

Systèmes embarqués <i>Embedded systems</i>			
Code cours <i>Course code: SEM</i>		Crédits ECTS <i>ECTS Credits: 3</i>	
Département <i>Department</i>	: IA	Cours <i>Lectures</i>	: 11h15
Coordonneurs <i>Lecturers</i>	: E. Grolleau, Y. Ouhammou, H. Bauer, M. Richard	T.D. <i>Tutorials</i>	: 11h15
Période <i>Year of study</i>	: 2 ^e année <i>2nd year</i>	T.P. <i>Laboratory sessions</i>	: 12h00
Semestre <i>Semester</i>	: 4 ^e semestre <i>4th semester</i>	Projet <i>Project</i>	:
Evaluation <i>Assessment method(s)</i>	: 1 écrit, 1 contrôle TP <i>1 written exam, 1 practical work test</i>	Non encadré <i>Homework</i>	:
Langue d'instruction <i>Language of instruction</i>	: Français <i>French</i>	Horaire global <i>Total hours</i>	: 34h30
Type de cours <i>Type of course</i>	: Obligatoire <i>Compulsory</i>		
Niveau <i>Level of course</i>	: Graduate		

Compétences attendues : Suivre un processus de l'ingénierie Système de la spécification à la conception produit. Suivre un cycle de développement logiciel permettant le développement sûr de programmes embarqués temps réel pour des systèmes critiques. Introduire les spécificités des logiciels temps réel.

Pré-requis : Cours d'informatique, cours Systèmes Embarqués 1. Bases d'architecture et de système d'exploitation : algorithmique & programmation, notion de tâches et processus, problèmes de base du parallélisme (producteur/consommateur, exclusion mutuelle) et sémaphore, fonctionnement d'un calculateur.

Contenu :

1. Introduction aux systèmes embarqués critiques

- Contraintes, exigences et certification,
- Redondance et tolérance aux pannes,
- Cycle de vie logiciel.

2. Introduction des concepts et de la méthode à travers un exemple

- Programmation embarquée séquentielle (exemple de programmation Arduino)
- Entrées-sorties analogiques, numériques, PWM, horloges, bus, représentation numérique
- Capteurs-actionneurs type sur drones
- Programmation événementielle basée interruptions matérielles
- Programmation cyclique
- Programmation multitâche
- Introduction à FreeRTOS
- Spécification fonctionnelle (utilisation de Capella)
- Modélisation par statecharts (UML FSM sur Yakindu <https://www.itemis.com/en/yakindu/state-machine/>)
- Conception logiciel-matériel utilisant le multitâche (utilisation d'AADL)
- Délais de bout-en-bout et performances

Bibliographie :

E. Grolleau, J. Hugues, Y. Ouhammou, H. Bauer, « Introduction aux systèmes embarqués temps réel », ed. Dunod, 2018
F. Cottet, E. Grolleau, S. Gérard, J. Hugues, Y. Ouhammou, « Spécification, conception, implémentation et validation temporelle », ed. Dunod, 2014
F. Cottet, E. Grolleau, « Systèmes temps réel de contrôle-commande », ed. Dunod, 2005
A. Tanenbaum, « Systèmes d'exploitation », ed. Pearson
P. Zanella, Y. Ligier, E. Lazard, « Architecture et technologie des ordinateurs »
P. Ward, S. Mellor, « Structured development for real-time systems », Yourdon press
H. Gomaa, « Software design methods for concurrent and real-time systems », Addison Wesley

Expected competencies: Follow a System Engineering design cycle. Use a software life-cycle to insure a safe, and fault-tolerant of critical real-time embedded systems. Introduce real-time specificities and constraints.

Prerequisites: Computer science programming, Introduction to Embedded Systems. Basics of computer architecture and operating systems: threads and processes, parallelism problems (producer/consumer, mutual exclusion) and semaphore, basic computer programming.

Content:

1. Introduction to critical and embedded systems

- Constraints, requirements and certification,
- Redundancy and fault-tolerance,
- Software life-cycle.

3. Introduction des concepts et de la méthode à travers un exemple

- Sequential embedded programming (based on a simple Arduino example)
- Analog Input/Output, Digital I/O, PWM, clock circuits, buses, digital representation
- Typical drone sensors and actuators
- Event-based programming using interrupts
- Cyclic programming model
- Multithread programming
- Introduction to FreeRTOS
- Functional specification (using Capella)
- Statecharts modelling (UML FSM using Yakindu <https://www.itemis.com/en/yakindu/state-machine/>)
- Hardware-software co-design for multithreading (using AADL)
- Performance and end-to-end delays

Recommended reading:

E. Grolleau, J. Hugues, Y. Ouhammou, H. Bauer, « Introduction aux systèmes embarqués temps réel », ed. Dunod, 2018

F. Cottet, E. Grolleau, S. Gérard, J. Hugues, Y. Ouhammou, « Spécification, conception, implémentation et validation temporelle », ed. Dunod, 2014

F. Cottet, E. Grolleau, « *Systèmes temps réel de contrôle-commande* », ed. Dunod, 2005

A. Tanenbaum, « *Systèmes d'exploitation* », ed. Pearson

P. Zanella, Y. Ligier, E. Lazard, « *Architecture et technologie des ordinateurs* »

P. Ward, S. Mellor, « *Strutured development for real-time systems* », Yourdon press

H. Gomaa, « *Software design methods for concurrent and real-time systems* », Addison Wesley