

# TD d'architecture matérielle : introduction aux architectures des microcontrôleurs

L. Werling<sup>a,b,\*</sup>, E. Dervieux<sup>a,b,c</sup>, M. Madec<sup>a,b</sup>, W. Uhring<sup>a,b</sup>

<sup>a</sup>Télécom Physique Strasbourg, Université de Strasbourg, France;

<sup>b</sup>Laboratoire ICube, équipe SMH, Université de Strasbourg/CNRS, France;

<sup>c</sup>Biosency, Cesson-Sévigné, France

\*Courriel : [lucas.werling@unistra.fr](mailto:lucas.werling@unistra.fr)

Cet article décrit un TD d'introduction aux architectures des microcontrôleurs réalisé avec les étudiants de Télécom Physique Strasbourg en première année du cycle de formation ingénieur (BAC + 3) en spécialisation « Informatique et Réseaux ». Il vise à introduire les concepts fondamentaux de l'architecture des microcontrôleurs sur un exemple concret, en l'occurrence la réalisation d'un microcontrôleur élémentaire à l'aide du logiciel LOGISIM. Les travaux dirigés se déroulent sur 4 séances au cours desquelles les étudiants mettent en œuvre un microcontrôleur 6 bits qu'ils devront ensuite programmer en assembleur.

## 1. CONTEXTE

Télécom Physique Strasbourg propose parmi son offre de formation quatre diplômes dont un de spécialité « Informatique et Réseaux ». Dans ce diplôme, les étudiants disposent d'un large spectre d'enseignements allant de l'électronique numérique à l'intelligence artificielle et la cybersécurité en passant par l'informatique, le développement logiciel et réseau, les mathématiques et le traitement du signal. Les étudiants de cette formation suivent un tronc commun de deux semestres, avant de se diriger vers l'une des deux spécialités qui leur sont proposées par le département Informatique et Réseaux de l'école (1) : Réseaux et Internet des Objets, ou Sciences des Données et Intelligence Artificielle. Le premier semestre du tronc commun de première année de cette formation vise à donner aux étudiants un socle de connaissances élémentaires dans les trois domaines d'expertise technique que sont le matériel, le logiciel et le réseau.

L'aspect matériel est abordé de manière progressive avec, en introduction, un module d'*Electronique Numérique*, commun à l'ensemble des filières de l'école. Ce module permet aux étudiants de découvrir dans un premier temps de manière théorique les différents aspects de la logique booléenne, ainsi que son utilisation dans la conception de circuit combinatoires et séquentiels synchrones. Ils ont ensuite la possibilité dans un second temps, de mettre en œuvre ces connaissances lors de travaux pratiques, à l'aide de circuits intégrés numériques élémentaires. Ce premier module matériel est mené en parallèle d'un module d'*Algorithmique et programmation C* où une emphase est mise sur le lien entre logiciel et matériel, avec notamment le stockage des variables en mémoire. Un deuxième module intitulé *Architecture matérielle* propose au travers de cours magistraux comme de travaux pratiques de créer le lien entre le module d'*Electronique Numérique* et l'architecture d'un microcontrôleur. Ce dernier présente les éléments principaux d'un processeur, à savoir : l'Unité Arithmétique et Logique, la mémoire vive et le séquenceur, avant de détailler leur fonctionnement conjoint, notamment en introduisant la notion de chemin de données. Quatre séances de travaux dirigés (TD) servent alors à illustrer directement cette partie théorique sur un système pédagogique minimaliste, avant de passer lors d'une séance de travaux pratiques sur un système plus largement employé avec la programmation d'une carte Arduino directement en langage assembleur. Ces aspects matériels sont ensuite brièvement

revus en deuxième année dans le cours de *Microcontrôleurs* et sont notamment étoffés avec le mécanisme d'interruption, passé sous silence en première année. Ce cours s'accompagne d'une série de travaux pratiques utilisant à nouveau la plateforme Arduino, mais programmée cette fois-ci en langage C. Le volet matériel reste cependant un sujet d'attention important lors des enseignements portant sur les systèmes embarqués et l'internet des objets présent dans la spécialité Réseaux et Internet des Objets que les étudiants peuvent choisir à partir de la deuxième année.

La présente publication se concentre sur les quatre séances (7 heures) de TD du module d'*Architecture Matérielle* de première année, qui propose aux étudiants la réalisation partielle d'un microcontrôleur 6 bits minimaliste à l'aide de l'outil Logisim. Le choix de ce logiciel a été motivé principalement parce qu'il s'agit d'un logiciel multiplateforme libre (licence GPL) et léger, dédié à une utilisation pédagogique.

Nous présentons d'abord le contenu des cours magistraux aboutissant audit TD, avant de nous pencher plus avant sur le contenu de ce dernier. Enfin, nous détaillons l'utilisation de Logisim pour mettre en avant ses différents atouts dans un contexte pédagogique.

## **2. CONTENU DES COURS MAGISTRAUX**

La partie théorique de ce module d'Architecture matérielle est composée de 10h30 de cours magistral scindant le cours en quatre parties :

1. La première partie fait le lien avec le cours d'*Electronique Numérique*, en rappelant un certain nombre de fonctions logiques élémentaires (multiplexeurs, bascules et mémoires, séquenceurs) ainsi que de circuits arithmétiques simples (demi-additionneur, additionneur complet, additionneur 2-bits).
2. La deuxième partie plonge au cœur de l'électronique intégré jusqu'à l'élément de base qu'est le transistor pour présenter son fonctionnement et comment il peut être utilisé pour former des portes logiques en technologies CMOS. Les technologies des mémoires sont également abordées dans ce cours.
3. La troisième partie présente l'historique du développement des architectures informatiques et des processeurs, en expliquant à chaque bond technologique les problématiques et besoins qui y ont mené. Cela permet d'amener en douceur une première version simplifiée du chemin de données, puis de le complexifier au fur et à mesure en évoquant les notions de pipeline, de cache, de système multicœurs et de jeu d'instruction SIMD. Le but est de faire comprendre aux étudiants que les systèmes informatiques les plus complexes peuvent émerger de la sophistication progressive d'un postulat de base relativement simple, qu'ils vont étudier en détail en dernière partie du cours.
4. Enfin, la quatrième partie décrit le fonctionnement d'un processeur 16 bits d'architecture d'instruction RAM-Accumulateur avec un séquenceur microcodé.

Cette partie théorique fait ensuite place à l'expérimentation, avec quatre séances de travaux dirigés (programmation en assembleur d'un microcontrôleur simplifié 6 bits, décrit ci-après) qui feront le lien avec la séance de travaux pratiques (programmation en assembleur d'un Arduino 8 bits).

## **3. CAHIER DES CHARGES DU MICROCONTROLEUR REALISE**

Au cours des quatre séances du TD, les étudiants sont amenés à compléter le circuit numérique d'un microcontrôleur 6 bits au jeu d'instructions minimaliste, dont le schéma complet est donné en Figure 1.

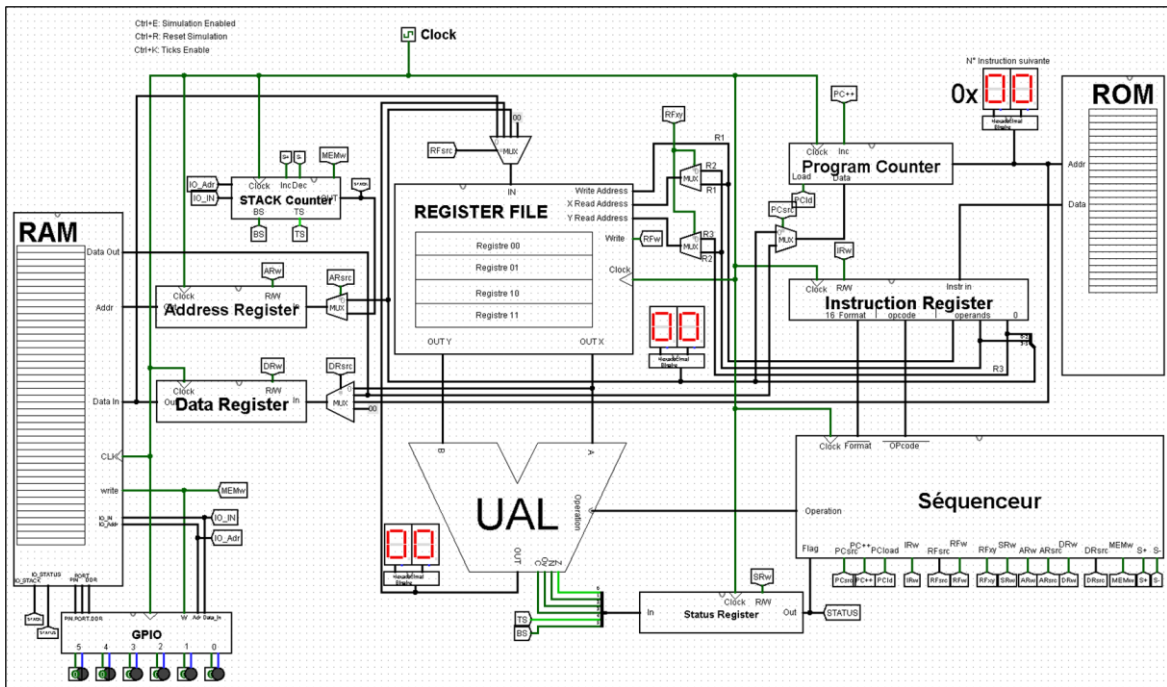


Fig.1. Schéma complet du microcontrôleur 6 bits. Il est constitué des éléments principaux d'un processeur: l'Unité Arithmétique et Logique, le séquenceur, et le banc de registres. Pour compléter ce microcontrôleur on trouve également des mémoires RAM et ROM pour le stockage des données et du programme, ainsi qu'un module d'entrée/sortie Global Purpose Input Output (GPIO). Gif on <https://osf.io/q67cw>.

L'ensemble de ce circuit sert à implémenter une architecture d'instruction registre-à-registre grâce à son chemin de données et son banc de registres de travail. Ce dernier fournit trois entrées d'adressage de registre sur 3 bits chacune : une pour le registre à écrire et deux autres, indépendantes, pour les registres à lire. Le séquenceur microcodé orchestre le bon déroulement de l'exécution des instructions de la Figure 2, en générant les 17 signaux de contrôles nécessaires à la manipulation des divers registres, multiplexeurs, mémoires et compteurs.

Description du jeu d'instructions (Instruction Set Architecture)									
Format	Description	Action	Instruction	Flag	Format bits	OpCode	R1	R2	R3
3R	Addition	$R1 \leftarrow R2 + R3$	ADD R1, R2, R3	Z,C	000	0 0000	rrr	rrr	rrr
	Soustraction	$R1 \leftarrow R2 - R3$	SUB R1, R2, R3	Z,N,O	000	0 0001	rrr	rrr	rrr
	ET logique	$R1 \leftarrow R2 \&\& R3$	AND R1, R2, R3	Z,N	000	0 0010	rrr	rrr	rrr
	OU logique	$R1 \leftarrow R2 \ \  R3$	OR R1, R2, R3	Z,N	000	0 0011	rrr	rrr	rrr
2R	Comparaison	$SR[Z] \leftarrow R1 == R2$	CMP R1, R2	Z,N,O	001	0 0100	rrr	rrr	000
	NON logique	$R1 \leftarrow \sim R2$	NOT R1, R2	Z,N	001	0 0101	rrr	rrr	000
	Décalage à gauche	$R1 \leftarrow R2 \ll 1$	SHL R1, R2	Z,N	001	0 0110	rrr	rrr	000
	2Décalage à droite	$R1 \leftarrow R2 \gg 1$	SHR R1, R2	Z,N	001	0 0111	rrr	rrr	000
RI	Transfert RAM vers registre	$R1 \leftarrow RAM[k]$	LDR R1, k	-	010	0 1000	rrr	kkk	kkk
	Transfert registre vers RAM	$RAM[k] \leftarrow R1$	STR R1, k	-	010	0 1001	rrr	kkk	kkk
	Ecriture dans un registre	$R1 \leftarrow k$	LDI R1,k	-	010	0 1010	rrr	kkk	kkk
1R	Saut inconditionnel	$PC \leftarrow k$	JMP k	-	011	0 1100	000	kkk	kkk
	Saut conditionnel si résultat nul	$PC \leftarrow PC+k \text{ ? } SR[Z]$	JMPZ k	-	011	0 1101	000	kkk	kkk
	Appel de fonction	$RAM[SP] \leftarrow PC+1, PC \leftarrow k, SP-1$	CALL k	-	011	0 1110	000	kkk	kkk
OR	Retour de fonction	$SP+1, PC \leftarrow RAM[SP]$	RET	-	100	1 0000	000	000	000
	No OPeration	-	NOP	-	100	1 0001	000	000	000

Fig.2. Jeu d'instructions du microcontrôleur 6 bits. Cinq formats d'instructions sont supportés pour : réaliser des instructions arithmétiques et logiques, interagir avec la mémoire, se déplacer en mémoire programme (sauts). Les flags ont la signification suivante : Z = zero, N = null, C = carry, O = overflow. rrr = bits d'adresse d'un registre de travail, kkkkkk = bits d'une valeur littérale.

Les instructions assembleurs se rapportant à des opérations arithmétiques ou logiques sont gérées par l'Unité Arithmétique et Logique (UAL), capable de réaliser deux opérations arithmétiques (addition et soustraction), trois opérations logiques (NON, ET, OU), une opération de comparaison et deux opérations de décalage. Deux instructions assembleur permettent les lectures/écritures en mémoire RAM des registres de travail afin d'assurer le stockage des variables, tandis qu'une instruction permet l'écriture d'une valeur directement dans un registre de travail. Enfin, des instructions de saut – conditionnel ou non – complètent le jeu d'instructions afin de garantir un panel d'actions suffisamment large pour réaliser de petits programmes et présenter le mécanisme de pile.

Un unique module périphérique d'entrée/sortie est également présent dans ce microcontrôleur pour permettre une interactivité du programme embarqué avec l'utilisateur (boutons poussoirs et LEDs), via l'accès à des registres de lecture et écriture accessible depuis la RAM.

#### 4. DEROULEMENT DU TD

Étant donné l'inexpérience des étudiants à ce stade de la formation, le TD se déroule de manière guidée et progressive pour pouvoir mener à bien plusieurs objectifs :

- Prendre en main le logiciel,
- Concevoir une Unité Arithmétique et Logique,
- Concevoir un banc de registres,
- Réaliser le chemin de données correspondant,
- Programmer en assembleur le microcontrôleur,
- Faciliter la compréhension du lien matériel/logiciel au travers d'un outil clair et visuel.

Le sujet de TD est écrit de façon à remplir ces différents objectifs tout en permettant l'indépendance de chaque séance vis-à-vis de la précédente, et ceci dans un but double : maintenir l'intérêt de l'étudiant tout au long du projet d'une part, et éviter les frustrations et découragements pouvant survenir en cas d'une trop forte dépendance entre les séances d'autre part. Le contenu des TD est susceptible de s'alléger ou de s'étoffer suivant les difficultés ou facilités que rencontreront les étudiants.

Le sujet de TD disponible en annexe (A) de cet article propose l'organisation suivante pour le déroulement des séances :

##### 1) Prise en main du logiciel avec la conception de l'Unité Arithmétique et Logique

Le premier travail demandé aux étudiants pour prendre en main le logiciel est la réalisation d'une simple porte OU-exclusive à 3 entrées. Cela leur permet de se familiariser à l'utilisation des différentes bibliothèques du logiciel, à la bonne gestion des circuits dans l'arborescence du projet, ainsi qu'à la hiérarchisation des circuits en sous-circuits. On instanciera ces derniers pour tester le circuit lors d'une simulation ultérieure, par exemple. L'exercice a aussi pour mission dans un second temps de présenter l'outil d'analyse combinatoire, qui est capable de générer un circuit logique fonctionnel à partir de sa description. Cette dernière pourra être réalisée au choix à l'aide d'une table de vérité, d'une équation logique ou d'un tableau de Karnaugh.

Le reste de la première séance de TD est dédié à la conception et au test de l'UAL et de ses sous-parties. Ainsi, les étudiants sont entre autres amenés à concevoir un additionneur 6 bits à propagation de retenue, des circuits de décalage binaire à gauche et à droite, ou encore la génération des flags Z=zero, N=null, C=carry, O=overflow.

## 2) Conception des registres et du banc de registres

La deuxième séance de TD est dédiée à la conception des différents éléments mémoire du microcontrôleur en commençant par les registres les plus simples, c'est-à-dire les registres de donnée, d'adresse, d'instruction, et le registre d'état. S'ensuit la conception des registres compteurs, du pointeur de pile et du compteur ordinal, puis vient la conception du banc de registres. Cette dernière pose plusieurs défis aux étudiants, puisqu'elle a pour singularité de présenter 3 entrées d'adresses : deux dédiées à la sélection des registres servant d'opérandes à l'unité arithmétique et logique, et une pour la sélection du registre dans lequel sauvegarder le résultat de l'instruction. Enfin, la séance se conclut en exposant l'architecture retenue pour la RAM, afin de mettre l'accent sur le mécanisme de *mapping* des registres dans cette dernière.

## 3) Conception du chemin de données

La troisième séance est l'occasion de faire le lien entre les différents blocs logiques conçus lors des deux séances précédentes et le jeu d'instructions du microcontrôleur. Ce lien se concrétise sous la forme du chemin de données et des signaux de contrôle nécessaire à son implémentation. Les étudiants seront alors amenés à construire ledit chemin de données instruction après instruction. La conception du séquenceur ne sera pas abordée, en raison du temps disponible lors des séances et de sa complexité. Toutefois, son fonctionnement sera expliqué aux étudiants qui devront charger le microprogramme dans le séquenceur et vérifier son bon fonctionnement pas à pas, à l'aide d'un programme de test.

## 4) Programmation du microcontrôleur

La dernière séance de TD est quant à elle centrée sur l'écriture de petits programmes en assembleur afin d'aborder plusieurs éléments du processus de compilation : la gestion des directives préprocesseur, la compilation d'un code C en langage assembleur, et l'assemblage du code. Cela permettra de s'intéresser à l'utilisation des variables en mémoire, et d'évoquer la notion de portée au sein du code. Enfin, nous aborderons le risque de débordement de pile (Stack Overflow) au travers d'un code d'exemple.

Tout au long de ces quatre séances, les étudiants auront l'occasion de découvrir progressivement l'utilisation du logiciel Logisim, dont les spécificités sont détaillées ci-après.

# 5. DESCRIPTION DU LOGICIEL LOGISIM

Le logiciel Logisim (2) utilisé lors des séances de travaux dirigés dispose d'une interface minimaliste (Figure 3) avec une barre d'outils composée principalement de trois outils : un outil d'interaction utile lors de la simulation – pour interagir avec les boutons-poussoirs, par exemple – un outil de sélection pour sélectionner déplacer et tracer des fils, et un outil d'annotation textuelle.

La partie de gauche de l'écran représente l'arborescence du projet en cours et les bibliothèques intégrées au logiciel. Ces dernières incluent des composants d'interconnexion, des composants numériques combinatoires ou séquentiels et des composants d'entrées/sorties interactifs. Les détails et paramètres du composant sélectionné apparaissent en bas de la section gauche de l'écran. La fenêtre principale, quant à elle, permet l'affichage au choix de l'éditeur de schéma ou de l'éditeur de symbole. Les symboles ainsi créés permettent l'encapsulation de fonctions de plus en plus complexes, et la hiérarchisation du circuit sur plusieurs niveaux de complexité.

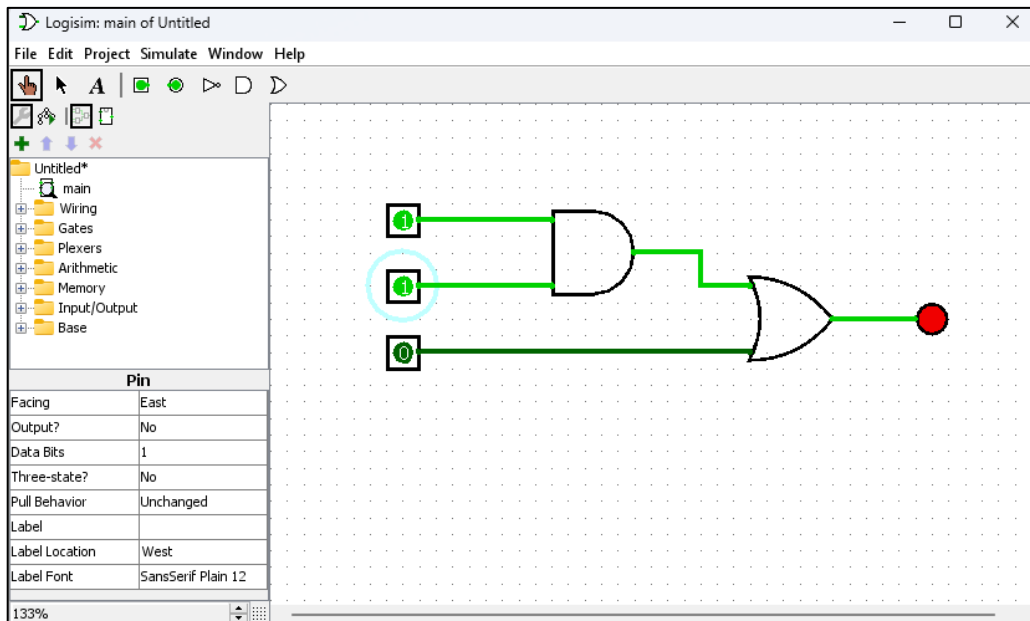


Fig.3. Interface du logiciel Logisim pour la réalisation d'un circuit combinatoire simple.

Le logiciel est adapté à la simulation de circuits de logique combinatoire ou séquentielle, qu'ils soient synchrones ou asynchrones. En effet, son algorithme de simulation évalue la sortie des portes logiques constituant le circuit. Pour ce faire, il prend en compte les équations logiques décrivant le fonctionnement desdites portes ainsi que les temps de propagation qui y sont associés. L'algorithme utilise d'ailleurs le délai de propagation d'une porte logique élémentaire comme pas de simulation. Le résultat de la simulation est visualisable au fur et à mesure de son exécution directement sur le schéma, à l'aide du code couleur retenu pour les fils d'interconnexion. Un fil au niveau logique 0 aura ainsi une couleur vert foncé, alors qu'elle sera vert clair pour un niveau logique 1. Enfin, pour s'aider dans le débogage des circuits complexes, il est possible d'enregistrer les valeurs des entrées et sorties du circuit à chaque pas de simulation dans un tableau, afin de vérifier chaque combinaison de sa table de vérité.

Le logiciel dispose de capacités d'évolutions grâce à l'ajout de bibliothèques utilisateurs, ce qui en fait un outil potentiellement polyvalent et évolutif capable de s'adapter à un large éventail de besoins pédagogiques.

## 6. CONCLUSION

Ces travaux dirigés, qui sont une nouveauté pour la période universitaire 2023-2024, ont été conçus pour illustrer progressivement le lien entre matériel et logiciel au sein d'un microcontrôleur. Ils permettent de mettre en lumière certaines notions parfois difficile à appréhender pour les étudiants, comme par exemple les notions de pile – et son dépassement éventuel – et de registre de travail. Pour ce faire, les deux premières séances traitent d'éléments matériels assez commun avant de les regrouper dès la troisième séance dans une architecture adaptée au jeu d'instructions à implémenter. La dernière séance assure quant à elle le lien avec la programmation tout en restant très proche du matériel.

L'utilisation du logiciel Logisim est particulièrement intéressante d'un point de vue pédagogique pour accompagner les étudiants dans leur compréhension du fonctionnement des circuits numériques. Sa simplicité garantit une prise en main facile et rapide tout en forçant à adopter une démarche rigoureuse et structurée. Son caractère libre et évolutif est par ailleurs un atout pour les étudiants curieux d'aller plus loin ou pour de futurs développements pédagogiques.

Au terme de ce module, une enquête sollicitera les étudiants pour récolter leur avis sur divers aspects des cours, des TD, et du TP afin d'ajuster au mieux leur contenu à l'avenir.

### **Remerciements**

Les auteurs remercient l'ensemble des collègues ayant contribué à la mise en place de ce TP au sein du département Informatique et Réseaux de l'Ecole. Le pôle MIGREST du GIP-CNFM a bénéficié du programme de soutien à l'innovation pédagogique, FINMINA (3), au cours des dernières années et du démarrage en 2023 du projet INFORISM (4), tous les deux gérés par l'ANR.

### **Références**

1. Télécom Physique Strasbourg–École d'ingénieur de l'Université de Strasbourg. Site web: <https://www.telecom-physique.fr/>
2. Site web du logiciel LOGISIM : <http://www.cburch.com/logisim/>
3. O. Bonnaud, FINMINA: A French National Project Dedicated to Educational Innovation in Microelectronics to Meet the Challenges of a Digital Society Smart Education and e-learning 2020, in V. Uskov, R. Howlett, L. Jain, (Eds.. Smart Innovation Systems and Technologies:188), Springer Nature Singapore Pte Ltd. 2020, p.31 (2020).
4. INFORISM : INgénierie de FORMations Innovantes et Stratégiques en Microélectronique, projet ANR-23-CMAS-0024-INFORISM au titre du programme France 2030. Ce projet à 5 ans démarre au cours de l'année académique 2023-2024.

### **Annexe**

- A. Lien du projet Open Science Framework (OSF) contenant l'ensemble des fichiers et présentation du TD : [https://osf.io/4zwa6/?view\\_only=529b3ffdbe6b426ab8c7338edf70d0be](https://osf.io/4zwa6/?view_only=529b3ffdbe6b426ab8c7338edf70d0be)