People's Democratic Republic of AlgeriaRépublique Algérienne Démocratique et Populaire
Ministry of Higher Education and Scientific Research
Ministry of Higher Education and Scientific Research
National Pedagogical Committee for Science and Technology
National Pedagogical Committee of the Science and Technology field

ACADEMIC MASTER's DEGREE HARMONISE

Update 2025

Domain	Stream	Specialty:
Sciences and Technologies	***Electronics***	***Embedded Systems Electronics***

People's Democratic Republic of AlgeriaRépublique Algérienne Démocratique et Populaire
Ministry of Higher Education and Scientific Research
Ministry of Higher Education and Scientific Research
National Pedagogical Committee for Science and Technology
National Pedagogical Committee of the Science and Technology field

Master Academy Harmonization

National Program

2025 Update

Specialization	Branch	The square
Electronics Embedded Systems	Electronics	Science & Technology

I – Master's identity sheet

Conditions of access

(Indicate the bachelor's specialties that can give access to the Master's degree)

Stream	Harmonized Master's Degree	Access Licenses Master's degree	Classification according to license compatibility	Coefficient assigned to the licence
Electronics	Embedded Systems Electronics	Electronics	**1**	**1.00**
		Telecommunications	**2**	**0.80**
		Biomedical Engineering	**2**	**0.80**
		Automatic	**3**	**0.70**
		Electrical engineering	**3**	**0.70**
		Electromechanics	**4**	**0.65**
		Other ST Domain Licenses	**5**	**0.60**

II – Semester organisation sheets for teaching of the specialty

Semester 1

Teaching unit	Materials	Credits	Coefficient	Weekly hourly volume			Semi-Annual Hourly Volume (15 weeks)	Complementary work in Consultation (15 weeks)	Evaluation method	
	Title			**Courses**	**TD**	**TP**			**Continuous assessment**	**Review**
Core UE Code: UEF 1.1.1 Credits: 10 Coefficients : 5	Microcontroller systems	6	3	3:00 pm	1h30		67h30	82:30 p.m.	40%	60%
	Advanced digital electronics: FPGA and VHDL	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
Core UE Code: UEF 1.1.2 Credits: 8 Coefficients : 4	Advanced signal processing	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
	Digital Servo Systems	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
Methodological UE Code: EMU 1.1 Credits: 11 Coefficients : 7	TP Microcontroller Systems	2	1			1h30	10:30 p.m.	27:30	100%	
	FPGA and VHDL TP	2	1			1h30	10:30 p.m.	27:30	100%	
	Advanced Signal Processing /Digital Servo Systems TP	2	1			1h30	10:30 p.m.	27:30	100%	
	Embedded C++ programming	3	2	1h30		1h30	45h00	30:00 pm	40%	60%
	Advanced programming in Python	2	2	1h30		1h30	45h00	5:00 pm	40%	60%
Discovery UE Code: UED 1.1 Credits: 1 Coefficients : 1	Choice of material	1	1	1h30			10:30 p.m.	02:30 am		100%
Total semester 1		**30**	**17**	**12:00 p.m.**	**6:00 pm**	**7:30 a.m.**	**382h30**	**367h30**		

Semester 2

Teaching unit	Materials	Credits	Coefficient	Weekly hourly volume			Semi-Annual Hourly Volume (15 weeks)	Complementary work in Consultation (15 weeks)	Evaluation method	
	Title			**Courses**	**TD**	**TP**			**Continuous assessment**	**Review**
Core UE Code: UEF 1.2.1 Credits: 10 Coefficients : 5	Embedded Processor Architecture	6	3	3:00 pm	1h30		67h30	82:30 p.m.	40%	60%
	Digital Signal Processors (DSPs)	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
Core UE Code: UEF 1.2.2 Credits: 8 Coefficients : 4	Computer Vision System	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
	Industrial programmable logic controllers.	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
Methodological UE Code: EMU 1.2 Credits: 9 Coefficients : 5	Embedded Processor Architecture Lab	2	1			1h30	10:30 p.m.	27:30	100%	
	TP Digital Signal Processors	2	1			1h30	10:30 p.m.	27:30	100%	
	Computer Vision System / TP Programmable Logic Controllers	2	1			1h30	10:30 p.m.	27:30	100%	
	Java Programming for Embedded Systems	3	2	1h30		1h30	45h00	30:00 pm	40%	60%
Transversal UE Code: UET 1.2 Credits: 3 Coefficients : 3	Elements of Applied AI	2	2	1h30	1h30		45h00	5:00 pm	40%	60%
	Compliance with the standards and rules of ethics and integrity	1	1	1h30			10:30 p.m.	02:30 am		100%
Total semester 2		**30**	**17**	**12:00 p.m.**	**6:00 pm**	**7:30 a.m.**	**382h30**	**367h30**		

Semester 3

Teaching unit	Materials	Credits	Coefficient	Weekly hourly volume			Semi-Annual Hourly Volume (15 weeks)	Complementary work in Consultation (15 weeks)	Evaluation method	
	Title			**Courses**	**TD**	**TP**			**Continuous assessment**	**Review**
Core UE Code: UEF 2.1.1 Credits: 10 Coefficients : 5	Real-Time Systems	6	3	3:00 pm	1h30		67h30	82:00 p.m.	40%	60%
	Embedded Artificial Intelligence	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
Core UE Code: UEF 2.1.2 Credits: 8 Coefficients : 4	Embedded systems	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
	Industrial Networks and Communications	4	2	1h30	1h30		45h00	55:00 pm	40%	60%
Methodological UE Code: EMU 2.1 Credits: 9 Coefficients : 5	Artificial Intelligence TP	2	1			1h30	10:30 p.m.	27:30	100%	
	Embedded Systems TP/ Real-Time Systems TP	2	1			1h30	10:30 p.m.	27:30	100%	
	TP Industrial Networks	2	1			1h30	10:30 p.m.	27:30	100%	
	Study and implementation of projects	3	2	1h30		1h30	45h00	30:00 pm	40%	60%
Transversal UE Code: UET 2.1 Credits: 3 Coefficients : 3	Reserve engineering	2	2	1h30	1h30 Workshop		45h00	5:00 pm	40%	60%
	Literature search and memory design	1	1	1h30			10:30 p.m.	02:30 am		100%
Total semester 3		**30**	**17**	**12:00 p.m.**	**6:00 pm**	**7:30 a.m.**	**382h30**	**367h30**		

General guidelines on the choice of discovery subjects:

The subjects discovered in the Master of Instrumentation Subject Repository (Table above) are left to the free choice of the institutions, which can choose their subjects from the list presented below according to their priorities.

Subjects with detailed programmes:

- Optoelectronics
- Autonomous energy systems
- Electroacoustics and vibration analyses
- Electromagnetic compatibility
- Industrial Instrumentation & Measurement
- Industrial safety
- Robotics
- Adjustment of Electric Drives
- Bio instrumentation and biosensors
- Methods and tools for non-destructive testing
- Tools for Instrumentation Maintenance
- Industrial maintenance and diagnostics
- Industrial Networks & Communication

Other subjects left to the free choice of the institutions (programs open after validation of the CPND)

- Display Systems(Discovery)
- Measuring Instruments(Discovery)
- High-Frequency Measurements(Discovery)
- Electroacoustics, Sound and HIFI(Discovery)
- Industrial Remote Management (SCADA) (Discovery)
- Industrial Systems Control Theory (Discovery)
- Smart Sensors in Industrial Instrumentation(Discovery)
- Operations Research

Semester 4

Internship in a company or in a research laboratory sanctioned by a dissertation and a defense.

	VHS	Coeff	Credits
Personal work	550	09	18
Internship in a company or laboratory	100	04	06
Seminars	50	02	03
Other (Framing)	50	02	03
Total Semester 4	750	17	30

This table is given for information purposes only

Evaluation of the Master's End of Cycle Project

- Scientific value (Jury's assessment) /6

- Writing of the Thesis (Jury's Assessment)	/4
- Presentation and answer to questions (Jury's assessment)	/4
- Supervisor's assessment	/3
- Presentation of the internship report (Jury's assessment)	/3

III - Detailed programme by subject of the S1 semester

Semester : 1
Teaching unit: UEF 1.1.1
Subject 1: Microcontroller systems
VHS: 67h30 (Lecture: 3h00, Tutorial: 1h30)
Credits: 6
Coefficient : 3

Objectives to be achieved:

This subject allows students to become familiar with the various features offered by microcontrollers in the real world. Indeed, the world is moving towards the aspect of artificial intelligence which requires designs involving microcontrollers as key parts to achieve industrial-scale applications recommended in particular in the embedded system. Thus, students will learn how to manipulate the different input/output devices, in this case: on/off inputs, sensor modules, supervision screens, motors and actuators, as well as the establishment of an appropriate communication layer, both simple as those using standardized protocols, UART, I^2C, SPI, Bluetooth, and complex protocols, namely: Modbus TCP/IP, MQTT, ESP-NOW, Zigbee. The basic learning is done with the famous "ATmega328p" microcontroller given the range of applications that can be offered to the user and which goes hand in hand with the Arduino platforms. On the other hand, it has better performance than that offered by older microcontrollers such as the "PIC-16Fxx" family, in addition, it lends itself well to accepting different types of compilers, including the C-Embedded in the user-friendly environment called "IDE". That being said, it allows students to evolve more easily in a fun way towards the new generations of microcontrollers such as the ESP32 or the ARM32, thus requiring a very substantial background in advanced programming.
In addition, the Proteus® software allows for interactive in-circuit simulation of these "AVR" microcontrollers, and explicit testing of the code and circuit diagram before thinking about building the target hardware. When the application is working properly, as it should, in simulation mode, the PCB can be designed to validate the design in question.
The application of the content of this subject will be carried out within the unit:
"UEM1.1: TP Microcontroller Systems".

Recommended Prerequisites:

L3 Microprocessor Systems, Digital Electronics, Finite State Machine, Basic Electronics, Assembly Language, C Language.

Content of the material:

Chapter 1 Basic Microcontroller Initiation (5 weeks)

- General architecture of the AVR32/PIC32 microcontroller.
- Setting a bit, clearing a bit, checking a bit, Toggling a bit and Macros.
- Architecture de la mémoire AVR (Flash, Boot section; Data memory, RAM, General Registers; EEPROM).
- C-Embedded in the IDE environment

- The general framework of an Arduino program (Header, declarative section, configuration and initialization, Main program and definition of subroutines)
- Instructions de bases (Typedef, enum, If-Else, Switch-Case, Struct, Union)
- Applications (Drop-down menu with three buttons Up, Down, OK).
- Digital output management (LED flashing using delay() and millis())
- Data Display (TM1637 4-digit 7-segment, 2x16 LCD)

Chapter 2. Advanced Microcontroller Features (5 weeks)

- Interruptions caused by external events (Rising, falling, change).
- Analog-to-digital conversion by programmed interruption.
- Interruptions des timers : Timer0, Timer1 et Timer2.
- Interrupt PWM generator and timer management (CTC mode and PWM generator).
- Examples: Incremental encoder dedicated to managing a drop-down menu
- AC power dimmer with a solid state relay (SSR)
- Example: Controlling a servo motor with OC1A/OC1B mode for different angular resolutions.
- Example: Variable speed drive of a DC motor with the L298N module.
- Use of internal EEPROM (Read, Write, Erase).
- Half- and full-duplex interrupt mode communication protocols: UART (data manipulation such as string, byte, int, double)
- I^{2}C Bus Implementation and Use (DH22, DS18b20, ds3231 Temperature Sensor)
- The SPI protocol
- Communication en Full-duplex
- Single master – multiple slave architecture
- High-speed communication [Hz, kHz or MHz]
- Master-slave configurations.
- Flexible protocol.
- Short-distance communication.
- Initializing an SD card and storing data.

Chapter 3. High-level structured programming (5 weeks)

- Multi-tasking systems (function pointers, implementation of a finite state machine "FSM", introductory example of the management of a two-way traffic light)
- Communication with Bluetooth (HC-05) using "MIT APP-Inventor" on Android and the Nano platform with the Atmega328P microcontroller).
- The CAN bus is its applications (design of a network, e.g. dashboard of a vehicle).
- The Modbus TCP/IP fieldbus is the example of the S7-1200 station with UNO equipped with the Ethernet Shield module.

Method of evaluation:

Exam: **60%,** Continuous assessment: **40%**

Bibliographical references:

[1] . *Arduino Workshop: A Hands-On Introduction with 65 Projects; De John Boxall; No Starch Press 2013.*

[2] . *C Programming for the Pc the Mac and the Arduino Microcontroller System; De Peter D Minns; AuthorHouse-2013*

[3] .*Raspberry Pi for Arduino Users: Building IoT and Network Applications and Devices; De James R. Strickland; Apress-2018*

[4] .*Practical AVR Microcontrollers: Games, Gadgets, and Home Automation with the microcontroller used in the Arduino; De Alan Trevennor; Apress-2012*

[5] .*Advances in Smart System Technologies: Select Proceedings of ICFSST 2019;*
Published by P. Suresh, U. Saravanakumar, Mohammed Saleh Hussein Al Salameh; Springer Nature.

[6] .*INTRENET OF THINGS WITH ARDUINO AND BOLD IOT: With Arduino and Bolt;*
By Ashwin Pajankar; Published by BPB Publications 2018.

[7] .*Arduino: A Technical Reference: A Handbook for Technicians, Engineers, and Makers; De J. M. Hughes; "O'Reilly Media, Inc." 2016.*

[8] .*Building Arduino PLCs: The essential techniques you need to develop Arduino-based PLCs; De Pradeeka Seneviratne; Apress-2017.*

[9] .*Arduino Robot Bonanza; De Gordon McComb; McGraw Hill Professional-2013.*

Semester: 1
Teaching unit: UEF 1.1.1
Subject 2: Advanced Digital Electronics: FPGA and VHDL
VHS: 45h00 (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

In this subject, students will have to study the different types of programmable circuits, as well as the different design methods in particular programming using hardware description languages.
The content of this subject is applied at the level of the subject "**UEM1.1: TP FPGA & VHDL**".

Recommended prior knowledge:

Digital electronics (combinatorial and sequential).

Material content:

Chapter 1. Programmable Circuit Basics **(1 week)**

- General Architecture of Programmable Logic Circuits: PAL, GAL, PLD, CPLD
- Examples of manufacturers and programming tools: Altera Quartus II, Xilinx ISE

Chapter 2. VHDL programming **(5 weeks)**

- History of the VHDL.
- Comparison between VHDL and programming languages.
- Different descriptions of an architecture: data flow, behavioral, structural.
- Identifiers and capitalization sensitivity.
- Comments.
- Representation of numbers in VHDL
- General structure of a VHDL code: Library, Entity, Ports, Architecture.
- Data types: Predefined, user-defined
- Operators: Logic, Relational, Offset, Concatenation
- Signal attributes: EVENT, ...
- Signal, variable and constant
- Process
- Component
- Instruction IF-THEN-ELSE
- Instruction CASE-WHEN
- Instruction WHEN-ELSE
- Instruction WITH-SELECT-WHEN

Chapter 3. Applications on FPGA Circuits **(5 weeks)**

- Multiplexer
- Bascule D
- Adder
- Universal counter with actions: activation, reset, load.
- Frequency divider.
- Frequency management with buttons: selection, division
- 7-segment decoder,
- Serial display on several 7 segments.
- 8-bit arithmetic-logical unit
- 8-bit comparator

Chapter 4. Advanced Design with Finite State Number Machines (FSM) **(4 weeks)**

- Introduction: Mealy and Moore structure
- Representation of an FSM machine
- FSM Design Examples

Evaluation method:

Exam: **60%,** Continuous assessment: **40%**

Bibliographical references:

[1] .Volnei A. Pedroni, « Circuit Design with VHDL », MIT Press, 2004.

[2] . Volnei A. Pedroni, « Circuit Design and Simulation with VHDL », 2ème édition, MIT Press, 2010.

[3] . Bryan Mealy, Fabrizio Tappero, « Free Range VHDL », 2018

[4] . Pong P. Chu, « FPGA prototyping by vhdl examples : Xilinx Spartan™-3 Version », John Wiley & Sons, 2008.

[5] . Jacques Weber, Sébastien Moutault, Maurice Meaudre, "The VHDL language: from language to the circuit, from the circuit to language", Dunod, 2007.

[6] . Christian Tavernier, "Programmable Logic Circuits", Dunod 1992.

Semester: 1
Teaching unit: UEF 1.1.2
Matter 3: Advanced Signal Processing
VHS: 45h00 (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

The student receives the basic notions that allow him to understand and apply signal processing methods concerning random signals and digital filters.

Recommended prior knowledge:

Knowledge of numerical processing of deterministic signals and probability is necessary to follow this subject. This knowledge is provided at the level of the third year of the Electronics Bachelor's degree.

Content of the material:

Chapter 1: Digital Filter Reminders (FIR and IIR) (3 weeks)

- Z Transform
- Structures, transfer functions, stability and implementation of digital filters (RIF and RII)
- Digital filter with minimum phase
- Methods of synthesis of RIF filters and RII filters
- Multi-rate digital filters

Chapter 2: Random Signals and Stochastic Processes (4 weeks)

- Reminder about random processes
- Stationarity
- Power spectral density
- Suitable filter, Wiener filter
- Periodogram, correlogram, averaged periodogram, smoothed periodogram
- Notions of stochastic processes
- Stationarities in the broad and strict sense and ergodicity
- Examples of stochastic processes (Poisson process, Gaussian process and Markov process)
- Higher-order statistics (Moments and cumulants, Polyspectra, non-Gaussian processes, nonlinear processing)
- Introduction to particle filtering

Chapter 3: Parametric Spectral Analysis and Adaptive Digital Filtering (4 weeks)

- Parametric methods
- AR model (Levinson, Yulewalker, Burg, Pisarenko, Music...)
- Model ARMA
- LMS stochastic gradient algorithm
- RLS recursive least squares algorithm

Chapter 4: Time-Frequency and Time-Scale Analysis (4 weeks)

- Time-frequency duality
- Short-term Fourier transform
- Continuous wavelets, discreet dyadic unstoplets
- Multi-resolution analysis and wavelet bases
- Wigner-Ville Transform
- Time-scale analysis.

Method of evaluation:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

[1] .Mori Yvon, "Random signals and stochastic processes", Lavoisier, 2014.

[2] .N. Hermann, "Probabilités de l'ingénieur: variables aléatoires et simulations Bouleau", 2002.

[3] .M., "Traitement Numérique des Signals", Dunod, Paris, 1981.

[4] .J. M Brossier, "Signal et Communications Numériques, Collection Traitement de Signal", Hermès, Paris, 1997.

[5] .M. Bellanger, "Traitement numérique du signal: Théorie et pratique", 8th edition, Dunod, 2006.

Semester: 1
Teaching unit: UEF 1.1.2
Material 4: Digital Servo Systems
VHS: 45h00 (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

Introduce the properties and representations of discrete-time linear dynamical systems. Give the fundamental elements of the control of linear systems represented as a Z-transfer function. Present the different methods of synthesis of discrete-time correctors.

Recommended prior knowledge:

Temporal and frequency analysis of continuous servo systems, Graphical and state representations, Corrector synthesis.

Material content:

Chapter 1:Study of the sampling of a signal (5 weeks)

Z-transform and modified Z-transform: Shannon's theorem, zero-order and one-order blockers, properties of the z-transform, Overview of the modified z-transform and its properties,... Theorem of the initial and final value of a sampled system

Sampled transfers, and recurrent equations: Discretization of a continuous transfer, Representation of discrete systems by recurrence equations, Properties, ...

Overview on the bilinear transformation of a sampled transfer: Relationship between the servo of continuous systems and the servo of sampled systems (study of the stability of a sampled system by the Routh criterion, ...).

Chapter 2:Analyzing Sampled Systems in the State Space (5 weeks)

Discretization of the equation of state of a continuous system: Relationship between the equation of state of a continuous system and that of a discrete system.

Representation and Solving the Equation of State of a Discrete System: Different Forms of the Evolution Matrix (Diagonal, Companion, Observer, Controller, Observability, and Controllability).

Stability and precision of a discrete system: Roots of the characteristic equation, controllable modes, observable modes from the state representation of the sampled systems, Responses of a sampled system, Stability examination by the Jurry criterion, ...

Notions of governability and observability for SISO and MIMO systems.

Chapter 3. Summary of a controller (5 weeks)

Pole placement by status feedback and output feedback: synthesis of simple control laws

State and Output Estimator: Cases of Inaccessible System States

Other synthesis methods: digital PID controller (1 degree of freedom structure), RST controller (2 degrees of freedom structure).

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

[1] .L. Maret, Régulation automatique, 1987.

[2] .Dorf & Bishop, Modern Control Systems, Addison-Wesley, 1995

[3] .J. L Abatut, Sampled Linear Systems and Servo Systems, Edition Dunod

[4] .J. Ragot, M. Roesch, Exercices et Problèmes d'Automatique, Edition Masson.
[5] .J. Mainguenaud, Cours d'automatique Tome3, Edition Masson.
[6] .T.J. Katsuhiko, Modern Control Engineering, 5th Edition, Prentice Hall.
[7] .H. Buhler, Sampled Adjustments Volume 1, Edition Dunod.
[8] .M. Rivoire, Cours d'automatique tome 2, édition Chihab.
[9] .Th. Kailath, Linear Systems, Prentice-Hall, 1980.

Semester : 1
Teaching unit: UEM 1.1
Subject 1: Practical work Microcontroller systems
VHS: 10:30 pm (TP: 1h30)
Credits: 2
Coefficient: 1

Objectives:

This subject is a supplement intended in particular to support and understand the above-mentioned subject **"UEF1.1.1: Microcontroller systems".** In addition, it offers students the opportunity to manipulate the knowledge acquired in a palpable way through well-targeted examples. In addition, it is preferable that the student can take advantage of designs based on the ATmega328P microcontroller to create reliable applications. The main objective can be to allow students to create their own applications according to the requirements imposed by the design itself.

Recommended Prerequisites:

Microprocessor systems, Digital electronics, FPGA, basic electronics, embedded C, Grafcet.

Main content:

Lab 1: Tutorial and getting started with the IDE and Proteus environment (Basic instructions: If-Else, Switch-case, delay(), millis(), typedef, enum, struct, union).

TP 2: External interrupts with two push buttons (control of four LEDs with two-way shift registers).

Lab 3: Implementation of a finite state machine using a simple push button to PWM control a DC motor with an H-bridge from an Analog module.

TP 4: Design and construction of a power dimmer (SSR solid state relay) via the Timer1 interrupt and the external interrupt to detect the zero-crossing signal.

Lab 5: Management of a drop-down menu with three push buttons and LCD2x16 display.

TP 6: Controlling a stepper motor via Timer1 in CTC "*Clear* Timer on *Compare" mode*.

Lab 7: ADC interfacing with operational amplifiers and analog sensors (Manipulation of internal registers).

TP 8: Multitasking using a function pointer and interrupt timer1 to manage the scrolling time of a group of LED diodes.

TP 9: Reading/writing data from an internal EEPROM and an external EEPROM type 24lc256 using the I^{2}C bus.

TP 10: Control of a DC type motor via the Bluetooth module "**HC-05"** using the MIT App Inventor "**Android".**

TP 11: Implementation of a PID controller dedicated to temperature control using the DS 18b20 sensor and an SSR solid relay mentioned above.

TP 12: Data exchange between Arduino (Slave) UNO and ESP32 (Master) platforms using the UART communication protocol.

TP 13: Data exchanges between Arduino (Clients) UNO and Raspberry pi 4 (Broker) platforms using the MQTT protocol.

TP 14: Data exchange between Arduino and an S7-1200 PLC via the Modbus TCP/IP protocol to control a Micro master 420 type speed drive.

Method of evaluation:

Continuous Control: **100%**

Bibliographical references:

[1] .Arduino Workshop: A Hands-On Introduction with 65 Projects; De John Boxall; No Starch Press 2013.

[2] .C Programming for the Pc the Mac and the Arduino Microcontroller System; De Peter D Minns; AuthorHouse-2013

[3] .Raspberry Pi for Arduino Users: Building IoT and Network Applications and Devices; De James R. Strickland; Apress-2018

[4] .Practical AVR Microcontrollers: Games, Gadgets, and Home Automation with the microcontroller used in the Arduino; De Alan Trevennor; Apress-2012

[5] .Advances in Smart System Technologies: Select Proceedings of ICFSST 2019;

[6] .Published by P. Suresh, U. Saravanakumar, Mohammed Saleh Hussein Al Salameh; Springer Nature.

[7] .INTRENET OF THINGS WITH ARDUINO AND BOLD IOT: With Arduino and Bolt;

[8] .By Ashwin Pajankar; Published by BPB Publications 2018.

[9] .Arduino: A Technical Reference: A Handbook for Technicians, Engineers, and Makers; De J. M. Hughes; "O'Reilly Media, Inc." 2016.

[10] . Building Arduino PLCs: The essential techniques you need to develop Arduino-based PLCs; De Pradeeka Seneviratne; Apress-2017.

[11] . Arduino Robot Bonanza; De Gordon McComb; McGraw Hill Professional-2013

[12] . Arduino Sketch for ESP32 Development Workshop; Agus Kurniawan; Published by PE Press-2018.

Semester: 1
Teaching Unit: UEM 1.1
Material2 :TP FPGA and VHDL
VHS: 10:30 pm (TP: 1h30)
Credits: 2
Coefficient: 1

Teaching objectives:

This subject is built around the subject "**UEF1.1.1: Advanced Digital Electronics: FPGA and VHDL"**; it allows students to apply the knowledge acquired in the form of specific examples.
This subject allows the student to design an electronic system using the VHDL description language and test each design on an FPGA board.

Recommended prior knowledge:

Combinatorial and sequential logic.

Material content:

TP1: Presentation of the development and simulation tool: Altera Quartus II or Xilinx ISE.
TP2: Exploitation of the development map through a given example of an adder.
TP3: First circuit examples: multiplexer, D flip-flop.
TP4: Simple 48-bit decimal counter.
TP5: 48-bit decimal counter with actions: activate, reset, charge.
TP6: Road traffic light.
TP7: Multiplier/divider with shift registers.
TP8: Serial display on several 7 segments.
TP9: Digital clock.
TP10: Frequency division.
TP11: Frequency division controllable with buttons.
TP12: VGA display.

Evaluation method:

Continuous Control: **100%**

Bibliographical references:

[1] .Volnei A. Pedroni, « Circuit Design with VHDL », MIT Press, 2004.
[2] .Volnei A. Pedroni, « Circuit Design and Simulation with VHDL », 2ème édition, MIT Press, 2010.
[3] .Bryan Mealy, Fabrizio Tappero, « Free Range VHDL », 2018
[4] .Pong P. Chu, « FPGA prototyping by vhdl examples : Xilinx Spartan™-3 Version », John Wiley & Sons, 2008.
[5] .Jacques Weber, Sébastien Moutault, Maurice Meaudre, "The VHDL language: from language to the circuit, from the circuit to language", Dunod, 2007.
[6] .Christian Tavernier, "Programmable Logic Circuits", Dunod 1992.

Semester: 1

Teaching unit: UEM1.1
Subject 3: Advanced Signal Processing/Digital Servo Systems TP
VHS: 10:30 pm (TP: 1h30)
Credits: 2
Coefficient: 1

Teaching objectives:

Hands-on work done in MATLAB to give a hands-on feel to complex theoretical concepts.

Recommended prior knowledge:

Mathematics (Probability Theory and Calculus, Complex Analysis)- Deterministic Signal Theory, Probability and Statistics.

Content of the material:

Advanced Signal Processing Lab

TP1: Synthesis and application of a low-pass RIF filter using the window method (Hanning, Hamming, Bessel and/or Blackman)
TP2: Synthesis and application of a low-pass RII filter by bilinear transformation
TP3: Parametric AR and/or ARMA spectral analysis of sound signals (example of non-audible signals) stationary)
TP4: Elimination of 50Hz interference by the LMS gradient algorithm
TP5: Denoising a signal by the discrete wavelet transform DWT.

TP Digital Servo Systems

Evaluation method:

Continuous assessment: 100%.

Bibliographical references:

[1] .Mori Yvon, "Random signals and stochastic processes", Lavoisier, *2014.*
[2] . 2. N. Hermann, "Probability of the Engineer: Random Variables and Birch Simulations", 2002.
[3] . 3. M., "Traitement Numérique des Signals", Dunod, Paris, 1981.
[4] . 4. M. Bellanger, "Digital Signal Processing: Theory and Practice", 8th edition, Dunod, 2006

Semester : 1
Teaching unit: UEM 1.1
Subject 4: Embedded C++ Programming
VHS : 45h00 (Cours : 1h30, TP : 1h30)
Credits: 3
Coefficient : 2

Teaching objectives:

C++ is one of the must-have languages, it is one of the most used languages in the industry. This is due to the power of **the object** language, its versatility, and its execution performance.
The aim of this subject is to discover the language with a focus on its specific use on embedded systems. The student will be able to assimilate object concepts, discover and implement the C++ language in projects and acquire the beginnings of autonomy in this language.

Recommended prior knowledge:

Digital systems, C programming.

Material content:

Chapter 1. The C++ Language - Introduction **(1 week)**
History, comparison to other languages, use of this language in the industry, standard and version of C++ (C++98 • C++03 • C++11 • C++14 • C++17 • C++20), new language features

Chapter 2. Compiler **(1 week)**
- GCC/G++ Open Source Compiler
- x86/ARM crossover compiler
- Dynamic and static linkage
- Debugging
- Makefile

Chapter 3. Types, Constants, Variables **(1 week)**
References and Pointers, declaring, scoping, initializing, array: declaring, initializing, namespace, dynamic allocation

Chapter 4. Notion of object **(2 weeks)**
- From C to C++, Classes and Objects
- Protection, Access
- Instance variable, Constructor, Destructor
- Surcharge
- "This" Operator
- UML/SysML Object and Modeling
- Automatic code generation

Chapter 5. Derived classes **(1 week)**
- Inheritance and instantiation
- Friends
- Virtual classroom
- Multiple inheritance

Chapter 6. Operator Overload **(1 week)**
- Functions Operator
- Surcharge
- Friend Function, Friendly Class

Chapter 7. Flow control **(1 week)**
Inputs, Outputs, Status, Overhead, File Management

Chapter 8. Exceptions **(1 week)**
Defining an Exception, Interception

Chapter 9. Data structure and STL **(1 week)**
Vecteur, Map, List, Pile, Algo standard

Chapter 10. Particularity of embedded C++ **(1 week)**

- Std Library
- Compilation et linkage
- Specific coding rules
- Virtual classroom

Chapter 11. Multithreading **(2 weeks)**

- Attribute thread_local
- Thread Class, Mutex Class
- Semaphore vs Mutex
- Conditions, Locks, Future and Promise
- Operator and atomic operation

Chapter 12. Smart Pointers **(1 week)**

- Impairment of auto_ptr
- unique_ptr,
- shared_ptr
- weak_p

Chapter 13. Principe sur les templates **(1 week)**

- Concept of generic programming
- Model class
- Template Functions

Method of evaluation:

Exam: **60%,** Continuous assessment: **40%**

Bibliographical references:

[1] . Michael Barr, Anthony Massa, Programming embedded systems: in C and C++, 2ème edition, O'Reilly, 2006.

[2] . Igor Viarheichyk, Embedded Programming with Modern C++ Cookbook: Practical recipes to help you build robust and secure embedded applications on Linux, 1st Edition, Kindle Edition, ISBN-13: 978-1838821043

[3] . Arkady Miasnikov, C++ for embedded systems, Kindle Edition, 2015

[4] . Christopher Kormanyos, Real-time C++: Efficient Object-oriented and Template Microcontroller Programming, Springer-Verlag Berlin and Heidelberg GmbH & Co, 2015

Semester: S1

Teaching Unit: UEM 1.1
Subject: Advanced programming in Python
VHS: 4:00 p.m. (Cours 1:30 a.m., Ho Chi Minh City 1:30 p.m.)
Credits: 2
Coefficient: 2

Objectives of the material:
Targeted skills:

- Use of computer tools for the acquisition, processing, production and dissemination of information
- Python and project management skills,
- Skills in automation and data visualization.

Objectives:

- Deepen the mastery of the Python language and introduce students to the basics of data analysis and artificial intelligence.
- Acquire the basics of solid computer science.
- Learn to program in Python, Excel
- Mastering task automation
- Mastering project management software

Materials needed:

- A computer with Python installed,
- Bibliothèques Python : NumPy, Pandas, Scikit-learn, Matplotlib, os.listdir, os.path.exists, os.mkdir, os.rmdir, Matplotlib, Seaborn, Plitly , Request, Beautiful Soup, Tkinter, PyQT, …
- Tensorflow, PyTorch, ...

Prerequisites : Python programming,
Content of the material:

Chapter 1: Reminders on Python Programming (02 Weeks)

1. Introduction: Basic concepts in computer science and digital tools, installation of Python.
2. Presentation of the notion of operating system: Roles, types (Linux, Woindows, ..) Priority management,
3. Presentations of computer networks (Principle, IP address, DNS, internet, ...)
4. Basic programming: Interactive mode and script mode, Variables, data types, operators. Conditional structures and loops (if, for, while).
5. Functions and Essentials: Predefined functions and creation of functions. Standard modules (math, random). Strings, lists, basic data manipulation.
6. Files, Lists, Tuples, Dictionaries,
7. Exercises:
   - Python Learning Exercises
   - Exercises on how to use the libraries seen in class (Math, Random, NumPy, Pandas,...)
   - ….

Chapter 2: Programming and Automation (04 weeks)

1. Task Automation Principles
   - Python libraries for automation:
     - ✓ Pandas and NumPy.
     - ✓ Os, shütil: manipulation of files and folders
     - ✓ Openpyxl or pandas: working with Excel or CSV files
   - Definitions and examples of automation (sending emails,...)

2. File manipulation with Python:
   - Use libraries to:
     - ✓ Browse a folder (os.listdir)
     - ✓ Verify the existence of a file or folder (os.path.exists)
     - ✓ Create or delete folders (os.mkdir, os.rmdir)
     - ✓ Visualize data: Matplotlib, Seaborn, Plitly
     - ✓ Request to react with Application Programming Interface (API)
     - ✓ Beautiful Soup for Data Scraping
     - ✓ Tkinter, PyQT to visualize graphical data
   - Copy or move files with shutil...
   - Simple search, sorting, and reporting.
   - Serialization and Deserialization (Using the Pickle Module).
   - Serialization of objects and processing of large files (streaming).
   - ….

3. **Exercises**:
   - Using openpyxl and pandas to read, edit, and write Excel or CSV files to:
     - ✓ Create automatic reports
     - ✓ Automatically extract data
     - ✓ ….
   - Writing scripts for:
     - ✓ process text files (search, sorting)
     - ✓ Automate engineering calculations
     - ✓ manage simple reports (PDF, Excel)
     - ✓ ….
   - Sorting, searching, and insertion sorting algorithms
   - Implement a list search function.
   - File Operation
   - Secure browsing (simple network configuration, password management)
   - ….

Chapter 3: Advanced Excel Learning (02 weeks)

1. Principles of macros and creating a simple macro,
2. Pivot tables,
3. Histograms,
4. Bar charts,
5. Spider,
6. etc.
7. **Excel exercises** ....

Chapter 4: Learning GanttProject (02 weeks)

1. Introduction to Project Management:
   - What is a project?
   - What are the challenges of managing a project?
   - Interface de GanttProject
2. Tasks (creation, modification, organization)

3. Time management (project start or end dates)
4. Resource Management
5. **Gantt Project** Exercises

Chapter 4: Advanced Object-Oriented Programming (03 weeks)

1. Code Organization:
   - Custom functions, parameters, return value.
   - Modules, importations et packages.
2. Complex data structures:
   - Lists, tuples, and dictionaries: creation, modification, deletion, traversal.
3. Fundamental concepts of Object-Oriented Programming (OOP):
   - Classes, objects, attributes, and methods.
   - Public, private and protected attributes.
4. Special methods:
   - **Init**, **S, Repr**, **Len**.
5. Advanced concepts:
   - Encapsulation, abstraction, inheritance, polymorphism.
   - Advanced heritage, decorators, design patterns, metaclasses.
6. **Exercises**

Chapter 5: Introduction to Data for AI (02 weeks)

1. Introduction aux Datasets courants en IA :
   - Iris, MNIST, CIFAR-10, Boston Housing, ImageNet.
2. Data preprocessing for machine learning:
   - Cleaning, normalization, encoding, data separation.
   - Validation croisée (cross-validation).
3. Techniques de Feature Engineering :
   - Selection, feature creation, dimension reduction.
4. Essential libraries for AI model development:
   - scikit-learn, TensorFlow, Keras, PyTorch
5. **Exercises**

Practical work:

Lab 01: Mastering the basics of programming in Python

(Control structures, types, loops, simple functions)

1. Initiation
2. Read and process text files
3. Manage simple reports (PDF, Excel)

TP 02 :

- Develop specifications for a mini task automation project with Python consisting of automatically identifying and sending reports by email with Python:

1. Load data from a file (e.g. experimental measurements),
2. Perform simple statistics on the data (mean, standard deviation with interpretation),

3. Generate a graph,
4. Sending the result with Python.

TP 03 :

1. Ex Excel programming of the dashboard seen in TD
2. Creating automated Excel tables
3. Simple macros,
4. Conditional formulas,
5. Research V.

TP 04 :

Organize a meeting in Ganttproject

1. Create a new project:
    - Project name: "Reunion.....
    - Start date: Date and time of the meeting
    - Estimated Duration: Total duration of the meeting
2. Task Definition
    - Agenda items (each agenda item becomes a task)
    - Subtasks: If a point is composed, then create the corresponding subtasks
    - Initial and final tasks (e.g., "Welcoming participants", "Closing the meeting")
3. Definition of resources:
    - Participants (each participant is a resource)
    - Equipment (computer, datashow, etc.)
4. Estimated durations:
    - Duration of each item: time required for each agenda item
    - Point-to-point transition time
5. Creating the Gantt chart:
    - View the agenda
    - Identify key points
6. Track progress in real time (Gantt Chart projection)

Lab 05: Advanced Structures and Code Organization

(Custom functions, dictionaries, modules and modular organization

Lab 06: Advanced Object-Oriented Programming in Python

(Encapsulation, inheritance, special methods, simple design patterns)

TP 07: File Manipulation and Data Analysis

(Read/write files, word processing, introduction to Pandas and NumPy)

TP 08: Data preparation and processing for artificial intelligence

(Loading AI datasets, cleaning, transforming, feature selection)

Final project

Title: Analysis and visualization of a dataset + simple predictive model
Skills mobilized: Data reading, OOP, advanced structures, Pandas, Scikit-learn. (Oral presentation + written report).

Method of evaluation:

exam 60% , CC=40%

Bibliography

[1] . E.Schultz and M.Bussonnier (2020): Python for the SHS. Introduction to Data Programming. Presses Universitaires de Rennes.

[2] . C.Paroissin, (2021): Data science practice with R: arranging, visualizing, analyzing and presenting data. Paris: Ellipses, DL 2021.

[3] . S.Balech and C.Benavent: NLP text minig V4.0, (Paris Dauphine – 12/2019): link: https://www.researchgate.net/publication/337744581_NLP_text_mining_V40_-_une_introduction_-_cours_programme_doctoral

[4] . Allen B. Downey Think Python: How to Think Like a Computer Scientist, O'Reilly Media, 2015;

[5] . Ramalho, L.. Fluent Python. " O'Reilly Media, Inc.", 2022;

[6] . Swinnen, G.. Learn to program with Python 3. Editions Eyrolles, 2012;

[7] . Matthes, E. Python crash course: A hands-on, project-based introduction to programming. no starch press, 2019

[8] . Cyrille, H. (2018). Learn to program with Python 3. Eyrolles, 6th edition. ISBN: 978-2212675214

[9] . Daniel, I. (2024). Learn to code in Python, I've read

[10] . Nicolas, B. (2024). Python, from the complete beginner to object programming Corrected courses and exercises, 3rd edition, Ellipses

[11] . Ludivine, C. (2024). Selenium Master your functional tests with Python, Eni

Online resources:

- Official Python documentation: docs.python.org
- Exercices Python sur Codecademy : codecademy.com/learn/learn-python-3
- W3Schools Python Tutorial : w3schools.com/python/

IV - Detailed programme by subject of the S2 semester

Semester : 2
Teaching unit: UEF 1.2.1
Topic 1: Embedded Processor Architecture
VHS: 67h30 (Lecture: 3h00, Tutorial: 1h30)
Credits: 6
Coefficient : 3

Teaching objectives:

This subject allows students to learn in less time how to exploit the architecture of ARM processors widely used in embedded systems. A first part showing how to configure and work with the development environment of the STM32 platform. A second part presents the basics of programming and the main aspects of the official HAL (Hardware Abstraction Layer). Towards the end, a more advanced part covering aspects such as the use of a real-time operating system, advanced applications like IoT and USB.

The application of the content of this subject is carried out at the level of the subject "**UEM1.2: TP Microprocessor Architecture for Embedded Systems**".

Recommended prior knowledge:

Microprocessor Systems, Microcontroller Systems, Embedded C/C++ Programming, Digital Electronics: FPGA & VHDL Programming.

Material content:

Chapter 1. Introduction to ARM-Cortex processors (4 weeks)

- The different types of ARM-Cortex processors
- ARM Cortex processor architecture
- The registers
- Memory
- The pipeline
- Interruptions and exceptions
- The instruction set
- Performance
- Introduction to STM32 families.
- The SM32CubeIDE development environment.
- Preview on the Nucleo development map.

Chapter 3. Leveraging ARM-Cortex processors (8 weeks)

- Configuring I/O Ports
- Outage Management
- Communication UART
- DMA Management
- Clock Management
- Using Timers
- Analog-to-Digital Conversion
- Digital-to-Analog Conversion
- Communication I²C
- Communication SPI
- Calcul des CRC (Cyclic Redundancy Check)
- Using the Watchdog Timer

- The real-time clock

Chapter 4. Advanced Applications of ARM-Cortex Processors (3 weeks)

- FreeRTOS
- IoT Development
- USB Development

Evaluation method:

Exam: **60%,** Continuous assessment: **40%**

Bibliographical references:

[1] . Carmine Noviello, « Mastering STM32 : A step-by-step guide to the most complete ARM Cortex-M platform, using the official STM32Cube development environment », 2ème édition, Lean Pub, 2022.

[2] . Donald Norris, « Programming With STM32: Getting Started With the Nucleo Board and C/C++ », McGraw-Hill, 2018.

[3] . Yifeng Zhu, « Embedded systems with ARM Cortex-M Microcontrollers in assembly language and C », 3ème édition, E-Man Press LLC, 2017.

[4] . Joseph Yiu, « Definitive guide to ARM Cortex-M23 and Cortex-M33 processors », Elsevier, 2021.

[5] . Joseph Yiu, « The definitive guide to ARM Cortex-M0 and Cortex-M0+ Processors », 2ème édition, Elsevier, 2015.

[6] . Joseph Yiu, « The definitive guide to ARM Cortex-M3 and Cortex-M4 Processors », 3ème édition, Elsevier, 2014.

[7] . Joseph Yiu, « The definitive guide to ARM Cortex-M0 », Elsevier, 2011.

[8] . Joseph Yiu, « The definitive guide to ARM Cortex-M3 », 2ème édition, Elsevier, 2010.

Semester: 2
Teaching unit: UEF 1.2.1
Subject 2: Digital Signal Processors (DSPs)
VHS: 45h00 (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

Be familiar with the internal architecture of a DSP and the hardware platform integrating that DSP as well as the development environment on a DSP-based platform. At the end of this subject, the student must master the design flow and must also be able to make an algorithm-architecture match for the implementation of algorithms on a platform based on DSP processors.

Recommended prior knowledge:

Microprocessor systems. Advanced digital signal processing. Programming in assembly language and C.

Material content:

Chapter 1: Fixed-Point and Floating-Point Arithmetic (2 weeks)

General information on DSP processors, Reminders on signal digitization (sampling, quantization), number representation formats, encoding of integers (positive or unsigned integers, complement to 1, complement to 2), representation of real numbers in a computer (fixed point, floating point)

Chapter 2: TMS320C6x DSP Architecture (4 weeks)

C6000 internal architecture, processor, memory mapping, functional units, *fetch* and execution packages, pipeline architecture, registers, control registers, peripherals (timers, PLLs, interrupts, HPI, GPIO), multichannel *buffered serial port*, instruction set overview

Chapter 3: Signal Processing Algorithms on DSP (5 weeks)

Algorithm-architecture fit. Quantification problems, real-time constraints, I/O management. Presentation of addressing modes (indirect, circular, inverted). Implementation RIF and RII filtering. Offset and circular buffers, Implementation of FFT on DSP (Reverse Addressing).

Chapter 4: Advanced DSP Memory Management Techniques (4 weeks)

Presentation of internal memories (L1 and L2 levels) and external memories (SRAM, Flash, DDRAM, ...). Management of memory sections via the *.cmd file (organization of sections). Management of external memory by EMIF (*External Memory InterFace*). Block transfer technique. Data organization for EDMA. Settings and options for EDMA. Example of data transfer using the DMA controller.

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

[1] . R. Chassaing, D. Reay, Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK, John Wiley & Sons, 2008.

[2] . D. Reay, Digital Signal Processing and Applications with the OMAP-L138 eXperimenter, John Wiley & Sons, 2012.

[3] . T.B. Welch, C.H.G. Wright and M.G. Morrow, Real-Time Digital Signal Processing from MATLAB to C with TMS320C6x DSPs, CRC Press, 2012.

[4] . Steven A Tretter, Communication System Design Using DSP Algorithms, Springer 2008.

[5] . N. Dahnoun, Digital Signal Processing Implementation using the TMS320 C6000 DSP platform, Prentice Hall, 2000.

[6] . N. Kehtarnaz, N. Kim, Real Time Digital Signal Processing Based on TMS320C6000, Newnes, 2004.

[7] . N. Kehtarnaz, M. Keramat, DSP System Design using TMS320C6000, Prentice Hall, 2006.

[8] . S. W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing.

[9] . G. Baudoin and F. Virolleau, DSPs: TMS320C54x Family.

[10] . L. Correvon, DSP and Real-Time: Industrial Application, Haute Ecole d'Ingénierie du Canton de Neuchâtel.

[11] . P. Laspsley , J. Bier , A. Shoham, E. A. Lee, DSP Fundamentals: Architecture and Features, Berkley Design Technology, Inc, 1994.

[12] . Oktay Alkin, Digital Signal Processing: A Laboratory Approach using. PC-DSP, Prentice Hall.

[13] . Digital Control Applications with the TMS320 Family: Selected Application notes, Texas Instruments, 1991.

[14] . M. Pinard, Les DSP, famille ADSP 218X: Principes et applications, Dunod, 2000.

[15] . B. Bouchez, Applications audio-numériques des DSP: théorie et pratique du traitement numérique du son, Publitronic, 2003.

[16] . Texas Instruments, TMS320C6000 Code Composer Studio Tutorial (Rev. C), http://www.ti.com/lit/ug/spru301c/spru301c.pdf, 2000.

[17] . Texas Instruments, Code Composer Studio Development Tools v3.3 Getting Started Guide (Rev. H), http://www.ti.com/lit/ug/spru509h/spru509h.pdf, 2008.

[18] . Texas Instruments, TMS320C6000 Programmer's Guide (Rev. K), http://www.ti.com/lit/ug/spru198k/spru198k.pdf, 2011.

[19] . Texas Instruments, TMS320C6000 CPU and Instruction Set Reference Guide (Rev. G), http://www.ti.com/lit/ug/spru189g/spru189g.pdf, 2006.

[20] . Texas Instruments, TMS320C6000 Chip Support Library API Reference Guide (Rev. J), http://www.ti.com/lit/ug/spru401j/spru401j.pdf, 2004.

[21] . Texas Instruments, TMS320C1X User's Guide. July 1991.

Semester: 2
Teaching unit: UEF 1.2.2
Subject: Computer Vision System
VHS: 45 h (Lecture: 01h30, TD: 01h30)

Credits:4
Coefficient:2

Teaching objectives:

To know the main components of a computer vision system and the role of each component in its design. To study the tools that allow the automatic reproduction of tasks performed by the human visual system and interpreted by the brain.

Recommended prior knowledge:

- ✓ Signal processing;
- ✓ Image processing.

Content of the material:

Chapter 1. Image acquisition and digitization (02 weeks)
Functional composition of a computer vision system, Image definition, Image sensors, image digitization**,** Human vision**,** 3D modeling and camera calibration.

Chapter 2. Reminders about image processing (02 weeks)
Point operations (Logarithmic transformation, Contrast inversion, histogram modification), Local Operations (Spatial Filtering, Frequency Filtering)

Chapter 3. Contours and segmentation (03weeks)
Edge Detection (Contour Definition, Gradient Approach, Lapalucian Approach, Canny Filtering, LOG Filtering, Active Edges) Segmentation (Histogram Thresholding, Region Approaches)

Chapter 4. Movement (02weeks)
Motion estimation, optical flow, Horn & Schunk method, Block matching Gunnar Farneback approach

Chapter 5. Feature detection (02 weeks)
Definition of a point of interest, geometric transformations, Moravec detector Harris detector ,SIFT
Chapter 6. Machine Learning (03weeks)
Machine Learning Definition, Supervised Classification (K-PPV, Naive Bayes, SVM), Unsupervised Classification (K-Means, Fuzzy C-Means)

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references :

[1] .*R.Gonzalez,* Digital Image Processing, *Upper Saddle River, N.J, Prentice Hall, 2002 (ISBN 978-0-201-18075-6).*

[2] . *M.Bergounioux, Introduction to Mathematical Image Processing: Deterministic Methods*, vol. 76, Berlin, Heidelberg, Springer Berlin Heidelberg, coll. "Mathematics and Applications", 19 March 2015

[3] . *A. Herbulot, Non-parametric statistical measurements for image and video segmentation and active edge minimization, Doctoral thesis defended at the University of Nice - Sophia Antipolis, October 10, 2007*

[4] . *C.Grava , Motion compensation by cellular neural networks: Application in medical imaging, PhD thesis, defended at INSA Lyon 2003.*

[5] . *J-P. Cocquerez and S. Phillip, Preface H.Maitre Analyse d'images: Filtering and segmentation, Edition Masson, 1995*

[6] . *H. P. Moravec, "Towards Automatic Visual Obstacle Avoidance". In Proceeding of Fifth International Joint Conference on Artificial Intelligence, Cambridge Massachusetts USA, pp. 584-587, 1977.*

[7] . C. Harris and M. Stephens. A combined corner and edge detector. In Proceedings of The Fourth Alvey Vision Conference, pages 147

[8] . D.Lowe, Object recogniton from local scale invariant features. In Proceedings of The IEEE international conference on computer vision, 1999

[9] . I. Rish, "An empirical study of the naive Bayes classifier". IJCAI Workshop on Empirical Methods in Artificial Intelligence

[10] . V.Vapnik, « Support Vector Regression Machines ». Advances in Neural Information Processing Systems 9, NIPS 1996, 155-161, MIT Press. 1997

[11] . E. Lebarbier, T. Mary-Huard, Lecture Notes, Unsupervised Classification, AgroParisTech, 2017

Semester : 2
Teaching unit: UEF 1.2.2
Material4: Industrial Programmable Logic Controllers
VHS: 45:00 (Lecture: 1h30, TD: 1h30)
Credits: 4

Coefficient : 2

Teaching objectives:

This course allows the student to understand the hardware and software organization of APIs, to choose an API and associated components according to the desired application, and to use a programming language adapted to the API.

Recommended prior knowledge:

Combinatorial and Sequential Logic, Microprocessors, Microcontrollers, Sensors, Industrial Networks and Communications.

Content of the material:

Chapter 1: PLC: Industrial Programmable Logic Controllers (2 weeks)

Definition of an API, Internal and external architecture of an API and characteristics. Choice of API. Types of I/O of an API and its characteristics.

Chapter 2: Materialization of industrial processes by APIs (3 weeks)

Definition of an automated system. The essential parts of an automated system (PO, PC, HMI, Interfacing). Principle of operation of an API and an automated order-information system. Wiring. Sensor-actuator concepts, industrial networks, ...

Chapter 3: API Programming (5 weeks)

Introduction to the Grafcet. Introduction to languages: LD, IL, FBD, SFC, SCL. Application: definition of PO-PC parts, development of the grafcet, ladder programming. Application exercises.

Chapter 4: Process Visualization (3 weeks)

Introduction to HMI (Human Machine Interface) and SCADA systems, process representation and control, alarm display, acceptance, archiving, user management, …. Application exercises.

Chapter 5: Programmable logic controller dedicated to safety (2 weeks)

Architecture, process and machine control, management of safety functions.

Method of evaluation:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references :

[1] . Frank D. Petruzella, Programmable Logic Controllers, 4th edition, Ed. Mc Graw Hill 2004.

[2] . William Bolton, Industrial Programmable Automata, Editions Dunod, l'Usine Nouvelle, 2010.

[3] . Ian G. Warnock, Programmable Controllers: Operation and Application, Prentice Hall.

[4] . Gilles Michel, Architecture and Applications of Industrial Programmable Logic Controllers, Dunod.

[5] . G. Michel, Industrial Programmable Automata, Dunod, 1979.

[6] . S. Thelliez and J.M. Toullote, Grafcet et logique industrielle programmee, Eyrolles, 1980.

[7] . J.C. Bossy, P. Brard, P. Faugère, C. Merlaud, Le Grafce: sa pratique et ses applications, Educalivre Ed. Casteilla, 1995.

[8] . Henri Ney, Eléments d'automatismes, Collection Electrotechnique et normalisation, Edition Nathan, 1996.

[9] . M. Diaz, Petri Nets - Fundamental Models. Traité IC2 - Série Informatique et Systèmes d'Information, Hermès Science 2001

[10] . A. Choquet-Geniet, Les réseaux de Pétri – Un outil de modélisation (Petri Networks – A Modelling Tool), Dunod, 2006. P a g e | 50

[11] . P. Ladet, Outils de modélisation des automatismes séquentiels, Les réseaux de Pétri, Techniques de l'ingénieur, 1990.

[12] . Siemens learning/training support, Module 041-101 TIA Portal WinCC Basic with KTP700 Basic and SIMATIC S7-1200.

*[13] .	IEC 61508-2 Functional safety of safety-related electrical/electronic/programmable electronic systems.*

Semester : 2
Teaching unit: UEM 1.2
Subject 1: Embedded Processor Architecture
VHS: 10:30 p.m. (TP: 1h30)
Credits: 2

Coefficient : 1

Teaching objectives:

This subject is built around the subject "**UEF1.2.2: Microprocessor Architecture for Embedded Systems**"; it allows students to apply the knowledge acquired in the form of specific examples.

Recommended prior knowledge:

Microprocessor systems, combinatorial and sequential logic, fundamental electronics, power electronics.

Material content:

Lab 1: Introduction to the STM Nucleo Development Board
Lab 2: Getting started with the SM32CubeIDE environment.
Lab 3: Implementation of a simple project (flashing an LED) on the STM Nucleo board
TP 4: Development of a project with several inputs/outputs: pushers, LEDs, etc.
Lab 5: Developing a Project with Interruptions
Lab 6: Configuring and using Timers devices.
Lab 7: Transmitting and Receiving with USART Serial Communication
Lab 8: Analog-to-Digital Conversion
Lab 9: PWM pulse width variation
TP 10: Demonstration of DMA operation
TP 11: Digital-to-Analog Conversion

Evaluation method:
Continuous Control: **100%**

Bibliographical references:

[1] . Carmine Noviello, « Mastering STM32 : A step-by-step guide to the most complete ARM Cortex-M platform, using the official STM32Cube development environment », 2ème édition, Lean Pub, 2022.
[2] . Donald Norris, « Programming With STM32: Getting Started With the Nucleo Board and C/C++ », McGraw-Hill, 2018.
[3] . Yifeng Zhu, « Embedded systems with ARM Cortex-M Microcontrollers in assembly language and C », 3ème édition, E-Man Press LLC, 2017.
[4] . Joseph Yiu, « Definitive guide to ARM Cortex-M23 and Cortex-M33 processors », Elsevier, 2021.
[5] . Joseph Yiu, « The definitive guide to ARM Cortex-M0 and Cortex-M0+ Processors », 2ème édition, Elsevier, 2015.
[6] . Joseph Yiu, « The definitive guide to ARM Cortex-M3 and Cortex-M4 Processors », 3ème édition, Elsevier, 2014.
[7] . Joseph Yiu, « The definitive guide to ARM Cortex-M0 », Elsevier, 2011.
[8] . Joseph Yiu, « The definitive guide to ARM Cortex-M3 », 2ème édition, Elsevier, 2010.

Semester: 2
Teaching unit: UEM 1.2
Subject 2: Digital Signal Processors (DSP)
VHS: 10:30 pm (TP: 1h30)
Credits: 2

Coefficient: 1

Teaching objectives:

Put into practice the theoretical knowledge acquired during the course and tutorial sessions. Familiarize the student with application programs on a DSP platform through an algorithm architecture. Programs can also be run using the simulator supplied with CCS.

Recommended prior knowledge:

Microprocessor systems. Advanced signal processing. Programming in assembly language and C.

Material content:

Training teams are asked to carry out at least 4 practical exercises (or more, if possible) depending on the type of DSP platform available. In addition, it is allowed to add or replace some of the labs from the attached list with other labs related to the subject.

Note*: Any changes made to this list must be reported to the CPND so that other establishments can benefit from them.*

TP1: Getting started with CCS and getting to know the DSKTMS320C6x evaluation board
Code Composer Studio (CCS) integrated development environment, compiling, loading, running, and debugging simple programs.
TP2: Acquisition, Processing and Playback of Audio Signals with the DSKTMS320C6x
Sampling, spectrum aliasing, quantization, data transfer to/from Codec and use in polling or interrupt mode.
TP3: Signal Generation with the DSKTMS320C6x
Wave generation, sine wave, AM modulation and FM modulation.
TP4: Implementation of Digital Filters by the DSKTMS320C6x
IIR and RIF filters.
Lab 5Implementation of the Discrete and Fast Fourier Transform
TFD and TFR
TP6: Using Matlab with the DSK TMS320C6x
Simulations: Matlab or Simulink, automatic code generation for the DSK using Simulink, Real Time Workshop, and Code Composer Studio.

Evaluation method:

Continuous Assessment: 100%

Bibliographical references:

[1] . R. Chassaing, D. Reay, Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK, John Wiley & Sons, 2008.

[2] . T.B. Welch, C.H.G. Wright and M.G. Morrow, Real-Time Digital Signal Processing from MATLAB to C with TMS320C6x DSPs, CRC Press, 2012.

[3] . Steven A Tretter, Communication System Design Using DSP Algorithms, Springer 2008.

Semester: 2
Teaching unit: UEM 1.2
Subject 1: Computer Vision
VHS: 10:30 pm (TP: 1h30)
Credits: 2
Coefficient: 1

Teaching objectives:

Present and discuss methodologies applied to computer vision. The concepts covered and their applications should prepare students for modern computer vision tools and lead them to a mastery of the ideas and techniques that will enable them to integrate a digital imaging and vision system into an industrial application

Recommended prior knowledge:

Signal Processing, Image Processing.

Material content:

TP1: Introduction to the use of OpenCV

- Representation of images and video in OpenCV
- Treatments on Color and palette

TP2: Camera calibration and 3D reconstruction

TP3: Edge detection and segmentation

TP4: Motion detection and estimation

TP5: Feature Detection

TP6: Object Pursuit

TP7: Machine Learning

Evaluation method:

Continuous Assessment: 100%

Bibliographical references

Semester: 02
Teaching unit: UEM 1.2
Material :TP Industrial Programmable Logic Controllers
VHS: 10:30 pm (TP: 1h30) – Once a fortnight
Credits: 2
Coefficient: 1

Teaching objectives:

Teach the student how to install, program, and use an API. Show them how to analyze and respect the technological and security constraints related to the interfacing of the various industrial elements with a PLC. To introduce him to master the tasks of editing and debugging programs as well as the correction of detected errors.

Recommended prior knowledge:

- ✓ Microcontrollers;
- ✓ API;
- ✓ Grafcet;
- ✓ Ladder.

Content of the subject: Choose at least 5 practical work.

TP01: Getting started with the API environment: Simulation of an automated system, Review of the different software. Introduction to Siemens Step7 Software

TP02: Implementation of an API: Hardware Configuration. Introduction to Ladder programming (On-Off of an actuator with hold). Use of digital inputs/outputs: Use of relays, contactor, etc. (possibly make the necessary wiring).

TP03: Checking the level of a tank. Use of special blocks (barriers)

Examples of applications: Carry out at least 2 practical exercises from the list of following practical exercises

TP04: Checking a Pneumatic Cylinder

TP05: Traffic light control for a simple intersection

TP06: Inspection of a bottle filling and transfer unit

TP07 : Checking an Automated Drill

TP08: Transfer and sorting of parts of different sizes

TP09: Inspection of a forging unit

TP10: Control of a Drug Manufacturing Unit

TP11 : Inspection of a Tunnel Kiln

Evaluation method:

Continuous assessment: 100%.

Bibliographical references:

[1] . Frank D. Petruzella, Programmable Logic Controllers, 4th edition, Ed. Mc Graw Hill 2004.

[2] . William Bolton, Industrial Programmable Automata, Editions Dunod, l'Usine Nouvelle, 2010.

[3] . Ian G. Warnock, Programmable Controllers: Operation and Application, Prentice Hall.

[4] . Gilles Michel, Architecture and Applications of Industrial Programmable Logic Controllers, Dunod.

[5] . G. Michel, Industrial Programmable Automata, Dunod, 1979.

[6] . S. Thelliez and J.M. Toullote, Grafcet et logique industrielle programmee, Eyrolles, 1980.

[7] . J.C. Bossy, P. Brard, P. Faugère, C. Merlaud, Le Grafce: sa pratique et ses applications, Educalivre Ed. Casteilla, 1995.

Semester: 3
Teaching unit: UEM 1.2
Subject: Java Programming for Embedded Systems
VHS: 45h00 (Lecture: 1h30 , TP: 1h30)–Choose between Python or Java
Credits: 3
Coefficient: 2

Teaching objectives:

According to the IEEE, Python and Java are the Best Programming Languages in 2021. Python is a widely used programming language that is easy to learn. It is suitable for both beginners and experts for its simplicity, readable syntax and variety of use. He is essential in all fields: embedded, a web developer, an analyst, a data scientist or a marketing expert,.. etc. Java is a pure object language, it is the most appropriate language for programming the real. It is present in the software kernels of almost all current technological devices. This subject allows the student to access an appreciable level in Java programming that allows him to deal with this aspect of modern technology.

Recommended prior knowledge:

- ✓ Programming (Pasacal/Matlab/C language);
- ✓ Computer Science 1, Computer Science 2, Computer Science 3;

Material content:

Java:

Chapter 1: Introduction to Java **(1 week)**
Chapter 2: Control Instructions **(3 weeks)**
A) Choice statements: The *if statement* (The *simple* if, *The* if *with* the else part, *nested* ifs and *if-else if*), 2.4 Logical operators, The *switch statement,* Conditional operators, 2.7 Priority and associativity of operators
B)Loops
2.9 La boucle *while*
2.10 La boucle *do... while*
2.11 *The for Loop*
2.12 Nested Loops
2.13 The Keywords Break and *Continue*
The *break instruction*
The Label Break Statement
The investigation continues
Instruction continues with label
Chapter 3: Mathematical Function, Characters and Strings **(3 weeks)**
3.2 Commonly Used Mathematical Functions
3.2.1 Trigonometric methods
3.2.2 Exponent Methods (Exponentials and Powers)
3.2.3 Rounding methods
3.2.4 The min, max and abs methods
3.2.5 The random method
3.3 Type of character data and operations
3.4 Le type String
3.4.1 Obtaining the length of a string
3.4.2 Obtaining characters from a string
3.4.3 Concatenation of Strings
3.4.4 Converting Strings
3.4.5 Playing a Channel from the Keyboard
3.4.6 Reading a Character from the Keyboard
3.5 Formatted Output on Screen (System.out.printf Statement)
Chapter 4: Methods **(2 weeks)**
4.2 Definition of a Method
4.3 Calling a Method
4.4 Void methods and methods allowing a return of value
4.5 Passing parameters by values
4.6 Scope of Variables
4.7 Method Overload
Chapter 5: Paintings **(3 weeks)**
A) One-dimensional tables

5.2 Table Basics
5.2.1 Reporting tables
5.2.2 Creating Tables
5.2.3 Table Size and Defaults
5.2.4 Access to Table Elements
5.2.5 Array Initializers
5.2.6 Boucle foreach
5.3 Transmit Tables to Methods
5.4 Returning an array from a method
5.5 Variable-Length Argument Lists
B) Two-dimensional tables
5.7 Basics of Two-Dimensional Arrays
5.7.1 Reporting and Creating Two-Dimensional Tables
5.7.2 Obtaining Lengths of Two-Dimensional Arrays
5.7.3 Shredded (Serrated) Arrays
C) Multidimensional Tables

Chapter 6: Objects and Classes **(3 weeks)**

6.2 Defining Classes for Objects
6.3 Example: Defining Classes and Creating Objects
6.4 Building Objects Using Constructors
6.5 Accessing Objects via Reference Variables
6.5.1 Reference Variables and Reference Types
6.5.2 Access to Object Data and Methods
6.6 Static variables, constants and static methods

Evaluation method:

Continuous assessment: ... 40 % ; Examination: 60%.

Bibliographical references:

[1] . Allen B. Downey Think Python: How to Think Like a Computer Scientist, O'Reilly Media, 2015;
[2] . Zed A. Shaw Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code, Addison-Wesley Professional, 2017;
[3] . Barry, P. Head first Python: A brain-friendly guide. " O'Reilly Media, Inc.", 2016;
[4] . Ramalho, L.. Fluent Python. " O'Reilly Media, Inc.", 2022;
[5] . Swinnen, G.. Learn to program with Python 3. Editions Eyrolles, 2012;
[6] . Le Goff, V.. Learn how to program in Python. Editions Eyrolles, 2019;
[7] . Matthes, E. Python crash course: A hands-on, project-based introduction to programming. no starch press, 2019;
[8] . Harvey Deitel, Java: How to Program, 9th Edition, Prentice Hall.;
[9] . Robert Sedgewick and Kevin Wayne, Introduction to Programming in Java: An Interdisciplinary Approach, Addison Wesley, 2007;
[10] . Claude Delannoy, Programming in Java, Editions Eyrolles;
[11] . J. Hunter, Java servlets, O'Reilly ;
[12] . P. Niemeyer, J. Knudsen, Introduction to Java, Ed. O'Reilly

Semester: S2
Teaching unit: UET 1.2.1
Subject: Elements of applied artificial intelligence
VHS: 4:00 p.m. (Cours 1:30 a.m., Ho Chi Minh City 1:30 p.m.)
Credits: 2
Coefficient: 2

Targeted skills:

- Identifying opportunities for artificial intelligence in engineering sciences

- Understand the ethical implications of AI and the best practices for its use.
- Ability to use AI techniques in problem solving

Objectives:

- Mastery of AI algorithms
- Introduction to the fundamental concepts, tools, and applications of modern artificial intelligence, with an emphasis on practice with Python and its libraries.
- Deepen the Python language,
- Understand AI approaches to problem solving,

Prerequisites :

Advanced Python Programming

Materials needed:

- A computer with Python installed,
- Bibliothèques Python : NumPy, Pandas, Scikit-learn, Matplotlib, os.listdir, os.path.exists, os.mkdir, os.rmdir, Matplotlib, Seaborn, Plitly , Request, Beautiful Soup, Tkinter, PyQT, …
- Tensorflow, PyTorch, ...

Content of the material:

Chapter 1: Introduction to Artificial Intelligence AI (01 week)

1. Definitions and scopes of AI.
2. Historical evolution of AI.
3. Introduction to the main areas:
   - Machine Learning
   - Deep Learning

Chapter 2: Basic Mathematics for AI (01 week)

1. **Linear algebra** : vectors, matrices, products, norms.
2. **Probability & Statistics** :
   - Variables, expectancy, variance.
   - Usual laws: normal, binomial, uniform.
3. **Simple linear regression** :
   - Formulation, cost, optimization.
   - Implementation with **Scikit-learn**.
4. **Exercises:**
   - Manipulating Matrices with the NumPy Library (Python)
   - Exercise on linear regression (use a Python library like Scikit-learn for example)
   - Explain the Matplotlib library (Python)
   - …

Chapter 3: Machine Learning (03 weeks)

1. Key concepts: Data, Models, features, labels, generalization.
2. Phases of a learning pipeline: training, validation, testing.
3. Types of learning:
   - Supervised
   - Unsupervised
   - By reinforcement *(preview)*

4. **Exercises:**
   - Deepen the concepts seen in the course
   - ….

Chapter 4: Supervised Classification (3 weeks)

1. Simple classification model training principle:
2. Models and algorithms:
   - SVM (Support Vector Machine)
   - Decision trees
3. Performance evaluation:
   - Confusion matrix, accuracy, recall, F1-score.
5. **Exercises:**
   - Explain how to use Scikit-learn?
   - Comparing Multiple Models on a Dataset
   - ….

Chapter 5: Unsupervised Learning

1. Notion de clustering.
2. Algorithms:
   - **K-means**
   - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
3. 2D visualization and interpretation of results.
4. **Exercises:**
   - Explain how to use a clustering algorithm on a Dataset
   - Explain how to visualize clusters.
   - ….

Chapter 6: Neural Networks

1. Architecture of a neural network:
   - Perception,
   - Layers and hidden layers, weights, biases.
   - Activation function: ReLU, Sigmoid, Softmax, ...
   - Application exercises
2. Introduction to **Deep Learning** :
   - Notion of deep layers.
   - Introduction to Convolutional Networks (CNN)
3. **Exercises :**
   - Explain Tensorflow and PyTorch
   - Analyze a Text Dataset and Predict Sentiment
   - ….

Chapter 6: Introduction Neural Networks

Chapter 7: Mini project (supervised personal work outside of class):

Creation of a complete classification or clustering model, with preprocessing, training and visualization; Select and process a project from start to finish from (to be distributed at the beginning of the semester):

- Handwriting Recognition
- Prediction of natural disasters
- Develop a Chatbot that can answer a company's frequently asked questions, in a natural way.
- Develop a system capable of distinguishing normal sounds of a machine from those indicating an anomaly (defective bearing, excessive vibration, etc.)
- Develop a system (mini AI) capable of analyzing the sentiments expressed in social media posts about a product, brand or event.
- …

Practical work :

TP 01: Initialization

TP 02 :

- Implementing a simple regression with Scikit-learn visualization with Matplotlib (for example)
- Visualize the results with Matplotlib
- …

TP 03 :

- **Machine learning pipeline and data separation**
- Deepen the concepts seen in the course

TP 04 :

- Using Scikit-learn to Train a Simple Classification Model
- …….

TP 05 :

- Implement a clustering algorithm on a Dataset
- Visualiser les clusters : Clustering non supervisé (K-means, DBSCAN)**.**
- ….

TP 06 :

- Building a simple neural network with TensorFlow or PyTorch or keras
- Build a simple CNN to classify images (example: Dataset MINIST)
- …

Method of evaluation:

exam 60% , CC=40%

Bibliography:

- Ganascia, J.Gabriel (2024): AI explained to humans. Paris, France- Edition le Seuil.

- English, Lise, Dilhac, Antione, Dratwa, Jim et al. (2023): Ethics at the heart of AI. Quebec Obvia.

- J.Robert (2024): Natural Language Processing (NLP): definition and principles – Datasciences. Link: https://datascientest.com/introduction-au-nlp-natural-language-processing

- What is natural language processing. Link: https://aws.amazon.com/fr/what-is/nlp/

- M.Journe: Elements of Discrete Mathematics – Ellipses

- F.Challet: Deep learning with Python – Eyrolles

- H.Bersini (2024): Artificial intelligence in practice with Python – Eyrolles

- B.Prieur (2024): Natural Language Processing with Python – Eyrolles
- V.Mathivet (2024) : Implementation in Python with Scikit-learn – Eyrolles
- G.Dubertret (2023): Introduction to cryptography with Python – Eyrolles
- S.Chazallet (2023): Python 3 – The fundamentals of language - Eyrolles
- H.Belhadef, I.Djemal: NLP Method – Course of the University of Msila - Algeria

Semester : 2
Teaching unit: UET 1.2.2
Subject: Compliance with the standards and rules of ethics and integrity.
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

To develop students' awareness of the respect of ethical principles and the rules that govern life at the university and in the world of work. Raise awareness of respect for and appreciation of intellectual property. Explain to them the risks of moral ills such as corruption and how to combat them, alert them to the ethical issues raised by new technologies and sustainable development.

Recommended prior knowledge:

Ethics and deontology (the basics)

Content of the material:

A.Respect for the rules of ethics and integrity,

1. **Reminder of the MESRS Charter of Ethics and Professional Conduct:** Integrity and Honesty. Academic freedom. Mutual respect. Requirement of scientific truth, Objectivity and critical spirit. Equity. Rights and obligations of the student, teacher, administrative and technical 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. The search for the general interest. Inappropriate conduct in the context of collective work
- Adopting responsible conduct and combating abuses: Adopting responsible conduct in research. Scientific fraud. Conduct against fraud. Plagiarism (definition of plagiarism, different forms of plagiarism, procedures to avoid unintentional plagiarism, plagiarism detection, sanctions against plagiarists, etc.). Falsification and fabrication of data.

3. Ethics and deontology in the world of work:

Legal confidentiality in business. Loyalty to the company. Corporate Accountability, Conflicts of Interest. Integrity (corruption in the workplace, its forms, consequences, methods of combating and sanctioning corruption)

B- Intellectual property

I- Fundamentals of intellectual property

1- Industrial property. Literary and artistic property.

2- Rules for citing references (books, scientific articles, communications in a congress, theses, dissertations, etc.)

II- Copyright

1. Copyright in the digital environment

Introduction. Copyright of databases, copyright of software. Specific case of free software.

2. Copyright in the Internet and Electronic Commerce

Domain name law. Intellectual property on the internet. E-commerce Site Law. Intellectual Property and Social Media.

3. Patent

Definition. Rights in a patent. Usefulness of a patent. Patentability . Patent application in Algeria and around the world.

III- Protection and enhancement of intellectual property

How to protect intellectual property. Violation of rights and legal tool. Valuation of intellectual property. Protection of intellectual property in Algeria.

C. Ethics, sustainable development and new technologies

Link between ethics and sustainable development, energy saving, bioethics and new technologies (artificial intelligence, scientific progress, human rights, human rights, human rights,

Method of evaluation:

Review: 100%

Bibliographical references:

[1] . The World Intellectual Property Organization website www.wipo.int

[2] . Charter of University Ethics and Professional Conduct, https://www.mesrs.dz/documents/12221/26200/Charte+fran__ais+d__f.pdf/50d6de61-aabd-4829-84b3-8302b790bdce

[3] . Orders No. 933 of 28 July 2016 laying down the rules relating to the prevention and fight against plagiarism

[4] . The ABCs of Copyright, United Nations Educational, Scientific and Cultural Organization (UNESCO)

[5] . E. Prairat, De la déontologie enseignante. Paris, PUF, 2009.

[6] . Racine L., Legault G. A., Bégin, L., Éthique et ingénierie, Montréal, McGraw Hill, 1991.

[7] . Siroux, D., Déontologie: Dictionnaire d'éthique et de philosophie morale, Paris, Quadrige, 2004, p. 474-477.

[8] . Medina Y., La déontologie, ce qui va changer dans l'entreprise, éditions d'Organisation, 2003.

[9] . Didier Ch., Penser l'éthique des ingénieurs, Presses Universitaires de France, 2008.

[10] . Gavarini L. and Ottavi D., Editorial. Professional Ethics in Training and Research, Research and Training, 52 | 2006, 5-11.

[11] . Caré C., Morale, éthique, déontologie. Administration and Education, 2nd Quarter 2002, No. 94.

[12] . Jacquet-Francillon, François. Concept: professional ethics. Le télémaque, May 2000, n° 17

[13] . Carr, D. Professionalism and Ethics in Teaching. New York, NY Routledge. 2000.

[14] . Galloux, J.C., Industrial Property Law. Dalloz 2003.

[15] . Wagret F. and J-M., Patent of invention, trademarks and industrial property. PUF 2001

[16] . Dekermadec, Y., Innovating through patents: a revolution with the internet. Insep 1999

[17] . AEUTBM. The engineer at the heart of innovation. University of Technology Belfort-Montbéliard

[18] . Fanny Rinck and Léda Mansour, Literacy in the Digital Age: Copying and Pasting Among Students, Université Grenoble 3 and Université Paris-Ouest Nanterre la Défense Nanterre, France

[19] . Didier DUGUEST IEMN, Citing your sources, IAE Nantes 2008

[20] . Similarity detection software: a solution to electronic plagiarism? Report of the Working Group on Electronic Plagiarism presented to the CREPUQ Subcommittee on Pedagogy and ICT

[21] . Emanuela Chiriac, Monique Filiatrault and André Régimbald, A Student's Guide: Intellectual Integrity, Plagiarism, Cheating and Fraud... avoid them and, above all, how to cite your sources, 2014.

[22] . Publication of the University of Montreal, Plagiarism Prevention Strategies, Integrity, Fraud and Plagiarism, 2010.

[23] . Pierrick Malissard, Intellectual Property: Origin and Evolution, 2010.

[24] . http://www.app.asso.fr/

[25] . http://ressources.univ-rennes2.fr/propriete-intellectuelle/cours-2-54.html

V - Detailed programme by subject of the S3 semester

Semester : 3
Teaching unit: UEF 2.1.1
Subject 1: Real-time systems
VHS: 67h30 (Lecture: 3h00, Tutorial: 1h30)
Credits: 6
Coefficient : 3

Teaching objectives:

Present the architecture of a real-time operating system and programming techniques in a real-time language.

Recommended prior knowledge:

- ✓ Design of microprocessor systems;
- ✓ Computer programming;

Content of the material:

Chapter 1: General, Concept and Terminologies on Operating Systems (01 weeks)

- Reminders about the architecture of a computer.
- The operating system (definition, role, structure, etc.)
- Notions of instruction and Macro-instruction
- Program, process, function...

Chapter 2: Introduction to Real-Time Systems (01 weeks)

- Definition of a real-time system
- Real-time constraints. Specifications
- Classification of real-time systems.
- Characteristics and Structure of a Control System.

Chapter 3: Scheduling in Classical Operating Systems (02 weeks)

- Concept of processes and process states
- Process Operation
- Process Scheduling Criteria
- FCFS, SJF, SRTF & Round Robin scheduling algorithm

Chapter 4: Scheduling in Real-time (04 weeks)

- Notion of real-time tasks. Task modeling and characterization.
- Scheduling independent tasks
- Scheduling dependent tasks
- Scheduling in an overload situation

Chapter 5: Scheduling in Multiprocessor real-time (02 weeks)

- Position and Formulation of the Problem
- Single-processor scheduling
- Multiprocessor Scheduling Anomalies
- Conditions of authorisation
- Les algorithmes Earliest Deadline et Least Laxity

Chapter 6: Memory and Communication Management (03 weeks)

- Virtual memory management and physical memory (pagination, addressing, ... allocation)
- competition problems, cooperation, synchronization
- Semaphore, monitors, ...
- Cross-tasking communication and messages

Chapter 7: Programming (02 weeks)

- Introduction to Concurrent Programming
- Management of multitasking aspects, Mutual exclusion, Synchronization, Communication... Real-time programming (real-time JAVA, ADA)
- Examples of applications.

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

[1] . T. Shanley and D. Anderson, PCI System Architecture, Adisson-Wessley.
[2] . H. Son Sang, Advances in Real-Time Systems, Prentice Hall.
[3] . J. W. S. Liu, Real-Time Systems, Prentice Hall, 2000.
[4] . D. Abbott, Linux for embedded and Real-Time systems, 2003, Architectural Press.

[5] . Nicolas Navet, Systèmes temps réel: Ordrement, réseaux et qualité de service, Hermès – Lavoisier, Volume 2, 2006.

[6] . Alan C. Shaw, Real-time systems and software, John Willey & Sons, Inc., 2001.

[7] . Francis Cottet and Emmanuel Grolleau, Real-Time Control System, Dunod 2005.

[8] . Nimal Nissanke, Real-Time Systems, Prentice Hall.

[9] . G.Bollela et al., The Real-Time Specification for Java, Ed. Addison-Wesley.

[10] . Cottet Francis, Joèlle Delacroix, Ordrement temps réel: Cours et exercices correctes, Hermès Science Publications, 2000.

[11] . A. Darseoil, P. Pillot, Le temps réel en milieu industriel, Dunod 1991.

[12] . Y. Trinquet, J.-P. Elloy, real-time executives, engineering techniques.

Semester : 03
Teaching unit: UEF 2.1.2
Subject 3: Embedded Artificial Intelligence
VHS: 45h00 (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

To allow students to become familiar with artificial intelligence (AI) techniques and its applications in embedded systems.

Recommended prior knowledge:

This subject requires prior knowledge of advanced numerical analysis methods, C programming, MatLab or Python.

Material content:

Chapter 1. Introduction to Artificial Intelligence and Embedded Systems (02 weeks)

Artificial Intelligence (AI); Areas of application; Main techniques related to artificial intelligence; Pros and Cons of AI; Embedded systems; Embedded systems and the prospects of AI; Challenges of implementing artificial intelligence in embedded design.

Chapter 2. Machine and Deep Learning Methods (05 weeks)

Supervised learning; Unsupervised learning; Semi-supervised learning; Reinforcement learning; Overall methods in machine learning (bagging, boosting and stacking), Neural networks (MLP, RBF, RNN, BNNs...); Convolutional neural networks (CNNs), deep neural networks (DCNN, VGG-16, ResNet, LSTM, GRU, ...).

Chapter 3. Machine and deep learning application (05 weeks)

Simple examples of the application of machine learning algorithms (Naive-Bayes, Decsion Tree, Random forest, k-NN, K-Means, svm, PCA, Q-Learning,...) and deep learning algorithms in regression, classification, control, data partitioning and dimension reduction problems.
Applications: signal processing (audio,..), imaging (object detection, pattern recognition, ,.. segmentation), natural language, text translation, detection, diagnosis, control, etc.

Chapter 4 Implementing Machine and Deep Learning in Embedded Systems (03 weeks)

Basic concepts about embedded machine learning; Pruning; Mixed quantitation and precision; Embedded architectures for machine and deep learning; Reconfigurable approaches (FPGAs), CPUs, and GPUs; Microcontroller approaches; Accelerator-oriented approaches; Efficient implementation of MAC units (multiply–accumulate); Software and hardware optimizations; Main levels of abstractions; TinyML; Embedded boards Coral, Jetson Nano, Raspberry Pi, ...

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

[1] . Warden, P. and Situnayake, D., 2019. Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers. O'Reilly Media.
[2] . Paluszek, Michael, and Stephanie Thomas. MATLAB machine learning. Apress, 2016.

[3] . Raschka, S., 2015. Python machine learning. Packt publishing ltd.

[4] . Liu, Y.H., 2017. Python Machine Learning By Example. Packt Publishing Ltd.

[5] . Ketkar, N. and Santana, E., 2017. Deep learning with Python (Vol. 1). Berkeley: Apress.

[6] . Kim, P., 2017. Matlab deep learning. With machine learning, neural networks and artificial intelligence, 130(21).

[7] . Warwick, K., 2013. Artificial intelligence: the basics. Routledge.

[8] . Gajski, D.D., Abdi, S., Gerstlauer, A. and Schirner, G., 2009. Embedded system design: modeling, synthesis and verification. Springer Science & Business Media.

[9] . Arora, Mohit. Embedded system design: Introduction to SoC system architecture. Learning Bytes Publishing, 2016.

[10] . Parab, J., Shinde, S.A., Shelake, V.G., Kamat, R.K. and Naik, G.M., 2008. Practical aspects of embedded system design using microcontrollers. Springer Science & Business Media.

[11] . Alippi, C., 2014. Intelligence for embedded systems (pp. 1-283). Berlin: Springer.

Semester: 3
Teaching unit: UEF 2.1.2
Material 1: Embedded Systems
VHS: 45h00. (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

In this subject, students will be able to specify and produce an application or embedded system on a Windows basis.

Recommended prior knowledge:

The content of this subject assumes mastery of programming techniques as well as an average knowledge of an operating system such as Windows or Unix and their network layers.

- ✓ Microcontrollers and Microprocessors
- ✓ Algorithmic;
- ✓ ASM,C/C++,... programming etc.
- ✓ Operating System Basics
- ✓ Integrated development software (Proteus Tool suite, Mplab, CCS)

Material content:

Chapter 1. Introduction to Embedded systems **(2 weeks)**

1.1. History of embedded systems
1.2. Definition of embedded systems
1.3. Development of embedded systems
1.4. Characteristics of embedded systems
1.5. Some properties of embedded systems
1.6. Design constraints of an embedded system
1.7. Some examples of embedded systems
1.8. Fields of application of embedded systems:
1.9. The challenges of embedded systems:

Chapter 2. Embedded Systems Architecture **(4 weeks)**

2.Hardware Aspect of an Embedded System
2.2 Von Neumann architecture
2.3 Harvard architecture
2.4 The microcontroller and its internal structure
2.5 The structure of a PIC assembler program
2.6 Disruption Management PIC16F84
2.7 Management of the PIC16F84's internal timers
(Timer and pulse counts)
2.8 Practical example of a microcontroller-based embedded system

Chapter 3. Programming Languages for Embedded Systems **(3 weeks)**

4.1 CCS C-PCW Compiler Basic Rules
4.2 Variables and constants
4.3 C-CCS Language Operators
4.4 Repetitive structures.
4.5 C-CCS functions adapted to microcontrollers
4.6 I/O Management

4.7 Managing timeouts
4.8 Outage Management
4.9 Serial Link Management
4.10 Examples

Chapter 4. Embedded and multitasking operating system **(4 weeks)**

5.Introduction
5.2 State Machine
5.3 Concepts of real-time operating systems for embedded systems (RTOS)
5.3.1 The scheduler
5.3.2 Services RTOS
5.3.3 Message synchronization and transmission tools
5.3.4 Example Application

Chapter 5. Case study :d development of an embedded application such as the implementation of a PID corrector for temperature control **(2weeks)**

Evaluation method:

Continuous assessment: 40%; Examination: 60%.

Bibliographical references:

[1] . http://beru.univ-brest.fr/~singhoff/supports.html
[2] . Tim Wilmshurst., Designing Embedded Systems with PIC Microcontrollers : Principles and applications.
[3] . Christian Tavernier , AVR Microcontrollers: from ATtiny to ATmega - Description and implementation
[4] . Christian Tavernier , Pic 18 microcontrollers - Description and implementation
[5] . Alexandre Nketsa; Programmable logic circuits: PLD, CPLD and FPGA memories, industrial computing
[6] . A. Dorseuil and P. Pillot. Real-time in an industrial environment. Edition DUNOD, Industrial Computer Collection, 1991.
[7] . Ch. Bonnet, I. Demeure , Introduction to real-time systems, Edition HERMES
[8] . Ivan Cibrario Bertolotti_ Gabriele Manduchi-Real-Time Embedded Systems_ Open-Source Operating Systems Perspective-CRC Press (2012).
[9] . Eugenio Villar, Maite Veiga (auth.), Juan Carlos LÃ³pez, RomÃ¡n Hermida, Walter Geisselhardt (eds.)-Advanced Techniques for Embedded Systems Design and Test-Springer US (1998)
[10] . Distributed Systems—Concepts and Design, 2nd Ed. Addison-Wesley Publishers Ltd., 1994.
[11] . I. Demeure and C. Bonnet. Introduction to real-time systems. Educational collection of telecommunications, Hermès, September 1999.
[12] . PIC Microcontroller and Embedded Systems By Muhammad Ali Mazidi
[13] . C Programming Language by Kernighan & Ritchie PDF
[14] . http://www.craslab.org
[15] . http://beru.univ-brest.fr/~singhoff/supports.html
[16] . https://www.ukonline.be/cours/embeddedsystems
[17] . https://www.techno-science.net/glossaire-definition/Systeme-embarque-page-3.html
[18] . http://www.embedds.com/
[19] . http://www.keil.com/rtos/
[20] . http://embedded-lab.com/

Semester: 3

Teaching unit: UEF 2.1.2
Subject 2: Industrial Networks and Communications
VHS: 45h00. (Lecture: 1h30, TD: 1h30)
Credits: 4
Coefficient: 2

Teaching objectives:

To allow the student to become familiar with the concepts of digital data transmission, more particularly the different types of networks existing in the industrial world. The focus will be on understanding the different topologies with their advantages and disadvantages vis-à-vis a given industrial plant.

Recommended prior knowledge:

- ✓ Local computer networks ;
- ✓ Sensors;
- ✓ Programmable logic controllers;

Material content:

Chapter 1. General information about fieldbuses (04 weeks)

1.1-Definitions (Bus, Field, Network, Local Area Network, Industrial Local Area Network (ILN), Enterprise Local Area Network, Some names of industrial local area networks). 1.2-Presentation of the industrial environment. 1.3-Architecture of an ILN (field networks, workshop networks, factory networks). 1.4-Characteristics of an ILR (number of nodes, amount of information, transmission time). 1.5-Characteristics of the data exchanged in an ILR (Nature of the messages exchanged, Size of the messages). 1.6-Role of an ILN in an industrial facility. 1.7-OSI and RLI architecture (Adaptation of the OSI model to RLIs, Physical layer characteristics for ILRs, MAC sublayer characteristics for ILRs).

Chapter 2. : Le bus 485 Modbus (02 weeks)

Reminder on the RS232 standard. The RS485 link. The Modbus protocol. Modbus addressing and frame.

Chapter 3. The bus CAN (Controller Area Network) (03 weeks)

Global view of CAN. OSI CAN models. CAN data frames and features. Methods of access and the principle of arbitration. Debits. CAN Hardware. Application layer services. CANopen.

Chapter 4. Profibus (03 weeks)

Overview of Profibus and features. The three types of Profibus (DP, FMS and PA). How to get there. Industrial Ethernet and Profinet. Debits.

Chapter 5. Overview of Industrial Wireless Networks (03 weeks)

Technologies, protocols and architectures of industrial wireless networks (WLAN 802.11, Bluetooth, HART protocols, Wireless Profibus, Bluetooth, ZigBee, ...). Security of industrial wireless communication networks.

Evaluation method:

Continuous assessment: 60%; Examination: 40%.

Bibliographical references:

[1] . Belgacem Jarray, Industrial Networks: Buses, Interfaces, Industrial Ethernet, Hart. Corrected courses and exercises. Ellipses, 2017.

[2] . J.F. Hérold, O. Guilloton and P. ANAYA, Industrial Computing and Networks in 20 sheets. Dunod, 2010.

[3] . Jean-Pierre Thomesse, Local Industrial Networks. Eyrolles, 1994.

[4] . Pascal Vrignat, Local Industrial Networks - Courses and practical work. Gaëtan Morin, 1999.

[5] . Ciame, Field networks: operational safety criteria. Lavoisier, 2009.

[6] . Ciame, Field networks: Description and selection criteria. Hermès, 2001.

Semester: 02
Teaching unit: UEM 2.1
Material :Artificial Intelligence Lab

VHS: 10:30 p.m. (TP: 1h30) – Once a fortnight
Credits: 2
Coefficient: 1

Teaching objectives:

To allow students to become familiar with the application of artificial intelligence (AI) techniques in embedded systems. Familiarize yourself with the ML and DL libraries i.e. Keras, Scikit-learn, Tensorflow, etc. under python as well as their implementation on development boards (MC, Raspberry Pi, or FPGA).

Recommended prior knowledge:

This subject requires prior knowledge of programming languages such as C, VHDL, MatLab or Python.

Material content:

TP1. Presentation of development kit, programming language (libraries, toolboxes,...), and databases (open source).

TP2. Application of machine learning (k-NN, RF, SVM,...) in regression, classification, control, data partitioning, and dimension reduction problems.

TP3. Application of deep learning (DCNN, VGG-16,..) in imaging (object detection, ,.. classification), natural language, text translation, anomaly detection, and diagnostics.

TP4. Examples on the implementation of machine and deep learning in embedded systems (embedded Coral board , Jetson Nano, Arduino, FPGA, Raspberry Pi, ...)

Evaluation method:

Continuous Assessment: 100%

Bibliographical references:

[1] . Warden, P. and Situnayake, D., 2019. Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers. O'Reilly Media.

[2] . Paluszek, Michael, and Stephanie Thomas. MATLAB machine learning. Apress, 2016.

[3] . Liu, Y.H., 2017. Python Machine Learning By Example. Packt Publishing Ltd.

[4] . Ketkar, N. and Santana, E., 2017. Deep learning with Python (Vol. 1). Berkeley: Apress.

[5] . Kim, P., 2017. Matlab deep learning. With machine learning, neural networks and artificial intelligence, 130(21).

[6] . Gajski, D.D., Abdi, S., Gerstlauer, A. and Schirner, G., 2009. Embedded system design: modeling, synthesis and verification. Springer Science & Business Media.

[7] . Arora, Mohit. Embedded system design: Introduction to SoC system architecture. Learning Bytes Publishing, 2016.

[8] . Parab, J., Shinde, S.A., Shelake, V.G., Kamat, R.K. and Naik, G.M., 2008. Practical aspects of embedded system design using microcontrollers. Springer Science & Business Media.

[9] . Alippi, C., 2014. Intelligence for embedded systems (pp. 1-283). Berlin: Springer.

Semester: 3
Teaching unit: UEM 2.1
Subject 2: Embedded Systems / Real-Time Systems Lab
VHS: 10:30 pm (TP: 1h30)
Credits: 2

Coefficient: 1

Teaching objectives:

To make students understand the practical aspects of embedded and real-time systems. Design of an embedded system dedicated to the automotive industry where analog and digital inputs/outputs, Timer, Interrupt, ADC-DAC conversion, LCD display, 7 segments, Serial communication (RS232, ...), WIFI communication, Bluetooth, Design of an HMI interface.

Recommended prior knowledge:

C programming, DOS commands.

Material content:

A.Embedded Systems TP - Once a fortnight

Simulation/Implementation of applications using the following hardware/software pairs: FPGA/VHDL, Arduino/C, Microprocessor/Assembler, PIC/C or Assembler, DSP/C, ...

Application example: **Design of a central locking system for an automobile**

Below is an example of the methodology applied to the design of a central locking system for a car. The teacher is free to choose another application.

Initial step: Develop a program that allows the four doors of a car to be opened and closed by means of an infrared or radio frequency control using the TOR control with the lighting (or flashing) of red LEDs and the emission of beeps of varying durations.
Variant 1: Take into consideration the actual closing of the doors (using stops (push button).
Variant 2: Anticipate the case of opening the doors by mistake (accidental action of the infrared control) without the door actually opening. Automatic conviction after a certain period of time.
Variant 3: Plan for the case of a door that is not closed properly while the car is in motion. Alert the driver: beep, display on the dashboard, etc.
Variant 4: Encryption of the control signal

Lead the student to imagine a global architecture and to propose different programming techniques (interrupt, polling, functions, communication (synchronous asynchronous), I/O management, Display, Alarm, etc.) and estimate the cost of the application each time.

B.Real-time systems lab - Once a fortnight

TP 01. Know how to use the basic commands of MS-DOS, Writing and executing a DOS batch script. File editing and running commands.
TP 02. Getting started with Linux commands: Managing processes: Create (launch), Visualize (listing), and Stop (kill) external processes. Memory Management in Linux (Understanding Dynamic Memory Allocation, Diagnosing Some Problems with Dynamic Allocation)
TP 03. Processing of a simple example (semaphore case) using one of the real-time languages.

Evaluation method:

Continuous Assessment: 100%

Bibliographical references:

Semester: S3
Teaching unit: UEM 2.1
Subject 3: Industrial Networks
VHS: 10:30 pm **(TP:** 1h30**)**
Credits: 2
Coefficient: 1

Teaching objectives:

At the end of this subject with the associated Lecture/Tutorial subject, the student will be able to use industrial software (Step-7, Unity Pro, Studio 5000, etc.) to configure and program an industrial network (Profibus, Profinet, CAN, Modbus, etc.) containing the main communication elements including: industrial PC, HMI, PLC and input/output modules. It will also have a good overview of the different protocols included in the OSI layers of the fieldbuses (access methods, frame structures, encoding, etc.).

Recommended prior knowledge:

- ✓ Programmable logic controllers (PLCs);
- ✓ PLC programming languages: Contact, Log, List, Stl, Graph (Grafcet);
- ✓ Local Area Networks (LANs);
- ✓ Sensors and actuators;

Material content:

TP1: PLC (Inputs/Outputs) with Step 7, TIA Portal or Unity Pro. Programming with Graph and/or Ladder and/or STL and/or Log and/or List.
TP2: Profibus with STEP 7 or TIA Portal. Master with passive slave and/or intelligent slave. Simulation and/or practice. Use of a protocol analyzer to decode Profibus messages.
TP3: Profinet with STEP 7 or TIA Portal. PLC1+PLC2+HMI or industrial PC. Simulation and/or practice. Use of a protocol analyzer to decode Profinet messages.
TP4: Serial Modbus with Unity Pro. Master + Slave + Operating Screen. Use a protocol analyzer to decode Modbus messages.
TP5: Modbus TCP/IP with Unity Pro. PLC1 + PLC2. Simulation and/or practice. Use a protocol analyzer to decode Modbus messages.

TP1: Implementation and implementation on RS232, RS485.
TP2: Implementation of the CAN Bus between Arduino modules.
TP3: Modbus dialog between a master PC and a slave device. Master Modbus Communication from a programmable logic controller.
TP4: Profibus industrial network study (based on Arduino or visit to a factory).
TP5: Data exchange via industrial Ethernet network.
TP6: Development of a wireless industrial local area network.

Evaluation method:

Continuous assessment: 100 %.

Bibliographical references:

[1] . https://www.se.com/;
[2] . https://support.industry.siemens.com/.
[3] . Belgacem Jarray, Industrial Networks: Buses, Interfaces, Industrial Ethernet, Hart. Corrected courses and exercises. Ellipses, 2017.
[4] . J.F. Hérold, O. Guilloton and P. ANAYA, Industrial Computing and Networks in 20 sheets. Dunod, 2010.
[5] . Jean-Pierre Thomesse, Local Industrial Networks. Eyrolles, 1994.
[6] . Pascal Vrignat, Local Industrial Networks - Courses and practical work. Gaëtan Morin, 1999.
[7] . Ciame, Field networks: operational safety criteria. Lavoisier, 2009.
[8] . Ciame, Field networks: Description and selection criteria. Hermès, 2001.

Semester: 3
Teaching unit: UEM 1.3
Subject 4: Study and implementation of projects
VHS: 4:00 p.m. (Cours: 1:30 a.m., Ho Chi Minh City: 1:30 a.m.)
Credits: 3
Coefficient: 2

Teaching objectives:

Produce an electronic board that constitutes an embedded system for a given application.
Code development and co-engineering design.

Recommended prior knowledge:

Programming, microprocessor and microcontroller systems.

Material content:

Chapter 1: Study of simulation software **(3 weeks)**
Getting started with a design environment (e.g. "*Proteus Design Suite*"), simulation, analysis of electronic circuits and development of printed circuit boards

Chapter 2: Definition and management of a project **(2 weeks)**
Study of the project (a simple embedded system meeting the needs of a particular sector such as medical, car or home automation) establishment of the specifications, technical choices, cost, schedule and planning of the execution of the work, documentation, choice of components.

Chapter 3: Realization of the electronic part **(3 weeks)**

Chapter 4: Realization of the software part **(3 weeks)**

Chapter 5: Simulation and Testing **(2 weeks)**

Chapter 6: Technical Report **(2 weeks)**
Drafting of the technical file and defense.

Evaluation method:
Continuous assessment: 100%; Review: 00%.

Bibliographical references:

[1] . https://labcenter.s3.amazonaws.com/downloads/Tutorials.pdf
[2] . https://en.wikipedia.org/wiki/Proteus_Design_Suite
[3] . http://www.coursexercices.com/PDF_Cours_Exercices_Telecharger.php?q=proteus+ares+tutorial
[4] . https://www.ele.uva.es/~jesman/BigSeti/ftp/Cajon_Desastre/Software-Manuales/EBook%20-%20Proteus%20Manual.pdf
[5] . https://www.arduino.cc/
[6] . https://www.manager-go.com/gestion-de-projet/
[7] . https://www.techno-science.net/definition/729.html
[8] . https://formation.aapq.org/etape.php
[9] . https://www.nutcache.com/fr/blog/demarche-de-projet/

Semester: 3
Teaching unit: UET 2.1
Material 1: Reverse Engineering
VHS: 45h00 (Course: 1h30 and Workshop: 1h30)
Credits: 2
Coefficient: 2

Teaching objectives:

- Understand the principles and objectives of Reverse Engineering (RE) in the field of science and technology (ST),
- To be introduced to the tools and methods of the ER in the specialty concerned.
- Understand the value and ethics of the principles of the RE in the design, manufacture and quality assurance of products,
- Encourage critical thinking, technical curiosity, reasoned reverse engineering and innovation,
- Learn how to analyze, document, and model an existing system without initial documentation.

Targeted skills

- Break down and analyze an existing system,
- Faithfully reproduce a technical diagram or a 3D model from an existing product,
- Apply diagnostic and simulation tools,
- Work in a group on an exploratory project,
- Identify the legal limits of reverse engineering

Adaptability to specialties in the field of Science and Technology **:**

- All specialties in the ST field are concerned according to
- Examples of tasks: Digital technical documentation, technology watch results, Technical project management, Collaboration around plans, Report analysis, Understanding of industrial processes, Production data monitoring, Reporting techniques, Prototyping, Testing)

Prerequisites:

- Fundamental knowledge in the specialty.

Content of the material:

1. Introduction to Reverse Engineering

- History, legal and ethical issues of the ER,
- Definitions and fields of application: Approaches (hardware, software, processes, etc.)
- Fields: maintenance, remanufacturing, cybersecurity, competitive intelligence

2. General Methodology

- Analysis of a "black box" system
- Functional decomposition
- Block diagrams, inputs/outputs, energy or information flows

3. Reverse engineering hardware

- Electronic boards: visual inspection, component identification
- Use of tools: multimeter, oscilloscope, logic analyzer
- Electronic schematic recognition
- Reconstruction of schematics in KiCad / Proteus

4. Reverse engineering software

- Static analysis of binaries (e.g. .exe, .hex)
- Decompilation, disassembly (introduction to Ghidra, IDA Free, or Hopper)
- Observation of behaviors: sniffing, monitoring (e.g. Wireshark)
- Microcontrollers: flash memory reading, firmware extraction

5. Mechanical reverse engineering

- 3D scanning: scanner, manual measurements
- Reproduction of CAD models from existing parts
- Software used: SolidWorks, Fusion360

6. Security and intrusion detection

- Reverse engineering in cybersecurity: malware detection, vulnerabilities
- Software signing, ROE protections (obfuscation, encryption)

7. Real-life case studies

- Scan an outdated or unknown product (mouse, power supply, Bluetooth module, etc.)
- Example of reverse engineering of a mechanical part or simple system (fan, housing)

Examples of practical work (basic the 4 Geniuses)

- **Electrical Engineering:**

- Reverse engineering of an electronic module without a schematic
- Example: Bluetooth module, timer relay
- Objectives: to identify how it works, to draw the diagram, to propose an improved variant.
- Identification of components (ICs, transistors, resistors, etc.).
- Use of tools: multimeter, oscilloscope, logic analyzer.
- Reading and extracting firmware from a microcontroller.
- Introduction to the detection of electronic counterfeits.

- **Mechanical Engineering:**

- Reverse engineering a simple mechanism
- Examples: hand pump, torque wrench, mini-press, etc.
- Mechanical dismantling of a system (pump, gear, cylinder, etc.).
- Measurements and reconstruction of plans or 3D models with CAD software (SolidWorks, Fusion360).
- Identification of materials and manufacturing methods.
- Functional simulation from the recreated model.

- **Civil Engineering:**

- Analysis of existing structures without plans (walls, slabs, structures, etc.).
- Examples: metal stairs, window sills, formwork)
- Study and reverse engineering of an existing structural element
- Identification of materials, assemblies and constraints.
- Modeling of the structure via Revit, AutoCAD or SketchUp.
- Rehabilitation study or reproduction of old structural elements.

- **Process Engineering:**

- Reverse engineering a laboratory module
- Examples: instruments, distillation, filtration, heat exchanger, simple reactors, etc.
- Analysis of existing industrial systems (distillation column, heat exchanger, reactor, etc.).
- Reconstruction of PFD and PID diagrams from the observation of an installation.
- Identification of sensors, actuators, control devices.
- Study of material/energy flow in a process.

Method of evaluation:

- TP techniques

- Mini-project of reverse engineering (report + defense)
- Final exam (MCQ + case study)

- Exam: 60% and CC TP: 40%

Bibliographical references:

- Reverse Engineering for Beginners – Dennis Yurichev (free online)
- The IDA Pro Book – Chris Eagle (software)
- Practical Reverse Engineering – Bruce Dang
- Documentation :
  - https://ghidra-sre.org
  - https://www.kicad.org
  - https://www.autodesk.com/products/fusion-360

Semester : 3
Teaching unit: UET2.1
Subject:Literature research and memory design
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

Give the student the necessary tools to look for useful information in order to better use it in his or her end-of-studies project. Help them go through the different steps leading to the writing of a scientific document. To show them the importance of communication and to teach them to present the work done in a rigorous and educational way.

Recommended prior knowledge:

Methodology of writing, Methodology of presentation.

Material content:

Part I-: Documentary research:

Chapter I-1: Definition of the Subject (02 weeks)

- Subject Title
- List of keywords relevant to the topic
- Gather basic information (acquisition of specialized vocabulary, meaning of terms, linguistic definition)
- The information sought
- Take stock of your knowledge in the field

Chapter I-2: Select sources of information (02 weeks)

- Type of documents (Books, Theses, Dissertations, Periodical Articles, Conference Proceedings, Audiovisual Documents, etc.)
- Type of resources (Libraries, Internet, etc.)
- Assess the quality and relevance of information sources

Chapter I-3: Locate documents (01 Week)

- Research techniques
- Search operators

Chapter I-4: Processing information (02 weeks)

- Work organization
- The initial questions
- Summary of the selected documents
- Links between different parties
- Final plan of the literature search

Chapter I-5: Presentation of the bibliography (01 Week)

- The systems of presenting a bibliography (the Harvard system, the Vancouver system, the mixed system, etc.)
- Presentation of documents.
- Citation of sources

Part II: Memory Design

Chapter II-1: Outline and stages of the thesis (02 weeks)

- Identifying and Defining the Subject (Summary)
- Problem and objectives of the dissertation
- Other useful sections (Acknowledgements, Table of Abbreviations, etc.)
- The introduction (*The writing of the introduction last)*
- State of the specialized literature
- Formulating the hypotheses
- Methodology
- Results
- Discussion
- Recommendations
- Conclusion and outlook
- The Table of Contents
- Bibliography
- Appendices

Chapter II-2: Writing techniques and standards (02 weeks)

- Formatting. Numbering of chapters, figures and tables.
- The cover page
- Typography and punctuation
- The editorial staff. Scientific language: style, grammar, syntax.
- Spelling. Improved general language proficiency in comprehension and expression.
- Backup, secure, archive your data.

Chapter II-3: Workshop: Critical study of a manuscript **(01 Week)**

Chapter II-4: Oral presentations and defenses (01 Week)

- How to Present a Poster
- How to present an oral communication.
- Defense of a thesis

Chapter II-5: How to Avoid Plagiarism ? **(01 Week)**

(Formulas, phrases, illustrations, graphs, data, statistics,...)

- The quote
- Paraphrasing
- Indicate the complete bibliographic reference

Method of evaluation:

Review: 100%

Bibliographical references:

1. *M. Griselin et al., Guide de la communication écrit, 2nd edition, Dunod, 1999.*
2. *J.L. Lebrun, Guide pratique de rédaction scientifique: comment écrire pour le lecteur scientifique international, Les Ulis, EDP Sciences, 2007.*
3. *A. Mallender Tanner, ABC de la rédaction technique: modes d'emploi, instructions d'utilisation, aides en ligne, Dunod, 2002.*
4. *M. Greuter, Bien rédaction son mémoire ou son rapport de stage, L'Etudiant, 2007.*
5. *Mr. Boeglin, reading and writing at university. From the chaos of ideas to the structured text. L'Etudiant, 2005.*
6. *M. Beaud, l'art de la thèse, Editions Casbah, 1999.*
7. *M. Beaud, l'art de la thèse, La découverte, 2003.*
8. *M. Kalika, Le mémoire de Master, Dunod, 2005.*

Proposal of some discovery materials

Semester:
Teaching unit: UED
Material 1: RFID RFID
VHS: 10:30 pm (Lecture: 1h30)
Credits: 1
Coefficient: 1

Teaching objectives:

Acquire sufficient technical and practical knowledge of RFID technology with a view to its implementation in projects on embedded systems.

Recommended prior knowledge:

Microprocessor System Architecture

Material content:

- Presentation, definition and history
- Ethics, privacy,
- Obstacles to the use of RFID: metallic environment, collisions,
- Classifications des tags RFID
- Working principle
- Read only or read/write?
- TTF and ITF protocols
- Les applications du RFID
- Field communication: NFC
- how NFC/RFID works
- Key features
- Les applications du NFC

Evaluation method:

Review: 100%.

Bibliographical references:

1. *D. Henrici, RFID Security and Privacy: Concepts, Protocols, and Architectures, Springer-Verlag 2008*
2. *K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, 3d edition, Wiley 2010*
3. *Syed Ahson and Mohammad Ilyas, RFID Handbook : Applications, Technology, Security, and Privacy, CRC Press 2008*
4. *http://www.centrenational-rfid.com/fonctionnement-dun-systeme-rfid-article-17-fr-ruid-17.html*
5. *https://fr.wikipedia.org/wiki/Protocoles_de_communication_RFID*
6. *https://fr.wikipedia.org/wiki/Radio-identification*
7. *https://fr.wikipedia.org/wiki/Communication_en_champ_proche*

Semester:
Teaching unit: UED
Material 2: Home automation
VHS: 10:30 pm (Lecture: 1h30)
Credits: 1
Coefficient: 1

Teaching objectives:

To give the student all the equipment of a smart home, its operation and its uses so that he is able to size and design a home automation installation.

Recommended prior knowledge:

Microprocessor Systems, Sensors,...

Material content:

Chapter 1. Comfort in buildings (1 week)
- Thermal, acoustic and visual,

Chapter 2. Notions about the safety of goods and people (7 weeks)
- Fire safety, Access control, Anti-intrusion, Video surveillance, Remote surveillance, ...

Chapter 3.Building management and communication (7 weeks)
- Lighting, Air conditioning, Heating, Regulation, Networks, Remote management, Supervision, BMS (technical management of the building), BMS (centralized technical management), ...

Evaluation method:

Review: 100%.

Bibliographical references:

1. *C. Locqueneux, The Guide to the Home and Connected Objects, Eyrolles 2016*
2. *F-X. Jeuland, La Maison communicante, Eyrolles, 2008 (2nd edition)*
3. *PROMOTELEC, Habitat communicant, Éditions Promotelec, 2006*
4. *E. A. Decamps, La Domotique, Presses universitaires de France, Collection "Que sais-je?", 1988.*
5. *M. Al-Qutayri, Smart Home Systems, In-Teh, Croatia 2010*
6. *C. Nugent, Smart Homes and Beyond, IOS Press, Netherlands 2006*

Semester :
Teaching unit: UED
Subject 3: Embedded systems for the automotive industry
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

The objective of this subject is to give students the necessary foundations to know how to develop and design applications of on-board electronics for the automotive industry, which is a discipline in its own right aimed at optimally controlling the traffic and safety of a vehicle.

Recommended prior knowledge:

Sensors and Instrumentation.

Content of the material:

Chapter 1: Introduction to Embedded Systems

Chapter 2: On-board sensors
Speed and flow sensors, Acceleration sensors, Temperature sensors, Pressure sensors, Proximity sensors, Gyro sensors.

Chapter 3: On-board actuators
Hydraulic Actuator, Air Bag Actuator, Air Conditioning System, Brake System.

Chapter 4: Vehicle System Architecture
Electronic Computer, CAN Communication Bus, Sensor Networks/Actuators.

Chapter 5: Embedded systems in the automotive industry
On-board sensor systems, Anti-lock Braking System (ABS), Wheel Traction Control (ASR), Electronic Dynamic Control (ESP), Measurement of wheel speed (encoder) and vehicle speed (Doppler effect).

Chapter 6: Typical architecture of a vehicle model manufactured in Algeria

Method of evaluation:

Review: 100%

Bibliographical references:

1. *S. Daly, Automotive Air Conditioning and Climate Control Systems, Elsevier, 2006.*
2. *J. Fenton Advances in Vehicle Design, Mechanical Engineering Publications Ltd, 1999.*
3. *B. Hollembeak, Today's Technician: Automotive Electricity and Electronics Classroom and Shop Manual Pack, 5th edition, Delmar, 2010.*
4. *N. Zaman, Automotive Electronics Design Fundamentals, Springer, 2015.*
5. *G. Asch, Data acquisition: from the sensor to the computer, Dunod, 2003.*
6. *G. Asch et al. Data Acquisition, 3rd edition, Dunod, 2011.*
7. *M. Bayart, B. Conrard, A. Chovin, M. Robert, Intelligent Sensors and Actuators, 2005.*
8. *P. Dassonvalle, Les Capteurs : Exercices et problèmes correctes, Dunod, 2005.*
9. *R. Frank, Understanding Smart Sensors (Artech House sensors library), 2nd edition.*
10. *F. Boudoin, M. Lavabre, Sensors: main uses, Edition Casteilla, 2007.*
11. *J. G. Webster, Measurement, Instrumentation and Sensors Handbook, Taylor & Francis Ltd*

Semester :
Teaching unit: UED

Subject 4: Embedded Systems Operating Systems
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

Allow students to learn the fundamentals of operating systems while studying how to put them into practice in an embedded system such as Android.

Recommended prior knowledge:

Basic notions in mathematics, algorithms and programming.

Content of the material:

Chapter 1. **(4 weeks)**
General presentation of operating systems and technical elements (e.g. OS, Android, Windows and Linux)

Chapter 2. **(3 weeks)**
Process Management

Chapter 3. **(3 weeks)**
Memory Management

Chapter 4. **(3 weeks)**
File Management

Chapter 5. **(2 weeks)**
Executable

Method of evaluation:

Review: 100%

Bibliographical references:

1. *Andrew Tanenbaum, Operating Systems, Pearson Publishing.*
2. *Michael Griffiths, Michel Vayssade, Architecture of operating systems, Hermès - Lavoisier*.
3. *P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer TinyOS: An Operating System for Sensor Networks in Ambient Intelligence, p. 115-148, Springer, 2005.*
4. *P. Levis, D. Gay, TinyOS Programming, Cambridge University Press, 2009.*
5. *TinyOS Open Technology Alliance :*
   http://www.cs.berkeley.edu/~culler/tinyos/ alliance/overview.pdf
6. *www.contiki-os.org/support.html*

Semester :
Teaching unit: UED
Material 5: Smart cards
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

Acquire sufficient technical knowledge of the technology, operation and use of smart cards with a view to its implementation in projects on embedded electronic systems.

Recommended prior knowledge:

Architecture of microcontroller and/or microprocessor systems.

Content of the material:

- General, History, Applications and Markets of the Smart Card.
- Semiconductors for smart cards, Technologies, Wired logic components, Microcomputers.
- Cryptology and Security, Principles of Cryptography, Symmetric Crypto Systems, Asymmetric Crypto Systems, Zero-Knowledge Systems, Physical and Logical Security of Smart Cards.
- Construction principles, Interconnection of components, Inserting, Connectors.
- Smart Card Operating Systems, General and Basic Mechanisms, Closed Operating Systems, Open Operating Systems.
- Contact communication, Radio frequency communication.

Method of evaluation:

Review: 100%

Bibliographical references:

1. *W.Rankl and W. Effing, Smart Card Handbook, Wiley, 2010.*
2. *C. Tavernier, Les cartes à puce, Dunod, 2011.*

Semester :
Teaching unit: UED
Subject 6: Mobile robotics
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

The objective of this subject is to give students the necessary foundations to know how to develop and design applications of embedded electronics in the service of mobile robotics, which is a discipline in its own right aimed at mastering movement.

Recommended prior knowledge:

Mathematics, Programming.

Content of the material:

Chapter 1: Classification and Modeling of Mobile Robots (R-M)	**(2 weeks)**
Chapter 2: Sensors used in R&M	**(3 weeks)**
Chapter 3: The location of R-M	**(2 weeks)**
Chapter 4: The Representation of the Environment of an R-M	**(2 weeks)**
Chapter 5: Trajectory planning techniques	**(2 weeks)**
Chapter 6: Navigation and SLAM techniques	**(2 weeks)**
Chapter 7: Humanoid robots	**(2 weeks)**

Method of evaluation:

Review: 100%

Bibliographical references:

1. *R. Siegwart, I.R. Nourbakhch, D. Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd Edition, MIT Press, 2011.*
2. *L. Jaulin, Mobile Robotics, Editions ISTE, 2015.*
3. *V. Maille, C. Accard, B. Breton, Robots: learning robotics through example, Editions Ellipse, 2016.*

Semester :
Teaching unit: UED
Subject 7: Wireless communication
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

To allow students to become familiar with wireless communication systems in order to use them in applications in embedded systems electronics. Among the various wireless networks, we will focus more specifically on the WIFI network of the IEEE 802.11 standard.

Recommended prior knowledge:

Basics in mathematics, statistics and signal processing.

Content of the material:

Chapter 1. Wireless networks **(4 weeks)**

Chapter 2. Presentation of WIFI (802.11): **(3 weeks)**
Different WIFI standards and equipment

Chapter 3. Implementing WIFI: **(3 weeks)**
Infrastructure mode, ad hoc modeand the establishment of a network

Chapter 4. Encryption: **(3 weeks)**
WEP, WAP ...

Chapter 5. Hacks and solutions: **(2 weeks)**
MAC address filtering, IP address definitions, and firewall installation

Method of evaluation:

Review: 100%

Bibliographical references:

1. *Tanenbaum, Networks, 4th edition, Prentice Hall, 2003.*
2. *R. Parfait, Les réseaux de télécommunications, Hermès science publications, 2002.*
3. *E. Hollocou, Techniques et réseaux de télécommunications, Armand Colin, 1991.*
4. *C. Servin, Réseaux et télécoms, Dunod, Paris, 2006.*
5. *D. Dromard and D. Seret, Network Architectures, Pearsont, 2009.*
6. *P. Polin, Les réseaux: Principes fondamentals, Edition Hermès.*
7. *D. Comer, TCP/IP, architectures, protocols and applications, Editions Interéditions.*
8. *D. Présent, S. Lohier, Transmissions et Réseaux, cours et exercices correctes, Dunod.*
9. *P. Clerc, P. Xavier, Principes fondamentals des Télécommunications, Ellipses, Paris, 1998.*
10. *D. Battu, Initiation aux Télécoms: Technologies et Applications, Dunod, Paris, 2002.*
11. *P. Rolin, G. Martineau, L. Toutain, A. Leroy, Les réseaux, principes fondamentals, éditions Hermès, 1997.*

Semester :
Teaching unit: UED
Subject 8: Robotics
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

To introduce the student to the fundamental aspects of robotics and to recent developments in the field of industrial robotics.

Recommended prior knowledge:

None.

Content of the material:

Chapter 1: General

Definitions, Constituents of a robot, Classification of robots, Characteristics of a robot, Generations of robots, Programming of robots.

Chapter 2: Degree of Freedom - Architecture

Positioning of a solid in space, Liaison, Mechanisms, Morphology of robots, manipulators

Chapter 3: Geometric Model of a Simple Chain Robot

Need for a Model, Operational Coordinates, Translation and Rotation, Homogeneous Transformation Matrices, Obtaining the Geometric Model, Modified Denavit-Hartenberg Parameters, Inversion of the Geometric Model - Paul's Method, Multiple Solutions - Workspace - Aspects

Chapter 4: Simplification Technique

Speed and acceleration of robots, Jacobean matrix and its usefulness, Definition of equations in direct and inverse, Meaning of singularities.

Method of evaluation:

Review: 100%

Bibliographical references:

1. *H. Asada, J.J.E. Slotine, Robot Analysis and Control, a Wiley Interscience Publication, 1986.*
2. *J.J. Craig, Introduction to Robotics, Mechanics and Control, Addison-Wesley, 1989.*

Semester :
Teaching unit: UED
Subject 9: Renewable energies: solar photovoltaics
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

This subject covers concepts relating to non-polluting renewable energies, photovoltaic (PV) devices, PV conversion, solar cell manufacturing processes, PV module assemblies, their degradation, etc.. It also addresses auxiliary systems: the battery, the fuel cell (with hydrogen as an energy carrier), converters, etc. The subject will also focus on the different loads to be supplied continuously or alternately by looking for all the possibilities of coupling with a PV generator, the description of a global PV system, its characteristics and the optimization of the operation of the system.

Recommended prior knowledge:

Notions about semiconductors, radiation physics, mathematics, electronics...

Content of the material:

Chapter 1: Renewable energies
Forms of energy, What is renewable energy, Main renewable energies, The world energy situation, ...

Chapter 2: The Solar Source
Solar radiation, Solar deposits, Solar energy (thermal, photovoltaic, thermodynamic)

Chapter 3: The photovoltaic source
Photovoltaic conversion, Solar cell technology, Properties of solar cells, Modeling of a photovoltaic cell (module) (electrical, thermal modeling...), Conversion efficiency, form factor..., Different connections (series, parallel, mixed), Impact of various factors on electrical characteristics, Degradation, Protections of photovoltaic modules, Photovoltaic energy applications (pumping, grid connection, ...).

Chapter 4: Photovoltaic Systems
Direct connection of photovoltaic generator – load, Storage (Battery), Fuel cell, Chopper, Inverter, Study of an example of a global system (hybridization), Sizing problem of a photovoltaic installation, *Maximum Power Point Tracker* (MPPT).

Method of evaluation:

Review: 100%

Bibliographical references:

1. *A. Vapaille, Semiconductor Devices and Integrated Circuits, Dunod, 1987.*
2. *M. Orgeret, les piles solaires, Masson, 1985.*
3. *A. Ricaud, Photopiles solaires, Presses polytechniques et universitaires romandes, 1997.*
4. *E.Lorenzo, G. Araflio, Solar Electricity - Engineering of Photovoltaic Systems.*
5. *Minano, R. Zilles, Stand alone photovoltaic Applications, JAMES & JAMES 1994.*
6. *B. Multon, Production of electrical energy by renewable sources, Techniques de l'Ingénieur, Traité de Génie Electrique, D4005/6, May 2003.*
7. *J. Nelson, The physics of solar cells, Imperial College Press.*
8. *A. Labouret, P. Cumune, Solar Cells, 5th edition - The basics of photovoltaic energy, Dunod, 2010*
9. *A. Labouret, Photovoltaic Solar Energy, 3rd edition, Dunod, 2006.*
10. *Deambi, Suneel, Photovoltaic System Design: Procedures, Tools and Applications, CRC Press, 2016.*

11. *O. Isabella, K. Jäger, A. Smets, R. Van Swaaij, MiroZeman, Solar Energy: The Physics and Engineering of Photovoltaic Conversion, Technologies and Systems, UIT Cambridge Ltd, 2016.*
12. *Gottfried H. Bauer, Lecture Notes in Physics 901, Photovoltaic Solar Energy Conversion, Springer-Verlag Berlin Heidelberg, 2015.*
13. *www.pveducation.org*
14. *http://www.cythelia.fr/nos-documents/*
15. *http://www.solems.com/depots-de-couches-minces*

Semester :
Teaching unit: UED
Subject 10: Autonomous energy systems
VHS: 10:30 p.m. (Lecture: 1h30)
Credit: 1
Coefficient : 1

Teaching objectives:

To arouse the student's interest in renewable energies in general and in energy systems exploiting solar or wind energy in particular. To help the student acquire a certain skill in the sizing of a wind or photovoltaic installation.

Recommended prior knowledge:

semiconductors, radiation physics, mathematics, electronics...

Content of the material:

Chapter 1: Electrical Power Generation Devices

Notions on energy transformations (mechanical, thermal, hydraulic, etc.), history (Volta, Oersted, Faraday, etc.), the alternator, the dynamo, the methods of producing electrical energy (hydraulic power plant, thermal power plants). Non-renewable energy sources (fossil and nuclear). Renewable energy sources.

Chapter 2: Wind Energy

History, principle and structure, Characteristics and sizing, Map of the wind farm in Algeria, Wind farms and power, Standards, Advantages and disadvantages. Example of a wind power installation.

Chapter 3: Hybrid Systems

Hybrid Systems (Tidal turbine, Principle of operation of the tidal turbine, The different types of tidal turbines and the operators,...)

Chapter 4: Photovoltaic solar energy

Principle of a photovoltaic installation, the solar deposit in Algeria, Photovoltaic cell technologies, Photovoltaic modules, MPPT, Photovoltaic characteristics and connectors, Standards. The inverter (role, principle, characteristics and efficiency). Example of a photovoltaic installation.

Chapter 5: Other Renewable Energy Sources

Renewable energy families (solar energy, wind energy, hydropower, biomass, geothermal energy). The different renewable energies in the world. Cost-effectiveness.

Method of evaluation:

Review: 100%

Bibliographical references:

1. *J. Vernier, Renewable Energies, PUF edition, 2012*
2. *E. Riolet, Le mini-éolienne, Eyrolles edition, 2010*
3. *A. Labouret and M. Villoz, Photovoltaic Solar Energy, Editions du Moniteur 2009*
4. *B. Fox, Wind Electric Energy: Production, Forecasting and Grid Integration, Technical and Engineering Collection, Dunod/L'Usine Nouvelle 2015 (2nd edition)*
5. *A. Damien, Biomass Energy: Definitions, Resources and Modes of Transformation, Technical and Engineering Collection, Dunod/L'Usine Nouvelle 2013 (2nd edition)*
6. *A. Labouret, M. Villoz, Photovoltaic installations: Design and sizing of installations connected to the grid, Technical and Engineering Collection, Dunod/Le Moniteur 2012 (5th edition)*

7. *http://www.cder.dz/spip.php?article1442*

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University...
Faculty...
Department...

Sector:..
Specialty:...
Academic year 20.... /20....

The............../.........../20....

MINUTES CONCERNING THE CHOICE OF DISCOVERY SUBJECTS IN THE FIRST YEAR OF THE MASTER'S DEGREE

The undersigned teachers, after deliberation, have decided on the choice of the subjects of discovery of the master's degree ..... proposed in the framework of this master's degree. In this regard, the following teachers* undertake to ensure the teaching of these subjects. In the event that the training team chooses a subject whose program is not available in the framework, the head of the course undertakes to send this program to the CPND-ST for enrichment and validation.

* For each subject, it is possible to indicate the name of the head teacher and possibly the name of a substitute teacher.

Semesters	Discovery Matters	Teachers
S1	Material 1: Material 2:	
S2	Material 1: Material 2:	

Observations :...
....

Teachers' first and last names		Subjects taught	Semester	Signatures
1				
2				
3				
4				
5				
6				
7				
8				
9				

The head of the sector The head of the department

***Reminders**: The nature of the discovery subjects must complement the training and must be chosen according to the needs of the local or regional socio-economic fabric and the availability of teachers specializing in the field.*

Copies to sales representatives/VDPs

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University..
Faculty...
Department...

Sector:..
Specialty:...
Academic year 20.... /20....

The............../.........../20....

MINUTES CONCERNING THE CHOICE OF DISCOVERY SUBJECTS FOR THE SECOND YEAR OF THE MASTER'S DEGREE

The undersigned teachers, after deliberation, have decided on the choice of the subjects of discovery of the master's degree ..... proposed in the framework of this master's degree. In this regard, the following teachers* undertake to ensure the teaching of these subjects. In the event that the training team chooses a subject whose program is not available in the framework, the head of the course undertakes to send this program to the CPND-ST for enrichment and validation.

* For each subject, it is possible to indicate the name of the head teacher and possibly the name of a substitute teacher.

Semesters	**Discovery Matters**	**Teachers**
S3	Material 1: Material 2:	

Observations :..
....

Teachers' first and last names		**Subjects taught**	**Semester**	**Signatures**
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

The head of the sector   The head of the
department

***Reminders**: The nature of the discovery subjects must complement the training and must be chosen according to the needs of the local or regional socio-economic fabric and the availability of teachers specializing in the field.*

Copies to sales representatives/VDPs