesley, 1983.
2. Brassard, Bratley, "*Fundamentals of Algorithmics*", Prentice Hall, 1993.
3. Cormen, Leiserson, Rivest, Stei,. "*Introduction to Algorithms*", MIT Press, 2001.
4. Horowitz, Sahni, Rajasekaran, "*Computer Algorithms*", Computer Science Press, 1998.

Semester: 2
Teaching unit: UEF 1.2.1
Subject 2: Operating Systems
VHS: 45h00 (Lecture: 3h00)
Credits:4
Coefficient:2

Teaching objectives:

The aim of this course is to show how a Software Package called an Operating System manages the many different resources of a Computer System (Processors, Memory, I/O, Secondary Storage *etc.).*

Recommended prior knowledge:

Programming 1&2 courses, Computer Architecture course and Data structures and Algorithm Course are all pre-requisite.

Material content:

- **Introduction to Operating Systems:** **(2 Weeks)**
  Definition of an OS, Fundamental Concepts of an OS, Dual-Mode of Operation, OS Services, System Calls, OS Components, Functions of an OS, Classifications of Operating Systems, Types of OS.

- **Process Management:** **(3 Weeks)**
  Definition of a Process, Process State, Process Control Block (PCB), Process Context Switch, Operation on Processes, UNIX/Linux OS System Calls for Process Management, Process Scheduling, Non-preemptive Scheduling Algorithms: FCFS, SJF, Priority Scheduler, Preemptive Scheduling Algorithms: Round Robin Scheduler, Multilevel Queues Scheduler, Multilevel Feedback Queues Scheduler.

- **Threads:** **(2 Weeks)**
  Multithreading, Threads and Processes, Definition of a Thread, Benefits of Threads, Thread Control Block (TCB), Concurrency vs. Parallelism, Types of Thread: User Level Threads and Kernel Level Threads, Multithreading Models, Thread Libraries: POSIX Threads (Pthreads).

- **On Processes and Threads: Synchronization:** **(3 Weeks)**
  Concurrency Control, Race Condition, The Critical-Section Problem, Solution to Critical-Section Problem, Peterson's and Bakery Algorithms, Disabling Interrupts and The TSL Instruction, Mutex Locks and Busy Waiting Solutions, Semaphores and Monitors, Deadlocks and Starvation, Classic Problems of Synchronization: The Producer/Consumer, The Readers/Writers, and The Dining-Philosophers Problems.

- **Memory Management:** **(2 Weeks)**
  Fundamental Concepts, Address Binding, Logical vs. Physical Address Space, Dynamic Loading, Static and Dynamic Linking, Swapping, Contiguous Allocation, Memory Protection, Process Partitions, Fragmentation, Segmentation, Paging.

- **Virtual Memory:** **(2 Weeks)**
  Virtual-address Space, Demand Paging, Page Fault, Page Replacement, Page and Frame Replacement Algorithms.

- **File Systems:** **(1 Week)**
  Files, Directories, File System Implementation, Examples of File Systems.

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. A. Shilberschatz, P. Galvin, G. Gagné: "*Operating Systems Concepts*". Ninth Edition. Prentice Hall, 2012.
2. W. Stallings: "*Operating Systems Concepts.Internals and Design Principles*". John Wiley & Sons, 2005.
3. A. Tanenbaum, "*Modern Operating Systems*". Pearson, 2009.

Semester: 2
Teaching unit: UEF 1.2.2
Subject 1: Advanced Microprocessor Interfacing
VHS: 45h00 (Lecture: 3h00, TD: 00h)
Credits: 5
Coefficient: 3

Teaching objectives:

The main objective of this course is to study the applications related to the microprocessor interfacing in advanced way (Programmable Peripheral Interface, Address decoding techniques, input/output Controller,...).

Recommended prior knowledge:

Digital Systems Design with VHDL I and II
Computer Architecture
Microprocessor Systems design

Material content:

- **Interfacing the μP to the outside world** **(1 Week)**
- **Different Address decoding techniques** **(2 Weeks)**
- **Intel (Motorola, Zilog) LSI I/O Controllers** **(2 Weeks)**
- **The Programmable Peripheral Interface (PPI) 8522** **(2 Weeks)**
- **The programmable Interval Time/Counter (PIT) 8254** **(2 Weeks)**
- **The Keyboard/Display Controller 8279** **(2 Weeks)**
- **The UART 8251** **(2 Weeks)**
- **Designing μP based systems** **(2 Weeks)**

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

The Z80 microprocessor GAONKAR, 2001

Semester:2
Teaching unit: UEF 1.2.2
Subject2: Machine Learning
VHS: 45h00 (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

As the trending side of artificial intelligence these days, machine learning allows learning from data examples and making decisions and predictions on new data. Machine learning models are being more and more deployed on embedded computing devices, such as IoT enabled gadgets and smart phones. This course aims to introduce the students to the elementary concepts of machine learning and their implementation.

Recommended prior knowledge:

Statistics and Probability, Linear Algebra, Programming.

Material content:

- **Introduction to machine learning: regression and classification (01 Week)**
- **Model evaluation and performance metrics (01 Week)**
- **Clustering and unsupervised learning (01 Week)**
- **K-nearest Nearest neighbour classifier (01 Week)**
- **Naïve Bayes classify (02 Weeks)**
- **Support vector machines classifier (02 Weeks)**
- **Decision Trees. (01 Week)**
- **Neural networks and deep learning. (03 Weeks)**
- **Boosting and ensemble methods (02 Weeks)**
- **Dimensionality reduction (01 Week)**

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. C. Bishop. "*Pattern Recognition and Machine learning*", Springer, 2006.
2. R. Duda, P. Storckand, D. Hart. "*Pattern Classification*", Prentice Hall, 2002.
3. S. Rogers and M Girolami. "*A first Course in machine learning*", CRC press, 2012.

Semester: 2
Teaching unit: UEM 1.2.1
Subject 1: Data structures and algorithms Lab
VHS: 22:30 (TP: 1h50)
Credits: 2
Coefficient: 1

Teaching objectives:

This is a companion course to the Data Structures and Algorithms course. Using a Programming Language such as C++ or Java, the student will learn how to implement the variety of different Data Structures and the Operations on these Data both elegantly and efficiently.

Recommended prior knowledge:

A first course in a programming Language like C.
A first course on calculus
Familiarity with some counting techniques

Material content:

- **Comparing the running times of Algorithms.**
- **Array and Pointer Implementation of General Lists.**
- **Array and Pointer Implementation of Binary Trees.**
- **Implementation of Depth First Search for Digraphs.**
- **Implementation of Depth First Search for graphs.**
- **Implementation of Dijkstra's Algorithm.**
- **Implementation of Floyd Warshall's Algorithm.**
- **Implementation of Merge Sort, Heap Sort and Bin Sort Algorithm.**

Evaluation method:

Continuous Assessment: 100%

Bibliographical references:

1. Aho, Hopcroft and Ullman. "*Data Structures and Algorithms*". Addison-Wesley, 1983
2. Brassard, Bratley. "*Fundamentals of Algorithms"*. Prentice Hall, 1993
3. Cormen, Liserson, Rivest, Stein." *Introduction to Algorithms*". MIT Press, 2001
4. Horowitz, Sahni, Rajasekaran. "*Computer Algorithms*". Computer Science Press, 1998.

Semester: 2
Teaching unit: UEM 1.2.1
Subject 2: Operating Systems Lab
VHS: 22H30 (TP: 1h30)
Credits: 2
Coefficient: 1

Teaching objectives:

This is a companion course to the Operating Systems Course offered in the same semester. The main objective is to show to the students how System Calls are implemented By the Unix OS.

Recommended prior knowledge:

Programming 1&2, Computer Architecture and Data structures and Algorithm Courses are all pre-requisite.

Material content:

- **Getting Started with *LINUX* Operating System**
- **Process Creation and Execution**
- **Scheduling Algorithms**
- **Using Pthread Library For Thread Manipulation**
- **Working With Threads – An Application –**
- **Threads Synchronization (Mutex lock)**
- **Threads Synchronization with POSIX Semaphores, Mutexes, and Condition Variables**
- **Memory Management**
- **File Management**

Evaluation method:

Continuous Assessment: 100%

Bibliographical references:

1. A. Shilberschatz, P. Galvin, G. Gagné: "*Operating Systems Concepts*". Ninth Edition. Prentice Hall, 2012.
2. W. Stallings: "*Operating Systems Concepts.Internals and Design Principles*". John Wiley & Sons, 2005.
3. A. Tanenbaum, "*Modern Operating Systems*". Pearson, 2009.

Semester:2
Teaching unit: UEM 1.2.2
Material 1: Advanced Microprocessor Interfacing Lab
VHS: 30h00 (TP: 02h00)
Credits: 2
Coefficient: 1

Teaching objectives:

This is a companion course to the Advanced Microprocessor Interfacing Course offered in the same semester. The main objective is to perform the applications related to the microprocessor interfacing in advanced way.

Recommended prior knowledge:

Digital Systems Design with VHDL I and II
Computer Architecture
Microprocessor Systems design.

Material content:

- **Implementation of the Z80 interrupt modes 0, 1 and 2**
- **The PPI 8255 and applications**
- **The 8254 and applications**
- **The 8279 and applications (keyboard and 7-seg display control)**
- **The 8251 UART**

Evaluation method:

Continuous Assessment: 100%

Bibliographical references:

1. *The Z80 microprocessor GAONKAR,* 2001
2. *The Altera DE2 Board*

Semester:2
Teaching unit: UEM 1.2.2
Subject2: Machine Learning Lab
VHS: 22h30 (TP: 1h30)
Credits: 2
Coefficient: 1

Teaching objectives:

Students will get hands-on experience in the design and optimization of machine learning models, as well as the analysis of the performances of these models and their deployment on hardware platforms, such as Arduino. The labs will mainly be based on python language and libraries.

Recommended prior knowledge:

Machine learning; programming.

Material content:

- **Prediction based on regression** **(02 Weeks)**
- **Unsupervised classification K-means** **(02 Weeks)**
- **Nearest neighbour classifier** **(02 Weeks)**
- **SVM linear vs kernel** **(02 Weeks)**
- **Classifier optimization and comparison** **(01 Week)**
- **Deep neural networks using Keras** **(03 Weeks)**
- **Deploying Machine learning model on Arduino** **(02 Weeks)**

Evaluation method:

Continuous assessment: 100%.

References:

1. C. Bishop, " *Pattern Recognition and Machine learning*", Springer, 2006.
2. R. Duda, P. Storck and D. Hart, " *Pattern Classification* ", Prentice Hall, 2002.
3. S. Rogers and M Girolami. "*A first Course in machine learning*", CRC press, 2012.
4. Scikit learn and Keras libraries.

Semester:2
Teaching unit: UEM 1.2.2
Subject3: System on Programmable Chip
VHS: 37h30 (Lecture: 1h00, TP: 1h30)
Credits: 2
Coefficient: 1

Teaching objectives:

This laboratory course introduces students to the practical aspects of designing, implementing, and debugging FPGA-based system on a chip (SoPC) designs with Nios II soft processor, RAM, Flash memory, and several peripherals. The students will use the Quartus-II and Qsys design tools to implements SoPC based systems on the DE2 FPGA boards, and program it using the Nios-II Software Build tools.

Recommended prior knowledge:

Advanced Digital Systems

Material content:

- **Building a basic Nios-II based SoPC system on the DE2 FPGA board using Quartus-II and Qsys design tools.**
- **Interrupt Programming and Debugging.**
- **Experiment with standard communication hardware such as RS232, I2C, SPI, USB, Ethernet.**
- **Experiment with an OS on SoPC system (Linux or FreeRTOS).**
- **Creating and integrating custom IP to SoPC system.**

Evaluation method:

Continuous assessment: 100%.

References:

1. Pong P. Chu, "*Embedded SoPC Design with NIOS II Processor and VHDL Examples*", 2012.
2. Hall, and Furman H., "*Rapid Prototyping of Digital Systems, SOPC Edition, by Hamblen*", 2008.
3. Volnei A. Pedroni, "*Circuit Design and Simulation with VHDL*", second edition, 2010.

Semester: 2
Teaching unit: UED 1.2
Subject 1: Elective Course 3
VHS: 22h30 (Lecture: 1:30)
Credits: 1
Coefficient: 1

Remark:

Members of "***Computer Engineering*** " (***Ingénierie Informatique***) curriculum committee are requested to select the course "**Elective Course 3"** from the list proposed in this document. Selection of one course or another depends on the need and priority.

Semester: 2
Teaching unit: UET 1.2
Subject 1: Standards and Rules of Ethics and Integrity
VHS: 22h30 (lecture: 1h30)
Credits: 1
Coefficient: 1

Teaching objectives:

Develop student awareness regarding ethics and rules that govern life at both university and in the professional world. The course presents the risks and consequences of corruption that are raised by new technologies and sustainable development and eventually how to fight them.

Recommended prior knowledge:

Ethics & Integrity (**Bachelor course**)

Content of the material:

A. Compliance with the rules of ethics and integrity: (06 Weeks)

1. Reminder of Ethics and Deontology of the MESRS: Integrity and Honesty. Academic freedom, objectivity and critical thinking. Equity. Rights and obligations of the student, teacher, and other staff,
2. Honest and responsible research
- Respect for the principles of ethics in teaching and research
- Responsibilities in teamwork: Professional equality of treatment. Conduct against discrimination.
- Plagiarism (definition of plagiarism, different forms of plagiarism, procedures to avoid unintentional plagiarism, etc.
3. Ethics and deontology in the professional life: Legal confidentiality in business. Loyalty to the company. Responsibility within the company, Conflicts of interest.

B. Intellectual Property: (04 Weeks)

1. Fundamentals of intellectual property
2. Copyright
3. Protection and enhancement of intellectual property

C. Ethics, sustainable development and new technologies: (05 Weeks)

Link between ethics and sustainable development, energy saving, bioethics and new technologies (artificial intelligence, scientific progress, humanoids, robots, drones

Method of evaluation:

Review: 100%

Bibliographical references:

1. Consult the two links : *www.wipo.int, and http://www.app.asso.fr/*
2. *Orders No. 933 of 28 July 2016 laying down the rules relating to the prevention and fight against plagiarism*
3. D. Carr, "*Professionalism and Ethics in Teaching*". New York, NY Routledge. 2000.

3. Detailed program by subject of the S3 semester

Semester: 3
Teaching unit: UEF 2.1.1
Material 1: Embedded Systems
VHS: 45h00 (Lecture: 3h00, TD: 00h)
Credits: 6
Coefficient: 3

Teaching objectives:

This course introduces students into the software/hardware design aspects of embedded systems. This module builds on the already accumulated knowledge from Advanced Digital Systems Design, Microprocessor Systems Design, and Computer Architecture courses. As an outcome, students should be able to design, implement, and test embedded systems based on microcontrollers using superloop and RTOS approaches.

Recommended prior knowledge:

Advanced Digital Systems

Content of the material:

Part I: Introduction to Embedded Systems:

- General Definitions, example embedded systems, real-time embedded systems.

Part II: Microcontroller Systems Design using MSP432 microcontroller:

- Microcontroller Architecture,
- ARM Assembly/C Programming,
- I/O Interfacing, Interrupts, Timers, ADC
- Basic Embedded software design using superloop approach
- Software implementation of FSMs
- MSP DriverLib Library
- Serial communication: UART, SPI, I2C

Part III: Real-Time Operating System (RTOS):

- Introduction to FreeRTOS: tasks, states, priorities, scheduling algorithms
- FreeRTOS synchronization: critical sections, semaphores, mutex, software timers
- FreeRTOS Inter-task communication and interrupt management

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. Ying Bai, "*Microcontroller Engineering with MSP432: Fundamentals and Applications*", 2016.
2. Dung Dang, Daniel J. Pack, and Steven F. Barrett, "*Embedded Systems Design with the Texas Instruments MSP43232-bit Processor*", 2016.
3. Pong P. Chu, "*Embedded SOPC Design with NIOS II Processor and VHDL Examples*", 2011.

Semester: 3
Teaching unit: UEF 2.1.1
Material 2: Digital Signal Processors
VHS: 45h00 (Lecture: 3h00, TD: 00h)
Credits: 5
Coefficient: 3

Teaching objectives:

Provide the know-how for the implementation and optimization of computationally intensive signal processing algorithms in the TMS320C6416 DSP Processor.

Recommended prior knowledge:

Computer Architecture
LinearSystems II
Microprocessor Systems design

Material content:

Chap. 01: Introduction to Digital Signal Processing
Chap. 02: TMS320C6000 Software Development Tools
Chap. 03: TMS320C6x Digital Signal Processors Architectures
Chap. 04: Discrete Time Signals and Systems
Chap.05: Design and Implementation of FIR Filters using TMS320C64x Processors
Chap. 06: Design and Implementation of IIR Filters using TMS320C64x Processors
Chap. 07: Design and Implementation of FFTAlgorithm using TMS320C64x Processors

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. NaimDahnoun ,"*DSP Implementation Using the TMS320C6711,C6713, and C6416 Processors*".
2. Rulph Chassaing, "*Digital Signal Processing and Applications with the C6713 and C6416 DSK*".

Semester: 3
Teaching unit: UEF 2.1.2
Subject 1: Database Systems
VHS: 45h00 (Lecture: 1h30, TD: 1h30)
Credits:5
Coefficient: 3

Teaching objectives:

At the end of the semester, the student will have a solid background in designing moderate-sized databases. In addition, he or she will be able to create, query and implement realistic databases and connect them to applications.

Recommended prior knowledge:

A first course on algorithms and computer programming.

Material content:

- **Introduction to database systems.**
  - Overview of databases and their significance
  - Types of database systems
  - Evolution of database management systems (DBMS)
  - Role of DBMS in modern applications
- **Relational Database Model.**
  - Introduction to the relational data model
  - Entity-Relationship Diagrams (ERD)
  - Relational algebra and calculus
  - Normalization techniques
- **SQL and Database Design.**
  - Introduction to SQL (Structured Query Language)
  - Data definition language (DDL) and data manipulation language (DML)
  - Database design principles and best practices
- **Transaction Management and Concurrency Control**
  - ACID properties of transactions
  - Transaction management in DBMS
  - Concurrency control techniques
  - Deadlocks and their prevention
- **Database Security and Integrity**
  - Access control and user privileges
  - Encryption and authentication in databases
  - Maintaining data integrity and consistency
- **Advanced Database Concepts**
  - Introduction to NoSQL databases

- Big Data and distributed databases
- Blockchain and database applications
- **UML for Database Design**
    - Introduction to UML and its role in database design
    - Use of UML diagrams in the database development lifecycle
    - Key UML diagrams for database modeling:
        - Class diagrams for entity-relationship modeling
        - Sequence diagrams for transaction flow
        - Activity diagrams for business processes
    - Hands-on exercises using UML for designing a database schema

Method of evaluation:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references :

1. R. Elmasri, S.B. Navathe, "*Fundamentals of Database Systems*", 2011.
2. J. Ullman, J. Widom, "*A first course in Database Systems* ", 2008.
3. J.Gehrke, "*Database Management Systems by R. Ramakrishnan* ", 2003.

Semester: 3
Teaching unit: UEF 2.1.2
Subject 2: Computer networks
VHS: 45h00 (Lecture: 3h00, TD: 00)
Credits: 5
Coefficient: 2

Teaching objectives:

The goal of this course is to bring the student to understand thoroughly the Network Protocol mechanisms, the roles and functions of the Intermediate Equipments, such as routers and switches.

Recommended prior knowledge:

The student should have an insight about numbering systems, basic numbering systems, basic boolean algebra and computer architecture.

Material content:

- Introduction to Networking **(1 Week)**
- Network Protocols and Communications **(2 Weeks)**
- IPv4 Addressing and subnetting **(3 Weeks)**
- Network Access layer and Ethernet **(2 Weeks)**
- Network Layer and Routing protocols **(2 Weeks)**
- Transport Layer **(2 Weeks)**
- IPv6 Networks and IPv6 subnetting **(2 Weeks)**
- Applications **(1 Week)**

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

1. Cisco CCNA V4.1, "*Official Exploration Course*". 2011-2012
2. A. Tanenbaum & al, "*Computer Networks*", 5th Edition. Prentice Hall, 2010

Semester:3
Teaching unit: UEM 2.1.1
Subject1: Embedded Systems Lab
VHS: 45h00 (TP: 3h00)
Credits: 2
Coefficient: 1

Teaching objectives:

Get hands on experiment using the TI MSP432 Launchpad and the Edubase-II board to design/implement embedded systems using the C programming language. The use of FreeRTOS™ real-time operating system is also covered.

Recommended prior knowledge:

Advanced Digital Systems

Material content:

- **Introduction to TI MSP432, Edubase-II Board, and Code Composer Studio CCS**
- **Basic Peripherals Interfacing: switches, LEDs, and 4x4 Keypad**
- **MSP432 Interrupts, SysTick Timer, and FSMs**
- **MSP432 ADC14 and UART**
- **Introduction to FreeRTOS on MSP432**
- **Embedded Systems Mini-project**

Evaluation method :

Continuous assessment: 100%.

References

1. *MSP432P4xx SimpleLink™ MicrocontrollersTechnical Reference Manual*, 2019.
2. *MSP432P401R SimpleLink™ Microcontroller LaunchPad™ Development Kit (MSP-EXP432P401R) User guide*, 2018.
3. *Edubase V2 user guide (http://www.trainer4edu.com/edubase_v2/index.html).*
4. *Mastering the FreeRTOS™Real Time Kernel161204*, 2016
5. *FreeRTOS Reference Manual*, 2016.

Semester:3
Teaching unit: UEM 2.1.1
Subject 2: Digital Signal Processors Lab
VHS: 45h00 (TP: 03h00)
Credits: 2
Coefficient: 1

Teaching objectives:

Provide hands-on experiment of computationally intensive signal processing algorithms on the TMS320C6416 DSP processor.

Recommended prior knowledge:

Computer Architecture
Linear Systems II
Microprocessor Systems design

Material content:

Lab 1: Introduction to DSK6416T Board and Code Composer Studio
Lab 2: Sine Wage Generation
Lab. 3: Code Analysis and Optimization using C, Linear Assembly and Assembly codes
Lab. 4: Design and Implementation of FIR Filters using TMS320C6416 Processors
Lab. 5: Design and Implementation of IIR Filters using TMS320C64x Processors
Lab. 6: Design and Implementation of FFT Algorithm using TMS320C64x Processors
Lab. 7: DSP Applications and Student Projects

Evaluation method:

Continuous Assessment: 100%

References

1. NaimDahnoun ,"*DSP Implementation Using the TMS320C6711,C6713, and C6416 Processors*".
2. Rulph Chassaing, "*Digital Signal Processing and Applications with the C6713 and C6416 DSK*".

Semester:3
Teaching unit: UEM 2.1.2
Material 1: Database Systems Lab
VHS: 37h30 (TP: 02h30)
Credits: 2
Coefficient: 1

Teaching objectives:

At the end of the semester, the student will be able to create, query and implement realistic databases and connect them to applications.

Recommended prior knowledge:

A first course on algorithms and computer programming.

Material content:

Lab 1: Database Design and Entity-Relationship Diagrams
- Practical exercises on creating ERDs for real-world scenarios.
- Introduction to database design tools.

Lab 2: SQL Basics and Query Optimization
- Hands-on exercises in writing SQL queries.
- Query optimization techniques and best practices.

Lab 3: Normalization and Database Optimization
- Normalization exercises on given datasets.
- Database optimization techniques and performance tuning.

Lab 4: Transaction Management and Concurrency Control
- Implementing transactions and analyzing their behavior.
- Handling, concurrency, issues and resolving conflicts.

Lab 5: Database Security and Access Control
- Configuring user privileges and access control.
- Implementing encryption and authentication mechanisms.

Lab 6: UML for Database Design
- Hands-on exercises using UML for database modeling.
- Translating UML diagrams into a relational database schema.

Lab 7: NoSQL Databases and Cloud Database Services
- Practical exploration of NoSQL databases (e.g., MongoDB, Cassandra).
- Setting up databases on cloud platforms (e.g., AWS, Azure).

Final Project: Real-World Database Application
- Students design and implement a database for a real-world scenario.
- Emphasis on application scalability, security, and performance.

Evaluation method:

Continuous Assessment: 100%

References

1. R. Elmasri, S.B. Navathe, "*Fundamentals of Database Systems*", 2011.
2. J. Ullman, J. Widom, "*A first course in Database Systems*",2008.
3. R. Ramakrishnan, J.Gehrke, "*Database Management Systems*", 2003.

Semester:3
Teaching unit: UEM 2.1.2
Subject 2: Computer Networks Lab
VHS: 22h30 (TP: 1h30)
Credits: 2
Coefficient: 1

Teaching objectives:

The Goal of this series of labs is to make sure the student will grasp the concepts of LANs, sub-networking and VLANs

Recommended prior knowledge:

No prerequisites

Material content:

The labs will be implemented using one of the two simulators: Packet Tracer of Cisco Academy or GNS3.

- Introduction to Packet Tracer and GNS3.
- Router Configurations used in LANs.
- Switch configuration used in LANs (VLAN configuration)
- LANs interconnections (MANs or/and WANs).

Evaluation method:

Continuous assessment: 100%

References

1. *Cisco Packet Tracer - Networking Simulation Tool*, (*https://www.netacad.com/courses/packet-tracer*).
2. A. Tanenbaum et al, " *Computer Networks,"* 5th Edition. Prentice Hall 2010.

Semester: 3
Teaching unit: UET 2.1
Subject 1: Project Management
VHS: 22h30 (Lecture: 1h30)
Credits: 1
Coefficient: 1

Teaching objectives:

The overarching objective of teaching project management is to impart students with a comprehensive skill set. This includes establishing a solid foundation in key processes and project management requirements, gaining nuanced expertise in project phases, stakeholder management, and understanding project dynamics. The course emphasizes practical application through tasks such as developing Project Management Plans, creating accurate Work Breakdown Structures, identifying critical paths, compute a project's earliest possible finish date by means of the two-pass method, time-cost analysis, and risk assessment. Ultimately, students will be equipped to proficiently navigate project initiation, planning, execution, monitoring and control, and closure, ensuring effective resource management and adaptability in dynamic project environments.

Recommended prior knowledge:

Basic Project Management Concepts
Fundamental Business Understanding
Leadership and Team Management Skills

Content of the material:

Chapter 1: Introduction to Project Management	**(02 Weeks)**
Chapter 2: Initiating a project-chartering	**(02 Weeks)**
Chapter 3: Project Planning and scope management	**(03 Weeks)**
Chapter 3: Project Schedule Management	**(03 Weeks)**
Chapter 4: Time and Cost Management	**(02 Weeks)**
Chapter 5: Project Execution/monitoring	**(01 Week)**
Chapter 6: Project Closure, and Case Studies	**(02 Weeks)**

Method of evaluation:

Review: 100%

Bibliographical references:

1. P. Lewis, "*Fundamentals of Project Management*", James, ISBN: 9780814408797
2. Harold, Kerzner, "*Project Management: A Systems Approach to Planning, Scheduling, and Controlling*" , (ISBN-10: 0471741876/ISBN-13: 978-0471741879).

Semester: 3
Teaching unit : UET 2.1
Subject 2: Communication skills
VHS: 22h30 (lecture: 1h30)
Credits: 1
Coefficient: 1

Teaching objectives:

This is an ESP course intended to equip the students with the necessary skills for an effective communication in both its aspects: oral and written. As the target is soon-to-be graduates, this course focuses on the following. Firstly, it addresses the writing of final year project report with all that it requires as advanced composition techniques, format of the report, the literature review along with the appropriate referencing skills, and of course basic, research methodology concepts. In addition, this course aims to assist the students with their oral skills through mock project presentations and the feedback they receive on the multiple aspects of the presentation. Finally, the soon-to-be graduates are introduced to business correspondence.

Recommended prior knowledge:

The students must have undertaken basic training in the four language areas: listening, speaking, reading and writing.

Content of the material:

1. **Transition from sentence production to the development of continuous prose.**
2. **Synthesizing, summarizing and patch-writing.**
3. **Academic technical writing: specifications, patterns, data analysis language and characteristics.**
4. **Serious writing pitfalls: comma splice, sentence fragments, etc.**
5. **Preliminary concepts on report writing: research problem, hypothesis, objectives, literature review, editing . . .**
6. **IMRaD Format: prior and post IMRaD sections, format variations, etc..**
7. **Reference Style: plagiarism, IEEE reference guide & others with their appropriate fields.**
8. **Oral communication techniques.**
9. **Oral presentations of students' graduation projects.**
10. **Business correspondence.**

Method of evaluation:

Review: 100%

Bibliographical references:

1. Craig Harkins and Daniel L. PlungEds. "*A Guide for Writing Better Technical Papers*". New York: IEEE Press, 1982.
2. James W. Souther and Myron L. White, "*Technical Report Writing*". 2nd edition, Washington: USA, 1977.
3. "*IEEE Reference Guide*". IEEE periodicals, USA, 2018.
4. Marta Grand, "*A Short Guide to the Oral Presentation in English*", *ENSIEG.*
5. John M. Swales & Christine B. Feak, "*Academic Writing for Graduate Students: Essential Tasks and Skills.*" 3rd edition, Michigan, USA, 2012.

4. Proposal of some subjects of discovery
(*3 courses must be selected from this list*)

Semester:
Teaching unit: UED
Material1: Printed Circuit Board Design and Technologies
VHS: 15h00 (Lecture: 1:00 am)
Credits: 1
Coefficient: 1

Teaching objectives:

Printed Circuit Boards (PCB) are widely used by electronics and electrical engineers in different applications. Generally the first step in a given electronic card production is making the PCB. This course provides the skills necessary to effectively design and implement a PCB. The course covers all the necessary aspect to design a PCB from an idea to a product by describing the steps involved in PCB Design and Fabrication process.

Recommended prior knowledge:

Experimental Labs of fundamental courses (Electricity, Active devices, Digital).
Some chemistry and physics principles.

Material content:

1. **Introduction to PCB: (02 Weeks)**
   Definition and Relevance of PCB, Background and History of PCB, Classes of PCB Design.
   PCB Design Process: (02 Weeks)
   PCB design rules and constraints, Steps involved in layout design, Designing of Multi-layer Boards, Design for manufacturability.
2. **PCB Manufacturing & Assembly: (04 Weeks)**
   Steps involved in fabrication of PCB, PCB Fabrication techniques-single, double sided and multilayer, Etching: chemical principles and mechanisms, PCB component assembly processes, Soldering, Inspection and electrical testing.
3. **PCB design Software tools: (03 Weeks)**
   Give in general the best PCB design software programs: KiCad, EasyEDA, Eagle..
4. **PCB design: Practical examples +Projects for the students (04 Weeks)**

Evaluation method:

Quizes: 50%; Project: 50%.

Bibliographical references:

1. D. L. Jones, "*PCB Design Tutorial*," June 29th 2004, *https://www.scs.stanford.edu/~zyedidia/docs/pcb/pcb_tutorial.pdf.*
2. M.I. Montrose, *"Printed circuit board design techniques for EMC compliance(A handbook for designers)"*, IEEE Press series on electronics technology, 2000.

Semester:
Teaching unit: UED
Subject 2: GPU Programming
VHS: 15h00 (Lecture: 1:00 am)
Credits: 1
Coefficient: 1

Teaching objectives:

This course provides a concise exploration of GPU (Graphics Processing Unit) computing principles and applications. Students will gain an understanding of GPU architecture, parallel programming models, and real-world applications in a variety of domains.

Course Objectives:

1. Understand the basics of GPU architecture.
2. Learn key concepts of parallel programming for GPUs.
3. Gain insights into GPU programming languages (e.g., CUDA, OpenCL).
4. Explore applications of GPU computing in various domains.

Recommended prior knowledge:

Programming I, Programming II, Mircoprocessor System Design.

Material content :

1. **Introduction to GPU Computing** **(02 Weeks)**
   - Overview of GPU architecture and its evolution
   - Parallelism and the role of GPUs in computation
   - Introduction to GPU programming languages (CUDA, OpenCL)

2. **GPU Architecture and Programming Models** **(03 Weeks)**
   - Understanding the key components of GPU architecture
   - Introduction to parallel programming models (SIMT, SPMD)
   - GPU, kernels and threads

3. **GPU Programming Languages** **(03 Weeks)**
   - Basics of CUDA programming (NVIDIA GPUs)
   - Overview of OpenCL programming (vendor-neutral)
   - Practical examples in GPU programming languages

4. **GPU Memory Management and Optimization** **(03 Weeks)**
   - Understanding GPU memory types and access patterns
   - Strategies for optimizing memory access in GPU programming
   - Profiling and basic optimization techniques

5. **GPU Applications in Various Domains** **(04 Weeks)**
   - GPU-accelerated applications in scientific computing
   - GPU applications in data science and machine learning

- Real-world case studies and examples

Evaluation method:

Review: 100%.

References:

1. Jason Sanders and Edward Kandrot, "*CUDA by Example: An Introduction to General-Purpose GPU Programming*".
2. David B. Kirk and Wen-mei W. Hwu,"Programming *Massively Parallel Processors: A Hands-on Approach*".
3. Aaftab Munshi, Benedict R. Gaster, and Timothy G. Mattson, "*OpenCL Programming Guide*".

Semester:
Teaching unit: UED
Subject 3: Academic Research and Tools
VHS: 15h00 (Lecture: 1:00 am)
Credits: 1
Coefficient: 1

Teaching objectives:

Research serves as the driving force behind transformative technological advancements, solving complex engineering problems, and the continual expansion of knowledge across various engineering disciplines. It acts as the cornerstone for innovation, developing new technologies, enhancing existing systems, and addressing critical societal needs. In today's interconnected world, where technological innovation is accelerating, having knowledge about the research field and familiarity with the different tools and platforms available are crucial for success. The objective of this course is to introduce students to the field of academic research, providing them with foundational knowledge and insights, essential methodologies, some practices, and requisite tools to conduct impactful research in the field of engineering.

Recommended prior knowledge:

Basic academic skills such as reading, comprehension, writing, critical thinking, and problem-solving. Some familiarity with handling softwares and web platforms.

Material content :

1. **Overview of Academic Research** **(01 Week)**

- Overview of academic research and its significance
- The research process: from problem identification to dissemination
- Research approaches.

2. **Literature Review and Information Retrieval** **(02 Weeks)**

- Purpose of literature reviews in research.
- Strategies for conducting literature searches.
- Tools and databases for information retrieval (Google Scholar, PubMed, Scopus, Web of Science, IEEE Xplore Digital Library... etc).

3. **Academic Writing** **(04 Weeks)**

- Principles of academic writing: definitions, characteristics, types, structure of disseratations and research papers.
- Academic writing tools: Grammar and writing style checkers. Word processing softwares. Reference management softwares. Collaborative writing platforms. Plagiarism detection tools.

4. Citation Management **(02 Weeks)**

- Concept of citation and referencing.
- Overview of citation styles (APA, MLA, Chicago, IEEE).
- Citation management tools (Zotero, Mendeley, EndNote, BibTex... etc).

5. Collaborative Research, Communication, and Publishing **(04 Weeks)**

- Collaborative research: Tools, platforms, and best practices.
- Communication of research findings: Strategies and tips.
- Conference presentations: Types of conference presentations, virtual conference platforms.
- Peer review process: Understanding the concept of peer reviewing, types of peer review.
- Publishing in academic journals: Publishing types, types of academic journals, understanding journal impact factors and other metrics, platformsand tools for selecting journals.

6. Research Ethics and Integrity **(02 Weeks)**

- Ethical considerations in research.
- Ensuring integrity in research practices.
- Addressing conflicts of interest and ethical challenges.

Evaluation method:

Quizes: 50%; Project: 50%.

References:

1. Taherdoost, H, "*Handbook on Research Skills: The Essential Step-By-Step; Guide on How to Do a Research Project*" (Kindle ed.): Amazon. (2021).
2. P. Pandey, M. M. Pandey, " *Research methodology: Tools and techniques*", Bridge Center, 2015.
3. Andy Stapleton, "*The best AI tools for research papers and academic research*.", *https://academiainsider.com/ai-tools-for-research-papers-and-academic-research/*,(Updated on: June, 2023).
4. N. Ebrahim, " *Introduction to the "Research Tools" for Research Methodology course*", 2016.

Semester:
Teaching unit: UED
Subject 4: Computer Vision
VHS: 15h00 (Lecture: 1:00 am)
Credits: 1
Coefficient: 1

Teaching objectives:

This course provides an introduction to the fundamentals of computer vision. Students will explore key concepts, techniques, and applications in computer vision, with an emphasis on real-world scenarios.

Course Objectives:

1. Understand the core principles and challenges of computer vision.
2. Learn basic image, processing techniques and feature extraction.
3. Explore key algorithms in computer vision, including object detection and recognition.
4. Gain insights into practical applications of computer vision in various fields.

Recommended prior knowledge:

Programming I, Programming II.

Material content :

1. **Introduction to Computer Vision** **(02 Weeks)**
   - Definition and scope of computer vision
   - Evolution and applications of computer vision
   - Key challenges and limitations

2. **Image Formation and Processing** **(03 Weeks)**
   - Basics of digital image formation
   - Image representation and pixel manipulation
   - Common image processing operations (blurring, sharpening, etc.)

3. **Feature Extraction and Image Segmentation** **(03 Weeks)**
   - Understanding image features and descriptors
   - Techniques for feature extraction
   - Image segmentation methods

4. **Object Detection and Recognition** **(03 Weeks)**
   - Overview of object detection algorithms (e.g., Haar cascades)
   - Introduction to image classification and recognition
   - Case studies in object detection and recognition

5. **Practical Applications of Computer Vision** **(04 Weeks)**
   - Computer vision in healthcare, automotive, and robotics
   - Augmented reality and virtual reality applications
   - Ethical considerations in computer vision

Evaluation method:
Review: 100%.

References:

- Richard Szeliski , "*Computer Vision: Algorithms and Applications*".
- E. R. Davies , "*Computer Vision: Principles, Algorithms, Applications, Learning*".
- David Forsyth and Jean Ponce , "*Computer Vision: A Modern Approach*".

Semester:
Teaching unit: UED
Subject 5: Internet of Things
VHS: 15h00 (Lecture: 1:00 am)
Credits: 1
Coefficient: 1

Teaching objectives:

This module aims to introduce the important notions of the Internet of Things and its applications and architecture models. The lecture can be considered as an Introduction to the technologies and mechanisms for sensing, actuation, processing and cyber-physical data communication.

Recommended prior knowledge:

Programming I, Programming II, Mircoprocessor System Design.

Material content :

- **Introduction to Internet of Things (IoT);** **(01 Week)**
- **Sensors, actuators and controllers;** **(02 Weeks)**
- **Wireless Technologies for IoT;** **(03 Weeks)**
- **IoT protocols;** **(03 Weeks)**
- **IoT data acquisition and storage;** **(03 Weeks)**
- **IoT applications.** **(03 Weeks)**

Evaluation method:

Review: 100%.

References:

1. Michael R. Miller , "*Internet of Things, The: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World*", book, Kindle Edition, Que Publishing; 1st edition (March 13, 2015).
2. Gary Smart, "*Practical Python Programming for IoT: BuildadvancedIoTprojectsusing a Raspberry Pi 4, MQTT, RESTful APIs, WebSockets, and Python 3*", Book, 978-1838982461, PacktPublishing (November 12, 2020).
3. Daniel Minoli, "*Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications*", ISBN: 978-1-118-47347-4, Willy Publications.

Semester:
Teaching unit: UED
Subject 6: Introduction to Renewable Energies
VHS: 15h00 (Lecture: 1:00 am)
Credits: 1
Coefficient: 1

Teaching objectives:

This course aims to provide a comprehensive understanding of renewable energy systems and their integration into power systems. Students will explore various renewable energy sources, including solar, wind, hydropower, and ocean energy, with a focus on the technological, control, and power engineering aspects.

Recommended prior knowledge:

A course in Electric Circuits
A course in Power Electronics

Material content:

1. **Introduction to Renewable Energies**(Fundamentals and key characteristics of wind, solar PV and hydro energy sources.)
2. **Solar Energy Systems**(Fundamentals of solar cells, Types of solar devices, Modelling Power electronics for PV systems, Maximum Power Point Tracking...)
3. **Wind Energy Systems**(Principle of operation, turbine types, power train, cost of energy, turbine modelling and control, wind turbine reliability...)
4. **Integration of Renewable Energy in Power Systems** (Renewable energy models, Power flow analysis, Dynamic simulations,Energy Storage Systems, Optimal Power Flow (OPF)...)
5. **Sizing of PV Systems (**Components selection, battery design, Controller design, Invereter design, ...)
6. **Hydropower Energy (**Types of hydropower plants, Control, Environmental considerations...)

Evaluation method:

Continuous assessment: 100%.

Bibliographical references:

1. Leon Freris, David Infield, "*Renewable energy in power systems*", John Wiley & Sons, 2008.

Semester:
Teaching unit: UED
Subject 7: Introduction to Robotics
VHS: 15h00 (Lecture: 1:00 am)
Credits: 1
Coefficient: 1

Teaching objectives:

The purpose of this course is to familiarize students with the field of robotics (Explain what robots are and what they can do, development of robotics in human society, explain what does sensors and actuators do and how they can be used according to the specifications of the problem and nature of the environments, design a simple robot with specified criteria).

Recommended prior knowledge:

A course in Linear Control Systems
Microprocessor Systems design.
A course in Differential equations

Content of the material:

- **Introduction**
- **Robot Classification and Sensor Specifications**
- **Sensor Attributes and Types**
- **Actuators, Hardware and Software**
- **Vector and Kinematics in Robotics**
- **Robot programming and Controlling**

Method of evaluation:

Continuous assessment : Examination : 100%.

Bibliographical references :

1. Saeed B. Niku, "*Introduction to Robotics: Analysis, Control, Applications*," 2nd Edition.
2. Nikolaus Correll, "*Introduction to Autonomous Robots: Kinematics, Perception, Localization and Planning*",1st Edition.