

Enseignement par la pratique de la conception conjointe Hard/Soft de systèmes numériques embarqués sur cible FPGA

F. Dadouche^a, W. Uhring^b,

^a Faculté de Physique et Ingénierie, Université de Strasbourg, Strasbourg, France

^b IUT de Haguenau, Université de Strasbourg, Strasbourg, France

Contacts email : dadouche@unistra.fr , wilfried.uhring@unistra.fr

Ce travail a pour but de partager une expérience d'enseignement dédiée à la conception de systèmes numériques embarqués sur cibles reconfigurables de type FPGA en première année de master au sein de l'université de Strasbourg. Cette dernière introduit une méthode d'apprentissage des concepts de la conception conjointe matérielle/logicielle (Hard/Soft) par la pratique. Dans cette méthode, la proportion des enseignements théoriques en cours magistraux est réduite au strict minimum pour laisser place aux travaux pratiques durant lesquels les étudiants sont confrontés à une situation réelle qu'ils doivent réaliser de manière individuelle dans le cadre d'un mini-projet. Ce dernier définit un seuil minimum de réalisation pour valider le projet, pour aller plus loin, des pistes d'amélioration sont suggérées par l'enseignant et au-delà les étudiants peuvent proposer leurs propres améliorations. Les retours, que nous avons eus, montrent que les étudiants sont très satisfaits de ce dispositif et souvent souhaitent la généralisation de la méthode à d'autres enseignements adaptés à un apprentissage par la pratique.

I. Introduction

Le master PAIP de la Faculté de Physique et Ingénierie de l'université de Strasbourg a pour objectif de former un flux de chercheurs ou d'ingénieurs de haut niveau dans quatre domaines spécifiques des sciences pour l'ingénieur (Systèmes microélectroniques (SM), Mécatronique et énergie (ME), Mécanique numérique en ingénierie (MNI) et Modélisation numérique avancée (MNA) ayant un spectre de connaissances spécialisées étendues allant de la physique aux applications et conceptions en ingénierie (1). Les deux premiers parcours de ce master partagent tout le socle d'enseignement autour des systèmes électroniques d'une manière générale (analogique, numérique, traitement du signal, électronique de puissance, automatique...) et tout particulièrement celui des systèmes embarqués. Dans ce cadre un pack d'enseignements en électronique numérique a été construit autour des compétences nécessaires à la validation du niveau master. Ce pack s'étale sur les trois dernières années du cursus : la troisième année de la licence SPI – Sciences Pour l'Ingénieur (parcours Systèmes Microélectronique ou Mécatronique) et les deux années des parcours SM et ME du master Physique Appliquée et Ingénierie Physique (PAIP).

Les prérequis d'accès direct au master pour les étudiants venant de l'extérieur correspondent aux compétences acquises en troisième année de licence SPI (parcours SE et Mécatronique) qui regroupent les concepts de base de l'électronique numérique combinatoire et séquentielle.

Dès la première année du master les étudiants sont amenés à apprendre la modélisation et la description des circuits et systèmes numériques en utilisant un langage de description de matériel en l'occurrence le langage VHDL. Ensuite les différents supports d'implémentation de fonctions numériques (PLD, CPLD, FPGA ...) sont étudiés dans une autre matière avant d'entamer une troisième unité d'enseignement consacrée à la conception de systèmes numériques embarqués sur cibles reconfigurables FPGA.

En deuxième année de master l'apprentissage de l'électronique numérique se termine par l'introduction des systèmes plus évolués intégrant un système d'exploitation.

Dans ce papier nous nous focaliserons sur l'enseignement de la première année du master consacré à la conception des systèmes numériques embarqués (SNE) contenant une partie matérielle et une partie logicielle et leur implémentation sur une cible FPGA. Après une introduction rapide des concepts clés de cet enseignement qui sont les systèmes numériques embarqués et la conception conjointe Hard/Soft dans la première section, la méthode d'apprentissage adoptée sera présentée dans la deuxième section. Les sections 3 et 4 détailleront le travail pratique réalisé en TD et en mini-projet. Enfin nous concluons sur la pertinence de cette méthode et les retours encourageants qui nous sont parvenus.

II. Systèmes numériques embarqués et Codesign Hard (HW) / Soft (SW)

Un système embarqué peut être défini de manière simple comme étant un système électronique autonome, dédié à une tâche bien précise, contenant un processeur dont l'utilisateur n'a souvent pas connaissance de son existence. Le rôle principal du processeur est d'apporter plus de flexibilité aux systèmes électroniques l'intégrant (2). De ce fait, de tels systèmes intègrent inéluctablement une partie matérielle représentée par l'ensemble des composants le constituant et d'une partie logicielle représentée par l'algorithme et/ou le code implémenté et exécuté par le processeur. La réalisation de tels systèmes est rendue possible grâce aux progrès réalisés dans les techniques d'intégration sur silicium permettent au jour d'aujourd'hui de concevoir des systèmes intégrant tous les éléments matériels et logiciels sur des puces électroniques avec des dimensions très réduites – systèmes sur puce ou System On Chip – SOC (processeurs, mémoires, timers, bus intégrés, contrôleurs et interfaces de communication avec des périphériques externes ...). D'où la nécessité de développer des méthodologies de conception à même d'exploiter au mieux les atouts des parties logicielles et matérielles. La conception conjointe matérielle/logicielle (communément appelé Codesign Hard/Soft) est ainsi introduite au début des années 90 comme nouvelle méthodologie de conception des SoCs pour développer parallèlement la partie matérielle et logicielle dans l'objectif d'optimiser le système afin de réduire le temps et le coût de conception (3)(4). Le codesign Hard/Soft couvre l'ensemble des opérations du flot de conception ; de la définition des spécifications à l'implémentation finale en passant par le partitionnement du système en partie HW et partie SW tout en gérant les interfaces entre ces deux parties (4)(5).

De plus, l'avènement des cibles d'implantation reconfigurables rend ces techniques flexibles, adaptables, évolutives et réutilisables. De ce fait, de nouveaux domaines d'application nécessitant des systèmes numériques entièrement reconfigurables et/ou reprogrammables sont devenus accessibles. En effet, depuis l'introduction des premiers dispositifs programmables (PLD) à programmation unique qui sont purement matériels, les technologies n'ont cessé d'évoluer et permettent aujourd'hui la réalisation de cibles FPGA contenant : (i) des ressources matérielles classiques pour l'implémentation de systèmes matériels, des blocs IP (Intellectual Property) pour l'implémentation de fonctions matérielles prêtes à l'emploi qui sont souvent paramétrables et (iii) de processeurs et DSP

HARD et /ou SOFT pour apporter une dimension logicielle synonyme de flexibilité, d'adaptabilité et de réutilisation. Ces caractéristiques ont fait des circuits FPGA une solution incontournable dans la conception des systèmes embarqués reprogrammables (soft) et/ou reconfigurables (hard) communément connus par les PSoC (Programmable System on Chip).

IntelFPGA anciennement Altera (un des pionniers du marché du FPGA) fournit une gamme large de cibles reconfigurables allant des cibles bas coût (familles Cyclone, MAX...), mais qui répondent à un large domaine d'applications courantes, aux cibles de très haute performances (famille Stratix) au prix d'un coût de revient assez élevé (6).

Xilinx premier producteur mondial de solutions FPGA et leader sur plusieurs nœuds technologiques (process nodes) propose une gamme comparable à celle de IntelFPGA dont Spartan représente la version bas coût et la version Vertex représente la solution haut de gamme avec de très hautes performances (7).

III. Principe de la méthode d'enseignement par la pratique du codesign des SNE

Comme introduit précédemment, dans le cadre du pack d'enseignement de l'électronique numérique nous avons créé en première année des parcours SME et ME du master PAIP une unité d'enseignement appelé Systèmes Numériques Embarqués (SNE). Cette dernière a pour objectif d'initier les étudiants aux concepts indissociables des systèmes numériques reprogrammables et/ou reconfigurables et la conception conjointe matérielle/logicielle ou le codesign Hard/Soft.

La matière SNE créée en 2013 était initialement composée de 14 heures de cours et de 16 heures de TP. Dans le but d'intégrer l'enseignement par la pratique, nous avons modifié la structure du cours pour en faire uniquement 6 heures de cours magistraux et le nombre d'heures de pratique a été augmenté à 24 heures. Ce qui permet, à la fois, d'initier les étudiants à la conception conjointe Hard/Soft à travers un TD intégré sur machine, et de traiter un mini-projet complet à travers lequel les étudiants approfondissent en pratique les techniques de conception de tels systèmes. L'enseignement est décliné sur trois volets dont deux réalisés par la pratique.

Les cours magistraux

Les CM ont pour objectif d'introduire les concepts nécessaires à la compréhension et au développement de systèmes embarqués ainsi que les outils professionnels de conception conjointe Hard/Soft utilisés en TD et en mini-projet.

Le TD sur machine

Le TD a pour rôle d'accompagner les étudiants pour faire leurs premiers pas dans le monde de la conception conjointe Hard/Soft. Le public visé dont les formations d'origine sont très variées et composées de profils et niveaux très hétérogènes (ME, SME et CMI – Coursus Master en Ingénierie), ce qui impose une démarche d'apprentissage à la fois didactique et évolutive. Cela passe par un socle commun que tous les étudiants doivent réaliser, mais avec un temps d'apprentissage dudit socle variable selon la compétence de l'étudiant. Pour cela, les étudiants ont pour mission dans un premier temps de réaliser un tutoriel retraçant l'ensemble des étapes du flot de conception d'un système embarqué avec une partie matérielle et une partie logicielle. La deuxième mission est de faire évoluer le tutoriel pour réaliser deux exercices qui leur permettront d'acquérir de nouvelles compétences.

Le temps de ce TD varie entre 4 et 8 heures selon le niveau et la compétence de chaque étudiant. Le restant du temps des 24 heures allouées à la pratique est consacré au mini-projet.

Le mini-projet

L'étudiant, dans une démarche de conception conjointe (codesign), doit concevoir et implémenter, sur une cible reconfigurable FPGA, un système réel contenant une partie matérielle (Hard) et une partie logicielle (Soft). Pour la partie matérielle, il doit sélectionner et configurer les composants matériels adéquats : le processeur, la mémoire et les contrôleurs des périphériques associés. La partie logicielle, quant à elle, consiste à développer un code en utilisant le langage C pour réaliser les fonctionnalités souhaitées.

Le travail réalisé en TD représente un point de départ du mini-projet qui définit un seuil minimum de réalisation pour valider le projet, puis pour aller plus loin, des pistes d'amélioration sont suggérées et au-delà les étudiants peuvent proposer leurs propres améliorations.

Pour réaliser la partie pratique de cet enseignement (TD + Mini-projet), nous avons opté pour l'utilisation d'une solution très bas coût de chez IntelFPGA. Il s'agit de la solution MAX10 intégrée dans le kit de développement à vocation éducative DE10-Lite par Terasic dont le schéma bloc est représenté par la figure suivante (8) :

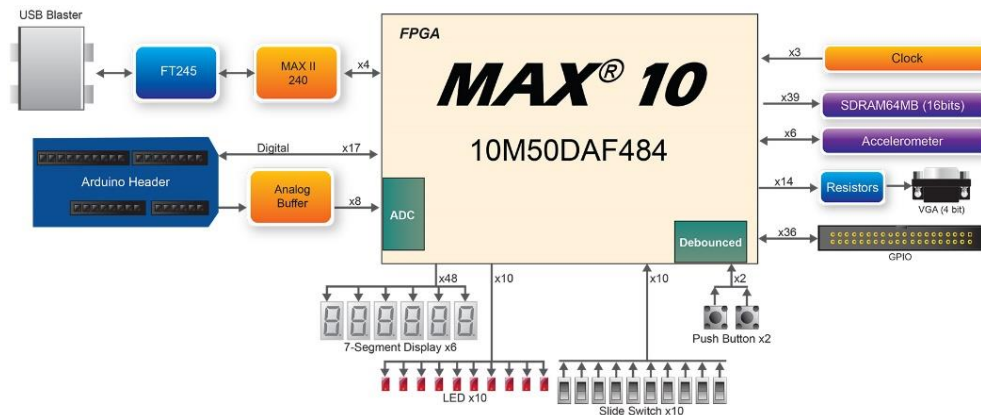


Fig. 1. Schéma bloc de la carte DE10-Lite.

La suite logicielle de conception associée fournie par IntelFPGA est Quartus Prime contenant de nombreux outils permettant la gestion des projets, la création des différents types de fichiers sources (VHDL, Verilog, BDF – Bloc Diagramm File...), la simulation, l'intégration des modules matériels créés à l'aide de l'outil Platform Designer, la configuration de la cible FPGA et le lancement de nombreux outils d'aide à la conception de systèmes numériques dont l'outil SBT for Eclipse pour le développement logiciel.

IV. TD sur Machine

Le TD vient compléter le cours des systèmes numériques embarqués pour initier les étudiants au Co-design Hard/Soft et l'implantation de systèmes numériques embarqués sur cible FPGA. Il leur permet d'assimiler le flot de conception de la suite logicielle Quartus Prime et de réaliser leur premier système embarqué sur kit de développement DE10 Lite intégrant un FPGA de la famille MAX 10.

Le logiciel de conception Quartus Prime de chez IntelFPGA intègre de nombreux outils d'aide à la conception, la configuration et le test des circuits implémentés, en particulier les deux outils qui sont utilisés dans cette UE qui sont :

- l'outil d'intégration de systèmes matériels (Platform Designer System Integration Tool – anciennement Qsys),
- et l'outil de construction logicielle pour éclipse (Nios II Software Build Tools for Eclipse – SBT).

Le flot de conception conjointe de la suite Quartus Prime est représenté sur la figure 2 ci-après (9).

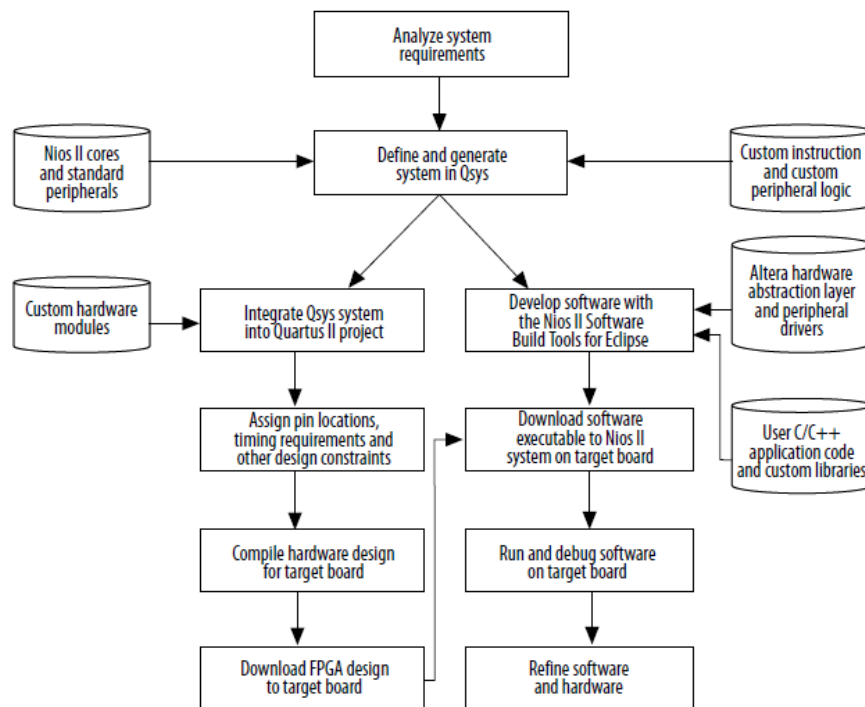


Fig. 2. Flot de conception Hard/Soft de la suite Quartus Prime d'IntelFPGA.

Le TD est structuré en trois parties :

Partie I

La première partie se résume en un tutoriel, appliquant étape par étape le flot de conception de la figure 2 permettant de réaliser un système basique de type microcontrôleur embarqué sur la carte de développement DE10-Lite.

Le système à réaliser a pour rôle de lire les positions des 10 interrupteurs (SW) et celles des 2 boutons poussoirs (Key) du kit DE10-Lite puis d'afficher et de faire défiler la position des interrupteurs (échantillon d'entrée) sur les afficheurs 7 segments et les LEDs avec un sens de rotation défini par la position des boutons poussoir.

Les éléments principaux de la partie matérielle sont : un microprocesseur NIOS II, une mémoire interne de 40 kB, une interface d'entrée/sortie ou PIO (Parallel Input Output) pour chaque périphérique d'entrée/sortie (SW, Key, LEDs et afficheurs 7 segments), une interface série JTAG pour configurer la carte par un PC via une liaison USB et un timer permettant de mesurer différents intervalles de temps.

La partie logicielle quant à elle consiste en un code en langage C fourni aux étudiants. Ce dernier effectue les opérations de lecture des données des périphériques d'entrées (SW

et Key) et les opérations de rotation puis affichage de l'échantillon binaire correspondant à la position des SW sur les périphériques de sortie (LEDs et 7 segments).

Partie II

Cette partie est une évolution du tutoriel pour apprendre l'utilisation de macros spécifiques fournies par IntelFPGA (Altera). Le travail demandé consiste à modifier le code du tutoriel précédent pour réaliser la même fonction, mais en utilisant les Macros d'Altera permettant de lire et d'écrire sur les PIOs à la place des pointeurs classiques. Ces dernières se trouvent dans la librairie "altera_avalon_pio_regs.h" à inclure dans le code.

Les macros les plus pertinentes à ce stade sont celles permettant la lecture et l'écriture des registres des périphériques d'entrée sortie :

- `IORD_ALTERA_AVALON_PIO_DATA(base)`
- `IOWR_ALTERA_AVALON_PIO_DATA(base, data)`
- `IORD_ALTERA_AVALON_PIO_EDGE_CAP(base)`
- `IOWR_ALTERA_AVALON_PIO_EDGE_CAP(base, data)`

Partie III

La troisième et dernière partie a pour finalité de tirer profit de l'exercice précédent pour réaliser une unité arithmétique simple baptisée « ALU du pauvre ».

Cette dernière, selon la position des deux boutons poussoirs (Key), effectue l'une des opérations représentées sur le tableau I. Les deux opérandes sont deux vecteurs de 5 bits chacun (2 bancs de 5 SW) et on affiche les résultats sur quatre afficheurs 7 segments :

- a) le signe du résultat sur l'afficheur HEX 3,
- b) la valeur absolue du résultat sur les afficheurs HEX 2-0,
- c) le reste en binaire sur les LEDs.

TABLEAU I. Table de fonctionnement de l'ALU du pauvre.

Key1	Key0	Opération
0	0	Addition
0	1	Soustraction
1	1	Multiplication
1	0	Division

A l'issue du TD, l'ensemble des étudiants ont acquis le flot de conception conjointe Hard/Soft mais aussi ont réalisé environ 80% de la partie matérielle du mini-projet qu'ils doivent terminer dans le temps restant.

V. Mini-projet

Après la réalisation du TD, les étudiants ont déjà acquis, à leur propre rythme, l'ensemble des compétences nécessaires et les techniques leur permettant de réaliser un projet de manière autonome. Ils sont alors confrontés à une situation réelle dont l'objectif est d'appliquer le flot de conception conjointe acquis en TD pour réaliser un réveil électronique sur une cible FPGA.

Ce mini-projet est évolutif en fonction des compétences et de l'avancement de chaque étudiant. Il s'articule autour de trois niveaux de compétences :

- le premier niveau consiste à répondre à un cahier de charges (ou spécifications) de base qui représente un seuil minimum de réalisation pour valider le projet,

- le deuxième niveau permet aux étudiants d'aller plus loin en leur suggérant des pistes d'amélioration qui viennent compléter le projet de base,
- et le troisième niveau leur donne la possibilité de faire leurs propres propositions d'amélioration.

Niveau 1 : répondre à un cahier des charges du projet de base

Dans cette étape, l'étudiant doit réaliser un réveil électronique illustré par la figure 3 dont le fonctionnement est résumé comme suit (10) :

- en mode normal, l'heure actuelle est affichée sur 6 afficheurs 7 segments (2 pour les heures, 2 pour les minutes et 2 pour les secondes) en mode 24 heures (de 00:00:00 à 23:59:59),
- le réveil dispose d'une fonction alarme permettant de générer un signal sonore de fréquence égale à 500 Hz (un signal carré périodique) lorsque l'heure actuelle coïncide avec l'heure d'alarme (de réveil). Ce signal est généré via une *pin* du port GPIO (General Purpose Input/Output),
- pour activer ou désactiver l'alarme, un interrupteur SW0 est utilisé, ce dernier permet d'activer l'alarme quand il est en mode ON et de la désactiver s'il est en mode OFF,
- pour indiquer si l'alarme est activée ou pas, une LED témoin est utilisée. Cette dernière est allumée si SW0 est mis en position ON,
- pour régler l'heure actuelle et l'heure d'alarme, on utilise deux interrupteurs SW1 et SW2 et deux boutons poussoirs KEY1 et KEY2 :
 - La sélection de l'heure actuelle se fait par l'activation de l'interrupteur SW1 (position ON),
 - La sélection de l'heure d'alarme est réalisée par la mise en position ON du SW2.
- les 2 boutons poussoirs KEY0 et KEY1 permettent d'ajuster, respectivement, les minutes et les heures,
- à la mise sous tension, l'heure actuelle et l'heure d'alarme sont mises à zéro.

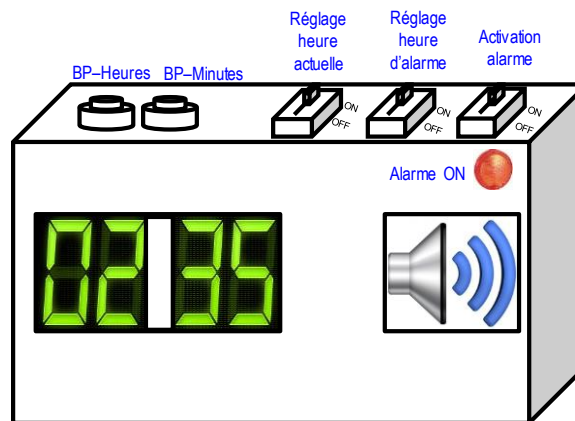


Fig. 3. Réveil électronique de base : seuil minimum du mini-projet.

Niveau 2 : Amélioration & évolution du cahier de charges

A ce stade la structure matérielle est complètement construite. Pour la partie logicielle, la lecture des données issues des périphériques d'entrée (SW et Key) ainsi que l'écriture sur les périphériques de sortie (LEDs, 7 segments et GPIO) sont déjà en place. Le travail

d'amélioration concernera essentiellement le code C dans le but d'intégrer de nouvelles fonctionnalités dont une liste non exhaustive est suggérée :

- choix du format d'affichage 24 ou 12 heures,
- réglage de l'heure par appuis successifs ou en continu (un seul appui) avec un pas de 0.5 seconde,
- utilisation de mélodies comme signaux d'alarme,
- sélection d'une mélodie parmi plusieurs,
- affichage avec réglage ou affichage seul.

Un exemple de configuration des SW pour la sélection des différents modes de fonctionnement est donné dans le tableau II.

TABLEAU II. Exemple de configuration des SW pour la sélection des différents modes de fonctionnement.

SW9	SW8	SW7	SW4	SW3	SW2	SW1	SW0	Opération
						0	0	Heure actuelle (format : 24 heures)
						1	1	Heure actuelle (format : 12 heures)
						0	1	Heure d'alarme (format : 24 heures)
						1	0	Heure d'alarme (format : 12 heures)
					1			Alarme activée, LED0 allumée
					0			Alarme désactivée, LED0 éteinte
				1				Mode affichage uniquement
				0				Mode affichage avec réglage
			1					Réglage de l'heure par appuis successifs
			0					Réglage continu avec un pas de 0.5 seconde
Mél.3	Mél.2	Mél.1						Sélection de la mélodie

Niveau 3 : Possibilité d'implémenter ses propres suggestions d'amélioration

Le troisième niveau est réservé aux étudiants qui sont en avance et désireux d'apporter leurs propres suggestions d'évolution du projet. L'idée que l'étudiant peut proposer des objectifs supplémentaires est en soi une motivation pour ces derniers. Même si le nombre d'étudiants capables d'apporter de nouvelles suggestions est généralement restreint, ces derniers se font réellement plaisir et vont chercher des solutions parfois inattendues. De plus ils représentent une source d'inspiration pour les autres ce qui permet souvent de tirer le niveau du groupe vers le haut.

A titre d'exemple voici quelques propositions faites par des étudiants :

- utilisation de l'accéléromètre embarqué dans le kit de développement pour arrêter la sonnerie par une simple secousse de la carte,
- utilisation du codec audio intégré à certains kits disponibles pour la génération d'une mélodie via les prises jacks,
- lecture d'un fichier mp3 en guise de signal d'alarme,
- utilisation de différentes techniques de codage pour améliorer la qualité des mélodies,
- ajout de nouvelles fonctionnalités de type minuterie, chronomètres ou affichage alternatif de la date du jour et de l'heure.

A travers ces différentes tentatives d'amélioration et l'envie de se distinguer, les étudiants font un réel travail de recherche de nouvelles propositions et trouvent dans ce défi une motivation supplémentaire à mettre en valeur leurs idées.

VI. Conclusion

A travers cette démarche de l'enseignement par la pratique nous nous sommes rendus compte de l'intérêt d'impliquer les étudiants dans leur apprentissage. En effet, l'introduction des concepts de manière progressive et de différents angles (CM, TD puis Mini-projet) fait que les étudiants n'ont pas le sentiment d'être gavés par du contenu théorique qu'ils risquent de ne pas pouvoir assimiler voire ne pas avoir la possibilité de l'utiliser dans leurs vies professionnelles. Le CM s'est limité à l'introduction des concepts nécessaires à la compréhension et au développement de systèmes embarqués ainsi que les outils de conception Hard/Soft utilisés en pratique. Le TD quant à lui complète le cours par l'introduction guidée et progressive des aspects. Enfin, le mini-projet est présenté de manière progressive et avec des exigences réparties par niveaux de compétences, ce qui permet à tout le monde de s'y retrouver.

Le fait de donner libre choix aux étudiants pour proposer des objectifs différents est ressenti à la fois comme un gage de confiance, mais aussi de responsabilité. Cela a montré que les étudiants se voient acteurs et de ce fait font preuve de beaucoup de sérieux et d'assiduité.

Cependant, il faut tout de même souligner que pour certains étudiants ayant une habitude de travail scolaire classique, la démarche de chercher par eux-mêmes des solutions aux problèmes rencontrés, voire de faire des propositions d'amélioration, présente une difficulté.

Enfin, globalement les retours que nous avons eus montrent que les étudiants sont très satisfaits de ce dispositif et souvent souhaitent la généralisation de la méthode à d'autres enseignements qui se prêtent bien à un apprentissage par la pratique.

Nous avons également appliqué cette technique d'apprentissage aux élèves ingénieurs en 2^{ème} année de la Formation d'Ingénieurs en Partenariat (Électronique et Informatique Industrielle) à l'école Télécom Physique Strasbourg qui ont montré aussi un intérêt particulier à cette méthode.

Remerciements

Nous remercions Denis Muller assistant d'enseignement, pour le support technique quotidien apporté aux étudiants, mais aussi à la sélection, l'achat, la préparation et la maintenance du matériel.

Nous remercions également le CNFM pour avoir mis en place le dispositif national « une carte - un étudiant » dans le cadre du programme IDEFI-FINMINA (11) qui nous a permis de fournir à chaque étudiant une carte individuelle pour travailler en toute autonomie.

Références

1. Faculté de Physique et Ingénierie, Université de Strasbourg, *Website* : <http://www.physique-ingenierie.unistra.fr/spip.php?rubrique21> (Accès : Octobre 2021)
2. J. O. Hamblen, Introduction to Embedded Systems using Windows Embedded CE, *Georgia Institute of Technology*, page 12, (2007).
3. Soonhoi Ha *et al.*, Handbook of Hardware/Software Codesign, *Soonhoi Ha Jürgen Teich Editors*, <https://doi.org/10.1007/978-94-017-7267-9>, page xi, (2017).
4. Patrick R. Schaumont, A Practical Introduction to Hardware/Software Codesign, *springer*, 2nd edition (2013). Chapter 1 - Introduction to Hardware/Software Codesign, pages 3-30.

5. Reiner Hartenstein, Reconfigurable Computing From FPGAs to Hardware Software Codesign, *João M. P. Cardoso. Michael Hübner (eds.) – Springer*, (2011), Chapter 2 - The Relevance of Reconfigurable Computing, pages 7-34.
6. Produits Intel / Intel FPGA, SOC FPGA and CPLD, Website : <https://www.intel.fr/content/www/fr/fr/products/details/fpga.html> (Accès : octobre 2021).
7. Xilinx / Devices / FPGAs & 3D ICs, <https://www.xilinx.com/products/silicon-devices/fpga.html> (Accès : octobre 2021).
8. Terasic DE10-Lite B <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1021> (Accès : octobre 2021).
9. Intel FPGA. Nios II Hardware Development Tutorial. AN-717 | 2014.09.22, <https://www.intel.com/content/www/us/en/programmable/documentation/sfo1429055121571.html> (Accès : octobre 2021).
10. Carl Hamacher *et al.*, Computer organization and embedded systems, *McGraw-Hill 6th edition*, (2012), ISBN-13: 978-0-07-338065-0, Chapter 11 - System-on-a-Chip – A Case Study, pages 428-430.
11. IDEFI-FINMINA : Initiative d'Excellence - Formation Innovante en Microélectronique et Nanotechnologies, ANR-11-IDFI-0017. Website: <http://www.cnfm.fr/VersionFrancaise/actualites/FINMINA.htm>