Semester 5 Modules: Specialization Embedded Software and Systems (Systèmes et Logiciels embarqués - SLE-)

code	Title	type	Coefficients	ECTS	Total work-load	Con-tact hours	Pri-vate study
AI.5.5	Robotics and soft computing	Optional	2	2	50	30	20
AI.5.12	Introduction to Deep Learning	Optional	1	1	25	15	10
DOS.5.4	Advanced Architectures and parallel program-ming models	Optional	2	2	50	30	20
DOS.5.5	Real Time Operating Systems	Compulsory	2	2	50	30	20
DOS.5.6	Distributed Systems and applications	Compulsory	2	2	50	30	20
DOS.5.7	Embedded Linux	Optional	2	2	50	30	20
DOS.5.8	Introduction to cloud computing	Compulsory	1	1	25	15	10
DOS.5.9	Software Development for robotics	Optional	1	1	25	15	10
DOS.5.10	Introduction to Digital EcoSystems: from IoT to Cloud computing	Optional	1	1	25	15	10
ESEP.5.1	Microcontroller-based systems	Compulsory	2	2	50	30	20
ESDV.5.2	System Integration (VHDL)	Compulsory	2	2	50	30	20
ESDV.5.3	Real-Time systems design and validation	Compulsory	2	2	50	30	20
ESDV.5.4	Cyber-Physical Systems	Optional	2	2	50	30	20
ESDV.5.5	Electronics for embedded systems	Compulsory	2	2	50	30	20
ESDV.5.6	Embedded Interfacing	Optional	1	1	25	15	10
ESDV.5.7	Integration project	Compulsory	1	1	25	15	10
ESDV.5.8	Microcontroller Project	Compulsory	1	1	25	15	10
ESDV.5.9	Reconfigurable Architectures	Optional	1	1	25	15	10
IAP.5.1	Soft Actuator Control and Applications	Optional	2	2	50	30	20
IAP.5.2	Programming connected objects	Optional	1	1	25	15	10
IAP.5.3	Positioning Systems and Applications	Optional	1	1	25	15	10
SEC.5.1	Diagnosis, safety and reliability of embedded sys-tems	Optional	2	2	50	30	20
SEC.5.2	IoT Security	Compulsory	2	2	50	30	20
SEC.5.3	Automotive Architecture and Security	Optional	1	1	25	15	10
SEC.5.4	IoT security project	Compulsory	1	1	25	15	10
ISA.5.12	Introduction to Datamining	Optional	1	1	25	15	10
	Complementary Module 1		2	2	50	30	20
	Complementary Module 2		2	2	50	30	20
	Complementary Module 3		2	2	50	30	20
	Complementary Module 4		1	1	25	15	10
	Complementary Module 5		1	1	25	15	10
	Complementary Module 6		1	1	25	15	10

AI.5.5 Robotics and Soft Computing

Module designation	AI.5.5 Robotics and Sof Computing
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Moncef TAGINA
Language	French
Relation to curriculum	optional
Teaching methods	lesson, lab works.
Workload (incl. contact hours, self-study hours)	Total workload: 50 contact hours: 30h (18h lecture, 6h exercise, 6h lab) private study : 20h
Credit points	2 ECTS
Required and recommended prerequisites for joining the module	MAT.1.1, M.1.2
Module objectives/intended learning outcomes	In terms of Knowledge: Soft computing is a set of so-called intelligent techniques that allow efficient control of automated systems and particularly autonomous mobile robots. The objective of this course is to present these techniques and make them accessible to students for implementation in practical cases. Competencies: C2, C3, C5, C9
Content	Introduction to Robotics and soft computing I. Classical commands in robotics (3 h) - Introduction and definitions - Position control - Speed control - Position and speed control - Limits of classical methods II. Fuzzy control (10.5 h) - Introduction and definitions - Fuzzy logic operator - Fuzzy rules - Inference - Fuzzy control process III. Neural control (9h) - Introduction and definitions - Deterministic Perceptron - Probabilistic Perceptron - Multi Layer Perceptron and error backpropagation IV. Path optimisationt (6h) - Implementation of neural control - Introduction and definitions - Genetic algorithm - Application in robotics
Examination forms	100% Final exam
Study and examination requirements	10/20

Reading list	Dieulot, J.-Y., Dubois, L., Borne, P., & Rozinoer, J. (1998). Introduction à la commande floue. Editions Technip. Faure, A. (2006). Classification et commande par réseaux de neurones. Editions Hermès. Santhosh, S. (2012). Soft-I-Robot: Soft Computing Techniques. LAP Lambert Academic Publishing.

AI.5.12 Introduction to Deep Learning

Module designation	AI.5.12 Introduction to Deep Learning
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Rym Besrour
Teaching team	Rym Besrour
Language	French
Relation to curriculum	optional
Teaching methods	Lesson and project
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact hours: 15h lessons Private study:10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	MAT.1.1: Probability and Statistics AI .3.1.: IA & Machine Learning Students must be competent in python.
Module objectives/intended learning outcomes	• Knowledge: • Understand generic machine learning terminology • Understand motivation and functioning of the most common types of deep neural networks • Understand the choices and limitations of a model for a given setting • Apply deep learning techniques to practical problems • Critically evaluate model performance and interpret results • Competences: Students are able to: • C1. To master in depth the basic sciences, in particular computer science and mathematics, essential for the design and production of computer applications. Competencies: C9

Content	Introduction Chapter1:Applied Math and Machine Learning Basics ● Linear Algebra ● Probability and Information Theory ● Numerical Computation ● Machine Learning Basics Chapter2:Modern Practical Deep Networks ● Deep Feedforward Networks ● Regularization for Deep Learning ● Optimization for Training Deep Models ● Convolutional Networks ● Sequence Modeling: Recurrent and Recursive Nets ● Practical Methodology ● Applications Chapter3 :Deep Learning Research ● Autoencoders ● Deep Generative Models Projects ideas : ● Smart routing ● Smart home security ● Smart energy managment
Examination forms	100% project evaluation
Study and examination requirements	10/20
Reading list	Géron, A. (2020). Deep Learning avec Keras et TensorFlow (2e édition). Dunod. Charniak, E. (2021). Introduction au Deep Learning. Dunod.

DOS.5.4 Advanced Architectures and parallel programming models

Module designation	DOS.5.4 Advanced Architectures and parallel programming models
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Lobna KRIAA
Teaching team	Chadlia Jerad
Language	French
Relation to curriculum	Optional
Teaching methods	Lesson & lab work.
Workload (incl. contact hours, self-study hours)	Total workload: 50h Contact hours: 30h (15.5h lessons, 5 hours exercises, 0.5h oral presentations, 9h labs) Private study:in hours: 20h
Credit points	2 ECTS
Required and recommended prerequisites for joining the module	EHA.2.1: Architecture & microprocessors EHA.3.1: Processor Design Methodology AP.1.3: Advanced C AP.2.1: Object-Oriented programming OS.3.03: Operating systems and concurrent programming Existing competencies: curiosity, autonomy
Module objectives/intended learning outcomes	• Knowledge: • Showcasing parallel architectures (advantages and disadvantages) • Ability to evaluate and analyse performances of different architectures (parallel and optimized) • Learning about parallel programming • Skills: Students know how to think parallel and try to write parallel programs Competencies: C5, C8, C13

Content	General Introduction Section I: Limits of classical architectures Section II: Definition of parallelism Section III Characteristics of parallel architectures Chapter I: Classification of Parallel Machines Section I: Flynn's classification (SISD, SIMD, MIMD) Section II: Raina's classification - Exercises related to SIMD and MIMD architectures (about 1h) Chapter II: Performance of computer architectures Section I: Importance of performance calculation Section II: Amdahl's law/Gustafson's law Section III: Units of performance measurement Section IV: SPEC standard - Exercises related to performance (about 2h with discussions) Chapter III: Cache coherence for parallel architectures Section I: Memory Cache o Definition o Type of cache memory mapping o Write problem Section II: Cache coherence protocols o Snoopy Bus o Directory-based protocols - Exercises related to performance (about 2h with discussions) Chapter IV: Basic Concepts for Parallel Programming Section I: Basic concepts Section II: Sharing for what purpose? Section III: Data Concurrency and Synchronization Chapter V : Programming with OpenMP - Introduction to OpenMP Section II: Studies of OpenMP pragmas - Private Variables vs Shared Variables - Parallel tasks - This chapter is fed by practical labs for each concept studied (about 9h) o Realization of the tests o Analysis of the results * Students in this context must choose another parallel programming model or language and make a presentation focusing on the parallel paradigm or concept offered by the chosen language
Examination forms	35% continuous evaluation + 65% written exam
Study and examination requirements	10/20

Reading list	Hennessy, J. L., & Patterson, D. A. (2021). Computer organization and design risc-v edition: The hardware software interface. Quemener, E. (November 2021). Des architectures parallèles aux programmes parallélisés - De la métrologie à la mesure de la performance [Course]. DOI:10.13140/RG.2.2.23467.72483 OpenMP. (n.d.). Retrieved from https://www.openmp.org/ Ayguadé, E., Copty, N., Duran, A., Hoeflinger, J., Lin, Y., Massaioli, F., ... & Zhang, G. (2008). The design of OpenMP tasks. IEEE Transactions on Parallel and Distributed Systems, 20(3), 404-418. Reed, D., Gannon, D., & Dongarra, J. (2022). Reinventing High Performance Computing: Challenges and Opportunities. arXiv preprint arXiv:2203.02544.

DOS.5.5 Real Time Operating Systems

Module designation	DOS.5.5 Real Time Operating Systems
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Latrach lassaad Mohamed Karim Bouafoura Jerad chedlia Masmoudi Mohamed
Teaching team	
Language	French
Relation to curriculum	Compulsory
Teaching methods	lab works, project
Workload (incl. contact hours, self-study hours)	Total workload:50h Contact hours 30h (9h lessons, 21h lab works) Private study:in hours: 20h
Credit points	2 ECTS
Required and recommended pre-requisites for joining the module	• Embedded System, operating system, Basic programming skills (preferrably C/C++). Hardware: Any STM32 Board
Module objectives/intended learning outcomes	After completion of this course, students should • Understand problems related to real-time applications and operating systems. • understand how to use real-time operating systems following the FreeRtos • Skills: Develop an Embedded Real Time software • Design real-time applications using freeRTOS Build real-time Competencies: C7, C8, C13

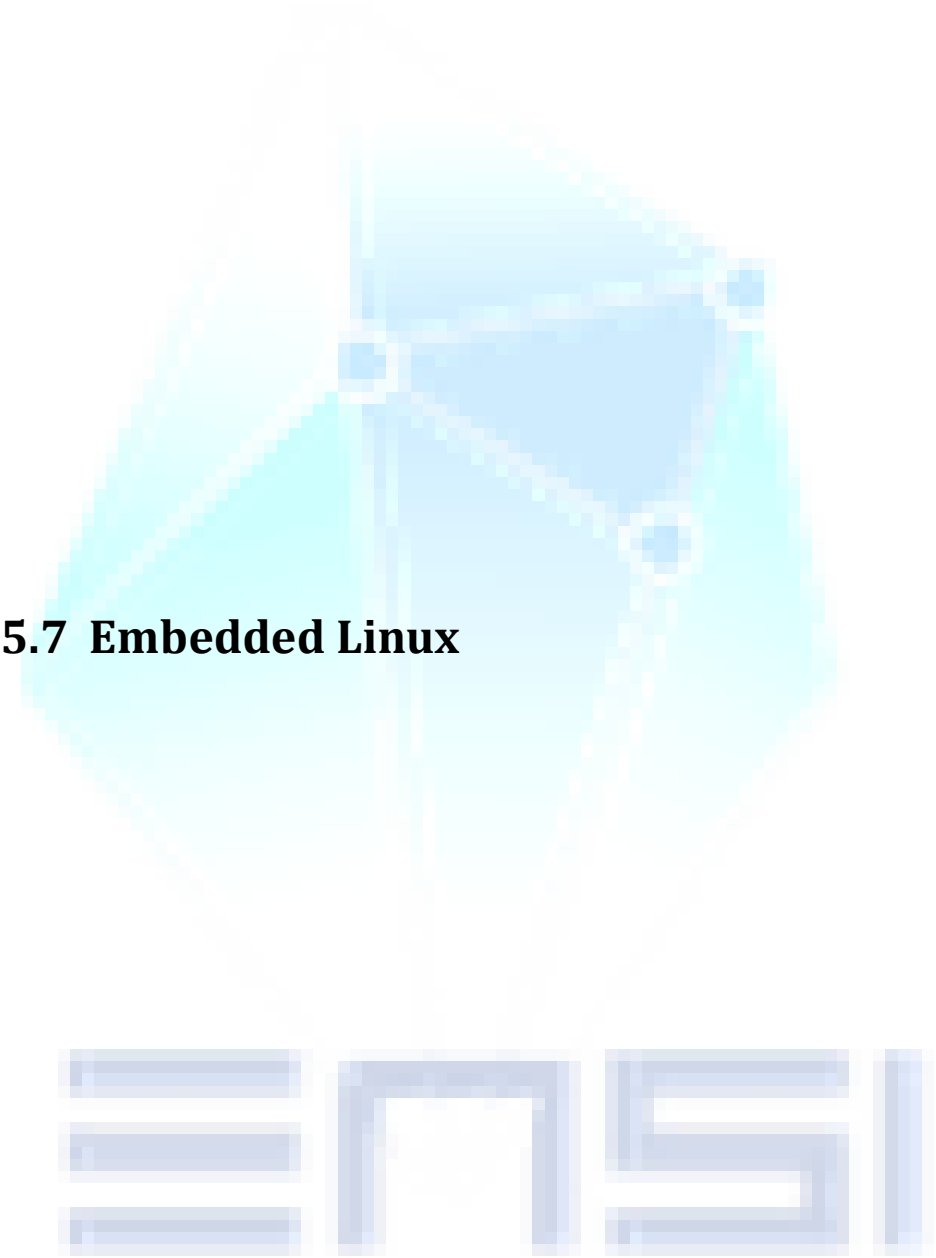
Content	Chapter I – Introduction 6. Introduction to Real-time systems and Embedded Real-time 7. Design objective for Real-time software 8. RTOS Task and Task state Chapter II - Real-time operating systems 9. Basic principles 10. Scheduling algorithms for periodic tasks: Rate Monotonic, Earliest Deadline First, Deadline Monotonic; 11. Scheduling algorithms for aperiodic tasks: schedulazione in background, Polling Server, Deferrable Server; Chapter III- System architecture of freeRTOS 12. Introduction to FreeRTOS 13. Task Management in Free RTOS 14. Synchronization in FreeRTOS 15. Creating FreeRTOS based project for STM32 MCUs 16. Inter Task Communication (FreeRTOS Queue) 17. Semaphores and Mutex (FreeRTOS) =>Lab Works (10h)
Examination forms	35% continuous assessment (project and mid-term exam) + 65% written exam
Study and examination requirements	10/20
Reading list	Buttazzo, G. C. Hard Real-time Computing Systems. Kluwer Academic Publishers. Building real-time embedded systems using FreeRTOS, STM32 MCUs, and SEGGER debug tools. Hands-On RTOS with Microcontrollers: Building real-time embedded systems using FreeRTOS, STM32 MCUs, and SEGGER debug tools.

DOS.5.6 Distributed Systems and Applications

Module designation	DOS.5.6 Distributed Systems and Applications
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Faïza NAJJAR
Language	French
Relation to curriculum	Compulsory
Teaching methods	Lessons, lab works, exercises & mini-project
Workload (incl. contact hours, self-study hours)	Total workload: 50H Contact hours: 30H (20H lessons, 10H lab works) Private study:in hours: 20H
Credit points	2 ECTS
Required and recommended pre-requisites for joining the module	Operating systems and Concurrent Programming (OS.3.03) Object-oriented Programming (AP.2.1)
Module objectives/intended learning outcomes	Objectives: With the omnipresence of distributed, multi-users and open computing, running constantly under different wide variations of platforms, this module provides a good comprehensive on fundamental concepts and issues of distributed systems. In particular, it masters in depth practical of the client-server programming applications and services upon (standard) middlewares (synchronous, asynchronous, transient, and persistent communications) with common and newest technologies. Competencies: C7, C8

Content	CHAP. 1. Client-server Programming (2 weeks) ● Introduction to distributed systems: basics concepts; architectures (C/S with request/response; Master-slaves; P2P; …) ● Client-server model: definitions; characteristics. (synchronous/asynchronous) communication ● Programming C/S with Remote Procedure Call (RPC): Interface description language –IDL; stubs; marshalling; binding; portmapper; XDR. LAB: C/S PROGRAMMING with RPC ● Synchronous Programming on laptops (with C on Linux) CHAP.2. gRPC Programming (1,5 weeks) ● Data serialization and Google protocol buffers –protobuf ● Message description Language (*.proto files) ● Message Encodage ● gRPC principles (https://grpc.io/) ● IDL Grpc; protoc compiler (from .proto to code) Programming Assignment: ● HelloWorld Application C/S (laptop & Smartphone-Android) with at least two different languages (Go; java; …). CHAP. 3 Message-oriented Middleware (MOM) (1 week) ● Communication types (Synchronous vs. Asynchronous ; Transient vs. Persistent) ● High-level middleware communication services: Point-to-point; Message queueing --MQ; publish-subscribe –pub-sub; pull and push delivery. ● MOM technology Java-message service –JMS. Mini-project: Asynchronous distributed programming with open MQ CHAP. 4. Peer-to-Peer systems and applications (2 weeks) ● Motivations: skype, Spotify, BitTorrent, ... ● Overlays on networks; Research issues (lookup/discovery, connectivity, communication, security, …). ● Types of P2P: centralized; decentralized; hybrid; semantic. ● Examples of P2P systems and applications: ● Distributed Hash table –DHT (on Chord) HOMEWORK : exercises on DHT CHAP. 5. Distributed file systems (2,5 weeks) ● Services and protocols ● A standard DFS: NFS ● A large scale DFS: HDFS ● Introduction to Big data ● Google FS –GFS: Master-slave architecture; Chunk; chunkserver; map-reduce distributed programming model. ● Hadoop distributed file systems: Namenode, Datanodes; JoTtracker, TaskTrackers; hdfs shells Mini-project: WordCount application with Hadoop map-reduce and hdfs

	Recapitulation (Review) and Cutting edge IT (last lesson before final exam)
Examination forms	40% CW + 60% Final Exam with CW -- Continuous work =Mean(test; (programming) assignments, HomeWorks; classroom exercises)
Study and examination requirements	10/20
Recommended readings	• A. S. Tanenbaum, M. V. Steen, Distributed Systems: Principles and Paradigms, Createspace Independent Pub; 2nd edition (26 February 2016), ISBN-10: 153028175X (ISBN-13: 978-1530281756). Free download onhttps://www.distributed-systems.net/index.php/books/ds3/ (third edition, 2017). • Max Hailperin, Operating Systems and Middleware: Supporting Controlled Interaction, Revised Edition 1.3.1, Max Hailperin, June 4, 2019 • Indranil Gupta, Advanced distributed systems, P2P systems, Spring 2018 • Martin Crane, Messaging on Distributed Systems (CA4006) LN (2017) (http://www.computing.dcu.ie/~mcrane/CA4006)



DOS.5.7 Embedded Linux

Module designation	DOS.5.7 Embedded Linux
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Lobna KRIAA
Teaching team	Industrial partner team (https://www.actia.com/fr/ : Tunisian pole) • Lassaad Jemai
Language	Frensh
Relation to curriculum	Optional
Teaching methods	Lesson and labs
Workload (incl. contact hours, self-study hours)	Total workload: 50h Contact hours: 30h (21h lessons, 9h labs) Private study: 20h
Credit points	2 ECTS
Required and recommended pre-requisites for joining the module	OS.2.1: Introduction to operating systems and Unix environment EHA.4.1: Introduction to embedded Systems EHA.5.2: Microcontroller-based system
Module objectives/intended learning outcomes	• Knowledge: • Understand the design of an embedded operating system • Install a cross-compilation chain • Use Buidlroot • Configure a cross-compilation chain • Generate an image for a specific chipset with Buildroot • Competences: Students are able to: • C5. Specify an information system in all its dimensions: software, hardware and architecture using the state-of-the-art available technologies and considering the company's strategic objectives Competencies: C7, C8, C13
Content	Introduction Section I: Linux OS presentation Section II: definition of embedded systems Chapter 1: Linux license and GNU/Linux architecture Section I: Linux license Section II: GNU/Linux architecture Section III: Bootloader Section IV: kernel Section V: Rootfs Chapter 2: Modules integration within Linux kernel Chapter 3: File system management Chapter 4: Processes and multithreading Chapter 5: Buildroot Practical labs will be proposed in order to: • Install a cross-compilation chain • Use Buidlroot • Configure a cross-compilation chain • Generate an image for a specific chipset with Buildroot

Examination forms	35% continuous evaluation +65% written exam
Study and examination requirements	10/20
Reading list	Blanc, G. (2011). Linux embarqué: comprendre, développer, réussir. Pearson Education France. Ficheux, P. (2017). Linux embarqué: Mise en place et développement. Eyrolles. OpenClassrooms. (n.d.). Créez un Linux embarqué pour la domotique. Retrieved from https://openclassrooms.com/fr/courses/5281406-creez-un-linux-embarque-pour-la-domotique

DOS.5.8 Introduction to cloud computing

Module designation	DOS.5.8 Introduction to Cloud Computing
Semester(s) in which the module is taught	5
Person responsible for the module (coordinator)	Dr. Mehrez Essafi
Teaching team	-
Language	French
Relation to curriculum	Compulsory
Teaching methods	• Lesson • Lab work
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact hours: 15h (12h lessons, 3h lab work) Private study: 10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	OS.2.1 Introduction to Operating systems and Unix environment NET.3.1 Local Networks NET.4.1 Computer Networks SE.4.2 Software Architecture SEC.4.1 Cybersecurity & cryptography
Module objectives/intended learning outcomes	• To understand the related architecture designs and technologies of cloud computing • To explain the overall architecture and key design principles of IoT systems, including both functional and non-functional aspects • To design cloud-based IoT applications using proper cloud services • To gain hands-on experience in key technologies for developing an IoT system, including sensor selection / interfacing, embedded system programming, use of suitable network protocols, and various cloud services (compute, storage, data analytics, management) • To gain design experience in solving a real-world problem using IoT and cloud technologies • To gain both the experience and confidence in learning a new IoT technology independently By the end of the course, students are expected to be able to: • Describe the IoT and Cloud architectures. • Deploy Cloud Services using different cloud technologies. • Implement cloud computing elements such virtual machines, web apps, mobileservices, etc. • Establish data migration techniques from IoT devices to the cloud. • Implement security features to protect data stored in the cloud. • Use visualisation techniques to show data generated from the IoT device. Competencies: C2, C3, C4, C5, C6, C13

Content	Unit 1 – Cloud Computing: main concepts ● General introduction ● Historical overview ● Cloud characteristics ● Business model ● Advantages and limits Unit 2 – Data centers ● Definitions ● Main components ● Green Computing ● Security ● High Availability Unit 3 – Cloud Services and deployment models ● IaaS (Infrastructure as a Service) ● PaaS (Platform as a Service) ● SaaS (Software as a Service) ● FaaS (Function as a Service) ● Other services ● Public Cloud ● Private Cloud ● Hybrid Cloud ● Community Cloud Unit 4 – Virtualization ● Definitions ● Architectures ● Solutions ● Servers virtualization ● Containers ● Storage virtualization Unit 5 – Application of IoT & Cloud ● IoT and cloud integration ● Application development and cloud processing ● Security and Privacy for IoT/Cloud Computing
Examination forms	● 20% labs // 80% written examination
Study and examination requirements	Student must achieve an overall minimum module mark of 10/20

Reading list	Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing (800-145). National Institute of Standards and Technology (NIST). Duncan, C. H. (2017). Cloud computing gateway, cloud computing hypervisor, and methods. International Conference on Cloud Computing. Hennion, R., Tournier, H., & Bourgeois, E. (2014). Cloud computing: Décider - Concevoir - Piloter – Améliorer. Plouin, G. (2014). Cloud Computing, Sécurité, stratégie d'entreprise et panorama du marché. Collection InfoPro, Dunod. Rapport Cigref. (2013). Fondamentaux du Cloud Computing: Le point de vue des Grandes Entreprises. Moyer, C. M. (2011). Building Applications in the Cloud: Concepts, Patterns, and Projects. Addison-Wesley. Marks, E. A., & Lozano, B. (2010). Executive's Guide to Cloud Computing. Wiley. Fagroud, F. Z., Benlahmar, E. H., Elfilali, S., & Toumi, H. (2019). IoT et Cloud Computing: état de l'art. Colloque sur les Objets et systèmes Connectés, Ecole Supérieure de Technologie de Casablanca (Maroc), Institut Universitaire de Technologie d'Aix-Marseille (France), CASABLANCA, Maroc. Odun-Ayo, I., Okereke, C., & Evwieroghene, O. (2018). Cloud Computing and Internet of Things - Issues and Developments. Christos, S., Kostas, P., Byung-Gyu, K., & Gupta, B. B. (2016). Secure Integration of Internet-of-Things and Cloud Computing. Future Generation Computer Systems. Vertiv. (2018). L'impact du cloud et de l'internet des objets sur la demande en datacenters, Livre blanc et noir.

DOS.5.9 Software Development for Robotics

Module designation	DOS.5.9 Software Development for Robotics
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Med Karim Bouaffoura
Teaching team	
Language	French
Relation to curriculum	Optional
Teaching methods	Lesson and Project
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact Hours: 15h (100% project development, testing, and validation) Private study:in hours: 10h
Credit points	1 ECTS
Module objectives/intended learning outcomes	• Knowledge: • Master the Robot Operating Systems • Deal with models and simulations for robots such as gazebo environment • Skills: • Work with a team • manage time • Systematically take initiative to realize creative ideas Competences: C4, C5, C7
Content	Lesson : 1. Introduction 2. The complexity of robotic systems 3. ROS-1 Introduction 4. Models and simulation 5. ROS-2 Introduction 6. Labs demonstration of different existing projects • Project description: • The module involves spending at least 4 hours per week on the project. • The teams are made up of 2 to 4 students (depending on the total number of students per class) • Each group must choose a robot application for different application domain E-health, E-agriculture, IIndustry4.0, etc. • The groups have to model and simulate them and probably implement them using hardware circuit such as raspberry, etc.
Examination forms	100% project evaluation
Study and examination requirements	10/20

Reading list	Koubâa, A. (Ed.). (2017). Robot Operating System (ROS) (Vol. 1, pp. 112-156). Cham: Springer. Dieber, B., Breiling, B., Taurer, S., Kacianka, S., Rass, S., & Schartner, P. (2017). Security for the robot operating system. Robotics and Autonomous Systems, 98, 192-203. Quigley, M., Gerkey, B., & Smart, W. D. (2015). Programming Robots with ROS: A practical introduction to the Robot Operating System. O'Reilly Media, Inc. Stasse, O. (2018). Robot Operating System introduction. LAAS-CNRS.

DOS.5.10 Introduction to Digital EcoSystems: from IoT to Cloud computing -DES

Module designation	DOS.5.10 Introduction to Digital EcoSystems: from IoT to Cloud Computing- DES
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Faïza NAJJAR
Language	English
Relation to curriculum	Optional
Teaching methods	Lessons, mini-projects and research presentations
Workload (incl. contact hours, self-study hours)	Total workload: 25H Contact hours (lectures, (classroom/homework) exercises, lab. System programming): 15H (10H lessons, 5H presentations) Private study:in hours: 10H
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	Distributed Systems and Applications
Module objectives/intended learning outcomes	Objectives: • A Roadmap from IoT to cloud (IoT-Cloud) • To Draw a panorama of emerging information and communication technologies (Cutting edge technologies) • To analyse issues raised and opportunities offered by these technologies and their integration • To Highlight open directions and research questions (top 10 trends IEEE, Gartner). Competencies: C7, C8 .
Content	1. A (Quick) Historical Panorama of Information Technologies 2. Digital Ecosystems: A (Rather) New Vision of IT 3. An Introduction to Large Scale Computing 4. Towards the Internet of Things --IoT? 5. An Introduction to Ubiquitous and Pervasive Computing 6. Multi-Scale Computing - Cloud-IoT: the Greatest Challenge for this Decade? Open Directions for Research and Innovation
Examination forms	30% CW + 70% Final Exam with CW --Continuous work = Mean(classroom exercises; Mini-projects; presentation)
Study and examination requirements	10/20
Reference	• Lionel Brunie, De l'Internet des objets au Cloud : une introduction aux nouveaux écosystèmes numériques, cours Mastère INSA-Lyon, 2018

ESDV.5.1 Microcontroller-Based Systems

Module designation	ESDV.5.1 Microcontroller Based Systems
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Latrach lassaad Mohamed Masmoudi
Teaching team	
Language	French
Relation to curriculum	Compulsory
Teaching methods	lab works, project
Workload (incl. contact hours, self-study hours)	Total workload: 50h Contact hours: 30h (9h lessons, 21h lab works) Private study: 20h
Credit points	2 ECTS
Required and recommended pre-requisites for joining the module	• Analog Ecletronic, Embedded System, Basic programming skills (preferrably C/C++). Hardware: Any STM32 Board
Module objectives/intended learning outcomes	Key question: what learning outcomes should students attain in the module? E.g. in terms of: • Knowledge to acquaint students with the interfaces between sensors, network devices and ARM Cortex devices; Skills: By the end of the course, students will know: • basics of programming for ARM Cortex devices • general principles and approaches to debugging and verification of embedded systems. • perform modeling, optimizing for IOT project Competencies: C5, C9
Content	Chapter I – Introduction to ARM® Cortex® (4h) 1. Based design. STM32CubeMX 2. Compiling, downloading, and running 3. simple programs on an evaluation board STM32. Chapter II - Processors and STM-Library – review (HAL/LL) (6) 1. tools and libraries (HAL/LL) for STM32-based design 2. Programming I/O, investigates some of the functions that configure I/O Chapter III- Peripheral Programming (20h) 2. Digital interfaces. 3. Graphic LCD Interfacing 4. Programming interface UART. I2C, PWM ,ADC SPI, CAN interfaces =>Lab Works (10h)
Examination forms	35% continuous assessment (project and mid-term exam) + 65% written exam
Study and examination requirements	10/20

Reading list	Furber, S. ARM System-on-Chip Architecture. Pearson Education. Yiu, J. (2013). The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors (3rd ed.). Newnes.

ESDV.5.2 System Integration (VHDL)

Module designation	ESDV.5.2 System Integration (VHDL)
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Karim Bouaffoura
Teaching team	Mohamed Masmoudi Mourad Fathallah
Language	Frensh
Relation to curriculum	Compulsory
Teaching methods	Lessons and labs with Modelsim quartus II and Vivado.
Workload (incl. contact hours, self-study hours)	Total workload: 50h Contact Hours: 30h (20h lesson, 10h labs) Private study:in hours: 20h
Credit points	2 ECTS
Required and recommended pre-requisites for joining the module	- Good coding knowledge - Digital Electronics
Module objectives/intended learning outcomes	• Knowledge: • Understand the VHDL language feature to realize the complex digital systems • Design sequential and concurrent techniques in VHDL • Modeling of digital systems using VHDL and design methodology • Understand behavioral, non-synthesizable VHDL and its role in modern design • Simulate and debug digital systems described in VHDL. • Students must be able to synthesize complex digital circuits at several levels of abstractions. • Ability to implement logic on an FPGA • Introduce to system on chip design and implementation Competences: C5, C7
Content	This course instructs the students in the use of VHDL ((Very High Speed Integrated Circuit Hardware Description Language) for describing the behavior of digital systems. • VHDL is a standardized design language used in computer/semiconductor industry. This course will teach students the • use of the VHDL language for the representation of digital signals • use of IEEE standard logic package/library • design description, and design of arithmetic, combinational, and synchronous sequential circuits. • Use of Modelsim environment for simulation
Examination forms	35% continuous evaluation +65% written exam
Study and examination requirements	10/20

Reading list	Zvolinski, M. (2003). Digital system design with VHDL. Prentice Hall. Hsu, Y. C., Tsai, K. F., Liu, J. T., & Lin, E. S. VHDL Modeling for Digital Design Synthesis. Kluwer Academic Publishers. Mazor, S., & Langstraat, P. A guide to VHDL. Kluwer Academic Publishers. Messerli, E. Manuel VHDL, synthèse et simulation. HEIG-VD. Doulos. The VHDL Golden Reference Guide, compatible IEEE std 1076-2002. Retrieved from http://www.doulos.com/

ESDV.5.3 Real-Time systems design and validation

Module designation	ESDV.5.3 Rela Time Systems Design and Validation
Semester(s) in which the module is taught	S5
Person responsible for the module	Leila Ben Ayed
Language	French
Relation to curriculum	Compulsory
Teaching methods	lesson, project (TINA validation Tool)
Workload (incl. contact hours, self-study hours)	Total workload: 50h Contact hours: 30h (20h Lessons, 10h Exercices and Project) Private study: 20h
Credit points	2 ECTS
Required and recommended pre-requisites for joining the module	Automata (AP.1.1), Operating Systems (OS.2.1) Compilation Techniques (AP.3.2), Formal Development Methods (SE.4.1)
Module objectives/intended learning outcomes	The objective of this lesson is to acquire the basic concepts and methods of process modeling, the theory of Petri nets and its applications in system modeling, design, and verification. To be able to use Petri net-based computer- aided tools in typical applications. Timed automata are one of the models of continuous-time reactive systems. The second part is devoted to timed automata and to the verification of the formulas of the timed tree logic TCTL. ● . ● Knowledge: ● Design models with Petri Net and Timed automata ● Parallelism and synchronization modeling of critical distributed systems ● Model Validation Competencies: C1, C2, C3
<u>Content</u>	Lesson I (06h). Introduction to Real-Time Systems ● Automated Real-Time systems ● Required behaviour properties in Real-Time systems ● Limits of finite state automata Lesson II (12h). Petri Net - Presentation - Modelling of synchronisation and parallelism - Bihaviour properties - Validation of PNt models (TINA project) - Colored Petri Net - Tiled Petri Net Lesson III (12h) .Timed automata - Presenation - Modeling in Tmed Aumata - Timed Computation Tree Logic (TCTL) - Verication with UPPAAL(an integrating tool environment for medeling, validation and verification of Real-Time systems

Examination forms	35% Continues evaluation + 65% Written exam
Study and examination requirements	10/20
Reading list	Choquet-Geniet. (1998). Les Réseaux de Petri: Un outil de modélisation. DUNOD. Schnoebelen, P. H. (1999). Vérification de logiciels: Techniques et outils du model-checking. Vuibert.

ESDV.5.4 Cyber-Physical Systems

Module designation	ESDV.5.4 Cyber-Physical Systems
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Chadlia Jerad
Teaching team	
Language	English
Relation to curriculum	Optional
Teaching methods	Lesson and Project
Workload (incl. contact hours, self-study hours)	Total workload: 50h Contact Hours: 30h (20h lesson, 10h labs) Private study: 20h
Credit points	2 ECTS
Required and recommended pre-requisites for joining the module	EHA.4.1:Introduction to embedded Systems Automates ; NET.3.1 Local Area Networks, NET4.1 Computer Networks, AP.1.3 Advanced C programming
Module objectives/intended learning outcomes	• Knowledge: • Identify a principled, scientific approach to designing and implementing Cyber-Physical Systems, with a focus on model-based system design, and on embedded software. • Skills: • Work with a team • manage time • Systematically take initiative to realize creative ideas Competences: C4, C5, C7

Content	Introduction Chapter 1: Motivation: Cyber Physical Systems ● Embedded Systems vs. Cyber Physical Systems ● The Hype Cycle; CPS Requirements and challenges ● Examples of CPS and IoT applications ● IoT Scene; Utility vs. Usability Chapter 2: Model Based Design ● Iterative process: modelling, design and analysis ● Model based design vs. Statistical methods ● The value of models, determinism, Models vs. implementations, Abstraction layers ● Models of Computation: State machines, FSM and their composition; Hybrid Automata; Dataflow and KPN ● Video Resource: Drone Simulation and Control (https://www.youtube.com/watch?v=gEmGfo36INc) Practical session 1: Practical exercises on Modeling Chapter 3: Sensors and Actuators ● Modelling issues with sensors and actuators ● Affine Model of Sensors ● Bias and Sensitivity ● Range and dynamic range ; Precision and quantization ● Faults in Sensors ; Sensor fusion ● Brief Overview of Actuators Chapter 4: Embedded Systems Programming ● Roles of Memory ● Memory Architecture issues ● Memory organization (statically allocated, stacks, heaps) ● Memory hierarchies (scratchpads, caches); Processors; Parallel and serial digital interfaces ● IO mechanisms in Software (Polling vs Interrupt) ; Concurrency Reading: Scientific paper selection from the proceedings of “Design Automation Conference” Chapter 5: Networking ● Communication technologies and protocols, and their use ● Wired Networks (CAN, TTP, FlexRay, TTEthernet, TSN (a.k.a. AVB)) ● Radio technologies (IEEE 802.15.4 (Zigbee, WirelessHART), Bluetooth, NFC) ● 6LowPAN; CoAP ● Ultrasonic ; Visible light ; Vibration ● IoT infrastructure and Cloud Service Composition (IFTTT) Chapter 6: IoT Ecosystem ● IoT visions, views, characteristics, requirements and building blocks ● Reference Architecture for IoT ● Resource management in IoT ● Resource partitioning: Container; Virtualization ● Scheduling and provisioning: Cloud vs. Fog ● Resource/Device discovery and selection Chapter 7: IoT Analytics ● Publish subscribe for IoT : MQTT ● Sensors reading and actuation ● MQTT broker, clients and topics ● Web of Things ● REST ● WoT Integration Patterns: Direct integration Pattern, Gateway Integration Pattern, Cloud Integration Pattern ● CoAP Practical sessions 2: ● Lab 1 : Hands-on Node-Red

	• Lab 2: Simple smart light control • Lab 3 : Hand gesture for smart light control
Examination forms	40% continuous evaluation +60% written exam
Study and examination require-ments	10/20
Reading list	Texbooks : Lee, E. A., & Seshia, S. A. (2015). Introduction to Embedded Systems – A Cyber-Physical Systems Approach (2nd ed.) [Ebook]. Buyya, R., & Dastjerdi, A. V. (Eds.). (2016). Internet of Things: Principles and Paradigms. Elsevier. Ptolemaeus, C. (Ed.). (2014). System Design, Modeling, and Simulation Using Ptolemy II [Ebook]. Ptolemy.org. UC Berkeley. Based on “Introduction to Embedded Systems”: EECS 149/249A UC Berkeley. Scientific papers: Design Automation Conference. Proceedings of the Design Automation Conference Tools and frameworks: • Ptolemy II tool:https://ptolemy.berkeley.edu/ptolemyII/ptII11.0/index.htm • Node-Red:https://nodered.org/

ESDV.5.5 Electronics for embedded systems

Module designation	ESDV.5.5 Electronics for Embedded Systems
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Latrach lassaad
Teaching team	
Language	French
Relation to curriculum	Compulsory
Teaching methods	lab works, project
Workload (incl. contact hours, self-study hours)	Total workload:50h Contact hours:30h (9h lessons, 21h lab works) Private study: 20h
Credit points	2 ECTS
Required and recommended prereq-uisites for joining the module	EHA.2.1 Architecture and microprocessors EHA.4.1 Introduction to Embedded Systems EHA.5.2: Microcontroller based systemnic
Module objectives/intended learning outcomes	Key question: what learning outcomes should students attain in the module? E.g. in terms of: ● Knowledge a student should be able to design and analyze digital systems, incorporating into a VLSI chip. They should be able to design for low power and design for performance, work in small groups and bring together design components into a full custom chip; Skills: Use of CAD tools for both schematic and layout of complex CMOS circuits; Design of arithmetic logic and memory cells; Large system integration. Competencies: C5, C9
Content	Chapter I – Introduction to CMOS Technology and Device 1. Design metrics 2. MOS Devices 3. CMOS Inverter. Chapter II - Design and Layout of Simple Logic Gates 1. Combinational logic, layout, design rules 2. Manufacturing process; 3. Simulation; CAD tools 4. Low Power design strategies Chapter III- Static, Dynamic sequential circuits 1. Deep sub-micron designs; design for performance 2. Wires; Coping with Interconnects 3. Adders, Multipliers, data paths; timing issues 4. Memory Circuits 5. Design and optimize complex functional blocks =>Lab Works (10h)
Examination forms	35% continuous assessment (project and mid-term exam) + 65% written exam

Study and examination requirements	10/20
Reading list	Brunvand, E. (2009). Digital VLSI Chip Design with Cadence and Synopsys CAD Tools. Addison Wesley. Weste, N., & Harris, D. (2010). CMOS VLSI Design: Circuits and Systems Perspective (4th ed.). Addison Wesley.

ESDV.5.6 Embedded Interfacing

Module designation	ESDV.5.6 Embedded Interfacing
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Latrach lassaad
Teaching team	
Language	French
Relation to curriculum	Optional
Teaching methods	lab works, project
Workload (incl. contact hours, self-study hours)	Total workload:25h Contact hours:15h (3h lessons, 13 h lab works) Private study: 10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	Microcontroller Based System, Analog Ecletronic
Module objectives/intended learning outcomes	Key question: what learning outcomes should students attain in the module? E.g. in terms of: ● Knowledge Altium Designer offers functions for: schematic entry, PCB design/routing,simulations, ● Skills: Design of embedded boards and electronic components Competencies: C5, C9
Content	Chapter I – Getting Started with Altium Designer (3h) 1 Introduction to Altium Designer 2 The Altium Designer environment 3 Working with projects and documents Chapter II - Schematic Editor Basics (4h) 1 Schematic Editor Basics 2 Schematic graphical objects 3 Schematic electrical objects =>Lab Works (3h) Chapter III Schematic Capture (9) 1Introduction to Schematic Capture 2The Schematic Editor workspace 3Libraries and components 4Placing and wiring 5 design of an electronic card based on stm32 =>Lab Works (6h)
Examination forms	100% Lab exam
Study and examination requirements	10/20
Reading list	Altium Designer PCB Design Official Guide (Basic Application) (EDA Engineering Technology Series) Author. Fast PCB Design with Altium Designer (Industrial Automation and Control)

ESDV.5.7 Integration Project

Module designation	ESDV.5.7 Integration Project
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Lobna KRIAA
Teaching team	Industrial partner team Alpha Connect International (https://www.linkedin.com/company/alpha-connect-international/) ● Mohamed Masmoudi
Language	French
Relation to curriculum	Compulsory
Teaching methods	Project
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact Hours: 15h (100% project development, testing, and validation) Private study: 10h
Credit points	1 ECTS
Required and recommended pre-requisites for joining the module	EHA.2.1: Architecture & microprocessors EHA.4.1: Introduction to embedded Systems EHA.5.2: Microcontroller-based system Integration System
Module objectives/intended learning outcomes	● Knowledge: ● Master the prototyping concept using FPGA ● Master VHDL language ● Deal with Codesign approaches ● Skills: Work with a team, manage time ● Competences: Students are able to: ● C4. Identify, formulate and solve complex or incompletely defined IT problems with the objective of ensuring maximum availability for users Competencies: C5, C7

Content	This is a project-based module. ● The students try to integrate their knowledge of the two modules "Systems Integration (VHDL)" and the module "Microcontroller-based systems". ● The idea is to generate an SoC design approach by integrating software and hardware parts into a prototype using FPGA prototyping platform. The class will be divided in groups: ● Each group will choose a system to implement and to develop. ● In this part, different types of projects for example ● Implementation of a mini microcontroller using one peripheral (e.g. SPI, I2C, UART, GPIO, Timer, etc.) and integrating a processor (e.g. NIOS, µBlaze, etc.) ○ Implementation of the software layer ● Implementation of an image processing application or a games ● Define an outline of the project ● Divide the work and assign task to each team member ● Integrate individual works ● The module involves spending at least 4 hours per week on the project. ● The teams are made up of 2 to 4 students (depending on the total number of the students per class) ● Hardware need: o FPGA boards baseds o Necessary connection cables o Necessary equipment to be provided by the participant: Laptop computer with administration rights
Examination forms	100% project evaluation
Study and examination requirements	10/20
Reading list	● Intel® Quartus® Prime Pro Edition User Guide: Platform Designer Updated for Intel® Quartus® Prime Design Suite: 22.3 ● Creating a System Design with Platform Designer: Getting Startedhttps://www.youtube.com/watch?v=d43Pqc_IZpg&list=PL0pU5hg9yniZ2ka-XBXROXNR0pAEAEFCB&index=10 (consulted October 2022) ● Introduction to Platform Designer (formerly Qsys)https://www.youtube.com/watch?v=6srZw3kGXA8&list=PL0pU5hg9yniZ2ka-XBXROXNR0pAEAEFCB&index=8 (consulted October 2022) ● How to generate a Soc Design with Quartus (https://www.youtube.com/watch?v=8BehnPg8IvM) (consulted October 2022) ● Adding Custom component using Qsyshttps://www.youtube.com/watch?v=mOc3ErBiuGY (consulted October 2022) ● Understanding Avalon MM Bursting (https://www.youtube.com/watch?v=8GAqT3nzHeQ (consulted October 2022) ● Peter J. Ashende,Jim Lewis. The Designer's Guide to VHDL. Third edition, Elseiver, 2008.

ESDV.5.8 Microcontroller Project

Module designation	ESDV.5.8 Microcontroller Project
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Mohamed Masmoudi - Industrial partner Alpha Connect International (https://www.linkedin.com/company/alpha-connect-international/)
Teaching team	
Language	French
Relation to curriculum	Compulsory
Teaching methods	Project
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact Hours: 15h (100% project development, testing, and validation) Private study: 10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	EHA.2.1: Architecture & microprocessors EHA.4.1: Introduction to embedded Systems ESDV.5.1 Microcontroller-based system
Module objectives/intended learning outcomes	• Knowledge: • Master STM32 microcontroller • Master peripheral interfaces • Skills: • Work with a team • manage time • Systematically take initiative to realize creative ideas Competences: C4, C5, C7

Content	This is a project-based module.
	● The students try to apply their knowledge of the learned course Microcontroller Based System
	● The idea is to develop a real embedded application based on STM32 using STM32 tools
	● Steps of the project
	  ● Define an outline of the project
	  ● Divide the work and assign task to each team member
	  ● Integrate individual works
	● The module involves spending at least 5 hours per week on the project.
	● The teams are made up of 2 to 4 students (depending on the total number of students per class)
	● Hardware need:
	  ● Electronic boards based on microcontrollers
	  ● Necessary connection cables
	  ● Other electronic components for the possible assemblies
	  ● Necessary equipment to be provided by the participant: Laptop computer with administration rights

Examination forms	100% project evaluation
Study and examination requirements	10/20
Reading list	● https://www.alldatasheet.com/view.jsp?Searchword=Stm32f4-discovery%20datasheet&gclid=EAIaIQobChMIr8rc_rnX-gIVRgWLCh2gbwosEAAYASAAEgLBavD_BwE ● https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html. (visited 10-06-2022)

ESDV.5.9 Reconfigurable Architectures

Module designation	ESDV.5.9 Reconfigurable Architectures
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Lobna KRIAA
Teaching team	Industrial partner Siemens Mentor Graphicshttps://eda.sw.siemens.com/en-US/ . Zied Marrakchi
Language	French
Relation to curriculum	Optional
Teaching methods	Lessons
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact Hours: 15h (10h lessons with 5 hours exercises) Private study:10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	EHA.4.1: Introduction to embedded Systems ESDV.5.2 Integration System
Module objectives/intended learning outcomes	• Knowledge: • Master the FPGA EDA steps • Know the optimization algorithm for mapping and routing steps • Deal with Codesign approaches Competences: C4, C5
Content	Introduction and Generalities Chapter 1: Mesh Architectures Chapter 2: Tree-based architectures Chapter 3: Configuration and algorithmic tools Chapter 4: Embedded FPGA
Examination forms	100% written exam
Study and examination requirements	10/20
Reading list	Farooq, U., Marrakchi, Z., & Mehrez, H. (2012). FPGA architectures: An overview. Tree-based heterogeneous FPGA architectures, 7-48. Farooq, U., Marrakchi, Z., & Mehrez, H. (2012). Tree-based heterogeneous FPGA architectures: Application specific exploration and optimization. Springer Science & Business Media. Saidi, H., Turki, M., Marrakchi, Z., Abid, M., & Obeid, A. (2021). Soft-core embedded FPGA based system on chip. Analog Integrated Circuits and Signal Processing, 109(3), 517-533.

IAP.5.1 Soft Actuator Control and Applications

Module designation	IAP.5.1 Soft Actuator Control and Applications
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Yasser GRITLI
Teaching team	Yasser GRITLI
Language	French
Relation to curriculum	optional
Teaching methods	Projects
Workload (incl. contact hours, self-study hours)	Total workload: 50h Contact hours: 30h (25h Lessons, 10h Exercise) Private study: 20h
Credit points	2 ECTS
Required and recommended prerequisites for joining the module	MAT.1.1, AP.1.2, MAT.1.2, EHA.1.1, EHA.1.2, NET.2.1, EHA.4.1, NET.4.1.
Module objectives/intended learning outcomes	This course is provided as Projects to develop skills and Knowledge in : - Different techniques of Smart Measurements; - New technologies of Soft and Smart Sensors; - Modern Soft and Smart Actuators; - Concepts of Soft Actuators Control Strategies implementation. Competencies: C1, C4, C5
Content	-- Project 1: Drone actuator Project 2: Soft-robotics in medical applications Project 3: Smart Agriculture -- Lesson 1* Smart Measurements - Basic Metrology concepts - Sensors Metrology Lesson 2* Soft and Smart Sensors - Basic Sensors principle - Soft Sensors - Smart Sensors and data communication systems Lesson 3* Soft and Smart Actuators - Basic actuators principle - Soft-Actuators topologies - Data communication system requirements Lesson 4* Soft Actuators Control Strategies - Basic Control Strategies - Control strategies and applications - Control implementation

	Practical exercises are set every week. Students have to work in fixed groups of 2–4 using the simulators dedicated to each Project. *: Lessons 1 to 4 are provided in parallel with the different projects progress.
Examination forms	100% project evaluation The exam at the end of term will test everything in the course, but will be especially closely tied to the practical exercises.
Study and examination requirements	10/20
Reading list	Sharma, K., & Saini, L. M. (2015). Performance analysis of smart metering for smart grid: An overview. Renewable and Sustainable Energy Reviews, 49, 720-735. Jiang, Y., et al. (2020). A review on soft sensors for monitoring, control, and optimization of industrial processes. IEEE Sensors Journal, 21(11), 12868-12881. Hines, L., et al. (2017). Soft actuators for small-scale robotics. Advanced Materials, 29(13),

IAP.5.2 Introduction to Programming Connected objects

Module designation	IAP.5.2 Introduction to Programming Connected Objects
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Latrach Lassaad
Teaching team	
Language	French
Relation to curriculum	Optional
Teaching methods	lab works, project
Workload (incl. contact hours, self-study hours)	Total workload:25h Contact hours: 15h (3h lessons, 13 h lab works) Private study: 10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	EHA.4.1- Embedded System EHA.1.2 - Analog Electronics
Module objectives/intended learning outcomes	- Know the technical knowledge and skills to build Internet of Things (IoT) systems Skills: programmation des Microcontrôleur ESP32 et raspberry pi avec implémentation sur cloud Competencies: C2, C3, C9
Content	Chapter I – Introduction to the Internet of Things (3h) 1. Technologies that led to the evolution of IoT 2. IOT and M2M 3. IoT and cloud platforms 4. ThingSpeak API and MQTT Chapter II - Sensor & Actuators with ESP32 (4h) 1.Overview of Sensors working 2.Analog and Digital Sensors 3. Interfacing of Temperature, Humidity, Motion, Light and Gas Sensor with ESP32 =>Lab Works (3h) Chapter III Networking with ESP32 WiFi module and Cloud Platforms for IOT (8h) 1. ESP8266 Wi-Fi Module and Various Wi-Fi library 2. Web server, installation, configuration 3. Posting sensor(s) data to web server 4. Thing Speak API and MQTT 5. Interfacing ESP8266 with Web services =>Lab Works (6h)
Examination forms	100% Lab exam
Study and examination requirements	10/20

Reading list	Pfister, C. Getting Started with the Internet of Things. Ramasamy, L. K., & Kadry, S. Internet of Things (IoT).

IAP.5.3 Positioning Systems and Applications -PSA.

Module designation	IAP.5.3 Positioning Systems and Applications- PSA
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Faïza NAJJAR
Language	English
Relation to curriculum	optional
Teaching methods	Lessons, mini-projects and research presentations
Workload (incl. contact hours, self-study hours)	Total workload: 25H Contact hours: 15H (10H lessons, 5H presentations) Private study:10H
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	Mobile communication and networks Advanced Networks
Module objectives/intended learning outcomes	Objectives: This course is a special interest of mobile and sensor computing; it will introduce fundamentals and theoretical foundations of location-based services (LBS), including the most common services such as wireless localization, tracking and navigation on outdoor (e.g. GNSS) and a special emphasis on indoor LBS (e.g. WiFi-based). Students may undertake projects with real world data to put in practice the knowledge acquired in wireless positioning techniques and algorithms. Competencies: C2, C3, C4,C6.

Content	1. Location based services –LBS: Fundamentals: Location; Mobile Technologies, techniques (proximity; trilateration; fingerprinting); Reference systems; outdoor, indoor 2. Introduction to Satellite-based systems --GNSS • Examples of GNSS (GPS, Glonass, Beidou, …) • Principles of the Global Positioning System • Programming assignment - Mini-Project: • Applications of GPS on Android Smartphones • Hot GNSS-based services? 3. Indoor localization • Classification • MEMS and multi-sensor positioning • Indoor map construction (standards) • Pedestrian positioning systems: examples • Lab. RSS-fingerprinting localization 4. Indoor Pedestrian Navigation using Smartphone: from research to production • Architecture of navigation Systems (Cartography, Pathfinding, guiding) • Mini-Project: WiFi-based navigation (ENSI)
Grading (Total 20 points)	35% CW + 65% Final Exam with CW --Continuous work = Mean (classroom exercises; Mini-projects; presentation)
Study and examination requirements	10/20

References	5. Data Management for Mobile Computing, Evaggelia Pitoura and George Samaras , Kluwer Academic Publishers, 1998 ,ISBN: 0-7923-8053-3 6. Alkhawaja, Fares & Jaradat, Mohammad & Romdhane, Lotfi. (2019). Techniques of Indoor Positioning Systems (IPS): A Survey. 1-8. 10.1109/ICASET.2019.8714291. 7. K. -J. Li, S. Zlatanova, J. Torres-Sospedra, A. Perez-Navarro, C. Laoudias and A. Moreira, "Survey on Indoor Map Standards and Formats," 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2019, pp. 1-8, doi: 10.1109/IPIN.2019.8911796. 8. A. Billa, I. Shayea, A. Alhammadi, Q. Abdullah and M. Roslee, "An Overview of Indoor Localization Technologies: Toward IoT Navigation Services," 2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT), 2020, pp. 76-81, doi: 10.1109/ISTT50966.2020.9279369.

SEC.5.1 Diagnosis, Safety and Reliability of Embedded Systems

Module designation	SEC.5.1 Diagnosis, Safety and reliability of Embedded Systems
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Yasser GRITLI
Teaching team	Yasser GRITLI
Language	French
Relation to curriculum	Optional
Teaching methods	Lesson, Lab
Workload (incl. contact hours, self-study hours)	Total workload:50h Contact hours: 30h (14h Lessons, 10h Laboratory tests, 06h Exercise) Private study:in hours[3]: 20h
Credit points	2 ECTS
Required and recommended prerequisites for joining the module	MAT.1.1, MAT.1.2, EHA.1.1, EHA.1.2, NET.2.1, EHA.4.1.
Module objectives/intended learning outcomes	This course develop advanced skills for on-line diagnosis and reliability processing dedicated to failure prediction and detection at incipient stage for ES. More specifically: - to develop skill and knowledge in diagnosis and fault detection techniques dedicated to ES; - to introduce the principals of independent/dependent components analysis in ES; - to investigate the on-line processing of reliability prediction in ES. Competencies:C1, C4
Content	Lesson 1 (01h). Introduction – Diagnosis, Safety and Reliability for Modern Embedded Systems (DSR-ES) Lesson 2 (02h). Diagnosis – Model-based techniques - Exercices (02) Lesson 3 (03h). Diagnosis – Signal-based techniques - Laboratory simulation (03) Lesson 4 (02h). Systems Analysis with independent components - Exercices (02) Lesson 5 (02). Systems Analysis with dependent components - Exercices (02) Lesson 6 (02). Application of dependability methodologies to ES - Laboratory simulation (03) Lesson 7 (02). Reliability prediction methodology for ES - Laboratory simulation (04)

3

Examination forms	35% Continues Laboratory evaluation + 65% Written exam.
Study and examination requirements	10/20
Reading list	Gao, Z., Cecati, C., & Ding, S. X. (2015). A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. IEEE Transactions on Industrial Electronics, 62(6), 3757-3767. Liggesmeyer, P., & Maeckel, O. (2001). Quantifying the reliability of embedded systems by automated analysis. In 2001 International Conference on Dependable Systems and Networks. IEEE. Huang, C. Y., & Chang, Y. R. (2007). An improved decomposition scheme for assessing the reliability of embedded systems by using dynamic fault trees. Reliability Engineering & System Safety, 92(10), 1403-1412. Han, Z., Tian, L., & Cheng, L. (2021). A deducing-based reliability optimization for electrical equipment with constant failure rate components duration their mission profile. Reliability Engineering & System Safety, 212, 107575.

SEC.5.2 IoT Security

Module designation	SEC.5.2 IoT Security
Semester(s) in which the module is taught	S5
Person responsible for the module	Mohamed Houcine Hdhili
Teachers team	Rihab Boussaada
Language	French
Relation to curriculum	Compulsory
Teaching methods	Lesson, Lab works
Workload (incl. contact hours, self-study hours)	Total workload:50h Contact hours: 30h (18h lesson, 12h lab works) Private study: 20h
Credit points	2 ECTS
Required and recommended prerequisites for joining the module	SEC.4.1 Cybersecurity and Cryptography NET.4.1 Computer Networks
Module objectives/intended learning outcomes	Knowledge: At the end of the course, the students will : ● Be able to discuss the main threats and attacks on IoT devices and services. ● Identify convenient cryptographic tools and protocols to secure devices, communications and IoT plateforms ● Be able to assess IoT threats and risks as they arise ● Be able to design a secure embedded system. Competencies: C6, C13

Content	Chapter 1 – IoT Security context 1. IoT security requirements 2. IoT security challenges 3. IoT potential risks in various use cases 4. IoT security risks assessments Chapter 2 - Securing IoT devices 5. IoT device hardware and firmware 6. Vulnerabilities and Attacks at the Hardware Layer 7. Threat mitigation of the Hardware Layer Chapter 3- Securing IoT network layer 8. IoT network layer vulnerabilities 9. Mitigating IoT network layer threats Chapter 4 - Securing IoT application layer 10. IoT application layer vulnerabilities 11. OWASP projets (Secure coding, OWASP Top 10) 12. Mitigating IoT application layer threats Chapter 5 – Securing IoT plateforms 13. Cloud security 14. Securing data at rest Chapter 6 – Security in embedded systems 15. security risks in embedded systems 16. Mitigating embedded systems threats 17. STM32 security features Lab Works: Lab1: Firmware security assessments using OWASP-fstm. Lab2: Testing commonly found vulnerabilities in IoT devices using the deliberately insecure firmware IoTGoat. Lab3: audit IoT hardware security using HardSploit Lab4: designing a secure embedded system, using the STM32Trust security framework.
Examination forms	50% continuous assessment (lab works) + 50% written exam
Study and examination requirements	10/20

Reading list	Hu, F. (2016). Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations. CRC Press. Seshadri, N., & Dhakshinamoorthy, S. (2020). Internet of things (IoT) and Security. International Journal of Engineering Research & Technology (IJERT), NCAIT – 2020, 8(15). IoTAA. (2016). IoTAA Security Guideline V1.2. Retrieved from https://www.iot.org.au/wp/wp-content/uploads/2016/12/IoTAA-Security-Guideline-V1.2.pdf OWASP. OWASP Internet of Things Project. Retrieved from https://owasp.org/www-project-internet-of-things/ STMicroelectronics. STM32Trust. Retrieved from https://www.st.com/content/st_com/en/ecosystems/stm32trust.html

SEC.5.3 Automotive Architecture and Security

Module designation	SEC.5.3 Automotive Architecture and Security
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Lobna KRIAA
Teaching team	Industrial partner team (https://group.mercedes-benz.com/en/) ● Rekik Youssef ● Rostom El Abed
Language	English
Relation to curriculum	Optional
Teaching methods	Lesson.
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact hours: 15h (12h lessons, 3h exercises) Private study:10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	EHA.2.1: Architecture & microprocessors EHA.4.1: Introduction to embedded Systems EHA.5.2: Microcontroller-based system

Module objectives/intended learning outcomes	- Knowledge:
	  - Familiarity with automotive architectures and automotive protocols
	  - Importance of the security issues in the automotive domain
	- Skills: cognitive and practical abilities for which knowledge is used
	- Competencies: C5, C6, C7, C8, C13

Content	Chapter 1: Automotive Overview Section I: Vehicle structure Section II: Electronic Control Unit (ECU) Section III: Network in Vehicle Chapter 2: AUTOSAR overview Section I: AUTOSAR overview Section II: Definition and history Section III: AUTOSAR vs OSI model Section IV: AUTOSAR Application Section V: AUTOSAR RTE Section VI: AUTOSAR BSW Chapter 3: Security in Automotive Section I: Automotive SPICE Section II: Functional Safety ISO 2626 Section III: Automotive Cybersecurity ● Challenges ● Terminology ● STRIDE ● Risk Determination ● Security Concepts and Engineering ● Hardware Security Modules ● Secure Boot ● Secure on-Board Communication

	• Security Testing (e.g. Fuzztesting) Chapter 4: CAN bus fundamentals (ISO 11898) Section I: CAN vs OSI model Section II: Addressing and physical layer Section III: Bus access Section VI: CAN framing and Data protection Section V: CAN FD Section VI: CAN tools and CAN Database
Examination forms	100% written exam
Study and examination requirements	10/20
Reading list	Subke, P. (2019). Diagnostic Communication with Road-vehicles and Non-road Mobile Machinery. SAE International. Etschberger, K., Hofmann, R., Stolberg, J., Schlegel, C., & Weiher, S. (2001). Controller area network: Basics, protocols, chips and applications. IXXAT Automation. Fuzztesting Whitepaper, Fending Off Cyber Attacks – Hardening ECUs by Fuzz Testing. (2018). Translation of a German publication in Hanser automotive, special edition “Safety & Security” August 2018. Retrieved from https://cdn.vector.com/cms/content/know-how/_technical-articles/Security_FuzzTesting_HanserAutomotive_201808_PressArticle_EN.pdf

SEC.5.4 IoT security project

Module designation	SEC.5.4 IoT Security Project
Semester(s) in which the module is taught	S5
Person responsible for the module	Mohamed Houcine Hdhili
Teachers team	
Language	French
Relation to curriculum	compulsory
Teaching methods	Project
Workload (incl. contact hours, self-study hours)	Total workload:25h Contact hours: 15h (lab works) Private study: 10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	SEC.4.1 Cybersecurity and Cryptography NET.5. 2 Communication Architectures and protocols for IoT SEC.5.1 IoT security
Module objectives/intended learning outcomes	Knowledge: At the end of the course, the students will : ● Be able to identify security requirements for a specific IoT application ● Be able to design and implement a security solution for a specific IoT application ● Identify convenient cryptographic tools and protocols to secure IoT applications Skills: Work with a team, manage time Competencies: C6, C13
Content	Project information: - This is a project-based module. Students apply - The instructor defines projects dealing with securing IoT applications (securing devices, embedded systems, network, and application layers) - Students will be divided into groups. - Each group chooses a project to develop (requirements, design, implementation, and tests) - Each group defines the project outline and divides the tasks among the team members. Project samples: - securing an e-health application - securing a smart-home application - securing a smart-hotel application
Examination forms	100% project assessment
Study and examination requirements	10/20

Reading list	Naoui, S., Elhdhili, M. E., & Azouz Saidane, L. (2020). Novel Enhanced LoRaWAN Framework for Smart Home Remote Control Security. Wireless Personal Communications, 110(4), 2109–2130. Boussada, R., Balkis, H., Elhdhili, M. E., & Saidane, L. A. (2019). Privacy-preserving aware data transmission for IoT-based e-health. Computer Networks, 162. Naoui, S., Elhdhili, M. E., & Saidane, L. A. (2019). Lightweight and Secure Password-Based Smart Home Authentication Protocol: LSP-SHAP. Journal of Network and Systems Management, 27(4), 1020–1042. IoTAA Security Guideline V1.2. (2016). Retrieved from https://www.iot.org.au/wp/wp-content/uploads/2016/12/IoTAA-Security-Guideline-V1.2.pdf https://owasp.org/www-project-internet-of-things/

ISA.5.12 Introduction to Data Mining

Module designation	ISA.5.12 Introduction to Data Mining
Semester(s) in which the module is taught	S5
Person responsible for the module (coordinator)	Rym Besrour
Language	French
Relation to curriculum	optional
Teaching methods	lesson, project.
Workload (incl. contact hours, self-study hours)	Total workload: 25h Contact hours: 15h lessons Private study: 10h
Credit points	1 ECTS
Required and recommended prerequisites for joining the module	MAT.1.1: Probability and Statistics AI .3.1.: IA & Machine Learning Students must be competent in python.
Module objectives/intended learning outcomes	Knowledge: ● To introduce students to the basic concepts and techniques of Data Mining. ● To develop skills of using recent data mining software for solving practical problems. ● To gain experience of doing independent study and research. Competencies: C1, C9
Content	Chapter1: Introduction to Data Mining ● What is data mining? ● Related technologies - Machine Learning, OLAP, Statistics ● Data Mining Goals ● Stages of the Data Mining Process ● Data Mining Techniques ● Knowledge Representation Methods ● Applications Chapter2: Data preprocessing ● Data cleaning ● Data transformation ● Data reduction ● Discretization and generating concept hierarchies Chapter3: Association rules ● Motivation and terminology ● Example: mining weather data ● Basic idea: item sets ● Generating item sets and rules efficiently ● Correlation analysis Chapter4 : Clustering

	• Basic issues in clustering • First conceptual clustering system: Cluster/2 • Partitioning methods: k-means, expectation maximization (EM) • Hierarchical methods: distance-based agglomerative and divisible clustering Projects ideas : • Text mining: extracting attributes (keywords), structural approaches (parsing, soft parsing). • Bayesian approach to classifying text • Web mining: classifying web pages, extracting knowledge from the web
Examination forms	100% Lab exam
Study and examination requirements	10/20
Reading list	Han, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques Morgan Kaufmann; 3rd edition.