الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne
Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement
Supérieur
et de la Recherche Scientifique

National Higher
School of
Autonomous
Systems
Technology

STATE ENGINEER TRAINING PROGRAM

Academic year: 2025/2026

Institution
National Higher School of Autonomous Systems Technology

Domain	Field	Specialty
Science and Technology	*Electronics*	*Autonomous embedded systems engineering*

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

اللجنة البيداغوجية الوطنية لميدان العلوم و التكنولوجيا
Comité Pédagogique National du Domaine Sciences et Technologies

عرض تكوين
مهندس دولة

السنة الجامعية: 2026/2025

المؤسسة
المدرسة الوطنية العليا لتكنولوجيا الأنظمة المستقلة

الميدان	الفرع	التخصص
علوم وتكنولوجيا	الإلكترونيك	هندسة الأنظمة المدمجة المستقلة

Table of contents

Copy of order no. 062of 22 January 2024

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

22 جانفي 2024

قرار رقم 062 المؤرّخ في
يحدد البرنامج البيداغوجي للتكوين القاعدي
لنيل شهادة مهندس دولة
بالمدرسة الوطنية العليا لتكنولوجيا الأنظمة المستقلة

إن وزير التعليم العالي والبحث العلمي،

- بمقتضى القانون رقم 99-05 المؤرّخ في 18 ذي الحجة عام 1419 الموافق 4 أبريل سنة 1999 والمتضمّن القانون التوجيهي للتعليم العالي، المعدّل والمتمّم،
- وبمقتضى المرسوم الرئاسي رقم 23-119 المؤرخ في 23 شعبان عام 1444 الموافق 16 مارس سنة 2023 والمتضمن تعيين أعضاء الحكومة، المعدل،
- وبمقتضى المرسوم التنفيذي رقم 13-77 المؤرّخ في 18 ربيع الأوّل عام 1434 الموافق 30 يناير سنة 2013 الذي يحدد صلاحيات وزير التعليم العالي والبحث العلمي،
- وبمقتضى المرسوم التنفيذي رقم 16-176 المؤرّخ في 9 رمضان عام 1437 الموافق 14 يونيو سنة 2016 الذي يحدد القانون الأساسي النموذجي للمدرسة العليا،
- وبمقتضى المرسوم التّنفيذي رقم 22-208 المؤرّخ في 5 ذي القعدة عام 1443 الموافق 5 يونيو سنة 2022 الذي يحدد نظام الدّراسات والتكوين للحصول على شهادات التعليم العالي،
- وبمقتضى المرسوم التّنفيذي رقم 23-434 مؤرخ في 18 جمادى الأول عام 1445 الموافق 2 ديسمبر 2023 يتضمن إنشاء مدرسة وطنية عليا لتكنولوجيا الأنظمة المستقلة،
- وبمقتضى القرار رقم 1564 المؤرخ في 05 أكتوبر سنة 2016 والمتضمن إنشاء اللجنة البيداغوجية الوطنية للمدارس العليا حسب الميدان ويحدد صلاحيتها وتشكيلتها وتنظيمها وسيرها،

يقـــــــرر:

المادة الأولـى: يحدّد البرنامج البيداغوجي للتكوين القاعدي لنيل شهادة مهندس دولة بالمدرسة الوطنية العليا لتكنولوجيا الأنظمة المستقلة، طبقا لملحق هذا القرار.

المـادة 2: تطبق أحكام هذا القرار على دفعات الطلبة المسجلين بالمدرسة الوطنية العليا لتكنولوجيا الأنظمة المستقلة، ابتداء من السنة الجامعية 2023-2024.

المـادة 3: يكلف كل من السيد المدير العام للتعليم والتكوين ومدير المدرسة الوطنية العليا لتكنولوجيا الأنظمة المستقلة، كلّ فيما يخصّه، بتطبيق هذا القرار الذي ينشر في النشرة الرسمية للتعليم العالي والبحث العلمي.

22 جانفي 2024
حـرر بالـجـزائـر في : ....

ع/وزير التعليم العالي والبحث العلمي

الأمين العام
عبد الحكيم بن تليس

I. Training Context and Objectives

1- Background and Strategic Vision of the Training program

The National Higher School of Autonomous Systems Technology offers a three-year engineering program in Autonomous Embedded Systems for students who have completed their preparatory cycle. This initiative aligns with the institution's mission to develop modern, rigorous engineering programs that respond to current technological needs.

The curriculum was designed in collaboration with academic and industrial partners to ensure both theoretical relevance and practical application. It integrates fundamental electronics principles with specialized knowledge in autonomous systems creating a comprehensive educational framework that prepares graduates for diverse professional paths.

This program is part of a coherent academic structure that includes complementary specializations in autonomous systems technologies:

- Autonomous Embedded Systems Engineering (present program)
- Unmanned Systems Navigation and Control
- Robotics and Autonomous Systems Design

The educational approach emphasizes balanced development of theoretical understanding and practical skills through laboratory work, project-based learning, and industrial internships. The program directly supports Algeria's strategic objectives for technological advancement and economic diversification while maintaining alignment with international standards.

2- Admission Requirements

Candidates for this engineering program are students from the National Higher School of Autonomous Systems Technology who have completed their preparatory cycle. Selection is based on merit ranking according to the overall average of the two preparatory years, subject to available places. Admission to the Autonomous Embedded Systems specialty follows the institution's internal progression protocols.

3- Program Objectives

The **Autonomous Embedded Systems Engineering (AESE)** specialty is designed to develop highly skilled engineers capable of conceiving, designing, and implementing sophisticated embedded systems with autonomous capabilities. This program addresses the growing demand for intelligent systems that can operate independently across various domains including transportation, healthcare, industrial automation, and smart infrastructure.

Our primary objective is to cultivate engineers with deep technical expertise in both hardware and software aspects of embedded systems, complemented by specialized knowledge in autonomy-enabling technologies. Graduates will master the integration of microcontrollers, sensors, actuators, and communication protocols with advanced algorithms for signal processing, control systems, artificial intelligence, and data fusion.

The curriculum balances theoretical foundations with practical implementation skills, preparing engineers to develop systems that can perceive their environment, make intelligent decisions, and execute actions with minimal human intervention. Particular emphasis is placed on critical aspects such as real-time processing, power efficiency, system reliability, and security—all essential for autonomous operations in complex environments.

By completing this specialty, engineers will be equipped to tackle emerging challenges in fields such as robotics, Internet of Things (IoT), smart grids, autonomous vehicles, and industrial automation. They will possess the technical versatility and innovative mindset needed to lead the development of next-generation autonomous systems that enhance productivity, safety, and quality of life across multiple sectors of the economy.

4- Targeted skills and Profiles

The Autonomous Embedded Systems Engineering (AESE) specialty aims to develop engineers with comprehensive technical expertise across both hardware and software domains, complemented by specialized knowledge in autonomy technologies. Our graduates will possess the skills necessary to innovate and lead in the rapidly evolving field of autonomous embedded systems.

Targeted Profiles:

- Embedded Systems Architect: These professionals excel at designing complete autonomous embedded solutions, integrating hardware components with sophisticated software layers. They understand system constraints and can architect solutions that balance performance, power consumption, and reliability requirements.
- Embedded Software Engineer: Specialists who develop optimized code for resource-constrained environments, with expertise in real-time operating systems, device drivers, and middleware. They implement robust algorithms that enable autonomous decision-making while maintaining deterministic behavior.
- Hardware Integration Specialist: Engineers skilled in electronic circuit design, sensor integration, and actuator control systems. They possess the technical knowledge to select, interface, and optimize the physical components required for autonomous functionality.

- Intelligent Systems Developer: Professionals who implement machine learning, computer vision, and data fusion algorithms on embedded platforms. They bridge the gap between theoretical AI approaches and practical embedded implementations with limited resources.
- Embedded Security Expert: Engineers focused on protecting autonomous systems from vulnerabilities, implementing secure communication protocols, and ensuring system integrity through hardware and software safeguards.
- Decision Autonomy Specialist: Engineers who design and implement advanced decision-making frameworks that allow embedded systems to operate independently in complex, uncertain environments. They develop algorithms that enable systems to assess situations, prioritize objectives, and adapt strategies without human intervention.
- Energy Autonomy Engineer: Professionals who specialize in optimizing power management for embedded systems, designing energy harvesting solutions, and implementing intelligent power consumption strategies. They ensure systems can maintain operational longevity through efficient energy utilization or self-sufficiency in diverse deployment scenarios.

These competencies are developed throughout the curriculum through:

- Foundational Lectureses in electronics, microcontrollers, and sensor technologies that build hardware expertise
- Software development modules covering embedded operating systems, real-time programming, and system optimization
- Advanced Lectureses in control systems, artificial intelligence, and data processing that enable autonomous capabilities
- Specialized training in decision-making algorithms and energy management systems
- Practical laboratory work and project-based learning that reinforce theoretical concepts with hands-on implementation
- Internships and capstone projects that provide exposure to real-world challenges and industry practices

Graduates will possess both technical depth and breadth, allowing them to contribute immediately in specialized roles while having the versatility to adapt to emerging technologies and cross-disciplinary challenges in sectors including transportation, industrial automation, healthcare systems, and smart infrastructure.

5- Regional and national employability potential

The **Autonomous Embedded Systems Engineering (AESE)** specialty addresses critical workforce needs across multiple strategic economic sectors in Algeria and the broader North African region and elsewhere. Graduates of this program will be positioned to meet growing demand across several key industry segments that are experiencing rapid technological transformation.

Algeria's economic diversification initiatives, particularly in the technology and industrial sectors, create substantial opportunities for specialists in autonomous embedded systems. The strategic national plans to strengthen digital infrastructure, expand industrial automation, and develop smart city capabilities necessitate engineers with precisely the skill set this program cultivates.

The employability potential for AESE graduates extends across several vital economic sectors:

- **Energy Sector Transformation** The ongoing modernization of Algeria's energy infrastructure—encompassing oil and gas operations, renewable energy integration, and smart grid development—requires embedded systems expertise. AESE graduates will contribute to developing monitoring systems, automated control solutions, and energy management platforms that enhance efficiency and sustainability in this foundational sector of the national economy.
- **Industrial Modernization and Automation** As Algerian manufacturing and processing industries implement Industry 4.0 principles, the demand for engineers who can design and deploy autonomous embedded systems continues to grow. Graduates will find opportunities in developing industrial control systems, automation solutions, and predictive maintenance platforms that increase productivity and competitiveness in national industrial operations.
- **Transportation and Logistics Enhancement** The expansion of intelligent transportation systems and logistics optimization represents another significant employment avenue. AESE engineers will contribute to traffic management systems, fleet tracking solutions, and autonomous vehicle technologies that improve the efficiency and safety of national transportation networks.
- **Agricultural Technology Advancement** Agricultural modernization initiatives create demand for precision farming technologies and monitoring systems. Graduates will find opportunities to develop autonomous irrigation controls, crop-monitoring platforms, and farm management systems that enhance productivity while conserving resources.
- **Healthcare Technology Development** The healthcare sector's digital transformation creates opportunities for embedded systems specialists to contribute to medical device development, patient monitoring systems, and healthcare logistics automation—all crucial elements in strengthening national healthcare delivery capabilities.
- **Telecommunications and Digital Infrastructure** The ongoing expansion of telecommunications networks and IoT infrastructure throughout Algeria and neighboring

countries requires engineers who understand embedded systems integration with network technologies. AESE graduates will support the development of the backbone systems needed for smart city initiatives and digital service expansion.

Beyond direct employment in these sectors, graduates will find opportunities with:

- National research centers and innovation hubs focused on autonomous systems
- Technology startups within Algeria's growing entrepreneurial ecosystem
- Multinational technology firms operating in the region
- Government agencies involved in digital transformation and industrial development
- Academic institutions supporting technological advancement through research and development

The cross-disciplinary skills developed in the AESE program are particularly valuable as Algeria continues to develop its innovation capacity and digital economy. By combining expertise in electronics, software development, artificial intelligence, and system integration, graduates will possess versatile capabilities that remain relevant as technologies evolve, ensuring sustainable career development and contributing to national technological sovereignty.

6- Bridges to other specialties

The Autonomous Embedded Systems Engineer specialty offers strong connectivity with related disciplines, facilitating academic mobility and career development. Students may request transfers to other electronics specializations within Algerian universities.

Our program's comprehensive curriculum creates multiple pathways for graduates. Those wishing to pursue academic careers can seamlessly transition into research teams across various domains including robotics, artificial intelligence, IoT systems, and advanced electronics. The foundational knowledge and specialized skills acquired throughout this program prepare students for both industry leadership and scholarly advancement in emerging technological fields in Algeria and beyond.

The interdisciplinary nature of autonomous embedded systems naturally creates bridges to specializations in telecommunications, control systems, computer engineering, and power electronics, allowing graduates to pivot their careers according to technological evolution and personal interest.

7- Program monitoring indicators

The Autonomous Embedded Systems Engineering program employs a comprehensive set of key performance indicators to ensure continuous quality improvement and alignment with industry needs. These metrics enable systematic monitoring of educational outcomes and program effectiveness across multiple dimensions:

Academic Performance Metrics:

- Application rates and selectivity ratios for incoming cohorts
- Program completion rates and graduation statistics
- Average time to graduation and credit accumulation patterns
- Academic performance distribution across core and specialized Lectures

Professional Integration Assessment:

- Employment rate of graduates within six months and one year of program completion
- Sectorial distribution of graduate employment across industries
- Percentage of graduates securing positions in their field of specialization
- Career progression trajectories of alumni over three and five-year periods

Research and Academic Advancement:

- Number of graduates pursuing doctoral studies
- Research publications stemming from capstone projects
- Participation rates in academic conferences and technical competitions
- Intellectual property development from student projects

Stakeholder Feedback Mechanisms:

- Structured evaluations from current students at mid-program and completion stages
- Alumni satisfaction surveys conducted at one and three years post-graduation
- Employer assessments of graduate preparedness and performance
- Industry partner feedback on curriculum relevance and emerging skill requirements

Partnership Development Indicators:

- Growth in industry partnerships for internships and capstone projects
- Expansion of research collaborations with national and international institutions
- Development of exchange opportunities with partner schools
- Engagement levels with the Algerian innovation ecosystem

The program administration conducts annual reviews of these indicators, enabling data-driven refinement of curriculum content, teaching methodologies, and industry alignment to ensure graduates remain at the forefront of autonomous systems engineering practice in Algeria and beyond.

8- Human resources available

a. Enrollment Capacity

The AESE enrollment capacity is 50 students per promotion.

b. Academic Supervision Team

- **Internal Supervision**

Name	Speciality	Rank	Intervention Type
Mourad ADNANE	Instrumentation / Signal Processing	Prof	Course/Tutorials/Lab
Faiza BOUMEDIENE	Mechanics / Robotics	Prof	Course/Tutorials/Lab
Ziane KECHIDI	Physics	Prof	Course/Tutorials/Lab
Ouassila HIOUAL	Informatics	Prof	Course/Tutorials/Lab
Kamel BOUDJIT	Electronics / Embedded Systems	MCA	Course/Tutorials/Lab
Salaheddine AOUDJ	Chemistry / Electrochemistry	MCA	Course/Tutorials/Lab
Tarek CHERIFI	Electronics / Embedded Systems	MCA	Course/Tutorials/Lab
Abdelkader HAMTTAT	Mathematics	MCA	Course/Tutorials/Lab
El Mahdi MEDIA	Physics	MCA	Course/Tutorials/Lab
Yasmine GUERBAI	Electronics / Artificial intelligence	MCA	Course/Tutorials/Lab
SAIB Abdessadek	Mathematics	MCA	Course/Tutorials/Lab
Leila ABBAD	Electronics / Networks	MCB	Course/Tutorials/Lab
Sarah CHENCHE	Computer sciences / Artificial intelligence	MCB	Course/Tutorials/Lab
Mohamed LAIDI	Statistics / Stochastic processes	MCB	Course/Tutorials/Lab
Halima LAMMARI	Mechanics / Robotics	MCB	Course/Tutorials/Lab
Ahmed ZEGLAOUI	Mathematics / Optimization	MCB	Course/Tutorials/Lab
Said REZIG	Mathematics	MAA	Course/Tutorials/Lab
Fatima KADIK	Mathematics	MAA	Course/Tutorials/Lab
ZAIR Mustapha	Informatics	MAA	Course/Tutorials/Lab

- **External Supervision**

Name	Specialty	Institution	Rank	Intervention Type
Moufid MANSOUR	Control Eng	USTHB	Prof	Course Responsible
Mohamed TADJINE	Control Eng / Robotics	ENP	Prof	Course Responsible
Abdelhalim ZAOUI	Electrical Eng	ENSTA	Prof	Course Responsible
Malika BOUTERFAS	Electronics / Sensors	ENSTA	MCA	Course Responsible
Tarek CHERIFI	Electronics / Embedded systems	ENSTA	MCA	Course Responsible
Souhila BOUTARFA	- Electronics/ Signal processing	ENSTA	MCB	Course Responsible
Nesrine ISSAAD	Bioinformatics / Artificial intelligence	U. Algiers	MCB	Course Responsible
kheira Lakhdari	Telecommunication / Artificial intelligence	ENSTA	MCB	Course Responsible
Fouad YACEF	Control Eng / Drones	CDTA	Dr	Course Responsible
Rabie Riadh BENREZKI	Navigation and Control	CDTA	Dr	Course Responsible
Abdellah KHELLOUFI	Navigation and Control	CDTA	Dr	Course Responsible
Mohamed ZAOUCHE	Control Eng / UAS	Expert	Dr	Course Responsible
Abdelkader BELAHCENE	Computer Science / Open Source software	ENSTA	MAA	Course Responsible

- **Comprehensive Summary of Human Resources**

Rank	Internal resource	External resource	Total
Professor	4	3	7
MCA	7	2	9
MCB	5	7	12

- **Permanent Support Staff**

Rank	Staff
Laboratory Engineer	1
Computer Engineer	1
Administrator	4
Administrative Assistant	2
Total	8

9- Available Material Resources

a. Educational Facilities

Location	Seating Capacity	Number	Total Capacity
Lecture Hall	200	4	800
Tutorial Room	30	15	450
Practical Lab	25	15	375
Library	30	1	30
Reading Rooms	40	2	80
Workshop	10	1	10
Computing Center	40	1	40
Internet Room	40	2	80

b. Educational Laboratories and Equipment

The National Higher School of Autonomous Systems Technology prioritizes hands-on learning experiences through well-equipped laboratory facilities. These educational environments are designed to support the practical application of theoretical concepts across all specializations offered by the institution.

As a newly established educational institution, the school is currently in the process of acquiring high-quality pedagogical equipment and materials to outfit its laboratories. This ongoing procurement initiative is strategically aligned with the specific requirements of each specialization and the overall educational objectives of the institution.

The equipment acquisition plan has been carefully developed to ensure that all laboratories will be furnished with state-of-the-art technology that meets international standards for engineering education. This deliberate approach to laboratory development reflects the institution's commitment to providing students with relevant, industry-aligned practical experiences that enhance their technical competencies and professional readiness.

Upon completion, these specialized laboratories will create comprehensive learning environments where students can effectively bridge theoretical knowledge with practical application, developing the hands-on skills essential for success in the autonomous systems sector.

- **Laboratory Name: Electricity Lab**

Student Capacity: 25

N°	Equipment	Quantity	Operational Status
1	Amperemeter	13	Excellent state
2	Galvanometer	02	Excellent state
3	Voltmeter	12	Excellent state
4	Multimeter	05	Excellent state
5	Wattmeter	07	Excellent state
6	Power supply	10	Excellent state
7	Power supply module	06	Excellent state
8	Low-Frequency Generator	06	Excellent state
9	Oscilloscope	10	Excellent state
10	Breadboards	20	Excellent state

- **Laboratory Name: Physics Lab 1**

Student Capacity: 25

N°	Equipment	Quantity	Operational Status
1	Force tables	01	Excellent state
2	Free Fall	04	Excellent state
3	Simple Pendulum	05	Excellent state
4	Air track	03	Excellent state
5	PasCars	02	Excellent state
6	Torsion pendulum	02	Excellent state
7	Balance of Coulomb	05	Excellent state
8	electric field strengthmeter	05	Excellent state
9	Tank Rheographic	04	Excellent state
10	Analogic voltmeter	07	Excellent state
111	Analogic Ammeter	07	Excellent state

- **Laboratory Name: physics Lab 2**

Student Capacity: 25

N°	Equipment	Quantity	Operational Status
01	Pohl's Pendulum	1	Excellent state
02	Wave Tank/Ripple Tank	1	Excellent state
03	Vibrating String/Cord	1	Excellent state
04	Kundt's Tube	1	Excellent state
05	Light Polarization (Equipment)	1	Excellent state
06	Light Diffraction (Equipment)	1	Excellent state

- **Laboratory Name: Chemistry Laboratory.**

Student Capacity: 25

N°	Equipment	Quantity	Operational Status
01	Fume hood	01	Excellent state
02	Oven/Incubator	01	Excellent state
03	Distiller	01	Excellent state
04	Ice generator	01	Excellent state
05	Calorimeter	04	Excellent state
06	Heating plates	04	Excellent state
07	Heater with stirrer	01	Excellent state
08	Scale/Balance	02	Excellent state
09	pH meter	04	Excellent state
10	Dosing equipment with pH meter	02	Excellent state
11	Thermometer	15	Excellent state

12	Stopwatch/Timer	06	Excellent state
13	Power supplies for calorimetry	02	Excellent state
14	Voltmeter for calorimetry	02	Excellent state
15	Ammeter for calorimeter	02	Excellent state

- **Laboratory Title: Fluid Mechanics.**

Student Capacity: 25

N°	Equipment	Quantity	Operational Status
01	Digital Hydraulic Bench	01	Excellent state
02	Center of Pressure Apparatus	02	Excellent state
03	Falling Sphere Viscometers	02	Excellent state
04	Venturi Tube	01	Excellent state
05	Notch Discharge Apparatus	01	Excellent state

- **Laboratory Title: Mechanics Laboratory.**

Student Capacity: 25.

N°	Equipment	Quantity	Operational Status
01	Materials Testing Apparatus	01	Excellent state

c. Internship and Corporate Training Sites

Institution	Student Number	Training Duration
Mobilis	**20**	15 days
CRTI	**20**	15 days
CDTA	**20**	15 days
DeepMinds	**5**	15 days
Sonatrach	**20**	15 days
SEAAL	**10**	15 days
SONELGAZ	**10**	15 days

d. Personal Workspaces and Information Technology Resources

The institution provides students with a well-equipped library featuring a spacious reading room designed for academic study. High-speed internet connectivity is available throughout all campus facilities, ensuring continuous access to digital resources.

10- Signatures of teachers involved in the course

a. Internal teaching team for the specialty

Name	Speciality	Rank	Intervention Type	Signature
Mourad ADNANE	Instrumentation / Signal Processing	Prof	Course/Tutorials/Lab	
Faiza BOUMEDIENE	Mechanics / Robotics	Prof	Course/Tutorials/Lab	
Ziane KECHIDI	Physics	Prof	Course/Tutorials/Lab	
Ouassila HIOUAL	Computer Sciences	Prof	Course/Tutorials/Lab	
Kamel BOUDJIT	Electronics / Embedded Systems	MCA	Course/Tutorials/Lab	
Salaheddine AOUDJ	Chemistry / Electrochemistry	MCA	Course/Tutorials/Lab	
Tarek CHERIFI	Electronics / Embedded Systems	MCA	Course/Tutorials/Lab	
Abdelkader HAMTTAT	Mathematics	MCA	Course/Tutorials/Lab	
El mahdi MEDIA	Physics	MCA	Course/Tutorials/Lab	
Yasmine GUERBAI	Electronics / Artificial Intelligence	MCA	Course/Tutorials/Lab	
SAIB Abdessadek	Mathematics	MCA	Course/Tutorials/Lab	
Leila ABBAD	Electronics / Networking	MCB	Course/Tutorials/Lab	
Sarah CHENCHE	Computer Sciences/Artificial Intelligence	MCB	Course/Tutorials/Lab	
Mohamed LAIDI	Statistics / Stochastic processes	MCB	Course/Tutorials/Lab	
Halima LAMMARI	Mechanics / Robotics	MCB	Course/Tutorials/Lab	
Ahmed ZEGLAOUI	Mathematics / Optimization	MCB	Course/Tutorials/Lab	
Said REZIG	Mathematics	MAA	Course/Tutorials/Lab	
Fatiha KADIK	Mathematics	MAA	Course/Tutorials/Lab	
ZAIR Mustapha	Computer Sciences	MAA	Course/Tutorials/Lab	

b. External teaching team for the specialty

Name	Specialty	Institution	Rank	Intervention Type	Signature
Moufid MANSOUR	Control Eng	USTHB	Prof	Course Responsible	
Mohamed TADJINE	Control Eng / Robotics	ENP	Prof	Course Responsible	
Abdelhalim ZAOUI	Electrical Eng	ENSTA	Prof	Course Responsible	
Malika BOUTERFAS	Electronics / Sensors	ENSTA	MCA	Course Responsible	
Tarek CHERIFI	Electronics / Embedded systems	ENSTA	MCA	Course Responsible	
Souhila BOUTARFA	- Electronics/ Signal processing	ENSTA	MCB	Course Responsible	
Nesrine ISSAAD	Bioinformatics / Artificial intelligence	U. Algiers	MCB	Course Responsible	
kheira Lakhdari	Telecommunication / Artificial intelligence	ENSTA	MCB	Course Responsible	
Fouad YACEF	Control Eng / Drones	CDTA	Dr	Course Responsible	
Rabie Riadh BENREZKI	Navigation and Control	CDTA	Dr	Course Responsible	
Abdellah KHELLOUFI	Navigation and Control	CDTA	Dr	Course Responsible	
Mohamed ZAOUCHE	Control Eng / UAS	Expert	Dr	Course Responsible	
Abdelkader BELAHCENE	Computer Science / Open Source software	ENSTA	MAA	Course Responsible	

II. Semester-based distribution of modules

- **Semester 1**

Teaching Unit TU	Semester hourly volume (15 weeks / Semester)					Coefficients	Credits	Module average		
	Lectures (h)	Tutorials (h)	Laboratory Work(h)	Total (h)	Personal work (h)			Continuous assessment		Final exam
								Tests	Practical Work	
Core Teaching Unit (UEF)										
UEF1.1.1	**4h30**	**3h00**	**4h30**	**12h00**	**7h15**	**12**	**12**			
Fundamental Electronics	1h30	1h30	1h30	4h30	2h30	4	4	20%	20%	60%
Instrumentation and Sensors	1h30	1h30	1h30	4h30	2h30	4	4	20%	20%	60%
Microprocessors	1h30	0	1h30	3h00	2h15	4	4	20%	20%	60%
UEF1.1.2	**4h30**	**3h00**	**4h00**	**11h30**	**5h15**	**9**	**9**			
Introduction to Signal Processing	1h30	1h30	1h00	4h00	1h45	3	3	20%	20%	60%
Operating Systems Essentials	1h30	0	1h30	3h00	1h45	3	3	20%	20%	60%
Networking fundamentals	1h30	1h30	1h30	4h30	1h45	3	3	20%	20%	60%
Methodology Teaching Unit (UEM)										
UEM1.1.1	**3h00**	**1h30**	**2h00**	**6h30**	**3h30**	**6**	**6**			
Introduction to artificial intelligence	1h30	0	1h00	2h30	1h45	3	3	20%	20%	60%
Database Essentials for Embedded Systems	1h30	1h30	1h00	4h00	1h45	3	3	20%	20%	60%
Cross-disciplinary Teaching Unit (UET)										
UET1.1.1	**0**	**0**	**1h00**	**1h00**	**1h15**	**2**	**2**			
Reverse Engineering & PCB Design	0	0	1h00	1h00	1h15	2	2	100%		0%
Discovery Teaching Unit (UED)										
UED1.1.1	**1h00**	**0**	**0**	**1h00**	**0h45**	**1**	**1**			
Engineering Ethics and Safety	1h00	0	0	1h00	0h45	1	1	100%		0%
Total Semester S1	**13h00**	**7h30**	**11h30**	**32h00**	**18:00**	**30**	**30**			

- **Semester 2**

Teaching Unit (UE)	Semester hourly volume (15 weeks / Semester)					Coefficients	Credits	Module average		
								Continuous assessment		Final exam
	Lectures (h)	Tutorials (h)	Laboratory Work(h)	Total (h)	Personal work (h)			Tests	Practical Work	
Core Teaching Unit (UEF)										
UEF1.2.1	**4h30**	**1h30h**	**4h00**	**10h00**	**3h45**	**9**	**9**			
Electronics Functions	1h30	1h30	1h00	4h00	1h15	3	3	20%	20%	60%
Microcontrollers	1h30	0	1h30	3h00	1h15	3	3	20%	20%	60%
Advanced sensors and Actuators	1h30	0	1h30	3h00	1h15	3	3	20%	20%	60%
UEF1.2.2	**4h30**	**3h00**	**4h00**	**11h30**	**3h45**	**9**	**9**			
Stochastic Processes and Estimation	1h30	1h30	1h00	4h00	1h15	3	3	20%	20%	60%
Operating Systems for Programmers	1h30	0	1h30	3h00	1h15	3	3	20%	20%	60%
Advanced Networking	1h30	1h30	1h30	4h30	1h15	3	3	20%	20%	60%
Methodology Teaching Unit (UEM)										
UEM1.2.1	**4h30**	**1h30**	**2h30**	**8h30**	**3h30**	**9**	**9**			
Control Systems	1h30	1h30	1h00	4h00	1h15	3	3	20%	20%	60%
Machine Learning	1h30	0	1h30	3h00	1h15	3	3	20%	20%	60%
Regulations & Standards for Unmanned System	1h30	0	0	1h30	1h00	3	3	20%	20%	60%
Cross-disciplinary Teaching Unit (UET)										
UET1.2.1	**0**	**0**	**1h00**	**1h00**	**2h00**	**2**	**2**			
Capstone Project I	0	0	1h00	1h00	2h00	2	2	100%		0%
Discovery Teaching Unit (UED)										
UED1.2.1	**0**	**0**	**0**	**0**	**6h00**	**1**	**1**			
Training Internship	/	/	/	/	6h00	1	1	100%		0%
Total Semester S2	**13h30**	**6h00**	**11h30**	**31h00**	**19h00**	**30**	**30**			

- **Semester 3**

Teaching Unit (UE)	Semester hourly volume (15 weeks / Semester)					Coefficients	Credits	Module average		
	Lectures (h)	Tutorials (h)	Laboratory Work(h)	Total (h)	Personal work (h)			Continuous assessment		Final exam
								Tests	Practical Work	
Core Teaching Unit (UEF)										
UEF2.1.1	**6h00**	**1h30**	**4h30**	**12h00**	**6h15**	**11**	**11**			
Power Electronics	1h30	0h45	1h00	3h15	1h45	3	3	20%	20%	60%
Industrial Networks and Communication Protocols	1h30	0h45	1h30	3h45	1h45	3	3	20%	20%	60%
Digital Signal Processors	1h30	0	1h00	2h30	1h45	3	3	20%	20%	60%
Multi-Sensor Data Fusion	1h30	0	1h00	2h30	1h00	2	2	20%	20%	60%
UEF2.1.2	**4h30**	**0**	**3h30**	**8h00**	**4h30**	**8**	**8**			
Introduction to Real-Time Operating Systems	1h30	0	1h30	3h00	1h45	3	3	20%	20%	60%
Digital Filtering	1h30	0	1h00	2h30	1h45	3	3	20%	20%	60%
Human-Machine Interface for Embedded Systems	1h30	0	1h00	2h30	1h00	2	2	20%	20%	60%
Methodology Teaching Unit (UEM)										
UEM2.1.1	**4h30**	**3h00**	**3h30**	**11h00**	**4h45**	**8**	**8**			
Digital Control Systems	1h30	1h30	1h00	4h00	1h45	3	3	20%	20%	60%
Deep Learning	1h30	0	1h30	3h00	1h45	3	3	20%	20%	60%
Operations Research	1h30	1h30	1h00	4h00	1h15	2	2	20%	20%	60%
Cross-disciplinary Teaching Unit (UET)										
UET2.1.1	**1h00**	**0**	**0**	**1h00**	**1h00**	**2**	**2**			
Introduction to embedded systems security	1h00	0	0	1h00	1h00	2	2	100%		0%
Discovery Teaching Unit (UED)										
UED2.1.1	**2h00**	**0**	**0**	**2h00**	**0h30**	**1**	**1**			
Project Management for Engineers	1h00	0	0	1h00	0h30	1	1	100%		0%
Total Semester S3	**17h00**	**4h30**	**11h30**	**33h00**	**17h00**	**30**	**30**			

- **Semester 4**

Teaching Unit (UE)	Semester hourly volume (15 weeks / Semester)					Coefficients	Credits	Module average		
	Lectures (h)	Tutorials (h)	Laboratory Work(h)	Total (h)	Personal work (h)			Continuous assessment		Final exam
								Tests	Practical Work	
Core Teaching Unit (UEF)										
UEF2.2.1	**6h00**	**1h30**	**5h30**	**13h00**	**4h30**	**12**	**12**			
Internet of Things (IoT)	1h30	0h45	1h30	3h45	1h00	3	3	20%	20%	60%
Electric Machines	1h30	0h45	1h00	3h15	1h00	3	3	20%	20%	60%
FPGA and Hardware Design	1h30	0	1h30	3h00	1h15	3	3	20%	20%	60%
Advanced Real-Time Operating Systems	1h30	0	1h30	3h00	1h15	3	3	20%	20%	60%
UEF2.2.2	**4h30**	**1h30**	**3h30**	**9h30**	**2h45**	**8**	**8**			
Cryptography	1h30	0h45	1h30	3h45	1h00	3	3	20%	20%	60%
Embedded Vision and Intelligent Image Processing	1h30	0	1h00	2h30	0h45	2	2	20%	20%	60%
Regulation and Control	1h30	0h45	1h00	3h15	1h00	3	3	20%	20%	60%
Methodology Teaching Unit (UEM)										
UEM2.2.1	**3h00**	**1h30**	**3h30**	**8h00**	**3h15**	**8**	**8**			
Fundamentals of Robotics	1h30	0h45	1h00	3h15	1h15	3	3	20%	20%	60%
Wireless Communication Essentials	1h30	0h45	1h00	3h15	1h15	3	3	20%	20%	60%
Embedded AI	0	0	1h30	1h30	0h45	2	2	100%	0%	
Cross-disciplinary Teaching Unit (UET)										
UET2.2.1	**0**	**0**	**1h00**	**1h00**	**2h00**	**2**	**2**			
Capstone Project II	0	0	1h00	1h00	2h00	2	2	100%		0%
Discovery Teaching Unit (UED)										
UED2.2.1	**0**	**0**	**0**	**0**	**6h00**	**1**	**1**			
Training Internship II	/	/	/	/	6h00	1	1	100%		0%
Total Semester S4	**13h30**	**4h30**	**13h00**	**31h30**	**18h30**	**30**	**30**			

- **Semester 5**

Teaching Unit (UE)	Semester hourly volume (15 weeks / Semester)					Coefficients	Credits	Module average		
	Lectures (h)	Tutorials (h)	Laboratory Work(h)	Total (h)	Personal work (h)			Continuous assessment		Final exam
								Tests	Practical Work	
Core Teaching Unit (UEF)										
UEF3.1.1	**6h00**	**0**	**5h30**	**11h30**	**8h15**	**12**	**12**			
Program Optimization and System Performance	1h30	0	1h30	3h00	2h00	3	3	20%	20%	60%
Power Supply and Auxiliary Peripherals	1h30	0	1h00	2h30	2h15	3	3	20%	20%	60%
Parallel Computing on GPUs	1h30	0	1h30	3h00	2h00	3	3	20%	20%	60%
Embedded Operating Systems	1h30	0	1h30	3h00	2h00	3	3	20%	20%	60%
UEF3.1.2	**4h30**	**0**	**3h00**	**7h30**	**6h00**	**9**	**9**			
Systems Security	1h30	0	1h00	2h30	2h00	3	3	20%	20%	60%
Network security	1h30	0	1h00	2h30	2h00	3	3	20%	20%	60%
Reliability and Safety of embedded systems	1h30	0	1h00	2h30	2h00	3	3	20%	20%	60%
Methodology Teaching Unit (UEM)										
UEM3.1.1	**4h30**	**0**	**3h30**	**8h00**	**4h00**	**6**	**6**			
Distributed Computing for Embedded Systems	1h30	0	1h30	3h00	1h15	2	2	20%	20%	60%
Fundamentals of Autonomous Navigation	1h30	0	1h00	2h30	1h15	2	2	20%	20%	60%
Introduction to quantum computing	1h30	0	1h00	2h30	1h30	2	2	20%	20%	60%
Cross-disciplinary Teaching Unit (UET)										
UET3.1.1	**0**	**0**	**1h00**	**1h00**	**2h00**	**2**	**2**			
Capstone Project III	0	0	1h00	1h00	2h00	2	2	100%		0%
Discovery Teaching Unit (UED)										
UED3.1.1	**1h00**	**0**	**1h00**	**2h00**	**0h45**	**1**	**1**			
Entrepreneurship and Startup Development	1h00	0	0	1h00	0h45	1	1	100%		0%
Total Semester S1	**16h00**	**0h**	**13h00**	**29h00**	**21h00**	**30**	**30**			

- **Semester 6**

Internships in companies or research laboratories culminating in a thesis and an oral defense.

		Semester Hourly Volume	Coefficients	Credits
Internship and Final Year Project	14 Weeks	420 hours	30	30
Total	14 Weeks	420 hours	30	30

III - Detailed Syllabus by Subject

a. Module codes

- **Semester 1**

Core Teaching Unit (UEF1)		Core Teaching Unit (UEF2)	
Module	**Code**	**Module**	**Code**
Fundamental Electronics	FE	Introduction to signal Processing	ISP
Instrumentation and Sensors	I/S	Operating Systems Essentials	OS-E
Microprocessors	µp	Database Essentials for Embedded Systems	DB

Methodology Teaching Unit (UEM		Cross-disciplinary Teaching Unit (UET)	
Module	**Code**	**Module**	**Code**
Networking fundamentals	NetF	Reverse Engineering & PCB Design	PCB
Introduction to artificial intelligence	IAI	**Discovery Teaching Unit (UED)**	
		Engineering Ethics and Safety	EES

- **Semester 2**

Core Teaching Unit (UEF1)		Core Teaching Unit (UEF2)	
Module	**Code**	**Module**	**Code**
Electronic Functions	EF	Stochastic Processes and Estimation	SPE
Microcontrollers	µC	Operating Systems for Programmers	OS-P
Advanced sensors and Actuators	ASA	Advanced networking	ANET

Methodology Teaching Unit (UEM)		Cross-disciplinary Teaching Unit (UET)	
Module	**Code**	**Module**	**Code**
Control Systems	CS	Capstone Project I	CP1
Regulations & Standards for Unmanned System	RSUS	**Discovery Teaching Unit (UED)**	
Machine Learning	ML	Training Internship	TI1

- **Semester 3**

Core Teaching Unit (UEF1)		Core Teaching Unit (UEF2)	
Module	**Code**	**Module**	**Code**
Power Electronics	PE	Introduction to Real-Time Operating Systems	IRTOS
Industrial Networks and communication Protocols	INCP	Digital Filtering	DF
Digital Signal Processors	DSP	Human-Machine Interface for embedded systems	HMI
Multi-Sensor Data Fusion	MSDF		

Methodology Teaching Unit (UEM)		Cross-disciplinary Teaching Unit (UET)	
Module	**Code**	**Module**	**Code**
Digital Control Systems	DCS	Introduction to embedded systems security	ICS
Deep Learning	DL	**Discovery Teaching Unit (UED)**	
Operations Research	OR	Project Management for Engineers	PME

- **Semester 4**

Core Teaching Unit (UEF1)		Core Teaching Unit (UEF2)	
Module	**Code**	**Module**	**Code**
Internet of things (IoT)	IoT	Cryptography	CRYP
Electric machines	EM	Embedded Vision and Intelligent Image Processing	VIP
FPGA and Hardware Design	FPGA	Regulation and Control	R&C
Advanced Real-Time Operating Systems	ARTOS		

Methodology Teaching Unit (UEM)		Cross-disciplinary Teaching Unit (UET)	
Module	**Code**	**Module**	**Code**
Fundamentals of robotics	ROBO	Capstone Project II	CP2
Wireless communication Essentials	WCE	**Discovery Teaching Unit (UED)**	
Embedded IA	EAI	Training Internship II	TI2

- **Semester 5**

Core Teaching Unit (UEF1)		Core Teaching Unit (UEF2)	
Module	**Code**	**Module**	**Code**
Program Optimization and System Performance	POSP	Systems Security	SySe
Power Supply and Auxiliary Peripherals	PSAP-	Network security	NCS
Parallel Computing on GPUs	GPU	Reliability and Safety of embedded systems	NSES
Embedded Operating Systems	EOS		

Methodology Teaching Unit (UEM)		Cross-disciplinary Teaching Unit (UET)	
Module	**Code**	**Module**	**Code**
Distributed Computing for Embedded Systems	DCES	Capstone Project III	CP3
Fundamentals of autonomous navigation	Nav	**Discovery Teaching Unit (UED)**	
Introduction to quantum computing	QC	Entrepreneurship and Startup Development	ESD

b. Detailed programs for the semester 1

Teaching Unit	Subject Title	Code	Semester
UEF1.1.1	Fundamental Electronics	FE	S1

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h30	4/4

Objectives

Introduce students to semiconductor devices (diodes, transistors), basic transistor operation, amplifier design (BJT, FET, op-amp), and data conversion (ADC, DAC). By the end, students will be able to:

- Understand p-n junctions, diode behavior, and transistor fundamentals.
- Design simple transistor amplifiers and operational amplifier circuits.
- Implement and analyze basic data converters (ADC/DAC) in embedded contexts...

Prerequisites

- Basic Electricity: Ohm’s law, Kirchhoff’s laws, resistor-capacitor circuits.
- Circuit Analysis: DC/AC analysis, familiarity with phasors and Bode plots is helpful.
- Introductory Embedded Systems: Some experience with microcontrollers (for ADC/DAC labs).

Program Outline

CHAPTER 1: Introduction to Semiconductors and Diodes (2 weeks)

- Semiconductor basics (intrinsic/extrinsic, doping)
- p-n junction behavior, diode I-V characteristics
- Diode applications (rectifiers, clamping, clipping)

CHAPTER 2: Introduction to Transistors (2.5 weeks)

- Bipolar Junction Transistors (BJT): structure, operation modes
- Field Effect Transistors (FET): JFET, MOSFET basics
- Biasing considerations, load lines, regions of operation

CHAPTER 3: Transistor-Based Amplifiers (2 weeks)

- BJT amplifier configurations (common-emitter, common-collector)
- FET amplifiers (common-source, common-drain)
- Frequency response, gain-bandwidth product
- Practical considerations (thermal stability, bias networks)

CHAPTER4: Operational Amplifiers (3 weeks)

- Ideal vs. real op-amp parameters (offset, slew rate, input bias)
- Inverting and non-inverting amplifier topologies

- Summing, differential, instrumentation amplifiers
- Compensation and stability in op-amp circuits.

CHAPTER5: Analog-to-Digital Converters (1.5 weeks)

- Sampling theorem, quantization basics
- ADC architectures (successive approximation, flash, sigma-delta)
- Performance metrics (resolution, SNR, conversion rate)
- Microcontroller-based ADC usage.

CHAPTER6: Digital-to-Analog Converters (1.5 weeks)

- R-2R ladder DAC and weighted resistor DAC
- DAC performance metrics (linearity, resolution, settling time)
- Voltage vs. current output DACs
- Practical integration in embedded systems (audio, control signals).

CHAPTER7: Multivibrators (2.5 weeks)

- Introduction to Multivibrators (Definition and classification).
- Astable Multivibrator (Operating Principle, Transistor-Based Astable Circuit, Oscillation Frequency Calculation, Astable Multivibrator with Integrated Circuit (555)).
- Monostable Multivibrator (Operating Principle, Transistor Monostable Circuit, Pulse Duration Calculation, Monostable with NE555).
- Bistable Multivibrator Applications (Operating Principle, Types of Bistables).
- Practical application.

Laboratory Work

- **Lab 1:** Diode Fundamentals
- **Lab 2:** Transistor Basics
- **Lab 3:** Transistor Amplifier
- **Lab 4:** Operational Amplifiers
- **Lab 5:** ADC Fundamentals on Microcontroller
- **Lab 6:** DAC Implementation

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 7th ed. New York, NY, USA: Oxford Univ. Press, 2015.
- P. Scherz and S. Monk, *Practical Electronics for Inventors*, 4th ed. New York, NY, USA: McGraw-Hill, 2021.

- D. J. Comer, *Electronics: Circuits, Amplifiers, and Gates*. Hoboken, NJ, USA: Wiley, 2022.
- A. P. Malvino, *Electronic Principles*, 8th ed. New York, NY, USA: McGraw-Hill, 2015.
- P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. Cambridge, UK: Cambridge Univ. Press, 2015.
- S. Franco, *Design with Operational Amplifiers and Analog Integrated Circuits*, 4th ed. New York, NY, USA: McGraw-Hill, 2014.
- J. F. Wakerly, Digital Design: Principles and Practices, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2018.

Teaching Unit	Subject Title	Code	Semester
UEF1.1.1	**Instrumentation and Sensors**	**I/S**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h30	4/4

Objectives

This course aims to provide students with a comprehensive understanding of the principles and operation of various sensors used in instrumentation systems. Students will learn to design, analyze, and condition sensor circuits such as Wheatstone bridges and impedance bridges. They will also explore a wide range of sensors, including mechanical, thermal, optical, and fluidic types, along with their applications in real-world scenarios. Additionally, the module focuses on practical aspects such as sensor interfacing, calibration, and noise filtering techniques, preparing students for hands-on problem-solving in engineering contexts.

Prerequisites

- Basic understanding of electronics, electrical circuits, and measurement principles
- Basic physical quantities such as temperature, pressure, and displacement will be helpful but not mandatory.

Program Outline

CHAPTER 1: Fundamentals of Sensors and Measurement Systems (5 weeks)

- Overview of measurement systems.
- Importance of instrumentation in engineering.
- Passive Sensors/ Active sensors
- Physical Principles of Active Sensors: photoelectric effect, Seebeck effect, piezoelectric effect, Hall Effect…
- Understanding sensor error vs. uncertainty.
- Optical Encoder

CHAPTER 2: Study of Advanced Sensors (4.5 weeks)

- Thermal Sensors: Thermocouples, RTDs, and thermistors.
- Mechanical Sensors: accelerometers, gyroscopes, and strain gauges.
- Optical Sensors: Photodiodes, phototransistors, and encoders.
- Fluidic Sensors: Pressure, velocity, and flow sensors.

CHAPTER 3: Signal Conditioning Electronics (5.5 weeks)

- Passive sensor conditioning circuits: Wheatstone bridge and impedance bridges for passive sensors.
- Amplifiers : Instrumentation amplifiers and isolation amplifiers.
- Filtering techniques: Butterworth, Chebyshev filters.

- Calibration techniques: Static and dynamic calibration.
- Noise filtering and signal stability.

Laboratory Work

- **Lab 1:** Temperature Measurement: Calibrate and condition thermocouples and RTDs.
- **Lab 2:** Mechanical Sensors: Experiment with accelerometers and gyroscopes.
- **Lab 3:** Filter Design: Implement Butterworth filters for noise reduction
- **Lab 4:** Sensor Interfacing: Interfacing MEMS sensors with microcontrollers
- **Lab 5:** Calibration and Noise Reduction: Analyze and filter noisy sensor signals

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- R. Pallas-Areny and J. G. Webster, *Sensors and Signal Conditioning*, 2nd ed. New York, NY, USA: Wiley, 2001.
- C. W. de Silva, *Sensors and Actuators: Engineering System Instrumentation*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2022.
- Y. Liu et al., *Smart Sensors and Systems: Technology and Applications*. New York, NY, USA: Springer, 2023.
- J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*, 5th ed. New York, NY, USA: Springer, 2016.
- E. O. Doebelin and D. N. Manik, *Measurement Systems: Application and Design*, 6th ed. New York, NY, USA: McGraw-Hill, 2010.
- J. G. Webster, Ed., *The Measurement, Instrumentation, and Sensors Handbook*. Boca Raton, FL, USA: CRC Press, 1999.
- G. C. M. Meijer, *Smart Sensor Systems*. Chichester, UK: Wiley, 2008.

Teaching Unit	Subject Title	Code	Semester
UEF1.1.1	Microprocessors	µp	S1

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	4/4

Objectives

Provide students with a solid understanding of microprocessor architectures, focusing on how general-purpose CPUs work at a hardware level. Students will learn about memory hierarchy, I/O systems, power management, and real-world applications, preparing them to design efficient and robust embedded solutions..

Prerequisites

- Basics of digital electronics (logic gates, combinational and sequential circuits)
- Introductory computer architecture concepts.

Program Outline

CHAPTER 1: Introduction to Embedded Systems and Microprocessors (2 weeks)

- Definition and key characteristics of embedded systems
- Role of microprocessors in embedded applications
- Comparison of embedded system architectures (Harvard vs. Von Neumann).

CHAPTER 2: Microprocessor Architectures (3 weeks)

- Instruction Set Architecture (ISA): CISC vs. RISC
- Internal structure: ALU, registers, control unit, pipeline basics
- Examples of microprocessor families (Intel x86, ARM Cortex-A).

CHAPTER 3: Memory Hierarchy and Interfacing (3 weeks)

- Types of memory (ROM, RAM, Flash)
- Cache levels, associativity, mapping
- Memory interfacing and Direct Memory Access (DMA).

CHAPTER4: Input/Output Systems (3 weeks)

- I/O concepts: polling vs. interrupts
- Peripheral interfacing: GPIO, UART, I^2C, SPI, CAN
- Bus systems: data, address, control.

CHAPTER5: Power Management in Microprocessors (2 weeks)

- Low-power design principles
- Sleep, idle, and hibernate modes
- Thermal design considerations.

CHAPTER6: Microprocessor Trends and Applications (2 weeks)

- Evolution of microprocessors and performance vs. efficiency trade-offs
- Multicore processors and AI accelerators
- Real-world applications in autonomous systems and IoT.

Laboratory Work

- **Lab 1:** Assembly Programming Basics
- **Lab 2:** Memory Interfacing and Data Flow
- **Lab 3:** GPIO and Interrupt Handling
- **Lab 4:** Cache and Performance Analysis
- **Lab 5:** Power Management Techniques
- **Lab 6:** Microprocessor-Based Mini Project

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2020.
- Y. Zhu, *Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C*, 4th ed. New York, NY, USA: Springer, 2022.
- J. P. Shen, *Modern Microprocessors: A 64-bit Perspective*. Cambridge, MA, USA: MIT Press, 2023.
- A. Sloss, D. Symes, and C. Wright, *ARM System Developer's Guide*. San Francisco, CA, USA: Morgan Kaufmann, 2004.
- S. Furber, *ARM System-on-Chip Architecture*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2000.
- J. Yiu, *The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors*, 2nd ed. New York, NY, USA: Newnes, 2015.
- M. Barr, *Programming Embedded Systems: With C and GNU Development Tools*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2006.

Teaching Unit	Subject Title	Code	Semester
UEF1.1.2	Introduction to Signal Processing	ISP	S1

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	3/3

Objectives

This module provides a comprehensive introduction to signal processing concepts essential for embedded systems. Students will explore the basics of continuous and discrete signals, revisit Fourier Transforms (with an emphasis on time-frequency duality), study sampling and quantization, and delve into the Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), and an introduction to the Laplace Transform. Emphasis is placed on both theoretical understanding and practical applications in embedded and real-time contexts.

Prerequisites

- Basic Fourier Transform concepts.
- Understanding of continuous-time signals.

Program Outline

CHAPTER 1: Introduction to Signals and Elementary Signals (2 weeks)

- Definitions: continuous-time vs. discrete-time signals, periodic vs. aperiodic.
- Elementary signals: impulse, step, exponential, sinusoidal.
- Signal properties: energy, power, symmetry.

CHAPTER 2: Vector Representation, Correlation, and Convolution (2.5 weeks)

- Vector Representation of Signals (Scalar product, orthogonally, norm, Euclidean distance, Signal expansion in orthogonal functions, least-squares approximation).
- Correlation (Definition, Autocorrelation, cross-correlation, properties, power spectral density, Detection of periodic signals in noise).
- Convolution (Definition, properties, calculation methods. Plancherel's theorem 'energy in time vs. frequency domain'.

CHAPTER3: Fourier Transform (1.5 weeks)

- Recap of the Continuous-Time Fourier Transform (CTFT).
- Time-Frequency Duality: mapping operations between time and frequency domains.
- Properties: linearity, shifting, convolution theorem.
- Physical interpretation of the frequency domain.

CHAPTER4: Sampling and Quantization Theory (4 weeks)

- Sampling theorem (Nyquist-Shannon), aliasing.
- Quantization: uniform vs. non-uniform, quantization error.

- ADCs, reconstruction filters, and anti-aliasing filters.

CHAPTER5: Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) (4 weeks)

- DFT: definition, properties, frequency resolution, leakage, windowing.
- FFT: Cooley-Tukey algorithm, computational complexity, practical efficiency gains.
- Implementation considerations for real-time and embedded systems..

Laboratory Work

- **Lab 1:** Introduction to signal processing tools
- **Lab 2:** Correlation, Convolution, and Signal Approximation
- **Lab 3:** Sampling and Quantization
- **Lab 4:** DFT and FFT Implementation
- **Lab 5:** Filter Design and Frequency-Domain Analysis

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- S. W. Smith, *Digital Signal Processing: A Practical Guide for Engineers and Scientists*, 3rd ed. San Diego, CA, USA: California Technical Publishing, 2022.
- N. Kehtarnavaz, *Real-Time Digital Signal Processing: Fundamentals, Implementations and Applications*. New York, NY, USA: Springer, 2023.
- J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2007.
- S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. San Diego, CA, USA: California Technical Publishing, 1999.
- L. F. Chaparro, *Signals and Systems Using MATLAB*, 3rd ed. Cambridge, MA, USA: Academic Press, 2018.
- S. Haykin and B. Van Veen, *Signals and Systems*, 2nd ed. Hoboken, NJ, USA: Wiley, 2002.

Teaching Unit	Subject Title	Code	Semester
UEF1.1.2	**Operating Systems Essentials**	**OS-E**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

Provide students with foundational knowledge of operating systems from a user and administrator perspective. By the end, students should be proficient in Linux system administration tasks, user management, file systems, and basic networking configuration..

Prerequisites

- Familiarity with basic computer architecture.
- Command-line interface (CLI) fundamentals..

Program Outline

CHAPTER 1: Introduction to Operating Systems (1.5 weeks)

- Definition and purpose
- Types of operating systems
- Computer System Architecture (Components and organization, CPU, memory, I/O devices, System bus architecture).

CHAPTER 2: Process Management (1.5 weeks)

- Process concept and states
- Process scheduling algorithms
- Inter-process communication.

CHAPTER 3: CPU Scheduling (1.5 weeks)

- Scheduling criteria and algorithms
- Preemptive vs. non-preemptive scheduling
- Performance evaluation

CHAPTER4: Memory Management (1.5 weeks)

- Memory hierarchy
- Virtual memory
- Paging and segmentation

CHAPTER5: File Systems (1.5 weeks)

- File concepts and operations
- Directory structures
- File allocation methods

- Type of Files
- Permissions.

CHAPTER6: I/O Systems (1.5 weeks)

- I/O hardware
- I/O software layers
- Disk scheduling algorithms.

CHAPTER7: Concurrency and Synchronization (1.5 weeks)

- Critical section problem
- Semaphores and monitors
- Deadlocks and starvation

CHAPTER8: Virtualization (1.5 weeks)

- Hypervisors and virtual machines
- Containers
- Resource virtualization

CHAPTER9: Security and Protection (1.5 weeks)

- Authentication and authorization
- Access control mechanisms
- Security threats and countermeasures.

CHAPTER10: Distributed Systems (1.5 weeks)

- Network operating systems
- Distributed file systems
- Distributed coordination

Laboratory Work

- **Lab 1:** Linux Installation and Basic Commands
- **Lab 2:** User and Group Management
- **Lab 3:** Filesystem Administration
- **Lab 4:** Process and Service Control
- **Lab 5:** Networking and Security
- **Lab 6:** Shell Scripting and Automation

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- A. Tanenbaum and H. Bos, *Modern Operating Systems*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2014.
- W. Shotts, *The Linux Command Line: A Complete Introduction*, 3rd ed. San Francisco, CA, USA: No Starch Press, 2021.
- C. Negus, *Linux Bible*, 11th ed. Hoboken, NJ, USA: Wiley, 2022.
- R. Love, *Linux System Programming: Talking Directly to the Kernel and C Library*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2013.
- W. Shotts, *The Linux Command Line: A Complete Introduction*, 2nd ed. San Francisco, CA, USA: No Starch Press, 2019.
- M. K. McKusick, G. V. Neville-Neil, and R. N. M. Watson, *The Design and Implementation of the FreeBSD Operating System*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2015.
- C. Negus, *Linux Bible*, 10th ed. Hoboken, NJ, USA: Wiley, 2020.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM1.1.1	**Database Essentials for Embedded Systems**	**DB**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**3/3**

Objectives

This course begins with the objective of enabling students to install, configure, and interact with a relational database management system. It is designed to teach and apply the SQL query language for both defining and manipulating databases. Additionally, the course focuses on utilizing a database modeling technique for a single entity class, ensuring that learners develop a solid foundation in designing effective database structures. Finally, it aims to impart the principles and concepts of data integrity, security, and confidentiality, which are essential for maintaining robust and secure database systems..

Prerequisites

- Basics of Programming
- Fundamental knowledge of data structures and algorithms.

Program Outline

CHAPTER 1: Conceptual and Logical Modeling (2Weeks)

- Introduction to data modeling
- Identification of entities, attributes, and relationships
- Transformation from conceptual (ER model) to logical (relational schema) modeling
- Primary and foreign key constraints.

CHAPTER 2: Relational Algebra (2Weeks)

- Common set operators in the context of a database.
- Specific operators: projection, selection, renaming, join, Cartesian product.
- Aggregation functions..

CHAPTER3: Relational Databases (05 weeks)

- Databases: Vocabulary (relation, attribute, domain, relational schema, and the concept of key).
- Primary key.
- Complex relational algebra operators.
- Normalisation of Database Tables

CHAPTER4: The SQL Language (06 weeks)

- **Data Definition Language (DDL):** creating and modifying tables**.**

- **Data Manipulation Language (DML):** manipulating data using SQL.

Laboratory Work

- **Lab 1**: SQLite basics: table creation, CRUD operations, and basic queries.
- **Lab 2**: Indexing and advanced queries with SQLite for embedded systems.
- **Lab 3**: Advanced SQL with SQLite for embedded systems.
- **Lab 4**: Transaction Management and Concurrency Control
- **Lab 5**: Setting up and querying InfluxDB for time-series sensor data.
- **Lab 6**: Capstone project: Integrated SQLite and InfluxDB solution.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- M. Kleppmann, *Designing Data-Intensive Applications*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2021.
- A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 8th ed. New York, NY, USA: McGraw-Hill, 2023.
- R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Pearson, 2016.
- J. D. Ullman and J. Widom, *A First Lecturese in Database Systems*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2007.
- P. Bailis, *Designing Data-Intensive Applications*. Sebastopol, CA, USA: O'Reilly Media, 2017.
- A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. New York, NY, USA: McGraw-Hill, 2020.
- M. Kleppmann, *Designing Data-Intensive Applications*. Sebastopol, CA, USA: O'Reilly Media, 2017.

- Elmasri & Navathe (2016, 7th Edition, Pearson)
- Silberschatz, Korth & Sudarshan (2019, 7th Edition, McGraw-Hill)

C.J. Date (2003, 8th Edition, Addison-Wesley)

Teaching Unit (UE)	Subject Title	Code	Semester
UEF1.1.2	**Networking Fundamentals**	**NetF**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h30	**3/3**

Objectives

This course provides foundational knowledge and practical skills in networking, with a focus on autonomous embedded systems and IoT. Students will learn about network protocols, architectures, and technologies, and how to apply them in real-world scenarios..

Prerequisites

- Basic understanding of computer systems and programming.

Program Outline

CHAPTER 1: Introduction to Networks (2 weeks)

- Fundamental concepts: nodes, links, packets, protocols, network topologies.
- Use of networks in embedded and autonomous systems.
- OSI Model and TCP/IP Stack: layers, functions, key differences.
- Client-server vs. peer-to-peer models.

CHAPTER 2: Ethernet and LAN Technologies (2.5 weeks)

- Data Link Layer (Layer 2) basics.
- Ethernet frame structure and MAC addressing.
- ARP (Address Resolution Protocol): role and operation.
- Introduction to switches and LAN architecture.
- VLAN basics and network segmentation (IEEE 802.1Q).

CHAPTER3: Wireless Networks (2.5weeks)

- Wireless technologies overview: Wi-Fi, Bluetooth, Zigbee (general introduction).
- Wi-Fi network structure (SSID, BSSID, channels, frequencies).
- Differences between wired and wireless networks.
- Embedded and IoT use cases for wireless communication.

CHAPTER4: IP Protocols and Addressing (3.5 weeks)

- IPv4 addressing: structure, classes, subnetting, CIDR notation.
- DHCP: automatic IP address assignment.
- ICMP and network diagnostic tools (ping, traceroute, ICMP errors).
- Static routing and default gateway.
- NAT and basic introduction to port forwarding.

- Introduction to IPv6: addressing scheme.

CHAPTER 5: Transport Protocols: TCP and UDP (3 weeks)

- TCP principles: reliable transport, three-way handshake, flow control.
- UDP principles: connectionless communication, low-latency transmission.
- TCP vs. UDP comparison in embedded and IoT environments.

CHAPTER 6: Application Protocols: HTTP (1.5 weeks)

- Application Layer protocols overview.
- Focus on HTTP (HyperText Transfer Protocol):
    - Client-server communication via HTTP.
    - HTTP request/response model.
    - Common HTTP methods: GET, POST, PUT, DELETE.
    - HTTP status codes (200 OK, 404 Not Found, 500 Server Error).
- Example: using HTTP in embedded systems to send sensor data to a server (simple RESTful interaction)..

Laboratory Work

- **Lab 1:** Basic Network Connection and Testing
- **Lab 2:** IP Addressing and Network Troubleshooting
- **Lab 3:** Routing, Default Gateway, and NAT
- **Lab 4:** TCP/UDP and Packet Analysis (Transport Protocols)
- **Lab 5:** HTLABSRequest Analysis and Testing.
- **Lab 6:** Name Resolution (DNS)

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 9th ed. Boston, MA, USA: Pearson, 2023.
- L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 7th ed. Cambridge, MA, USA: Morgan Kaufmann, 2021.
- A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2011.
- B. A. Forouzan, *Data Communications and Networking*, 5th ed. New York, NY, USA: McGraw-Hill, 2012.
- W. Stallings, *Data and Computer Communications*, 10th ed. Upper Saddle River, NJ, USA: Pearson, 2013.
- L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2020.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM1.1.1	**Introduction to artificial intelligence**	**IAI**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**3/3**

Objectives

This Lecturese provides foundational knowledge of Artificial Intelligence (AI), focusing on core concepts, algorithms, and real-world applications. Students will explore intelligent agents, search algorithms, reasoning, planning, and major AI application areas, with practical exposure to implementing AI techniques.

Prerequisites

Data Structures and Algorithms.

Program Outline

CHAPTER1: Introduction to Python (4weeks)

- Definition and history of AI.
- Goals and challenges of AI.
- Overview of AI applications in real-world systems.

CHAPTER2: Introduction to AI (1week)

- Definition and history of AI.
- Goals and challenges of AI.
- Overview of AI applications in real-world systems.

CHAPTER3: Intelligent Agents (4weeks)

- Definition and structure of intelligent agents.
- Types: reactive, goal-based, utility-based, and learning agents.
- Environments and properties (deterministic/stochastic, fully/partially observable).

CHAPTER4: Search and Problem Solving (3 weeks)

- Problem-solving as search.
- Uninformed search: Depth-First Search (DFS), Breadth-First Search (BFS).
- Informed search: A* and greedy algorithms.
- Heuristic functions and optimization.

CHAPTER5: Adversarial Search and Games (2weeks)

- Game theory basics and AI applications.
- Minimax algorithm and alpha-beta pruning.
- Applications: chess, tic-tac-toe, and real-time strategy games.

CHAPTER6: Knowledge Representation and Reasoning (2 weeks)

- Representing knowledge: propositional and first-order logic.
- Inference mechanisms: forward and backward chaining.
- Ontologies and semantic networks.

Laboratory Work

Lab 1: introduction to python
Lab 2: Python basics and AI libraries (e.g., NumPy, scikit-learn).
Lab 3: Implementing search algorithms (e.g., BFS, A*).

Lab 4: Building a simple game-playing agent (e.g., tic-tac-toe using Minimax).
Lab 5: Constraint satisfaction problems (e.g., solving Sudoku).
Lab 6: Knowledge representation and reasoning (e.g., building a simple rule-based system).

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2022.
- E. Alpaydin, *Introduction to Machine Learning*, 5th ed. Cambridge, MA, USA: MIT Press, 2021.
- M. Tim Jones, *AI Application Programming*, 2nd ed. Boston, MA, USA: Charles River Media, 2008.
- E. Alpaydin, *Introduction to Machine Learning*, 4th ed. Cambridge, MA, USA: MIT Press, 2020.
- C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

Teaching Unit (UE)	Subject Title	Code	Semester
UED1.1.1	**Engineering Ethics and Safety**	**EES**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h00	0	0	**1/1**

Objectives

This module focuses on instilling a strong ethical foundation and safety awareness in engineering students. By the end of this Lecturese, students will:

- Understand the role of ethics in engineering practice and decision-making.
- Recognize and apply ethical principles in professional scenarios.
- Gain knowledge of safety regulations, risk management, and the importance of safe design in engineering systems.
- Be equipped to identify and mitigate potential hazards in engineering workplaces.

Prerequisites

No prerequisites

Program Outline

- CHAPTER 1: Introduction to Engineering Ethics
- CHAPTER 2: Ethical Decision-Making in Engineering
- CHAPTER 3: Professional Responsibility and Codes of Conduct
- CHAPTER 4: Safety Fundamentals in Engineering
- CHAPTER 5: Workplace Safety and Risk Management
- CHAPTER 6: Ethical and Safety Challenges in Emerging Technologies

Laboratory Work

No laboratory works

Evaluation Method

Module average = 40% continuous assessment + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- Charles B. Fleddermann, Engineering Ethics.
- Mike W. Martin and Roland Schinzinger, Ethics in Engineering.
- Roland P. Blondé, Safety and Risk in Engineering Practice.
- Deborah G. Johnson, Engineering Ethics: Contemporary Issues and Challenges.

Relevant standards and guidelines (e.g., IEEE Code of Ethics, ISO 45001).

Teaching Unit (UE)	Subject Title	Code	Semester
UET1.1.1	**Reverse Engineering & PCB Design**	**PCB**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	1h00	**2/2**

Objectives

This module aims to provide students with the knowledge and skills required to reverse engineering, design and fabricate Printed Circuit Boards (PCBs) for embedded and electronic systems. By the end of the module, students will be able to:

- Understand the role and methodologies of Reverse Engineering (RE) in product development.
- Apply 3D scanning and surface reconstruction techniques.
- Understand PCB design principles and best practices for layout and routing.
- Use EDA (Electronic Design Automation) tools effectively to create schematics and PCB layouts.
- Integrate Design for Manufacturing (DFM) considerations into PCB projects.
- Oversee PCB fabrication processes and perform assembly, soldering, and testing.

Prerequisites

- Basic knowledge of electronics (circuit analysis, components).
- Familiarity with schematic design software or circuit simulators is recommended.

Program Outline

- Introduction to Reverse Engineering & Rapid Prototyping **(2weeks)**
- Reverse Engineering Process **(2weeks)**
- Introduction to PCB Design **(2weeks)**
- EDA Tools and Schematic Design **(2weeks)**
- Layout Techniques and Constraints **(2weeks)**
- Design for Manufacturing (DFM) and Rules Checking **(2weeks)**
- PCB Fabrication and Assembly **(1.5weeks)**
- Testing, Inspection, and Quality Assurance **(1.5 weeks)**

Laboratory Work

- **LAB1:** Schematic Capture and Library Creation
- **LAB2:** PCB Layout Basics
- **LAB3:** Advanced Layout and DRC
- **LAB4:** Fabrication and Assembly
- **LAB5:** Testing and Troubleshooting

Evaluation Method

Module average = 40% continuous assessment + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- Coombs, C.F. Jr. (Ed.). Printed Circuits Handbook. McGraw-Hill.
- KiCad/Altium/Eagle official documentation and tutorials.
- Bosshart, W. Printed Circuit Boards: Design and Technology. McGraw-Hill.
- IPC Standards (e.g., IPC-2221, IPC-7351) for PCB design and assembly guidelines.

c. Detailed programs for the semester 2

Teaching Unit	Subject Title	Code	Semester
UEF1.2.1	**Electronic functions**	**EF**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**3/3**

Objectives

This module introduces oscillator design, analog modulation (AM, FM, PM), and Phase-Locked Loop (PLL) fundamentals. By the end, students will be able to:

- Understand and implement common oscillator circuits (RC, LC, crystal).
- Explain the need for modulation, compare analog vs. digital approaches, and design basic AM/FM/PM systems.
- Grasp PLL principles, components (phase detector, VCO, loop filter), and apply them to frequency synthesis, clock recovery, and demodulation.

Prerequisites

- Fundamental Electronics: Familiarity with transistor amplifiers, op-amp basics, diode behavior.
- Circuit Analysis: AC analysis, frequency response, Bode plots.
- Introductory Signals & Systems: Fourier transforms, basic knowledge of frequency domain..

Program Outline

CHAPTER 1: Oscillators (2 weeks)

- Principle of Oscillation: Barkhausen criterion, feedback loop concepts.
- RC and LC Oscillators: Phase-shift, Wien bridge, Colpitts, Hartley.
- Crystal Oscillators: Piezoelectric resonance, stability.
- Frequency Control: Tuning elements, temperature compensation.

CHAPTER 2: Introduction to Modulation (3 weeks)

- Need for Modulation: Signal bandwidth, transmission efficiency.
- Comparison: Analog vs. Digital: Basic differences, advantages, typical use cases.
- Spectrum and Bandwidth: Carrier concept, baseband vs. passband signals.

CHAPTER 3: Amplitude Modulation (AM) (2 weeks)

- Concept and Math Representation: Carrier, sidebands, envelope.
- Single Sideband (SSB) and Double Sideband (DSB): Efficiency, practical generation.
- Detection and Demodulation: Envelope detection, synchronous detection.
- Applications and Limitations: Broadcasting, power considerations.

CHAPTER4: Frequency and Phase Modulation (FM/PM) (2 weeks)

- Frequency Modulation (FM): Mathematical representation, deviation, Carson’s rule.
- FM Generation/Detection: VCO-based modulators, discriminators, PLL demod.

- Phase Modulation (PM): Basic concept, comparison with FM, typical PM applications..

CHAPTER5: Phase-Locked Loop (PLL) - Principles and Components (2 weeks)

- Introduction to PLL: Basic operating principle, lock range vs. capture range.
- Core Components: Phase detector, VCO, loop filter.
- Design Considerations: Loop bandwidth, stability, noise performance.

CHAPTER6: PLL Applications and Advanced Topics (2 weeks)

- Frequency Synthesis: Integer-N vs. fractional-N PLL, practical examples.
- Clock Recovery in Digital Communication: Jitter, synchronization.
- FM Demodulation and Other Uses: Lock-in detection, advanced demod techniques.
- Practical Implementations: Simulations, typical ICs (74HC4046, LMX series)..

Laboratory Work

- **Lab 1:** RC Oscillator Design
- **Lab 2:** LC/Crystal Oscillator
- **Lab 3:** AM Generation and Detection
- **Lab 4:** FM/PM Generation and Detection
- **Lab 5:** Basic PLL Implementation
- **Lab 6:** PLL Applications

Evaluation Method .

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- S. Haykin, *Communication Systems*, 6th ed. Hoboken, NJ, USA: Wiley, 2021.
- P. Horowitz and W. Hill, *The Art of Electronics*, 4th ed. Cambridge, UK: Cambridge Univ. Press, 2023.
- H. Taub and D. Schilling, *Principles of Communication Systems*, 3rd ed. New York, NY, USA: McGraw-Hill, 2011.
- S. Franco, *Design with Operational Amplifiers and Analog Integrated Circuits*, 4th ed. New York, NY, USA: McGraw-Hill, 2014.
- U. L. Rohde and A. Poddar, *The Design of Modern Microwave Oscillators for Wireless Applications*. Hoboken, NJ, USA: Wiley, 2005

Teaching Unit	Subject Title	Code	Semester
UEF1.2.1	**Microcontrollers**	**µC**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

Introduce students to ARM-based microcontrollers, emphasizing real-time operation, peripheral integration, and low-power design. By the end of this module, students will be able to develop firmware and interface sensors/actuators for resource-constrained, real-time embedded applications..

Prerequisites

- Fundamentals of C programming for embedded systems
- Basic knowledge of digital electronics and microprocessor concepts.

Program Outline

CHAPTER 1: Introduction to Microcontrollers (2 weeks)

- Definition and scope of microcontrollers vs. microprocessors
- Common microcontroller architectures (8-bit, 16-bit, 32-bit)
- Typical embedded applications (sensors, actuators, control systems)

CHAPTER 2: ARM Cortex Microcontroller Architecture (3 weeks)

- ARM Cortex-M series overview (M0, M3, M4, M7)
- RISC principles, pipeline basics
- Core components: ALU, registers, memory bus, interrupts

CHAPTER 3: Memory and I/O Organization (3 weeks)

- Flash, SRAM, and EEPROM in microcontrollers
- Memory mapping and vector tables
- GPIO organization, port configuration, pull-up/pull-down resistors
- Basic bus interfaces: AHB, APB

CHAPTER4: Peripherals and Real-Time Features (3 weeks)

- Timers, counters, watchdog timers
- PWM generation for motor control or signal generation
- Nested Vectored Interrupt Controller (NVIC)
- Basics of RTOS on ARM microcontrollers.

CHAPTER5: Communication Protocols (2 weeks)

- UART/USART, SPI, I²C, and CAN
- Serial communication setup and data handling

- DMA usage for high-speed data transfer

CHAPTER6: Low-Power Modes and Advanced Features (2 weeks)

- Sleep, stop, standby modes on ARM Cortex-M
- Clock management and PLL configuration
- Advanced features: floating-point unit (FPU), DSP instructions, cryptographic accelerators.

Laboratory Work

- **Lab 1:** Setting Up the ARM Development Environment
- **Lab 2:** GPIO and Interrupts on ARM MCU
- **Lab 3:** Timers and PWM
- **Lab 4:** Communication Protocols
- **Lab 5:** Low-Power Modes
- **Lab 6:** Mini Project

Evaluation Method .

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*, 6th ed. New York, NY, USA: Springer, 2022.
- G. C. M. Meijer, *Smart Sensor Systems: Emerging Technologies and Applications*. Chichester, UK: Wiley, 2023.
- J. Yiu, *The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors*, 2nd ed. New York, NY, USA: Newnes, 2015.
- M. Predko, *Programming and Customizing the PIC Microcontroller*, 3rd ed. New York, NY, USA: McGraw-Hill, 2007.
- D. W. Lewis, *Fundamentals of Embedded Software with the ARM Cortex-M3*. Upper Saddle River, NJ, USA: Pearson, 2013.
- J. W. Valvano, *Embedded Systems: Introduction to ARM Cortex-M Microcontrollers*, 5th ed. Austin, TX, USA: Jonathan W. Valvano, 2019.
- T. Noergaard, *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*, 2nd ed. Burlington, MA, USA: Newnes, 2012.

Teaching Unit	Subject Title	Code	Semester
UEF1.2.1	**Advanced sensors and Actuators**	**ASA**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

This module aims to introduce students to the principles and working mechanisms of actuators, including electric, hydraulic, and mechanical systems. It also focuses on the integration of sensors and actuators in embedded systems, providing students with the skills to design, interface, and calibrate sensor-actuator networks. Emphasis is placed on noise filtering, signal conditioning, and the implementation of control loops for real-time applications in automation and robotics. By the end of the module, students will have the expertise to work with complex mechatronic systems..

Prerequisites

- Students are expected to have prior knowledge of basic instrumentation and sensor systems.
- Familiarity with microprocessor programming.

Program Outline

CHAPTER1: Advanced Sensors overview (3 weeks)

- GNSS, LiDAR, and Radar: Principles and applications in autonomous systems.
- Accelerometers and Gyroscopes: MEMS-based inertial measurement units (IMUs).
- Camera Sensors: Visual and infrared imaging and vision systems for robotics and drones.

CHAPTER 2: Introduction to Actuators (1 weeks)

- Overview and classification of actuators (electric, hydraulic, pneumatic, mechanical).
- Performance metrics: Force, speed, precision, and energy efficiency.

CHAPTER 2: Electric Actuators overview (2 weeks)

- DC motors: Brushed vs. brushless motors, PWM control.
- Stepper motors: Full-step, half-step, and microstepping modes.
- Servo motors: Position control using feedback systems.

CHAPTER 3: Hydraulic, Pneumatic, and Mechanical Actuators (3.5 weeks)

- Hydraulic Actuators: Principles of hydraulic systems, Applications in heavy machinery, aerospace, and robotics. Hydraulic pumps and valves for control.
- Pneumatic Actuators: Basics of pneumatic systems and compressed air usage. Types of pneumatic actuators (cylinders, rotary actuators). Applications in industrial automation and robotics. Advantages and limitations (cost, force, precision).
- Mechanical Actuators: Gear systems: Speed reduction and torque amplification. Linkages and cams: Principles and applications in mechanical motion.

CHAPTER4: Sensor-Actuator Integration (3 weeks)

- Feedback control systems: PID control for position, speed, and force.
- Interfacing techniques for real-time control..

CHAPTER5: Noise Filtering and Signal Conditioning (2.5 weeks)

- Electrical noise and mechanical vibrations: Sources and impact.
- Filtering techniques: Low-pass, high-pass, and band-pass filters.
- Signal conditioning: Amplification and linearization.

Laboratory Work

- **Lab 1:** DC Motor Control: Speed and torque control using PWM.
- **Lab 2:** Stepper Motor Control: Test and analyze microstepping.
- **Lab 3:** Servo Motor Experiment: Implement feedback-based position control.
- **Lab 4:** Hydraulic Simulation: Simulate a hydraulic actuator system.
- **Lab 5:** Signal Conditioning: Design and implement filters for actuator signals

Evaluation Method .

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*, 6th ed. New York, NY, USA: Springer, 2022.
- G. C. M. Meijer, *Smart Sensor Systems: Emerging Technologies and Applications*. Chichester, UK: Wiley, 2023.
- R. Pallas-Areny and J. G. Webster, *Sensors and Signal Conditioning*, 2nd ed. New York, NY, USA: Wiley, 2001.
- G. C. M. Meijer, *Smart Sensor Systems*. Chichester, UK: Wiley, 2008.
- E. O. Doebelin and D. N. Manik, *Measurement Systems: Application and Design*, 6th ed. New York, NY, USA: McGraw-Hill, 2010.
- J. G. Webster, Ed., *The Measurement, Instrumentation, and Sensors Handbook*. Boca Raton, FL, USA: CRC Press, 1999.

Teaching Unit	Subject Title	Code	Semester
UEF1.2.2	**Stochastic Processes and Estimation**	**SPE**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**3/3**

Objectives

This module introduces stochastic processes, probabilistic modeling, and advanced estimation techniques. Students will learn to analyze random signals (autocorrelation, PSD), perform spectral estimation (classical + parametric), and implement adaptive filters (Wiener, Kalman). By the end, they will be able to design and evaluate estimation algorithms for embedded and communication systems..

Prerequisites

- Basic probability theory (random variables, probability distributions).
- Knowledge of Fourier Transform and convolution from previous signal processing Lectureses.
- Familiarity with linear time-invariant systems.

Program Outline

CHAPTER 1: Introduction to Stochastic Processes (2 weeks)

- Deterministic vs. random signals; stationarity, ergodicity
- Classification: discrete-time vs. continuous-time
- Common examples: white noise, random walk, Gaussian processes

CHAPTER 2: Probability and Statistical Descriptions (2 weeks)

- Probability density/cumulative distribution functions (PDF/CDF)
- Moments and expectations: mean, variance, higher-order moments
- Joint PDFs, conditional probabilities for multivariable processes
- Noise modeling in communication systems.

CHAPTER3: Autocorrelation, Cross-Correlation, and Power Spectral Density (2 weeks)

- Definitions: autocorrelation, cross-correlation, and their properties.
- Wiener-Khinchin theorem: relationship between autocorrelation and power spectral density (PSD).
- Stationarity and spectral analysis of random signals.
- Applications: analyzing noise in frequency domain, detection, and filtering.

CHAPTER4: Spectral Estimation (3 weeks)

- Bias, variance, and estimation error metrics
- Classical methods: periodogram, correlogram
- Parametric methods (AR, MA, ARMA)

- Yule-Walker, Levinson-Durbin algorithms.

CHAPTER5: Estimation Methods and Modeling (3 weeks)

- Estimation approaches: method of moments, least squares, maximum likelihood
- Model fitting for random signals
- Practical aspects: dealing with limited data, model selection (AIC, BIC)
- Implementation considerations in embedded contexts.

CHAPTER6: Introduction to Adaptive Filtering (3 weeks)

- Wiener filter fundamentals (MMSE, optimum filter)
- Kalman filter basics: state-space formulation, recursive updates
- Sensor fusion and real-time adaptation
- Example applications: noise cancellation, IMU-GPS fusion

Laboratory Work

- **Lab 1:** Stochastic Processes and Basic Statistics
- **Lab 2:** Autocorrelation and PSD
- **Lab 3:** Classical vs. Parametric Spectral Estimation
- **Lab 4:** Estimation Methods
- **Lab 5:** Wiener Filter
- **Lab 6:** Kalman Filter Application

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill, 2002.
- S. Haykin and B. Van Veen, *Signals and Systems*, 2nd ed. Hoboken, NJ, USA: Wiley, 2002.
- J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2007.
- N. C. Beaulieu, *Stochastic Processes and Random Signals*. Hoboken, NJ, USA: Wiley, 2013.
- M. Schwartz, *Information Transmission, Modulation, and Noise*, 4th ed. New York, NY, USA: McGraw-Hill, 1990.

Teaching Unit	Subject Title	Code	Semester
UEF1.2.2	**Operating Systems for Programmers**	**OS-P**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

Introduce students to OS internals and programming interfaces that developers use for system-level and application-level programming. Students will learn about process management, threading, inter-process communication, and debugging at the OS level, primarily in Linux.

Prerequisites

- Completion of "Operating Systems for Users and Administration" or equivalent knowledge.
- Proficiency in C programming.

Program Outline

CHAPTER 1: OS Internals Overview (2 weeks)

- Kernel vs. user space.
- System calls and API layers.
- Overview of Linux kernel modules.

CHAPTER 2: Process, Thread, and Scheduling (3 weeks)

- Process vs. thread concepts.
- Linux scheduling policies (CFS, real-time).
- Thread libraries (pthreads) and synchronization (mutexes, semaphores).

CHAPTER 3: Memory Management (3 weeks)

- Virtual memory, paging, swapping.
- Memory allocation (malloc, mmap).
- Shared memory segments and dynamic libraries.

CHAPTER4: Inter-Process Communication (IPC) (3 weeks)

- Pipes, FIFOs, message queues.
- Sockets for network communication.
- Signals and event-driven architectures.

CHAPTER5: File I/O and Device Drivers (2 weeks)

- File descriptors, blocking vs. non-blocking I/O.
- Character vs. block devices.
- Basics of writing a simple device driver module.

CHAPTER6: Debugging and Profiling (2 weeks)

- Tools: gdb, strace, ltrace.
- Performance profiling (perf, valgrind).
- Identifying memory leaks and bottlenecks.

Laboratory Work

- **Lab 1:** System Calls and Kernel Space vs. User Space
- **Lab 2:** Multithreading and Synchronization
- **Lab 3:** IPC Mechanisms
- **Lab 4:** Memory Management
- **Lab 5:** Basic Device Driver
- **Lab 6:** Debugging and Profiling

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- A. Tanenbaum and H. Bos, *Modern Operating Systems*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2014.
- R. Love, *Linux System Programming: Talking Directly to the Kernel and C Library*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2013.
- W. Stallings, *Operating Systems: Internals and Design Principles*, 9th ed. Upper Saddle River, NJ, USA: Pearson, 2017.
- M. K. McKusick, G. V. Neville-Neil, and R. N. M. Watson, *The Design and Implementation of the FreeBSD Operating System*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2015.
- B. W. Kernighan and R. Pike, *The Unix Programming Environment*. Upper Saddle River, NJ, USA: Pearson, 1984.

Teaching Unit (UE)	Subject Title	Code	Semester
UEF1.2.2	**Advanced networking**	**ANET**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h30	**3/3**

Objectives

This course aims to provide students with advanced networking skills specifically tailored for embedded systems and IoT environments. Learners will deepen their understanding of modern IP networking (IPv6), virtualization techniques, secure remote access (VPN), network programming, and network automation. Through theoretical and hands-on activities, students will be able to configure, manage, and automate networked embedded systems within industrial and IoT applications..

Prerequisites

- Basic understanding of computer systems and programming.
- Networking fundamentals for embedded systems

Program Outline

CHAPTER 1: IPv6 and Modern IP Networking (3 weeks)

- IPv6 protocol overview: addressing, subnetting, and features.
- Address configuration techniques: SLAAC, DHCPv6, and NDP.
- IPv4 to IPv6 transition techniques: tunneling, dual-stack, NAT64.
- IPv6 configuration on Linux-based embedded systems.

CHAPTER 2: Virtualization and Network Segmentation (3 weeks)

- VLAN (advanced concepts) and VXLAN (overlay networks).
- Network segmentation for industrial and IoT environments.
- Introduction to Docker networking
  - Bridge, host, and overlay networks.
  - Port mapping and container-to-container communication.

CHAPTER3: VPN and Secure Remote Access (3 weeks)

- VPN principles and architectures.
- IPsec VPN and OpenVPN: principles and configurations.
- VPN use cases in industrial and embedded systems.

CHAPTER4: Network Programming (Socket API) (3 weeks)

- Client-server model applied to socket programming.
- TCP and UDP socket programming.
- Basic services implementation: simple server/client app.

- Embedded systems and socket communication scenarios.

CHAPTER 5: Introduction to Network Automation (3 weeks)

- Fundamentals of network automation: objectives and benefits.
- Basic automation techniques for network management tasks .
- Application of automation in embedded and IoT systems.

Laboratory Work

- **Lab 1**: IPv6 Configuration.
- **Lab 2**: VLAN, VXLAN, and Docker Networking.
- **Lab 3**: VPN Setup (OpenVPN or IPsec).
- **Lab 4**: Simple TCP/UDP Client-Server in Python.
- **Lab 5**: Basic Network Automation with Bash or Python

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 8th ed. Boston, MA, USA: Pearson, 2020.
- A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2011.
- B. A. Forouzan, *Data Communications and Networking*, 5th ed. New York, NY, USA: McGraw-Hill, 2012.
- W. Stallings, *Data and Computer Communications*, 10th ed. Upper Saddle River, NJ, USA: Pearson, 2013.
- L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2020.

Teaching Unit	Subject Title	Code	Semester
UEM1.2.1	**Control Systems**	**CS**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**3/3**

Objectives

Provide an in-depth understanding of classical and advanced control systems, with a particular focus on applications in drones and embedded autonomous systems. Students will learn to model, analyze, and design control systems, then implement and validate them in both simulation and real embedded hardware.

Prerequisites

- Applied Mathematics: Linear algebra, Laplace/Fourier transforms, differential equations.
- Programming and Simulation: Basics of Python or C/C++ for embedded development..

Program Outline

CHAPTER 1: Fundamentals of Control Systems (2 weeks)

- Open-loop vs. closed-loop systems
- Laplace Transform basics for control (transfer functions)
- Real-world examples: drone stabilization, BLDC motor speed control.

CHAPTER 2: Modeling and Representation of Dynamic Systems (3 weeks)

- Differential equations, transfer functions, state-space representation
- Modeling mechanical, electrical, and embedded systems
- Practical case: Mathematical modeling of a drone.

CHAPTER 3: Stability and Performance Analysis (3 weeks)

- Stability criteria (Routh-Hurwitz, Nyquist, Bode)
- Time-domain and frequency-domain performance metrics
- Example: Stability verification for drone attitude control.

CHAPTER4: Classical Controllers: PID and Variants (3 weeks)

- P, PI, PD, and PID controllers
- Manual tuning vs. automatic methods (Ziegler-Nichols)
- Practical case: PID tuning for altitude or attitude hold in a drone.

CHAPTER5: Advanced Control Strategies for Drones (2 weeks)

- State-feedback control (LQR, $H\infty$, Backstepping)
- Model Predictive Control (MPC) for trajectory optimization
- Robust and adaptive control for uncertain environments.

CHAPTER6: Practical Implementation and Simulation (2 weeks)

- FS-Simulink connection to test PID controllers in simulated flight conditions
- Telemetry data analysis for control adjustment.

Laboratory Work

- **Lab 1:** Modeling and Simulation of a Drone
- **Lab 2:** PID Controller Design and Implementation
- **Lab 3:** LQR Controller on a Simulated Drone
- **Lab 4:** Drone Control in Flight Simulator
- **Lab 5:** PID Controller on STM32/ESP32
- **Lab 6:** Advanced Control (LQR or MPC) on a Real Drone

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 8th ed. Upper Saddle River, NJ, USA: Pearson, 2019.
- R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 13th ed. Upper Saddle River, NJ, USA: Pearson, 2016.
- N. S. Nise, *Control Systems Engineering*, 8th ed. Hoboken, NJ, USA: Wiley, 2019.
- J. J. D'Azzo and C. H. Houpis, *Linear Control System Analysis and Design with MATLAB*, 6th ed. Boca Raton, FL, USA: CRC Press, 2017.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM2.1.1	**Machine Learning**	**ML**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

This Lecturese provides foundational knowledge and practical skills in applying machine learning techniques within autonomous embedded systems. Students will learn how to process and analyze data generated by IoT devices, drones, and smart vehicles, integrating real-time decision-making capabilities into resource-constrained platforms.

Prerequisites

- Data Structures and Algorithms
- Basic understanding of Programming (Python or C++)
- Introductory knowledge of Mathematics (Linear Algebra, Probability)

Program Outline

CHAPTER 1 : Fundamentals of Machine Learning for Embedded Systems (2.5 weeks)

- Definition and scope of machine learning.
- Types of machine learning: supervised, unsupervised, reinforcement.
- Challenges of machine learning in embedded systems (e.g., resource constraints, real-time processing)..

CHAPTER 2: Linear Models for Regression and Classification (2.5 weeks)

- Linear regression: Theory, implementation, and evaluation.
- Logistic regression for binary and multi-class classification.
- Regularization techniques: L1 (Lasso) and L2 (Ridge).

CHAPTER3: Decision Trees and Ensemble Methods (2.5 weeks)

- Decision trees: Theory, implementation, and interpretation.
- Ensemble methods: Random Forests, Gradient Boosting Machines (GBM).
- Applications in embedded systems: Predictive maintenance, decision-making.

CHAPTER4: Support Vector Machines (SVM) and Instance-Based Learning (2.5 weeks)

- Support Vector Machines (SVM): Theory, kernel functions, and hyperparameter tuning.
- Instance-based learning: k-Nearest Neighbors (k-NN) algorithm.
- Applications in embedded systems: Object detection, anomaly detection.

CHAPTER 5: Introduction to Neural Networks in Embedded Applications (2.5 weeks)

- Basics of neural networks: Perceptrons, activation functions, and backpropagation.
- Shallow vs. deep neural networks.
- Applications in embedded systems: Image recognition, time-series forecasting.

CHAPTER 6: Performance Analysis, Benchmarking, and Case Studies (2.5 weeks)

- Model evaluation metrics: Accuracy, precision, recall, F1-score, ROC-AUC.
- Cross-validation and hyperparameter tuning.
- Case studies

Laboratory Work

- **Lab 1:** Introduction to machine learning tools for embedded systems.
- **Lab 2:** Implementing linear and logistic regression models.
- **Lab 3:** Building decision trees and ensemble models.
- **Lab 4:** Implementing SVM and k-NN algorithms.
- **Lab 5:** Introduction to neural networks and deployment on embedded system.
- **Lab 6:** Performance analysis

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.
- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
- K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM1.2.1	**Regulations & Standards for Unmanned System**	**RSUS**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	0	**2/2**

Objectives

This seminar-based module aims to provide students with a broad understanding of the regulatory frameworks, technological norms, and national/international standards that govern unmanned systems (UAVs, UGVs, UMVs, etc.). Through flexible sessions led by guest professionals, students will::

- Explore the latest national and international regulations for unmanned systems, adapting to evolving legal landscapes.
- Discover relevant safety and quality standards (ISO, CE, EASA, FAA, etc.) and the associated certification processes.
- Analyze how these regulations and norms affect the technical design, deployment, and operation of drones and autonomous vehicles.
- Discuss real-world case studies and ethical/legal implications, guided by the expertise of invited speakers.

Prerequisites

No prerequisites

Program Outline

Because each session's content is shaped by invited experts, the following chapters represent thematic goals rather than a fixed schedule. The exact emphasis on regulatory, technological, or national/international norms is adapted to each guest's expertise.:

- **Foundations of Unmanned Systems Regulations**
  - Overview of global regulatory bodies (ICAO, EASA, FAA, etc.).
  - National vs. international norms, typical legal frameworks.
  - Ethical considerations: privacy, safety, data protection.
- **National & International Standards**
  - ISO, IEC, CE compliance for UAVs and autonomous systems.
  - Industry best practices: fail-safe designs, reliability standards.
  - Case examples: how specific countries approach UAV certification.
- **Technological Norms & Certification**
  - Processes for **safety certification** (e.g., DO-178C for avionics software).
  - Design constraints: sensor redundancy, flight termination systems.
  - Guest lecture from an industrial manufacturer or certification agency.
- **Operational Constraints & Airspace Integration**
  - Integrating UAVs into controlled airspace or roads (depending on domain).
  - Notions of UTM (UAS Traffic Management) for drones.
  - Communication with authorities, obtaining flight permissions.
- **Case Studies & Emerging Technologies**
  - Real examples of cargo drones, inspection drones, or driverless vehicles.

    - Challenges in advanced tech (BVLOS, swarms, AI-based autonomy).
    - Panel discussion on next-generation standards (5G, detect-and-avoid systems).
- **Legal, Ethical, and Social Implications**
    - Liability issues in accidents or misuse of autonomous vehicles.
    - Cross-border operations: dealing with multiple jurisdictions.
    - Roundtable with legal experts on future policy directions.

Laboratory Work

No laboratory works

Evaluation Method

Module average = 100% continuous assessment.

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject. The maximum value of the level threshold is set at 10/20.

Teaching Unit (UE)	Subject Title	Code	Semester
UET1.2.1	Capstone Project I	**CP1**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	1h00	**2/2**

Objectives

- Form teams and identify project ideas.
- Conduct feasibility analysis and initial design documentation.
- Establish milestones and present progress to faculty advisors.
- Teamwork and collaboration.
- Problem identification and solution structuring.
- Technical writing and presentation skills.
- Project planning and feasibility assessment.

Prerequisites

No prerequisites

Program Outline

1. **Team Formation and Brainstorming (6 hours)**
   - Introduction to project selection.
   - Team roles and responsibilities.
   - Idea generation techniques.
2. **Feasibility Analysis and Initial Design Documentation (7.5 hours)**
   - Market and technical feasibility.
   - Risk assessment.
   - Drafting project proposals and documentation.
3. **Regular Milestones and Faculty Presentations (9 hours)**
   - Setting up project goals and deadlines.
   - First milestone presentation.
   - Refining project scope based on feedback.

Laboratory Work

No laboratory works

Evaluation Method

Module average = 100% continuous assessment.

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject. The maximum value of the level threshold is set at 10/20.

Teaching Unit (UE)	Subject Title	Code	Semester
UED1.2.1	Training Internship I	**TI1**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	0	**1/1**

Objectives

- Introduce students to professional work environments.
- Develop basic technical and soft skills in an industry setting.
- Understand company structure, workflow, and project dynamics.

Prerequisites

No prerequisites

Program Outline

1. **Introduction to the Company**
   - Company overview and mission.
   - Workplace safety and regulations.
2. **Observation and Shadowing**
   - Understanding team roles and responsibilities.
   - Exposure to ongoing projects.
3. **Hands-on Experience**
   - Assisting in minor tasks and operations.
   - Learning basic tools and methodologies.
4. **Final Review and Feedback**
   - Supervisor feedback session.
   - Internship report preparation.

Laboratory Work

No laboratory works

Evaluation Method

Module average = 100% continuous assessment.

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject. The maximum value of the level threshold is set at 10/20.

d.Detailed programs for the semester 3

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.1	**Power Electronics**	**PE**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0h45	1h00	**3/3**

Objectives

Provide students with an in-depth understanding of power semiconductor devices and power converter topologies (AC/DC, DC/DC, AC/AC, DC/AC). By the end of this module, students will be able to design, analyze, and implement power electronic circuits for embedded systems such as drones, electric vehicles, and industrial drives..

Prerequisites

- Basic electronics (diodes, transistors, operational amplifiers)
- Fundamentals of electrical circuits (R, L, C, and simple AC analysis)
- Introductory knowledge of microcontroller-based control (PWM, gating signals)

Program Outline

CHAPTER 1: Introduction and Power Semiconductors (02 weeks)

- Mechanical and Solid-State Relays: fundamentals, switching characteristics
- Power Semiconductor Devices: diode, thyristor (SCR), power transistor (BJT, MOSFET, IGBT)
- Static and dynamic characteristics, safe operating areas
- Heat dissipation, packaging, and cooling methods.

CHAPTER 2: AC/DC Converters (Rectifiers) (03 weeks)

- Uncontrolled Rectifiers: single-phase and three-phase with R, RL, RLE load; freewheel diode usage
- Controlled Rectifiers: single-phase and three-phase with R, RL, RLE load; thyristor-based control
- Analysis of average output voltage, ripple, and power factor

CHAPTER3: DC/DC Converters (Choppers) (03 weeks)

- Buck, Boost, and Buck-Boost Topologies
- Series vs. parallel choppers with various loads (R, RL, LC, RC, RLC)
- Continuous vs. discontinuous conduction mode
- PWM control strategies and efficiency considerations

CHAPTER4: AC/AC Converters (Dimmers) (02 weeks)

- Single-Phase AC Voltage Controllers: R, RL, RLC loads
- Three-Phase AC Voltage Controllers: unidirectional and bidirectional dimmers
- Applications: lighting control, motor speed control (AC induction motors).

CHAPTER5: DC/AC Inverters (03 weeks)

- Single-Phase Inverters: half-bridge, full-bridge with R, RL loads
- Three-Phase Inverters: 120°, 180° conduction modes, voltage control (SPWM)
- Harmonic analysis, THD, and filtering techniques.

CHAPTER 5: Applications and Advanced Topics (02 weeks)

- Power factor correction (PFC) basics
- Battery charging circuits and advanced converter topologies (resonant converters)
- Introduction to ESCs for brushless DC motors (BLDC), multi-level inverters
- Practical design considerations (EMI, snubbers, gate drivers).

Laboratory Work

- **Lab 1:** Semiconductor Characterization.
- **Lab 2:** Uncontrolled and Controlled Rectifiers.
- **Lab 3:** DC/DC Buck and Boost Converters.
- **Lab 4:** AC Voltage Control (Dimmer).
- **Lab 5:** Single-Phase Inverter.
- **Lab 6:** Advanced Converter or ESC for BLDC.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- M. H. Rashid, *Power Electronics: Circuits, Devices, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2013.
- N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd ed. Hoboken, NJ, USA: Wiley, 2002.
- D. W. Hart, *Power Electronics*, 1st ed. New York, NY, USA: McGraw-Hill, 2010.
- B. K. Bose, *Modern Power Electronics and AC Drives*. Upper Saddle River, NJ, USA: Pearson, 2001.
- R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, 2nd ed. New York, NY, USA: Springer, 2001.

Teaching Unit (UE)	Subject Title	Code	Semester
UEF2.1.1	**Industrial Networks and Communication Protocols**	**INCP**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0h45	1h30	**3/3**

Objectives

This Lecturese provides a comprehensive understanding of industrial networks and communication protocols in modern industrial systems, including real-time constraints, fieldbus communication, SCADA integration, and smart grid applications. The focus is on Industry 4.0, industrial IoT (IIoT), and embedded systems communication.

Prerequisites

- Basic knowledge of network protocols and communication systems
- Basic programming and data communication concepts

Program Outline

CHAPTER 1: Introduction to Industrial Networks (2 weeks)

- Definitions and key concepts of industrial networking
- Evolution of industrial technologies towards Industry 4.0
- CIM Pyramid: Industrial automation hierarchy
- EPA Layer Model: Standardized industrial communication model
- Classification of industrial systems and applications.

CHAPTER 2: Basic Access Protocols (4 weeks)

- UART-based protocols: RS232, RS485, RS422
- IEEE 802-derived protocols: Ethernet, Wi-Fi, TSN
- Low-level communication protocols: I2C, SPI, 1-wire, HART

CHAPTER3: Fieldbus and Industrial Local Networks (4 weeks)

- Definitions and key challenges in industrial networking
- Architectures, topologies, and industrial constraints
- DCS vs PLC: Distributed Control Systems vs Programmable Logic Controllers
- Fieldbus communication protocols: Modbus (RTU/TCP), PROFIBUS/PROFINET, CAN, IEC 61850

CHAPTER4: SCADA Systems and Industrial Automation (04 weeks)

- SCADA Systems: Definition and key challenges

- SCADA architectures, topologies, and evolution
- Reliability and security in industrial networks (IEC 62351)
- Industrial transport protocols: DNP3, IEC 60870-5-101, IEC 60870-5-104, Pur2.4

Laboratory Work

- **Lab 1:** UART-based protocols
- **Lab 2:** IEEE 802-derived protocols
- **Lab 3:** Low-level communication protocols.
- **Lab 4:** Fieldbus Communication.
- **Lab 5:** Time Synchronization with PTP.
- **Lab 6:** Industrial transport protocols.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- T. Sauter, *Industrial Communication Technology Handbook*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2014.
- G. Clarke and D. Reynders, *Practical Modern SCADA Protocols: DNP3, 60870.5, and Related Systems*. Burlington, MA, USA: Newnes, 2004.
- J. D. Decotignie, *Ethernet-Based Real-Time and Industrial Communications*. Hoboken, NJ, USA: Wiley, 2005.
- S. Mackay, E. Wright, D. Reynders, and J. Park, *Practical Industrial Data Networks: Design, Installation, and Troubleshooting*. Burlington, MA, USA: Newnes, 2004.
- K. C. Chang, *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*, 2nd ed. Waltham, MA, USA: Syngress, 2014.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.1	**Digital Signal Processors**	**DSP**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**3/3**

Objectives

Enable students to understand and utilize Digital Signal Processors for high-speed, real-time signal processing tasks. By mastering DSP architecture, fixed/floating-point arithmetic, and specialized instructions, students can implement efficient filtering, transforms, and adaptive algorithms in embedded environments.

Prerequisites

- Basics of signal processing (Fourier transforms, convolution, sampling theory)
- Fundamentals of microcontroller architectures.

Program Outline

CHAPTER 1: Introduction to DSPs and Applications (2 weeks)

- What is a Digital Signal Processor (DSP)?
- DSP vs. general-purpose processors
- Use cases: audio, image processing, control systems, telecommunications

CHAPTER 2: DSP Architectures (3 weeks)

- Core features: MAC unit, parallelism, pipelining
- Harvard architecture in DSPs
- Examples of popular DSP families (TI C6000, ARM Cortex-M4/M7 DSP extensions)

CHAPTER 3: Fixed-Point vs. Floating-Point DSPs (3 weeks)

- Precision, scaling, and quantization in fixed-point arithmetic
- Floating-point DSPs for high-performance tasks
- Trade-offs between fixed-point and floating-point implementations

CHAPTER4: Signal Processing Algorithms on DSPs (3 weeks)

- FFT and real-time frequency analysis
- FIR and IIR filters: implementation and optimization
- Adaptive filtering: LMS algorithm basics.

CHAPTER5: DSP Peripherals and Real-Time Considerations (2 weeks)

- DMA for high-speed data transfer
- Interfacing DSPs with ADCs/DACs
- Scheduling and interrupt handling in real-time DSP applications

CHAPTER6: Trends and Case Studies in DSPs (2 weeks)

- DSP in IoT and edge computing
- Integration of DSP features in ARM Cortex-M processors
- Case studies: audio equalizers, radar systems, motor control.

Laboratory Work

- **Lab 1:** DSP Development Environment
- **Lab 2:** FIR Filter Implementation
- **Lab 3:** FFT Implementation and Signal Analysis
- **Lab 4:** Interfacing DSP with ADC/DAC
- **Lab 5:** Adaptive Filtering on DSP
- **Lab 6:** Mini Project

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, 4th ed. New York, NY, USA: McGraw-Hill, 2010.
- A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2007.
- R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. San Diego, CA, USA: California Technical Publishing, 1999.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.1	**Multi-Sensor Data Fusion**	**MSDF**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**2/2**

Objectives

This module covers fundamental concepts, advantages, and applications of multi-sensor data fusion. Students will learn about sensor models, data acquisition, preprocessing, fusion methods (e.g., Bayesian, Kalman filter), architectures (centralized vs. distributed), and performance evaluation. By the end, they will be able to design, implement, and assess multi-sensor data fusion strategies for embedded systems, robotics, and autonomous applications..

Prerequisites

- Basic probability, statistics, and linear algebra
- Microcontroller programming (C/C++),
- Sensor interfacing (SPI, I2C).

Program Outline

CHAPTER 1: Introduction to Multi-Sensor Data Fusion (2 weeks)

- Basic concepts and definitions
- Advantages and applications in robotics, drones, IoT
- Overview of fusion levels (signal, feature, decision)

CHAPTER 2: Sensor Models and Data Acquisition (3 weeks)

- Types of sensors (IMU, GPS, cameras, LiDAR, etc.)
- Key sensor characteristics (accuracy, resolution, sampling rate)
- Techniques for sensor data acquisition (synchronization, buffering)

CHAPTER 3: Data Preprocessing (3 weeks)

- Filtering noisy data (low-pass, median, etc.)
- Sensor calibration (offset, scaling)
- Correction of systematic errors

CHAPTER4: Data Fusion Techniques (3 weeks)

- Basic methods: averaging, weighted combination
- Bayesian approaches, Kalman filtering (and variants)
- Static vs. dynamic fusion approaches.

CHAPTER5: Data Fusion Architectures (2 weeks)

- Centralized vs. distributed fusion
- Hierarchical and cascading architectures

- Communication and synchronization among multiple sensors

CHAPTER6: Performance Evaluation and Case Studies (2 weeks)

- Metrics for assessing fusion performance (RMSE, variance, etc.)
- Error analysis and system limitations
- Real-world case studies (autonomous vehicles, UAVs, sensor networks).

Laboratory Work

- **Lab 1:** Basic Fusion Concepts
- **Lab 2:** Sensor Calibration and Preprocessing
- **Lab 3:** Kalman Filter Introduction
- **Lab 4:** Multi-Sensor Fusion Architecture
- **Lab 5:** Real-Time Data Fusion on Embedded Hardware
- **Lab 6:** Performance Evaluation and Case Study

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- H. B. Mitchell, *Data Fusion: Concepts and Ideas*, 2nd ed. New York, NY, USA: Springer, 2012.
- D. L. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2001.
- L. A. Klein, *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making*, 2nd ed. Bellingham, WA, USA: SPIE Press, 2004.
- D. Blasch, E. Bosse, and D. A. Lambert, *High-Level Information Fusion Management and Systems Design*. Norwood, MA, USA: Artech House, 2012.
- M. E. Liggins, D. L. Hall, and J. Llinas, *Multisensor Data Fusion: Theory and Practice*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2017.

Teaching Unit (UM)	Subject Title	Code	Semester
UEF2.1.2	**Introduction to Real-Time Operating Systems**	**IRTOS**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

This module aims to provide students with a thorough understanding of real-time operating systems (RTOS) and the constraints that guide the design of real-time systems. By the end of the Lecturese, students will:

- Understand Real-Time Concepts: Distinguish between hard and soft real-time requirements, and recognize the unique constraints of embedded systems.
- Master RTOS Fundamentals: Learn about microkernel vs. monolithic architectures, deterministic behavior, and how an RTOS differs from a general-purpose OS.
- Design and Schedule Tasks: Implement priority-based scheduling, apply real-time scheduling algorithms (RMS, EDF, DMS), and handle task synchronization (mutexes, semaphores).
- Manage Resources and Memory: Explore static vs. dynamic allocation, memory pool strategies, and handle deadlock prevention and priority inversion.
- Handle Interrupts and Timing: Minimize interrupt latency, design ISRs, use timers/watchdogs, and ensure time-critical tasks meet their deadlines..

Prerequisites

- Operating Systems Basics: Familiarity with processes, threads, scheduling, and basic OS architecture.
- Programming Skills: Proficiency in C/C++ for embedded/RTOS coding.
- Embedded Systems Fundamentals: Knowledge of microcontrollers, memory constraints, and peripheral interfacing is recommended..

Program Outline

CHAPTER 1: Introduction to Real-Time Systems (1 weeks)

- Real-time concepts and terminology
- Hard vs. soft real-time requirements
- Embedded system constraints.

CHAPTER 2: RTOS Architecture Fundamentals (1 weeks)

- Microkernel vs. monolithic designs
- Deterministic behavior characteristics
- RTOS vs. general-purpose OS differences

CHAPTER 3: Task Management in RTOS (1.5 weeks)

- Task states and lifecycle
- Priority-based scheduling
- Context switching mechanisms.

CHAPTER4: Real-Time Scheduling Algorithms (1.5 weeks)

- Rate Monotonic Scheduling (RMS)
- Earliest Deadline First (EDF)
- Deadline Monotonic Scheduling (DMS).

CHAPTER5: Task Synchronization and Communication (1.5 weeks)

- Mutex and semaphore implementations
- Priority inversion and inheritance
- Message queues and mailboxes

CHAPTER6: Memory Management for Embedded Systems (1.5 weeks)

- Static vs. dynamic allocation
- Memory pool strategies
- Stack usage analysis and optimization

CHAPTER7: Interrupt Handling in RTOS (2 weeks)

- Interrupt latency considerations
- Interrupt service routines
- Deferred processing techniques

CHAPTER8: Timer Services and Time Management (2 weeks)

- High-resolution timers
- Timeouts and watchdogs
- Clock synchronization

CHAPTER9: Resource Management and Deadlock Prevention (2 weeks)

- Resource allocation strategies
- Deadlock detection and avoidance
- Bounded priority inversion solutions

Laboratory Work

- **Lab 1:** Basic RTOS Setup and Task Creation
- **Lab 2:** Real-Time Scheduling and Deadline Adherence
- **Lab 3:** Synchronization and Priority Inversion
- **Lab 4:** Interrupt Handling and Timer Services
- **Lab 5:** Memory Management and Resource Control
- **Lab 6:** Embedded Linux Deployment

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- C. Hallinan, *Embedded Linux Primer: A Practical Real-World Approach*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- K. Yaghmour, *Embedded Android: Porting, Extending, and Customizing*. Sebastopol, CA, USA: O'Reilly Media, 2013.
- J. Masters, *Mastering Embedded Linux Programming*, 2nd ed. Birmingham, UK: Packt Publishing, 2017.
- D. Abbott, *Linux for Embedded and Real-Time Applications*, 4th ed. Burlington, MA, USA: Newnes, 2016.
- M. Tim Jones, *Linux Application Development*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004.

Teaching Unit (UT)	Subject Title	Code	Semester
UEF2.1.2	**Digital Filtering**	**DF**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**3/3**

Objectives

This module aims to provide students with the theoretical foundations and practical skills needed to design and implement digital filters. Students will learn about filter specifications, types, and the trade-offs involved in digital filter design. They will also gain hands-on experience with tools for implementing filters in software and hardware.

Prerequisites

- Fourier Transform (continuous and discrete).
- Familiarity with convolution and frequency-domain representations of signals.

Program Outline

CHAPTER 1: Introduction to Digital Filters (2 weeks)

- Overview of filtering and its role in digital signal processing.
- Classification of filters: FIR (Finite Impulse Response) and IIR (Infinite Impulse Response).
- Applications of digital filters in communications, audio, and image processing.

CHAPTER 2: The Z-Transform and Its Applications (3 weeks)

- Ideal filter characteristics and design specifications (passband, stopband, ripple).
- Window-based FIR design methods: rectangular, Hamming, Hann, Blackman.
- Frequency sampling method.

CHAPTER 3: FIR Filter Design (3 weeks)

- Ideal filter characteristics and design specifications (passband, stopband, ripple).
- Window-based FIR design methods: rectangular, Hamming, Hann, Blackman.
- Frequency sampling method.

CHAPTER4: IIR Filter Design (3 weeks)

- Introduction to IIR filters: Butterworth, Chebyshev, Elliptic filters.
- Pole-zero placement and stability considerations.
- Design methods: impulse invariance, bilinear transformation.

CHAPTER5: Implementation of Digital Filters (2 weeks)

- Structures for implementing digital filters: direct form, cascade form, parallel form.
- Fixed-point vs. floating-point implementations.
- Introduction to real-time filtering on embedded platforms.
- Hands-on: Implementation of real-time filters on software/hardware.

CHAPTER6: Filter Performance Analysis and Optimization (2 weeks)

- Analysis of filter performance: frequency response, phase response, group delay.
- Computational efficiency and trade-offs in real-time filtering.
- Advanced concepts: multi-rate filtering, decimation, and interpolation.

Laboratory Work

- **Lab 1:** FIR Filter Design and Simulation
- **Lab 2:** IIR Filter Design and Stability Analysis
- **Lab 3:** Real-Time Digital Filtering

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2007.
- S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, 4th ed. New York, NY, USA: McGraw-Hill, 2010.
- R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. San Diego, CA, USA: California Technical Publishing, 1999.

Teaching Unit (UT)	Subject Title	Code	Semester
UEF2.1.2	**Human-Machine Interface (HMI) for embedded systems**	**HMI**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**2/2**

Objectives

This module aims to train engineers in the design and development of human-machine interfaces (HMI) tailored for embedded systems, with a focus on drones and autonomous aircraft. The approach includes case studies and practical applications using Microsoft Flight Simulator (MSFS) for real-time visualization and interaction

Prerequisites

- Basic Knowledge of Embedded Systems:
- Experience with Programming and Development Tools
- Basic Knowledge of Communication Protocols

Program Outline

CHAPTER 1: Introduction to HMIs in Embedded Systems (2 weeks)

- Definition and role of HMIs in embedded systems.
- Technical constraints and challenges (latency, limited resources, interoperability).
- Examples of interfaces for drone piloting and telemetry
- Future Trends: Touch, voice, AR/VR (brief overview)

CHAPTER 2: Principles of UI/UX Design for Embedded (2 weeks)

- Ergonomics & UX: Designing for small screens, constrained environments
- UI Frameworks & Tools: Qt for embedded, LVGL, TouchGFX
- Display of Critical Parameters: Altitude, speed, attitude, battery level
- Interactive Controls: Touch, haptics, physical buttons
- Performance & Aesthetics: Balancing design complexity with resource limits.

CHAPTER 3: Technologies & Communication Protocols (3 weeks)

- Hardware Interfaces: LCD/TFT displays, touchscreens, mechanical buttons
- Communication with Drones: MAVLink basics, UART, SPI, I2C usage
- Sensor & Actuator Integration: Reading from IMU, sending control signals
- Real-Time Data Handling: Buffering, polling vs. interrupts, concurrency issues.

CHAPTER4: Embedded HMI Development & Implementation (3 weeks)

- Programming Languages: C/C++ and Python for UI logic
- Framework Usage: Step-by-step with Qt, LVGL, or TouchGFX
- Performance Optimization: Memory footprint, CPU usage, double buffering
- Hardware Platforms: STM32, ESP32, Raspberry Pi (brief comparison).

CHAPTER5: Practical Experimentation with Flight Simulators (2 weeks)

- MSFS for Aeronautical Simulation: Overview, real-time data streams
- Connecting MSFS: SimConnect/FSUIPC for flight data exchange
- HMI Supervision Interface: Displaying altitude, speed, heading, battery status
- Virtual Cockpit: Secondary displays, interactive controls, advanced telemetry
- Testing Autonomous Missions: Real-time flight data tracking, mission planning.

CHAPTER6: Security & Resilience of Embedded HMIs (2 weeks)

- Cybersecurity: Potential attacks (injection, spoofing), encryption, authentication
- Fault Tolerance: Handling UI crashes, fail-safe modes in critical environments
- Testing & Validation: Simulation vs. real hardware, stress tests, user acceptance
- Best Practices: Input validation, secure data channels, hardened OS layers.

Laboratory Work

- **Lab 1:** Basic HMI Concept & Setup
- **Lab 2:** UI/UX Design & Touch Input
- **Lab 3:** Communication Protocols with Drone Data
- **Lab 4:** Embedded HMI Implementation
- **Lab 5:** MSFS Integration & Virtual Cockpit
- **Lab 6:** Security & Resilience Testing

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*, 5th ed. Hoboken, NJ, USA: Wiley, 2019.
- B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, and N. Elmqvist, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 6th ed. Upper Saddle River, NJ, USA: Pearson, 2017.
- A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2003.
- D. Norman, *The Design of Everyday Things*, Revised and Expanded ed. New York, NY, USA: Basic Books, 2013.
- J. Johnson, *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*, 3rd ed. Cambridge, MA, USA: Morgan Kaufmann, 2020.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM2.1.1	**Digital Control Systems**	**DCS**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**3/3**

Objectives

Focus on digital control systems and their embedded implementation for drones and autonomous systems. Students will learn discretization techniques, analyze stability in the discrete domain, and program real-time digital controllers on microcontrollers..

Prerequisites

- Basic analog control knowledge (stability, PID)
- Dynamic systems modeling (differential equations)
- Mathematics and Signal Theory (Laplace, Fourier, sampling)
- Microcontrollers (STM32, ESP32) and sensor fundamentals

Program Outline

CHAPTER 1 : Fundamentals of Digital Control (2.5 weeks)

- Analog vs. digital control systems
- Sampling and the Nyquist-Shannon theorem
- Applications to drones and embedded systems.

CHAPTER 2: Z-Transform and Modeling of Digital Systems(2.5 weeks)

- Definition of the Z-transform
- Discretization methods (Euler, Tustin)
- Example: Converting an analog PID to a discrete PID.

CHAPTER3: Stability and Performance in Discrete Time(2.5 weeks)

- Pole-zero analysis in the z-plane
- Time-domain performance (rise time, overshoot, steady-state error)
- Frequency-domain considerations for discrete systems.

CHAPTER4: Design and Implementation of Digital Controllers (2.5 weeks)

- Discrete PID and advanced variants
- Effects of sampling and computation delay
- Practical case: implementing a discrete PID.

CHAPTER 5: Optimal and Robust Digital Control (2.5 weeks)

- Discrete Model Predictive Control (MPC)
- State feedback (discrete LQR, H∞)
- Practical case: Optimal control for a drone in discrete time.

CHAPTER 6: Practical Simulation and Embedded Implementation (2.5 weeks)

- Flight Simulator (MSFS) integration for digital controllers
- Testing discrete PID, LQR, MPC under simulated conditions
- Analyzing real-time discrete responses with actual data

Laboratory Work

- **Lab 1:** Discretization of a PID Controller.
- **Lab 2:** Discrete PID on a Simulated Drone.
- **Lab 3:** Discrete LQR for Autonomous Drones
- **Lab 4:** Kalman Filtering for IMU Data.
- **Lab 5:** Testing Digital Controllers in Flight Simulator
- **Lab 6:** Model Predictive Control (MPC) for Drones

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- K. Ogata, *Discrete-Time Control Systems*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 1995.
- G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 1997.
- C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2014.
- M. Gopal, *Digital Control and State Variable Methods*, 4th ed. New York, NY, USA: McGraw-Hill, 2020.
- R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 13th ed. Upper Saddle River, NJ, USA: Pearson, 2016.

Teaching Unit (UE)	Subject Title	Code	Semester
UET1.2.1	**Deep learning**	**DL**	**S3**

	Lectures	TD	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

This Lecturese provides foundational knowledge and practical skills in applying deep learning techniques within autonomous embedded systems. Students will learn how to design, train, and optimize deep neural networks for tasks such as image recognition, time-series forecasting, and real-time decision-making in resource-constrained platforms.

Prerequisites

- Basic understanding of Machine Learning (e.g., linear models, decision trees).
- Programming experience in Python.
- Introductory knowledge of Mathematics (Linear Algebra, Calculus).

Program Outline

CHAPTER 1: Introduction to Deep Learning in Embedded Systems (2.5 weeks)

- Key concepts: Perceptrons, activation functions, and backpropagation.
- Challenges of deep learning in embedded systems (e.g., computational constraints, memory limitations).

CHAPTER 2: Training Deep Neural Networks: Optimization(2.5 weeks)

- Gradient descent and its variants: SGD, Adam, RMSprop.
- Loss functions and their role in training.
- Challenges in training deep networks: Vanishing/exploding gradients.

CHAPTER3: Teaching Deep Learners to Generalize: Regularization(2.5 weeks)

- Overfitting and underfitting in deep learning.
- Regularization techniques: Dropout, weight decay, batch normalization.
- Data augmentation for improving generalization.

CHAPTER4: Convolutional Neural Networks (CNNs) (2.5 weeks)

- Basics of CNNs: Convolutional layers, pooling, and feature maps.
- Applications in embedded systems: Image recognition, object detection.
- Lightweight CNN architectures for embedded systems (e.g., MobileNet, SqueezeNet).
- Optimization of CNNs for embedded systems (MobileNet, EfficientNet, Edge TPU).

CHAPTER 5: Recurrent and Recursive Neural Networks(2.5 weeks)

- Basics of RNNs: Sequence modeling, LSTM, and GRU.
- Applications in embedded systems: Time-series forecasting, natural language processing.
- Challenges of RNNs in embedded systems (e.g., memory usage, latency).

CHAPTER 6: Deep Generative Models and Transformers(2.5 weeks)

- Introduction to generative models: GANs, Boltzmann Machines, Autoencoders.
- Transformers and attention mechanisms: Theory and applications.
- Applications in embedded systems: Data synthesis, anomaly detection.

Laboratory Work

- **Lab 1:** Introduction to deep learning tools for embedded systems.
- **Lab 2:** Training and optimizing deep neural networks.
- **Lab 3:** Implementing regularization techniques.
- **Lab 4:** Building and deploying CNNs for image recognition.
- **Lab 5:** Implementing RNNs for time-series forecasting.
- **Lab 6:** Exploring generative models and transformers.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- F. Chollet, *Deep Learning with Python*, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2021.
- C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. New York, NY, USA: Springer, 2018.
- A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
- Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from Data: A Short Lecturese*. Pasadena, CA, USA: AMLBook, 2012.

Teaching Unit (UE)	Subject Title	Code	Semester
UET2.1.1	**Operations Research**	**OR**	**S3**

	Lectures	TD	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**2/2**

Objectives

This Lecturese provides foundational knowledge and practical skills in Operations Research, focusing on Linear Optimization and Graph Theory. Students will learn to model and solve optimization problems, analyze graphs, and apply these techniques to real-world problems in logistics, scheduling, and resource allocation.

Prerequisites

- Basic understanding of Linear Algebra.
- Programming experience.
- Introductory knowledge of Algorithms.

Program Outline

PART 1: Linear Optimization (8 weeks)

CHAPTER 1: Generalities of Linear Programming (2.5 weeks)

- Definition and scope of linear programming.
- Formulating linear programming problems: Objective function, constraints, and feasible region.
- Graphical solution method for two-variable problems.

CHAPTER 2: Simplex Algorithm (3 weeks)

- Theory of the simplex algorithm: Basic feasible solutions, pivot operations.
- Solving linear programming problems using the simplex method.
- Handling special cases: Degeneracy, unbounded solutions, and infeasibility.

CHAPTER 3: Post-Optimal Analysis and Duality(2.5 weeks)

- Sensitivity analysis: Changes in objective function coefficients and constraint values.
- Dual problem formulation and interpretation.
- Complementary slackness and economic interpretation of duality.

PART 2: Graph Theory (6 weeks)

CHAPTER 4: Definitions and Basic Concepts (2 weeks)

- Basic definitions: Graphs, vertices, edges, paths, and cycles.
- Types of graphs: Directed vs. undirected, weighted vs. unweighted.
- Graph representations: Adjacency matrix and adjacency list.

CHAPTER 5: Connectivity in Graphs. (2 weeks)

- Connectivity and components in graphs.
- Eulerian and Hamiltonian paths and cycles.
- Applications in network design and routing.

CHAPTER 6: Scheduling Problems (2 weeks)

- Definition and scope of scheduling
- GANTT Chart
- PERT Method
- MPM Method
- Applications

Laboratory Work

- **Lab 1:**Formulating and solving linear programming problems using Python..
- **Lab 2:**Implementing the simplex algorithm for linear optimization.
- **Lab 3:**Performing sensitivity analysis and solving dual problems.
- **Lab 4:**Implementing graph representations and basic algorithms (BFS, DFS).
- **Lab 5:**Solving connectivity problems and finding Eulerian/Hamiltonian paths.
- **Lab 6:**Implementing graph coloring algorithms and solving scheduling problems.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

1. F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 11th ed. New York, NY, USA: McGraw-Hill, 2020.
2. H. A. Taha, *Operations Research: An Introduction*, 10th ed. Upper Saddle River, NJ, USA: Pearson, 2017.
3. R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, 5th ed. New York, NY, USA: Springer, 2020.
4. W. L. Winston, *Operations Research: Applications and Algorithms*, 4th ed. Boston, MA, USA: Cengage Learning, 2003.
5. M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 4th ed. Hoboken, NJ, USA: Wiley, 2009.

Teaching Unit (UE)	Subject Title	Code	Semester
UET2.1.1	**Introduction to embedded system security**	**ICS**	**S3**

	Lectures	TD	LABS	Crédit/ Coeff
VHH	1h00	0	0	**2/2**

Objectives

This module introduces students to the fundamental principles of embedded system security. It provides an understanding of the unique challenges, threats, and vulnerabilities in embedded systems and covers basic security concepts like confidentiality, integrity, and availability (CIA) in embedded environments.

Prerequisites

- Basic knowledge of embedded systems (architecture and operation).
- Fundamentals of programming.
- Familiarity with microcontroller-based systems

Program Outline

CHAPTER 1: Security Challenges in Embedded Systems (3 weeks)

- Threats and vulnerabilities specific to embedded systems.
- Real-world examples.
- Security lifecycle: Threat modeling and risk assessment..

CHAPTER 2: Security Principles and Best Practices (3 weeks)

- CIA Triad: Confidentiality, Integrity, Availability.
- Secure design principles: Least privilege, defense in depth, and fail-safe defaults.
- Secure coding practices for embedded software.

CHAPTER3: Cybersecurity Frameworks and Standards (3 weeks)

- Overview of cybersecurity standards: ISO 27001, NIST Cybersecurity Framework.
- Risk management frameworks: Identifying, assessing, and mitigating risks.
- Incident response and recovery planning..

CHAPTER4: Securing Systems and Networks (3 weeks)

- Security layers: Firewalls, antivirus, intrusion detection systems (IDS).
- Basics of network security: VPNs, TLS/SSL protocols.
- Introduction to Zero Trust Architecture.

CHAPTER 5: Emerging Trends and Case Studies (2 weeks)

- Overview of cyber threats in IoT and embedded systems.
- Real-world case studies: Cyberattacks and countermeasures.

- Future challenges in cybersecurity: AI-driven attacks, quantum computing risks.

Laboratory Work

No labs.

Evaluation Method

Module average = 40% continuous assessment + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- W. Stallings, “Computer Security: Principles and Practice”, 4th ed., Pearson, 2018.
- C. P. Pfleeger, S. L. Pfleeger, “Security in Computing”, 5th ed., Pearson, 2015.
- M. Bishop, “Computer Security: Art and Science”, 2nd ed., Pearson, 2018.
- R. Anderson, “Security Engineering: A Guide to Building Dependable Distributed Systems”, 3rd ed., Wiley, 2020.
- B. Schneier, “Secrets and Lies: Digital Security in a Networked World”, Wiley, 2004.

Teaching Unit (UE)	Subject Title	Code	Semester
UED2.1.1	Project Management for Engineers	**PME**	**S3**

	Lectures	TD	LABS	Crédit/ Coeff
VHH	1h00	0	0	**1/1**

Objectives

By the end of this course, students will:

- Understand the fundamentals of project management in an engineering context.
- Learn project planning, execution, and control techniques.
- Develop skills for managing resources, risks, and timelines effectively.
- Apply project management tools and methodologies to engineering projects.

Gain knowledge of industry standards and best practices in project management.

Prerequisites

- Basic understanding of engineering processes.
- Familiarity with technical documentation and reporting.
- Basic knowledge of organizational and business principles

Program Outline

- **Chapter 1: Introduction to Project Management (3 weeks)**

1.1 Definition and importance of project management in engineering.

1.2 Project lifecycle: Initiation, Planning, Execution, Monitoring, and Closure.

1.3 Key stakeholders and roles in a project.

- **Chapter 2: Project Planning and Scheduling (3 weeks)**

2.1 Work Breakdown Structure (WBS) and task dependencies.

2.2 Scheduling techniques: Gantt charts, Critical Path Method (CPM), and PERT.

2.3 Resource allocation and budget estimation.

- **Chapter 3: Risk Management and Quality Control (3 weeks)**

3.1 Identifying and assessing risks in engineering projects.

3.2 Risk mitigation strategies.

3.3 Quality assurance and control techniques.

- **Chapter 4: Execution, Monitoring, and Control (3 weeks)**

4.1 Project tracking and performance evaluation.

4.2 Handling project changes and challenges.

4.3 Communication and documentation in project management.

- **Chapter 5: Case Studies and Best Practices (3 weeks)**

5.1 Engineering project success and failure analysis.

5.2 Lessons learned and continuous improvement.

5.3 Agile methodologies in engineering project management..

Laboratory Work

No labs.

Evaluation Method

Module average = 40% continuous assessment + 60% final exam

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- Harold Kerzner, Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Wiley, 2017.
- Erik W. Larson & Clifford F. Gray, Project Management: The Managerial Process, McGraw-Hill, 2020.
- PMI, A Guide to the Project Management Body of Knowledge (PMBOK Guide), Project Management Institute, 2021.
- F. L. Harrison & Dennis Lock, Advanced Project Management: A Structured Approach, Routledge, 2017.
- Bent Flyvbjerg, How Big Things Get Done, Currency, 2023.

e. Detailed programs for the semester 4

Teaching Unit (UE)	Subject Title	Code	Semester
UEF2.2.1	**Internet of Things (IoT)**	**IoT**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0h45	1h30	**3/3**

Objectives

The objective of this course is to introduce students to equip students with the fundamental concepts, technologies, and best practices required to design, deploy, and optimize robust IoT solutions in autonomous embedded environments. Students will learn to acquire sensor data, communicate using multiple protocols (WiFi/MQTT, HTTP, LPWAN, XBee), process and store data both locally and in the cloud, and implement energy-efficient strategies. This course lays the groundwork for advanced topics in distributed computing for embedded systems

Prerequisites

- Basic knowledge of embedded systems and microcontroller programming.
- Fundamentals of computer networking (IP addressing, routing, TCP/UDP, client-server model).
- Familiarity with Linux environment and basic shell commands.
- Introduction to Python or C programming (basic scripting is sufficient)

Program Outline

CHAPTER 1: Introduction to the IoT Ecosystem (2 Weeks)

- Definitions, concepts, and use cases
- Presentation of hardware platforms (ESP32, Raspberry Pi)
- Components of an IoT system: sensors, actuators, gateways, servers
- Main challenges in IoT: energy consumption, latency, reliability
- Applications in autonomous systems (e.g., drones, Industry 4.0)

CHAPTER 2: IoT Communication Protocols, Models, and M2M (3 Weeks)

- MQTT (Message Queuing Telemetry Transport): Publish/Subscribe model, QoS.
- CoAP (Constrained Application Protocol): Lightweight alternative to HTTP.
- HTLABS& WebSockets: Use cases in IoT.
- LPWAN protocols: LoRaWAN and NB-IoT for long-range, low-power networks..

CHAPTER3: IoT Architecture, Edge Processing, and Energy Optimization (3 Weeks)

- Role of IoT gateways and edge nodes in local data preprocessing
- Implementation of edge computing: filtering, aggregation, and data preprocessing
- Energy optimization techniques: Deep Sleep modes, wake-up on interrupt
- Interfacing ESP32/Raspberry Pi with various sensors.

CHAPTER4: IoT Deployment and Cloud Integration (3 Weeks)

- Data preprocessing and formats (filtering, aggregation, JSON, CSV)
- Local storage solutions for embedded devices (SQLite, InfluxDB)
- Introduction to Node-RED and other dashboard tools for orchestrating and visualizing data.

CHAPTER 5: IoT Data Flow and Cloud Integration (2 weeks)

- Data transfer via MQTT or HTTP to cloud platforms (e.g., AWS IoT, Azure IoT Hub)
- Cloud integration strategies and data visualization

Laboratory Work

- **Lab 1**: Setting Up IoT Platforms and Sensor Data Acquisition
- **Lab 2**: Implementing MQTT Communication for IoT Devices.
- **Lab 3**: IoT REST API and HTTP Communication.
- **Lab 4**: Bridging Edge to Cloud with an MQTT Bridge.
- **Lab 5**: Real-Time Data Visualization with Node-RED.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Amsterdam, Netherlands: Elsevier, 2014.
- R. Minerva, A. Biru, and D. Rotondi, *Towards a Definition of the Internet of Things (IoT)*. IEEE Internet Initiative, 2015.
- A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- D. Uckelmann, M. Harrison, and F. Michahelles, *Architecting the Internet of Things*. Berlin, Germany: Springer, 2011.
- O. Hersent, D. Boswarthick, and O. Elloumi, *The Internet of Things: Key Applications and Protocols*. Hoboken, NJ, USA: Wiley, 2012.

Teaching Unit (UE)	Subject Title	Code	Semester
UEF2.2.1	**Electric Machines**	**EM**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0h45	1h00	**3/3**

Objectives

Enable students to understand, model, and drive electric machines (transformers, DC motors, stepper motors, asynchronous and synchronous machines). By the end, they will be able to select and implement motors for drones, robotic platforms, and industrial drives, as well as grasp basic control strategies.

Prerequisites

- Basics of electromagnetic circuits (Hopkinson's law, flux, inductance)
- Fundamentals of circuit analysis (AC, DC)
- Introductory knowledge of power electronics (switching devices, inverters)

Program Outline

CHAPTER 1: Magnetic Circuits (2 weeks)

- Hopkinson's law, reluctance, flux
- Heterogeneous magnetic circuits, leakage flux
- Permanent magnet circuits, losses (hysteresis, eddy currents)

CHAPTER 2: Single-Phase Transformer (3 weeks)

- Transformer construction and equations
- Ideal vs. real transformer, equivalent circuit
- Voltage drop calculation, Kapp diagram, efficiency
- Special transformers (isolation, pulse, autotransformer)..

CHAPTER3: Direct Current (DC) Machines (3 weeks)

- Structure and principle (generator vs. motor)
- Electromotive force, electromagnetic torque
- Excitation modes (separate, shunt, series, compound)
- Armature reaction, starting the DC motor

CHAPTER4: Stepper Motors (2 weeks)

- Types of stepper motors (permanent magnet, variable reluctance, hybrid)
- Torque, speed, and stepping modes (full, half, microstepping)
- Basic driver circuits and control logic

- Applications in robotics, CNC, 3D printing.

CHAPTER 5: Alternating Current (AC) Machines

- Rotating magnetic field creation, electromechanical energy conversion
- Asynchronous (Induction) Machines: construction, equivalent circuit, speed control, braking
- Synchronous Machines: construction, operation as alternator or motor, phasor diagrams, stability.

CHAPTER 6: Introduction to Modeling and Control of Electric Machines (2 weeks)

- State-space modeling basics
- Control strategies: scalar control (V/f), vector control (FOC) for induction and PMSM
- Practical considerations: sensorless control, real-time implementation
- Efficiency and optimization in drives

Laboratory Work

- **Lab 1**: Magnetic Circuit and Transformer Basics
- **Lab 2**: DC Motor Operation.
- **Lab 3**: Stepper Motor Control.
- **Lab 4**: Asynchronous Motor Characterization.
- **Lab 5**: Synchronous Machine as Alternator.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- S. J. Chapman, *Electric Machinery Fundamentals*, 5th ed. New York, NY, USA: McGraw-Hill, 2011.
- P. C. Sen, *Principles of Electric Machines and Power Electronics*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
- A. E. Fitzgerald, C. Kingsley, and S. D. Umans, *Electric Machinery*, 7th ed. New York, NY, USA: McGraw-Hill, 2013.
- G. R. Slemon, *Electric Machines and Drives*. Upper Saddle River, NJ, USA: Pearson, 1992.
- B. K. Bose, *Modern Power Electronics and AC Drives*. Upper Saddle River, NJ, USA: Pearson, 2001.

Teaching Unit	Subject Title	Code	Semester
UEF2.2.1	**FPGA-and Hardware design**	**FPGA**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

Teach students the foundations of Field-Programmable Gate Array (FPGA) design, from architecture to implementation. Emphasis is on HDL-based development, IP core usage, timing analysis, and real-time applications, enabling rapid prototyping of custom hardware solutions..

Prerequisites

- Basics of digital design (logic gates, combinational and sequential circuits).

Program Outline

CHAPTER 1: Introduction to FPGAs and Applications (2 weeks)

- Overview of programmable logic devices (PLDs)
- FPGA vs. ASIC vs. Microcontrollers
- FPGA use cases: AI accelerators, high-speed signal processing, IoT

CHAPTER 2: FPGA Architecture (3 weeks)

- Configurable Logic Blocks (CLBs), Look-Up Tables (LUTs), Flip-Flops
- Routing fabric and programmable interconnects
- Memory blocks, DSP slices, and clock management.

CHAPTER 3: FPGA Development Workflow (3 weeks)

- HDL languages (VHDL, Verilog)
- Synthesis, placement, routing, and bitstream generation
- Development tools: Xilinx Vivado, Intel Quartus

CHAPTER4: Designing with IP Cores (3 weeks)

- Vendor-provided IP cores (e.g., FFT, FIR filters)
- Interfacing peripherals (UART, I²C, SPI)
- High-Level Synthesis (HLS) basics

CHAPTER5: Timing Analysis and Debugging (2 weeks)

- Static timing analysis: setup, hold times, and constraints
- Debugging with simulation tools (ModelSim, Vivado Logic Analyzer)

CHAPTER6: FPGA Applications and Trends (2 weeks)

- Case studies: AI accelerators, video processing, cryptography
- SoC FPGAs (Xilinx Zynq, Intel Stratix)
- Future of reconfigurable computing.

Laboratory Work

- **Lab 1:** Introduction to FPGA Development Tools
- **Lab 2:** Sequential Logic Design
- **Lab 3:** Using IP Cores
- **Lab 4:** Real-Time Signal Processing
- **Lab 5:** Debugging and Timing Optimization
- **Lab 6:** Mini Project

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*, 3rd ed. New York, NY, USA: McGraw-Hill, 2013.
- M. D. Ciletti, *Advanced Digital Design with the Verilog HDL*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- P. Ashenden, *The Designer's Guide to VHDL*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2008.
- C. H. Roth and L. L. Kinney, *Fundamentals of Logic Design*, 7th ed. Boston, MA, USA: Cengage Learning, 2013.
- J. O. Hamblen, T. S. Hall, and M. D. Furman, *Rapid Prototyping of Digital Systems: SOPC Edition*. New York, NY, USA: Springer, 2008.

Teaching Unit	Subject Title	Code	Semester
UEM2.2.1	**Advanced Real-Time Operating Systems (RTOS)**	**ARTOS**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

Enable students to understand real-time constraints and how specialized operating systems (like RT Linux or FreeRTOS) handle scheduling, interrupts, and synchronization. By the end, students will be able to develop deterministic embedded applications requiring guaranteed response times..

Prerequisites

- Knowledge of embedded Linux or microcontroller programming.
- Familiarity with OS concepts: scheduling, concurrency.

Program Outline

CHAPTER 1: I/O Systems for Embedded Devices (2 weeks)

- Device driver architecture
- Polling vs. interrupt-driven I/O
- DMA controller utilization

CHAPTER 2: Power Management in Embedded RTOS (1.5 weeks)

- Sleep modes and wake-up sources
- Dynamic voltage and frequency scaling
- Battery-aware scheduling.

CHAPTER 3: File Systems for Embedded Devices (1.5 weeks)

- Flash memory considerations
- Wear leveling algorithms
- Journaling vs. log-structured file systems.

CHAPTER4: Networking in Real-Time Systems (1.5 weeks)

- Real-time network protocols
- Deterministic Ethernet
- Wireless communication challenges

CHAPTER5: Fault Tolerance and Reliability (1.5 weeks)

- Watchdog mechanisms
- Error detection and recovery
- Redundancy techniques.

CHAPTER6: RTOS Performance Analysis (1.5 weeks)

- Worst-case execution time analysis
- Response time analysis
- Jitter measurement and minimization.

CHAPTER7: RTOS Porting and BSP Development (1.5 weeks)

- Hardware abstraction layers
- Board support package implementation
- Cross-compilation environments

CHAPTER8: Certification and Standards (1 weeks)

- Safety-critical development (IEC 61508)
- MISRA C guidelines
- DO-178C for aviation systems.

CHAPTER9: RTOS and Multicore Processing (1.5 weeks)

- Task affinity and migration
- Inter-core synchronization
- Cache coherency challenges.

Laboratory Work

- **Lab 1:** I/O Systems for Embedded Devices
- **Lab 2:** Power Management in Embedded RTOS
- **Lab 3:** File Systems for Embedded Devices
- **Lab 4:** Networking in Real-Time Systems
- **Lab 5:** Fault Tolerance and Reliability
- **Lab 6:** RTOS Performance Analysis
- **Lab 7:** RTOS Porting and BSP Development
- **Lab 8:** Real-Time Application

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- Q. Li and C. Yao, *Real-Time Concepts for Embedded Systems*. Boca Raton, FL, USA: CRC Press, 2003.
- J. W. S. Liu, *Real-Time Systems*. Upper Saddle River, NJ, USA: Pearson, 2000.
- D. Kalinsky, *Introduction to Real-Time Operating Systems*. Embedded Systems Conference, 2002.
- M. Barr and A. Massa, *Programming Embedded Systems: With C and GNU Development Tools*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2006.

Teaching Unit (UE)	Subject Title	Code	Semester
UEF2.2.2	**Cryptography**	**CRYP**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0 h45	1h30	**3/3**

Objectives

This module provides an in-depth understanding of cryptographic techniques used to secure embedded systems. Students will learn symmetric and asymmetric encryption, hash functions, and key management tailored for constrained environments.

Prerequisites

- Introduction to Embedded System Security
- Familiarity with programming.
- Basic understanding of mathematics.

Program Outline

CHAPTER 1: Introduction to Cryptography (3 weeks)

- Importance of cryptography in embedded systems.
- Goals of cryptography: Confidentiality, integrity, authentication, and non-repudiation.
- Overview of classical cryptography: Caesar cipher, Vigenère cipher.
- Key concepts: Symmetric vs. asymmetric cryptography, key management, and entropy.

CHAPTER 2: Symmetric Cryptography (3 weeks)

- Block ciphers: DES, AES (Advanced Encryption Standard).
- Stream ciphers: ChaCha and RC4 (focus on ChaCha for lightweight encryption).
- Lightweight cryptography for embedded systems: Introduction to AES-GCM and ChaCha20-Poly1305.

CHAPTER3: Asymmetric Cryptography (3 weeks)

- Public-key cryptography: RSA, Diffie-Hellman key exchange.
- Introduction to elliptic-curve cryptography (ECC) for embedded systems.
- Digital signatures: RSA, ECDSA, and their applications.

CHAPTER4: Cryptographic Hash Functions (3 weeks)

- Overview of hash functions: MD5, SHA family (SHA-2, SHA-3).
- Lightweight hash functions for IoT (e.g., Blake2).
- Applications: Password storage, integrity checks, and digital signatures.

CHAPTER 5: Cryptanalysis and Future Directions (2 weeks)

- Basics of cryptanalysis: Brute force, side-channel attacks.

- Quantum-safe cryptography: Post-quantum cryptographic algorithms.
- Case study.

Laboratory Work

- **Lab 1:** Implementing Classical Ciphers.
- **Lab 2:** Symmetric Encryption with AES and ChaCha20.
- **Lab 3:** Lightweight Cryptography in Embedded Systems.
- **Lab 4:** Public-Key Cryptography with RSA and ECC.
- **Lab 4:** Cryptographic Hash Functions for Data Integrity.
- **Lab 4:** Cryptanalysis and Side-Channel Attack Simulation.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Upper Saddle River, NJ, USA: Pearson, 2016.
- B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. Hoboken, NJ, USA: Wiley, 2015.
- A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.
- N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications*. Hoboken, NJ, USA: Wiley, 2010.
- J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2020.

Teaching Unit	Subject Title	Code	Semester
UEF2.2.2	**Embedded Vision and Intelligent Image Processing**	**VIP**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**2/2**

Objectives

This module aims to provide students with theoretical and practical knowledge of image processing and computer vision techniques. Students will learn the principles of digital image processing, key vision algorithms, and their applications in embedded systems, enabling them to process and analyze images in real-time for autonomous systems.

Prerequisites

- • Fourier transform and its application to signals.
- Basics of convolution and filtering techniques.
- Linear algebra (matrices, eigenvalues, eigenvectors).
- Probability and statistics (important for feature extraction and object detection).

Program Outline

CHAPTER 1: Introduction to Image Processing (2 weeks)

- Overview of image processing and computer vision.
- Basics of digital images: pixels, resolution, and color spaces (RGB, Grayscale, HSV).
- Image file formats and compression techniques.
- Applications in embedded systems: autonomous vehicles, drones, and IoT devices..

CHAPTER 2: Image Enhancement and Filtering (3 weeks)

- Image enhancement in spatial and frequency domains.
- Smoothing filters (mean, Gaussian) and sharpening filters (Laplacian).
- Histogram equalization and contrast adjustment..

CHAPTER3: Feature Extraction and Image Segmentation (4 weeks)

- Edge detection: Sobel, Prewitt, and Canny.
- Corner detection (Harris corner detector).
- Thresholding techniques (global, adaptive).
- Region-based segmentation: Watershed algorithm.

CHAPTER4: Motion Analysis and Object Tracking (3 weeks)

- Optical flow estimation (Horn-Schunck, Lucas-Kanade).
- Background subtraction and motion segmentation.
- Basics of object tracking algorithms (e.g., Kalman filter, mean-shift, correlation trackers).

CHAPTER5: Introduction to Machine Vision for Autonomous Systems (3 weeks)

- Object detection and classification using Haar cascades and HOG descriptors.
- Basics of Convolutional Neural Networks (CNNs) for vision tasks.
- Applications: Face detection, pedestrian detection, and lane detection in autonomous vehicles.

Laboratory Work

- **Lab 1:** Image Basics and Histogram Equalization
- **Lab 2:** Filtering and Image Smoothing
- **Lab 3:** Edge and Feature Detection
- **Lab 4:** Image Segmentation Techniques
- **Lab 5:** Motion Detection and Optical Flow
- **Lab 6:** Object Detection and Tracking

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing. Pearson.
- Szeliski, R. (2011). Computer Vision: Algorithms and Applications. Springer.
- Pratt, W. K. (2007). Digital Image Processing: PIKS Inside. Wiley.
- Forsyth, D. A., & Ponce, J. (2011). Computer Vision: A Modern Approach. Pearson.
- Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.

Teaching Unit	Subject Title	Code	Semester
UEF2.2.2	**Regulation and Control**	**R&C**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0h45	1h00	**3/3**

Objectives

Equip students with the knowledge to design, analyze, and implement regulation and control systems in embedded environments. Emphasis is on drones and autonomous systems, covering everything from basic PID to robust and optimal control, plus sensor fusion techniques.

Prerequisites

- Linear and dynamic systems
- Basic electronics and programming (C/C++, Python)
- Fundamentals of signal processing and control theory.

Program Outline

CHAPTER 1: Introduction to Regulation and Control (2 weeks)

- Definitions: open-loop vs. closed-loop
- Examples in drones/autonomous systems
- Mathematical models for dynamic systems

CHAPTER 2: Modeling and Analysis of Control Systems (3 weeks)

- State-space and differential equation modeling
- Stability analysis (Routh-Hurwitz, root locus, Bode/Nyquist)
- Drone and autonomous system modeling.

CHAPTER3: PID Control and Its Variants (4 weeks)

- P, PI, PD, PID design and analysis
- Discretization for digital systems
- Implementation on STM32/ESP32.

CHAPTER4: Optimal and Robust Control (3 weeks)

- Linear Quadratic Regulator (LQR)
- $H\infty$ techniques for uncertainty handling
- Applications to drone and vehicle control

CHAPTER5: Filtering and Estimation in Control Systems (3 weeks)

- Introduction to Kalman filtering for dynamic systems
- Data fusion (GPS, IMU) in control loops
- Improving sensor accuracy for stable control

Laboratory Work

- **Lab 1:** Modeling and Stability Analysis
- **Lab 2:** Implementing a PID on an Embedded Platform
- **Lab 3:** Kalman Filtering for Sensor Fusion
- **Lab 4:** LQR Controller Design
- **Lab 5:** Flight Simulator Integration

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- K. Ogata, "Modern Control Engineering", 5th ed., Pearson, 2010.
- G. F. Franklin, J. D. Powell, A. Emami-Naeini, "Feedback Control of Dynamic Systems", 8th ed., Pearson, 2019.
- R. C. Dorf, R. H. Bishop, "Modern Control Systems", 13th ed., Pearson, 2016.
- N. S. Nise, "Control Systems Engineering", 8th ed., Wiley, 2019.
- J. J. D'Azzo, C. H. Houpis, "Linear Control System Analysis and Design with MATLAB", 6th ed., CRC Press, 2017.

Teaching Unit	Subject Title	Code	Semester
UEM2.2.1	**Fundamentals of Robotics**	**ROBO**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0h45	1h00	**3/3**

Objectives

This Lecturese provides a comprehensive treatment on the fundamentals of robotic manipulators and mobile robots

By the end of this Lecturese, students will be able to:

- Analyze and understand the structure and mobility of robotic systems.
- Model robotic mechanisms using geometric, kinematic, and dynamic approaches.
- Implement trajectory generation techniques for robotic motion planning.
- Apply control techniques for motion control.
- Kinematic and dynamic modeling of mobile robots.
- Apply control techniques for motion control
- Localization and mapping
- Path planning and obstacles avoidance..

Prerequisites

- Basic knowledge of rigid body mechanics
- Kinetics and dynamics of rigid bodies
- State estimation and control theory.

Program Outline

CHAPTER 1: General definitions (2 weeks)

- - General structure, Components and classification of a robotic manipulator
- - Articulated mechanical structure
- - Robot characteristics and Classification.

CHAPTER 2: Geometric modeling of robots (3 weeks)

- Homogeneous transformations for translation and rotation
- -Orientation représentations (Euler angles, Cosine matrix and Quaternion)
- -Forward and Inverse geometric model.

CHAPTER3: Kinematics and dynamics of robot manipulators (3 weeks)

- Forward and inverse kinematic model
- -Dynamic Modeling.

CHAPTER4: Motion generation and control (3 weeks)

- Motion generation in the joint and operational spaces
- -Motion Control in the joint and operational spaces.

CHAPTER 5: Wheeled mobile robots (2 weeks)

- Drive configurations
- - Kinematics and dynamic models
- -Kinematics control.

CHAPTER 6: Motion control for wheeled mobile robots (2 weeks)

- Localization
- -Trajectory tracking
- - Path Planning and Obstacles avoidance.

Laboratory Work

- **Lab 1:** Robot Programming with ROS
- **Lab 2:** Kinematics and simple motion control
- **Lab 3:** Control of PUMA 560 Robotic arm
- **Lab 4:** Motion control of differential drive mobile robot
- **Lab 5:** State estimation of differential drive mobile robot

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2017.
- R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 2017.
- K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge, UK: Cambridge Univ. Press, 2017.
- L. Jaulin, *Mobile Robotics*. Hoboken, NJ, USA: Wiley, 2019.
- R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.

Teaching Unit	Subject Title	Code	Semester
UEM2.2.1	**Wireless Communication essentials**	**WCE**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0h45	1h00	**3/3**

Objectives

This module provides students with a comprehensive understanding of wireless communication principles and the technologies used in embedded and IoT applications. It is divided into two parts:

1. **Fundamentals of Wireless Communication**: Covers channel characteristics, modulation, coding, link adaptation, and antenna basics.
2. **Wireless Technologies for IoT and Embedded Systems**: Explores the most common wireless standards and protocols (cellular, Bluetooth, Wi-Fi, Zigbee, LoRa, LoRaWAN, etc.) and their integration in embedded systems..

Prerequisites

- Basic knowledge of signals and systems (Fourier transforms, basic filtering).
- Fundamentals of digital communication (bit rates, bandwidth…).
- Familiarity with embedded systems (microcontrollers, basic programming).

Program Outline

PART I: Fundamentals of Wireless Communication

CHAPTER 1: Wireless Channel & Propagation (3 weeks)

- Propagation Mechanisms: Reflection, diffraction, scattering.
- Channel Characteristics: Path loss, fading, multipath effects.
- Link Budget: Power calculations, SNR, link margin.

CHAPTER 2: Modulation, Coding, and Link Adaptation (3 weeks)

- Analog Modulation Schemes: AM, PM, FM.
- Digital Modulation Schemes: ASK, PSK, FSK,QAM.
- Coding Techniques: Error-correcting codes.
- Adaptive Modulation & Coding: Balancing throughput and reliability.

CHAPTER3: Antennas and RF Basics (3 weeks)

- Antenna Parameters: Gain, directivity, radiation patterns.
- Antenna Types: Dipole, patch, Yagi, and embedded antennas.
- RF Front-End Considerations: Filters, amplifiers, matching networks.

PART II: Wireless Technologies for IoT and Embedded Systems

CHAPTER4: Cellular & Short-Range Wireless (6 weeks)

- Cellular (2G, 3G, 4G, 5G, NB-IoT): Architecture, coverage, data rates.
- Bluetooth & BLE: Topology, power consumption, pairing & security.
- Wi-Fi (802.11 variants): Channel access, security (WPA2/3), embedded use cases.
- Zigbee: Network topology, cluster library, typical IoT applications.
- HF Radio Communication: ISM bands, typical HF modules for embedded.
- Protocol Stacks: Interfacing with MCUs, choosing the right protocol for a use case.

CHAPTER 6: LoRa & LoRaWAN (2 weeks)

- LoRa Modulation: Chirp Spread Spectrum, link budget considerations.
- LoRaWAN Architecture: End-devices, gateways, network server.
- Use Cases: Long-range IoT applications, sensor networks, coverage vs. data rate trade-offs.

Laboratory Work

- **Lab 1:** Antenna measurements
- **Lab 2:** BLE/Wi-Fi setup on an embedded device
- **Lab 3:** LoRaWAN deployment and range testing

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2001.
- A. Goldsmith, *Wireless Communications*. Cambridge, UK: Cambridge Univ. Press, 2005.
- J. D. Kraus and R. J. Marhefka, *Antennas: For All Applications*, 3rd ed. New York, NY, USA: McGraw-Hill, 2002.
- D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, UK: Cambridge Univ. Press, 2005.
- W. Stallings, *Wireless Communications and Networks*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004.

Teaching Unit	Subject Title	Code	Semester
UEM2.2.1	**Embedded IA**	**EAI**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	1h30	**2/2**

Objectif

This module provides hands-on experience in applying AI techniques to real-world embedded systems problems. Students will work on practical projects, from data collection to model deployment, using industry-standard tools and platforms. The module focuses on optimizing AI models for resource-constrained devices and preparing students for industry challenges.

Pré-requis

- Completion of Machine Learning and Deep Learning modules.
- Programming experience in Python.
- Familiarity with embedded systems platforms

Contenu

LAB 1: Real-Time Object Detection on Embedded Hardware

LAB 2: Predictive Maintenance for IoT Devices

LAB 3: Speech Recognition for Embedded Systems

LAB 4: Energy Optimization in Autonomous Systems

LAB 5: Anomaly Detection in Sensor Data

LAB 6: Gesture Recognition for Human-Machine Interaction

LAB 7: End-to-End Project: Autonomous Navigation

Travaux pratiques

- **LAB 1:** Real-Time Object Detection on Embedded Hardware.
- **LAB 2:** Predictive Maintenance for IoT Devices.
- **LAB 3:** Speech Recognition for Embedded Systems
- **LAB:** Energy Optimization in Autonomous Systems
- **LAB 5:** Anomaly Detection in Sensor Data.
- **LAB 6:** Gesture Recognition for Human-Machine Interaction
- **LAB 7:** End-to-End Project: Autonomous Navigation

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Références

- Aurelien Geron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 2nd Edition.
- Christopher M. Bishop, "Pattern Recognition and Machine Learning", 2006.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", 2016.
- Andrew R. Webb, Keith D. Copsey, Statistical Pattern Recognition, 3rd Edition, 2011
- Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2000.
- C. Bishop : Pattern Recognition and Machine Learning. Springer: New York,2006.
- A. Cornuéjols and L. Miclet : Apprentissage artificiel – Concepts et algorithmes. Eyrolles, 2010.
- T. Hastie, R. Tibshirani and Friedman, The Elements of Statistical Learning, Springer. [ebook]
- G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning , Springer, 2013
- K. P. Murphy : Machine Learning: a Probabilistic Perspective. MIT Press, 2012.
- Scikit-learn Documentation: https://scikit-learn.org TensorFlow Lite Documentation: https://www.tensorflow.org/lite.

Teaching Unit (UE)	Subject Title	Code	Semester
UET1.2.1	Capstone Project II	**CP2**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	1h00	**2/2**

Objectives

- Form teams and identify project ideas.
- Conduct feasibility analysis and initial design documentation.
- Establish milestones and present progress to faculty advisors.

- Teamwork and collaboration.

- Problem identification and solution structuring.
- Technical writing and presentation skills.
- Project planning and feasibility assessment.

Prerequisites

No prerequisites

Program Outline

1. **Team Formation and Brainstorming (6 hours)**
   - Introduction to project selection.
   - Team roles and responsibilities.
   - Idea generation techniques.
2. **Feasibility Analysis and Initial Design Documentation (7.5 hours)**
   - Market and technical feasibility.
   - Risk assessment.
   - Drafting project proposals and documentation.
3. **Regular Milestones and Faculty Presentations (9 hours)**
   - Setting up project goals and deadlines.
   - First milestone presentation.
   - Refining project scope based on feedback.

Evaluation Method

Module average = 100% continuous assessment.

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject. The maximum value of the level threshold is set at 10/20.

Teaching Unit (UE)	Subject Title	Code	Semester
UED1.2.1	Training Internship II	**TI2**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	0	**1/1**

Objectives

- Introduce students to professional work environments.
- Develop basic technical and soft skills in an industry setting.
- Understand company structure, workflow, and project dynamics.

Prerequisites

No prerequisites

Program Outline

1. **Introduction to the Company**
   - Company overview and mission.
   - Workplace safety and regulations.
2. **Observation and Shadowing**
   - Understanding team roles and responsibilities.
   - Exposure to ongoing projects.
3. **Hands-on Experience**
   - Assisting in minor tasks and operations.
   - Learning basic tools and methodologies.
4. **Final Review and Feedback**
   - Supervisor feedback session.
   - Internship report preparation.

Laboratory Work

No laboratory works

Evaluation Method

Module average = 100% continuous assessment.
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject. The maximum value of the level threshold is set at 10/20.

f. Detailed programs for the semester 5

Teaching Unit	Subject Title	Code	Semester
UEF3.1.1	**Program Optimization and System Performance**	**POSP**	**S5**

	Lectures	TD	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

This module focuses on foundational and practical techniques for optimizing software performance on embedded systems. Students will learn how to identify bottlenecks, manage resources (memory, power), and apply both low-level and system-level optimizations to ensure reliability and efficiency under real-time constraints.

Prerequisites

- Embedded Systems Fundamentals.
- Proficiency in C/C++ for low-level programming
- Understanding of CPU pipelines, caches, and memory hierarchies.

Program Outline

CHAPTER 1: Foundations of Program Optimization (3.5 weeks)

- Resource management in embedded systems (memory, power, performance).
- Compiler optimizations: loop unrolling, function inlining, instruction scheduling.
- Profiling and benchmarking techniques (time-based, event-based).

CHAPTER 2: System Performance Tuning (4 weeks)

- Performance metrics and bottleneck identification.
- Debugging performance issues with tools (gprof, Valgrind).
- Multithreading and concurrency considerations in embedded systems.
- Cache optimization and memory alignment.

CHAPTER3: Energy Efficiency (3.5 weeks)

- Power-aware programming: dynamic voltage and frequency scaling (DVFS).
- Sleep modes and scheduling for low-power applications.
- Balancing performance and energy constraints in real-time systems..

CHAPTER4: Case Studies and Advanced Topics (3 weeks)

- Real-time OS optimizations (FreeRTOS, RT-Linux).
- Hardware acceleration (GPU, DSP) for performance-critical tasks.
- Practical examples in automotive, robotics, and IoT domains.

Laboratory Work

- **Lab 1:** Profiling and Optimization
- **Lab 2:** Cache and Memory Optimization
- **Lab 3:** Power Management and DVFS
- **Lab 4:** Advanced System Tuning

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- M. Kerrisk, *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*. San Francisco, CA, USA: No Starch Press, 2010.
- R. Bryant and D. O'Hallaron, *Computer Systems: A Programmer's Perspective*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2015.
- G. Shaw, *Software Optimization for Embedded Systems*. Boca Raton, FL, USA: CRC Press, 2015.
- J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2017.
- S. S. Muchnick, *Advanced Compiler Design and Implementation*. San Francisco, CA, USA: Morgan Kaufmann, 1997.

Teaching Unit	Subject Title	Code	Semester
UEF3.1.1	**Power Supply and Auxiliary Peripherals**	**PSAP**	**S5**

	Lectures	TD	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**3/3**

Objectives

Offer students a complete perspective on designing and implementing power solutions for embedded systems, from linear/switching supplies to UPS, battery management, auxiliary peripherals, and now solar energy integration. By the end of this module, students can select, design, prototype, and maintain power circuits in drones, robotics, IoT, and more.

Prerequisites

- Basic electronics (diodes, transistors, operational amplifiers)
- Fundamentals of electrical circuits (AC, DC, passive components)
- Introductory knowledge of power electronics (switching devices, rectifiers).

Program Outline

CHAPTER 1: Introduction to Power Supplies (2 weeks)

- Basic Concepts: Roles of power supplies, linear vs. switching, efficiency, voltage/current ratings
- Key Components: Transformers, diodes, capacitors, inductors, regulators
- Power Semiconductors: MOSFET, IGBT, thyristors (brief review)

CHAPTER 2: Linear Power Supplies (3 weeks)

- Operating Principle: Energy dissipation for regulation, advantages and limitations
- Design of Linear Supplies:
  - Transformer-rectifier-filter approach
  - Fixed voltage regulators (7805, 7812, LM317)
  - Ballast transistor for high-power linear supplies
- Applications and Heat Management:
  - Analog circuits, audio equipment
  - Heat dissipation solutions (heatsinks, cooling fans)

CHAPTER3: Switching Power Supplies (3 weeks)

- SMPS Principles: Differences with linear supplies, PWM-based regulation
- Topologies: Buck, Boost, Buck-Boost, SEPIC, Flyback, Forward, Half-Bridge, Full-Bridge
- Design Considerations:

- Component selection (Schottky diodes, MOSFETs, inductors)
- Magnetic design (high-frequency transformers, cores)
- Protection (overvoltage, overcurrent), EMI filtering

CHAPTER4: Introduction to UPS (Uninterruptible Power Supply) (2 weeks)

- UPS Architectures: Standby, line-interactive, online
- Battery Technologies: Lead-acid, NiCd, Li-ion basics
- Battery Chargers: Automated battery monitoring, efficiency considerations
- Applications: Critical environments (servers, medical, industrial).

CHAPTER5: Auxiliary Peripherals and Battery Management (2.5 weeks)

- Auxiliary Peripherals: Safety relays, e-fuses, fan controllers, sensor power rails
- Battery Management Systems (BMS): Balancing, protection, state-of-charge (SoC) estimation
- Implementation in Embedded Systems: Microcontroller-based monitoring, data logging, fault handling
- Practical Integration: Combining BMS with UPS or SMPS in drones, robotics

CHAPTER6: Solar Energy Systems (2.5 weeks)

- Solar Energy Basics: PV cell operation, IV curves, efficiency
- Small-Scale PV Systems: off-grid vs. grid-tied, charge controllers, MPPT
- Embedded Solar Integration: powering drones, IoT nodes with solar panels
- Design Considerations: panel sizing, battery storage, environmental factors
- Case Studies: solar-powered embedded systems, stand-alone UAV charging stations

Laboratory Work

- **Lab 1:** Linear Power Supply Design
- **Lab 2:** Basic Switching Converter / UPS Demonstration
- **Lab 3:** Battery Management and Auxiliary Circuits
- **Lab 4:** Solar Mini-Project

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- M. H. Rashid, *Power Electronics: Circuits, Devices, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2013.
- N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd ed. Hoboken, NJ, USA: Wiley, 2002.
- D. W. Hart, *Power Electronics*, 1st ed. New York, NY, USA: McGraw-Hill, 2010.
- B. K. Bose, *Modern Power Electronics and AC Drives*. Upper Saddle River, NJ, USA: Pearson, 2001.

Teaching Unit	Subject Title	Code	Semester
UEF3.1.1	**Parallel Computing on GPUs**	**GPU**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectives

Familiarize students with GPU architecture and parallel programming models (CUDA, OpenCL) to harness massive parallelism. Students will learn to optimize algorithms for high-throughput computing in areas like image processing, AI, and scientific simulations.

Prerequisites

- Basics of C/C++ programming
- Introduction to computer architecture

Program Outline

CHAPTER 1: Introduction to Parallel Computing (2 weeks)

- Importance of parallelism in modern computing
- Types of parallelism: task, data, pipeline
- GPU vs. CPU parallelism

CHAPTER 2: GPU Architecture (3 weeks)

- CUDA cores, streaming multiprocessors
- Warp execution, memory hierarchy (global, shared, local)
- Differences between CPU and GPU concurrency.

CHAPTER 3: Programming GPUs with CUDA (3 weeks)

- CUDA programming model: kernels, threads, blocks
- Memory management (shared, global, constant)
- Synchronization and thread safety

CHAPTER4: Parallel Algorithms on GPUs (3 weeks)

- Matrix multiplication and reduction algorithms
- Image processing basics (convolution, filtering)
- Performance optimization

CHAPTER5: Applications of GPU Computing (2 weeks)

- AI acceleration: training and inference on GPUs
- Scientific computing and simulations
- Real-time graphics and rendering

CHAPTER6: Trends and Case Studies in GPUs (2 weeks)

- Multi-GPU systems and frameworks (OpenCL, Vulkan)
- Case studies: self-driving cars, weather simulations, cryptography
- Future directions in GPU computing.

Laboratory Work

- **Lab 1:** Introduction to GPU Programming
- **Lab 2:** Memory Management
- **Lab 3:** Basic Parallel Algorithms
- **Lab 4:** Image Processing on GPUs
- **Lab 5:** Optimization Techniques
- **Lab 6:** Mini Project

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- D. B. Kirk and W. W. Hwu, *Programming Massively Parallel Processors: A Hands-On Approach*, 3rd ed. Cambridge, MA, USA: Morgan Kaufmann, 2016.
- J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Upper Saddle River, NJ, USA: Pearson, 2010.
- N. Wilt, *The CUDA Handbook: A Comprehensive Guide to GPU Programming*. Upper Saddle River, NJ, USA: Pearson, 2013.
- M. Garland and D. B. Kirk, *Parallel Computing for Data Science: With Examples in R, C++, and CUDA*. Boca Raton, FL, USA: CRC Press, 2015.
- J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable Parallel Programming with CUDA," *ACM Queue*, vol. 6, no. 2, pp. 40–53, 2008.

Teaching Unit	Subject Title	Code	Semester
UEF3.1.1	**Embedded Operating Systems**	**EOS**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**3/3**

Objectif

Equip students with the skills to build, customize, and optimize Linux for embedded systems. Emphasis is placed on kernel configuration, building minimal distributions, cross-compilation, and optimizing Linux for specific hardware constraints.

Prerequisites

- Familiarity with Linux system administration and OS internals (from previous modules).
- Experience with C/C++ cross-compilation.

Program Outline

CHAPTER 1: Embedded Linux Fundamentals (2 weeks)

- Characteristics of embedded Linux vs. desktop/server Linux.
- Typical embedded boards (Raspberry Pi, BeagleBone, STM32MP1).
- Toolchains and cross-compilation basics.

CHAPTER 2: Kernel Configuration and Compilation (3 weeks)

- Kernel menuconfig, defconfig, patching.
- Selecting drivers and kernel features for minimal footprint.
- Building a custom kernel for an ARM board.

CHAPTER 3: Root Filesystem and Boot Process (3 weeks)

- Bootloaders (U-Boot, Barebox) and initialization steps.
- Creating a minimal root filesystem (BusyBox, systemd).
- Managing device trees (.dts) for hardware configuration.

CHAPTER4: Device Drivers in Embedded Linux (3 weeks)

- Writing and integrating custom drivers.
- Interfacing with GPIO, I2C, SPI in the kernel.
- Kernel modules vs. built-in drivers.

CHAPTER5: Optimization and Distribution Building (2 weeks)

- Build systems: Yocto Project, Buildroot.
- Reducing footprint: removing unused packages and features.
- Performance tuning: CPU frequency scaling, power management.

CHAPTER6: Case Studies and Advanced Topics (2 weeks)

- Real-world embedded Linux examples (routers, automotive, consumer electronics).
- Security considerations (SELinux, AppArmor).
- Containerization in embedded environments (Docker on ARM).

Laboratory Work

- **Lab 1:** Cross-Compilation Basics
- **Lab 2:** Custom Kernel Build
- **Lab 3:** Minimal Root Filesystem
- **Lab 4:** Device Tree and Driver Configuration
- **Lab 5:** Build Systems
- **Lab 6:** Embedded Linux Deployment

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- C. Hallinan, *Embedded Linux Primer: A Practical Real-World Approach*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
- K. Yaghmour, *Embedded Android: Porting, Extending, and Customizing*. Sebastopol, CA, USA: O'Reilly Media, 2013.
- J. Masters, *Mastering Embedded Linux Programming*, 2nd ed. Birmingham, UK: Packt Publishing, 2017.
- D. Abbott, *Linux for Embedded and Real-Time Applications*, 4th ed. Burlington, MA, USA: Newnes, 2016.
- M. Tim Jones, *Linux Application Development*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004.

Teaching Unit (UE)	Subject Title	Code	Semester
UEF3.1.2	**System security**	**SySe**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**3/3**

Objectives

This module equips students with skills to secure embedded operating systems, firmware, and hardware interfaces. Students will learn to implement techniques to protect against software and hardware-based attacks.

Prerequisites

- Introduction to Embedded System Security
- Cryptography.
- Familiarity with embedded operating systems and firmware architecture.

Program Outline

CHAPTER 1: Threats and Security Challenges in Autonomous Systems (02 weeks)

- Security challenges in embedded operating systems (e.g., FreeRTOS, Linux).
- Case studies of cyberattacks on drones, robots, and IoT botnets..

CHAPTER 2: Microcontroller and Firmware Security (04 weeks)

- Physical security threats: firmware extraction, side-channel attacks.
- Implementing secure boot and protected OTA (Over-The-Air) updates.
- Secure key storage and embedded data protection techniques.

CHAPTER3: Security of Embedded Operating Systems (04 weeks)

- Hardening embedded OS (Yocto, Linux-based embedded OS).
- Securing processes and application isolation techniques.
- Memory protection mechanisms against buffer overflow and ROP (Return-Oriented Programming) attacks.

CHAPTER4: Advanced Topics in System Security (4 weeks)

- Intrusion detection and prevention in embedded systems.
- Case studies of system security breaches..

Laboratory Work

- **Lab 1:** Firmware attack and defense on an embedded system
- **Lab 2:** Implementing and verifying a Secure Boot process.
- **Lab 3:** Securing an embedded Linux system: File permissions and user management.
- **Lab 4:** Threat detection and prevention on a robot/autonomous deployment

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- W. Stallings, *Computer Security: Principles and Practice*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2018.
- C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2015.
- M. Bishop, *Computer Security: Art and Science*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2018.
- R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Hoboken, NJ, USA: Wiley, 2020.
- B. Schneier, *Secrets and Lies: Digital Security in a Networked World*. Hoboken, NJ, USA: Wiley, 2004.

Teaching Unit (UE)	Subject Title	Code	Semester
UEF3.1.2	**Network security**	**NCS**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**3/3**

Objectives

This module introduces students to securing communication in embedded systems and IoT networks. It covers fundamental network security threats, secure communication protocols, VPNs, network segmentation, and anomaly detection techniques. Through hands-on labs, students will gain practical experience in protecting communication channels, securing IoT protocols, and mitigating cyberattacks.

Prerequisites

- Introduction to Embedded System Security.
- Cryptography.
- Networking basics.

Program Outline

CHAPTER 1: Fundamentals of Embedded Communication Security (01 weeks)

- Overview of common network threats: MITM, sniffing, injection, replay attacks.
- Specific challenges in IoT and Edge Computing communications.
- Key concepts: confidentiality, integrity, and availability in embedded systems...

CHAPTER 2: Secure Communication Protocols (03 weeks)

- Overview of secure protocols: TLS, DTLS, and their applications in IoT.
- Securing IoT-specific protocols: MQTT with certificates, LoRaWAN, and Zigbee.
- Authentication and access control for embedded communication systems.

CHAPTER3: VPNs and Network Segmentation for Embedded Systems (03 weeks)

- Lightweight VPN solutions for IoT: WireGuard and OpenVPN.
- Network segmentation strategies: VLANs and Zero Trust Network Architecture (ZTNA).
- Protecting data exchanges between Edge devices and Cloud platforms.

CHAPTER4: Anomaly Detection and Cyberattack Prevention (3 weeks)

- Techniques for detecting anomalies in IoT and Edge networks.
- Preventing command injection, replay, and remote access attacks.
- Simulation and mitigation of IoT protocol vulnerabilities.

CHAPTER5: Anomaly Detection and Cyberattack Prevention (3 weeks)

- Optimizing Edge-Cloud communication for secure data transfer.

- Performance trade-offs: Real-time processing on Edge vs. secure storage and analysis on Cloud.
- Techniques for data aggregation, filtering, and encryption for hybrid architectures

Laboratory Work

- **Lab 1:** Capturing and Analyzing Network Attacks with Wireshark.
- **Lab 2:** Implementing TLS-secured MQTT communication.
- **Lab 3:** Setting Up a Lightweight VPN to Secure an Edge Gateway.
- **Lab 4:** Detection and Prevention of Attacks on an IoT Network (Replay, Injection).

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Upper Saddle River, NJ, USA: Pearson, 2016.
- C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2002.
- B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. Hoboken, NJ, USA: Wiley, 2015.
- W. Mao, *Modern Cryptography: Theory and Practice*. Upper Saddle River, NJ, USA: Pearson, 2003.
- J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2020.

Teaching Unit	Subject Title	Code	Semester
UEF3.1.2	**Reliability and Safety of Embedded Systems**	**NSES**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**2/2**

Objectives

This module aims to equip students with the foundational concepts and methodologies required to design and evaluate reliable and safe embedded systems. Students will learn key techniques to identify, assess, and mitigate risks and faults in critical embedded applications, ensuring system functionality and safety in real-world environments.

Prerequisites

- Embedded Systems Fundamentals.
- Digital Electronics and Circuits
- Probability and Statistics.

Program Outline

CHAPTER 1: Introduction to Reliability and Safety (2.5 weeks)

- Definition of reliability and safety in embedded systems.
- Importance of reliability and safety in critical systems (e.g., automotive, medical, aerospace).
- Key concepts: Faults, errors, failures, and hazards.
- Standards and regulations: IEC 61508, ISO 26262, and DO-178C.
- Reliability and safety metrics: MTBF, MTTF, and MTTR..

CHAPTER 2: Fault Tolerance in Embedded Systems (4 weeks)

- Types of faults: transient, permanent, and intermittent.
- Fault detection techniques: error-correcting codes, parity checks, watchdog timers.
- Fault recovery mechanisms: redundancy (hardware, software, time).
- Examples of fault-tolerant designs in embedded systems.
- Evaluating fault tolerance: fault injection testing.

CHAPTER3: Safety Analysis Techniques (3.5 weeks)

- Risk assessment methodologies: FMEA (Failure Modes and Effects Analysis) and FTA (Fault Tree Analysis).
- Hazard and Operability Analysis (HAZOP).
- Safety Integrity Levels (SILs) and their application in system design.
- Design principles for safety-critical systems: fail-safe, fail-operational, and safe-state designs.
- Case study: Safety analysis of an automotive braking system.

CHAPTER4: System Verification, Validation, and Certification (4 weeks)

- Verification vs. validation in safety-critical systems.
- Testing strategies: unit testing, integration testing, and system testing.
- Certification process for embedded systems: ISO 26262 (automotive), DO-178C (aerospace).
- Formal methods for safety-critical software development.
- Tool support for reliability and safety verification

Laboratory Work

- **Lab 1:** Fault Detection and Recovery Techniques
- **Lab 2:** Safety Analysis of an Embedded System
- **Lab 3:** Fault Injection Testing and Evaluation
- **Lab 4:** Verification and Validation of Safety-Critical Systems

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- N. Storey, *Safety-Critical Computer Systems*. Upper Saddle River, NJ, USA: Pearson, 1996.
- J. B. Bowles, *Software Engineering for Safety-Critical Systems*. Boca Raton, FL, USA: CRC Press, 2017.
- L. L. Pullum, *Software Fault Tolerance Techniques and Implementation*. Norwood, MA, USA: Artech House, 2001.
- N. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA, USA: MIT Press, 2011.
- J. McDermid, *Software Engineer's Reference Book*. Boca Raton, FL, USA: CRC Press, 1993.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM3.1.1	**Distributed Computing for Embedded Systems**	**DCES**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h30	**2/2**

Objectives

This course introduces students to the principles of distributed computing, focusing on how edge and cloud resources can be leveraged to optimize autonomous embedded systems and industrial IoT solutions. The emphasis is placed on designing lightweight, distributed architectures capable of handling real-time data processing, device coordination, and scalable remote management. Through practical applications, students will learn to deploy distributed services and pipelines to enhance embedded system performance and resilience..

Prerequisites

- IoT for Embedded Systems course
- Advanced Networking course
- Basic programming knowledge (Python, Bash or C)
- Familiarity with Linux-based embedded platforms (e.g., Raspberry Pi)

Program Outline

CHAPTER 1: Introduction to Distributed Computing for Embedded Systems (3 weeks)

- What is Distributed Computing?
- Differences between Cloud Computing and Edge Computing.
- Benefits and constraints in embedded/IoT contexts.
- Industrial applications: Smart Factory, Predictive Maintenance, Autonomous Systems

CHAPTER 2: Edge Computing in Autonomous Systems (3 weeks)

- Edge node roles: local data preprocessing, filtering, real-time decision-making.
- Edge architectures: Edge Gateway, Edge Node, Fog Computing.
- Edge Computing platforms: Raspberry Pi, Jetson Nano, etc

CHAPTER3: Cloud Computing for IoT & Embedded Systems (3 weeks)

- Cloud fundamentals: compute, storage, and basic services.
- IoT Platform-as-a-Service (PaaS): AWS IoT Core, Azure IoT Hub (conceptual overview).
- Integrating cloud services for device management and data storage.

CHAPTER4: Distributed Data Processing Pipeline (3 weeks)

- Data ingestion, processing, and visualization (Edge-to-Cloud workflow).
- Lightweight messaging systems: MQTT bridging, HTTP REST APIs.
- Event-driven processing and simple analytics at the Edge.

CHAPTER 5: System Design for Resilience and Scalability (2 weeks)

- Load balancing between Edge and Cloud.
- Distributed fault tolerance concepts.
- Latency optimization and offline-first approaches in embedded systems.

Laboratory Work

- **Lab 1:** Deploying an Edge Node with MQTT and Local Data Filtering (Raspberry Pi).
- **Lab 2:** Bridging Edge to Cloud: MQTT Broker on Pi + Cloud API Integration.
- **Lab 3:** Real-Time Data Dashboard: Collect and visualize sensor data (Edge + Cloud).
- **Lab 4:** Designing a Resilient Edge Computing System (handling disconnections).
- **Lab 5:** Lightweight Distributed Processing: Edge pre-processing + cloud post-processing.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems: Concepts and Design*, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2011.
- A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2017.
- N. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann, 1996.
- M. Raynal, *Distributed Algorithms for Message-Passing Systems*. New York, NY, USA: Springer, 2013.
- P. Jain and M. K. Singh, *Distributed Computing: Principles, Algorithms, and Systems*. Cambridge, UK: Cambridge Univ. Press, 2008.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM3.1.1	**Fundamentals of Autonomous Navigation**	**Nav**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**2/2**

Objectives

This module teaches the fundamental and advanced principles of autonomous system navigation, focusing on sensor fusion and navigation algorithms for drones. By the end, students will be able to:

- Understand different navigation techniques (visual, inertial, satellite-based, odometry).
- Implement sensor fusion algorithms (Kalman filters, complementary filters).
- Test and validate navigation algorithms on a flight simulator (MSFS) and embedded microcontrollers (STM32/ESP32)..

Prerequisites

- Familiarity with Laplace/Fourier transforms for system/signal analysis.
- Basic knowledge of Kalman filters and sensor fusion concepts.
- Embedded Systems and Microcontrollers

Program Outline

CHAPTER 1: Introduction to Navigation (2 weeks)

- Types of navigation: visual, inertial, satellite, odometry
- Basic vehicle modeling and sensor overview
- Data fusion overview for improved accuracy.

CHAPTER 2: Errors and Correction (3 weeks)

- Sensor noise, calibration, and drift
- Kalman filter basics and complementary filter
- Real-world constraints (wind, vibration, environmental factors).

CHAPTER3: Inertial Navigation and Sensor Fusion (3 weeks)

- IMU principles (accelerometer, gyroscope, magnetometer)
- Extended Kalman Filter (EKF) for GPS/IMU
- Implementing INS for drone pose/orientation

CHAPTER4: Satellite Navigation (GNSS) and RTK (3 weeks)

- GNSS fundamentals (GPS, Galileo, Glonass, BeiDou)
- Integration with INS (standard vs. RTK-GPS)
- Improving accuracy via advanced filtering.

CHAPTER 5: Vision-Based Navigation and SLAM (2 weeks)

- Vision-based techniques (feature extraction, obstacle detection)
- SLAM fundamentals (mapping, localization)
- Drone-specific challenges (indoor/outdoor, lighting changes).

CHAPTER 6: Practical Experimentation and Embedded Deployment (2 weeks)

- Flight Simulator (MSFS) integration for algorithm testing
- Embedded coding on STM32/ESP32 with sensor data (SPI/I2C)
- Validation and case studies (logging flight data, analyzing results)

Laboratory Work

- **Lab 1:** Sensor Data and Noise Simulation.
- **Lab 2:** Complementary Filter Implementation.
- **Lab 3:** Basic Kalman Filter.
- **Lab 4:** EKF for Drone Trajectory.
- **Lab 5:** FS Integration
- **Lab 6:** Embedded Navigation Project.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.
- S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
- H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed. New York, NY, USA: Springer, 2017.
- J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Boston, MA, USA: Kluwer Academic Publishers, 1992.

Teaching Unit (UE)	Subject Title	Code	Semester
UEM3.1.1	Introduction to Quantum Computing	**QC**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	0	1h00	**2/2**

Objectives

Provide a basic introduction to quantum computing for students with a simple math/embedded background. By the end, they will:

- Understand qubits, superposition, and entanglement.
- Implement small quantum circuits (Deutsch–Jozsa, Grover's) on simulators.
- Gain awareness of quantum error correction and quantum hardware constraints.
- Explore potential embedded and future applications (quantum cryptography, optimization)..

Prerequisites

- Basic Linear Algebra: Vectors, matrices, complex numbers.
- Introductory Programming: Ability to code in Python/C++ for lab simulators.
- Elementary Probability: Basic probability distributions and expectation..

Program Outline

CHAPTER 1: Classical vs. Quantum Computation (02 weeks)

- Classical circuit model, reversible logic basics
- Qubits vs. classical bits: superposition, measurement postulate
- Quantum circuit model, universal gate sets
- Simple example: Deutsch's problem

CHAPTER 2: Core Quantum Concepts (03 weeks)

- Single-qubit gates (Pauli, Hadamard, phase gates)
- Multi-qubit gates (CNOT, controlled-phase)
- Bloch sphere representation
- Small quantum circuits (2–3 qubits).

CHAPTER3: Basic Quantum Algorithms (03 weeks)

- Deutsch–Jozsa algorithm: constant vs. balanced functions
- Simon's algorithm (hidden subgroup problem) – high-level overview
- Mathematical parenthesis: factoring integers, period finding (brief mention)
- Continued fraction expansions (only conceptual).

CHAPTER4: Grover's Search and Intro to Error Correction (3 weeks)

- Grover's algorithm: concept, speedup, small example
- Noise models (decoherence, amplitude/phase damping)
- Simple repetition code concept

- High-level mention of Shor or Steane codes.

CHAPTER5: Quantum Hardware and Tools (3 weeks)

- Current quantum hardware: superconducting qubits, trapped ions
- Noise levels, gate fidelity, qubit counts
- Simulators and cloud platforms (IBM Quantum, Qiskit, Cirq)
- Limitations of near-term quantum devices (NISQ era)

CHAPTER5: Quantum Hardware and Tools (3 weeks)

- Potential use cases: quantum cryptography (BB84), quantum optimization, machine learning
- Shor's algorithm (factorization) – conceptual
- Embedding quantum solutions in autonomous/embedded contexts (high-level)
- Research challenges, future outlook (fault-tolerant QC)

Laboratory Work

- **Lab 1:** Quantum Circuit Basics.
- **Lab 2:** Multi-Qubit Operations.
- **Lab 3:** Deutsch–Jozsa Algorithm.
- **Lab 4:** Grover's Search.
- **Lab 5:** Noise and Simple Error Correction.
- **Lab 4:** Advanced Topics and Mini-Project.

Evaluation Method

Module average = 20% continuous assessment + 20% practical work + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed. Cambridge, UK: Cambridge Univ. Press, 2010.
- N. D. Mermin, *Quantum Computer Science: An Introduction*. Cambridge, UK: Cambridge Univ. Press, 2007.
- E. Rieffel and W. Polak, *Quantum Computing: A Gentle Introduction*. Cambridge, MA, USA: MIT Press, 2011.
- C. Bernhardt, *Quantum Computing for Everyone*. Cambridge, MA, USA: MIT Press, 2019.
- J. Preskill, "Quantum Computing in the NISQ Era and Beyond," *Quantum*, vol. 2, p. 79, 2018.
- Bera, Rajendra K. *The Amazing World of Quantum Computing*. Springer, 2020.
- Hassi Norlén. *Quantum Computing in Practice with Qiskit® and IBM Quantum Experience®: Practical recipes for quantum computer coding at the gate and algorithm level with Python*, Packt, November 23, 2020

Teaching Unit (UE)	Subject Title	Code	Semester
UED3.1.1	Entrepreneurship and Startup Development	**ESD**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h00	0	0	**1/1**

Objectives

This module aims to equip students with the foundational knowledge and skills required to navigate the entrepreneurial journey, from ideation to startup creation. Through theoretical and practical insights, students will:

- Understand the fundamentals of entrepreneurship and the startup ecosystem.
- Learn how to identify opportunities, develop business ideas, and validate them in the market.
- Acquire knowledge in business modeling, funding strategies, and essential skills for managing startups.
- Develop critical thinking and problem-solving skills to address real-world business challenges.

Prerequisites

- Basic knowledge of business concepts and management.
- Familiarity with the fundamentals of project planning and teamwork..

Program Outline

CHAPTER 1: Introduction to Entrepreneurship (2 weeks)

- Definition of entrepreneurship.
- Traits of successful entrepreneurs.
- The role of startups in economic development.

CHAPTER 2: Idea Generation and Opportunity Identification (3 weeks)

- Methods for generating business ideas.
- Identifying and analyzing market opportunities.
- Techniques for validating startup ideas (Lean Startup methodology, customer discovery).

CHAPTER 3: Business Model Development (4 weeks)

- Introduction to business models.
- Understanding the Business Model Canvas.
- Case studies of successful business models.

CHAPTER 4: Funding and Financial Planning (2 weeks)

- Types of funding: bootstrapping, angel investors, venture capital, crowdfunding.

- Developing a financial plan for startups.
- Managing cash flow and budgeting for early-stage companies.

SEMINAR 5: Startup Management and Growth Strategies (4 weeks)

- Building a team: roles and responsibilities in a startup.
- Marketing and customer acquisition strategies.
- Scaling up: growth hacking and expansion techniques..

Laboratory Work

- **No Lab**

Evaluation Method

Module average = 40% continuous assessment + 60% final exam
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject.

The maximum value of the level threshold is set at 10/20.

Bibliography

- E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, 2011.
- B. Horowitz, *The Hard Thing About Hard Things: Building a Business When There Are No Easy Answers*. HarperBusiness, 2014.
- P. Thiel and B. *Masters, Zero to One: Notes on Startups, or How to Build the Future*. Crown Business, 2014.
- S. Blank and B. Dorf, *The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company*. K&S Ranch, 2020.
- R. Branson, *Screw It, Let's Do It: Lessons in Life and Business*. Virgin Books, 2016.

Teaching Unit (UE)	Subject Title	Code	Semester
UET3.1.1	Capstone Project III	**CP3**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	1h00	**2/2**

Objectives

- Form teams and identify project ideas.
- Conduct feasibility analysis and initial design documentation.
- Establish milestones and present progress to faculty advisors.

- Teamwork and collaboration.

- Problem identification and solution structuring.
- Technical writing and presentation skills.
- Project planning and feasibility assessment.

Prerequisites

No prerequisites

Program Outline

1. **Team Formation and Brainstorming (6 hours)**
   - Introduction to project selection.
   - Team roles and responsibilities.
   - Idea generation techniques.
2. **Feasibility Analysis and Initial Design Documentation (7.5 hours)**
   - Market and technical feasibility.
   - Risk assessment.
   - Drafting project proposals and documentation.
3. **Regular Milestones and Faculty Presentations (9 hours)**
   - Setting up project goals and deadlines.
   - First milestone presentation.
   - Refining project scope based on feedback.

Laboratory Work

No laboratory works

Evaluation Method

Module average = 100% continuous assessment.

The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject. The maximum value of the level threshold is set at 10/20.

Teaching Unit (UE)	Subject Title	Code	Semester
UED1.2.1	Training Internship III	**TI3**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	0	0	1h00	**1/1**

Objectives

- Introduce students to professional work environments.
- Develop basic technical and soft skills in an industry setting.
- Understand company structure, workflow, and project dynamics.

Prerequisites

No prerequisites

Program Outline

1. **Introduction to the Company**
   - Company overview and mission.
   - Workplace safety and regulations.
2. **Observation and Shadowing**
   - Understanding team roles and responsibilities.
   - Exposure to ongoing projects.
3. **Hands-on Experience**
   - Assisting in minor tasks and operations.
   - Learning basic tools and methodologies.
4. **Final Review and Feedback**
   - Supervisor feedback session.
   - Internship report preparation.

Laboratory Work

No laboratory works

Evaluation Method

Module average = 100% continuous assessment.
The level threshold is a grade equal to 60% of the average of grades obtained by all students in the cohort for the subject. The maximum value of the level threshold is set at 10/20.

g.Detailed programs for the semester 6

Teaching Unit (UE)	Subject Title	Code	Semester
	Internship and Final Year Project	**Final**	**S6**

			Crédit/ Coeff
VHH	15 Weeks	0	**30/30**

Objectives

The Internship and Final Year Project serve as critical educational components designed to consolidate academic knowledge while developing professional competencies. These experiences provide students with structured opportunities to apply theoretical concepts in practical settings, develop technical problem-solving abilities, and demonstrate their capacity for independent work. Through these professional experiences, students refine their analytical skills, enhance their project management capabilities, and establish meaningful industry connections that facilitate their transition from academia to professional practice.

The Final Year Project focuses on guiding engineering students toward addressing a specific engineering and/or research challenge. This culminating project is partially conducted within a company or laboratory setting in the form of an internship, creating a seamless integration between academic requirements and real-world professional experience. This approach ensures graduates possess the comprehensive skill set required by employers in the autonomous systems sector while providing them with authentic exposure to industry practices and methodologies.

Evaluation Method

- **Overall Module Evaluation Formula**

Module Average = 20% Internship Assessment + 80% Final Year Project

- **Final Year Project Grading Criteria**

1. Manuscript Quality (Document Formatting, Writing Quality)
2. Oral Presentation (Speech Quality and Delivery, Slide Quality and Visual Presentation, Understanding and Response to Questions)
3. Supervisor Assessment (Student Engagement and Commitment)
4. Innovation (Novelty and Originality of Solutions Provided)

Note: The detailed evaluation criteria for the Final Year Project will be defined by the school's scientific committees.

Agreements and conventions

Decisions and Approvals of Administrative and Advisory Bodies

Department
Scientific Board
School Direction