

Objet Connecté Personnel : ObCP

Joël Imbaud, David Vernier, Philippe Abbé, Fabrice Sthal
fsthal@ens2m.fr
ENSMM, 26 Rue de l'Épitaphe, 25030 Besançon cedex

RESUME : Cet article présente la réalisation d'un Objet Connecté Personnel (ObCP) dans le cadre d'une formation ingénieur consacrée à la conception et réalisation d'objets connectés. Les étudiants réalisent leur ObCP en début de formation, l'utilisent pour réaliser leur projet d'option et le conservent à l'issue de la formation. L'ObCP est une carte basée sur un microcontrôleur STM32F411RE et un connecteur de carte additionnelle compatible arduino.

Mots clés : PCB, conception électronique, STM32, programmation C embarquée, Bluetooth, IOT, objet connecté.

1 INTRODUCTION

L'École Nationale Supérieure de Mécanique et des Microtechniques (ENSMM) est une école d'ingénieurs qui recrute principalement à Bac + 3 sur les concours des grandes écoles. Elle forme des ingénieurs généralistes à dominante mécanique [1]. La dernière réforme pédagogique initiée à la rentrée 2018/2019 a permis la création d'une option de 3^{ème} année ingénieur dédiée à la « Conception et Réalisation d'Objets Connectés » (CROC) qui se déroule sur le premier semestre de l'année scolaire. Elle est basée sur quatre modules d'enseignements thématiques de 60 heures constituant ainsi l'unité d'enseignement scientifique (UES), une UE Sciences humaines de 120 heures et une UE projet de 90 heures. Les quatre modules de l'UES s'articulent autour de la conception, design et ergonomie (CROC1), les composants des objets connectés (CROC2), les contrôles et réseaux (CROC3) et enfin le traitement de données (CROC4).

Un fil rouge d'enseignement pratique a été mis en œuvre sous la forme d'un Objet Connecté Personnel (ObCP) qui doit être réalisé et mis en situation par l'étudiant.

Dans CROC1, l'étudiant voit les notions d'architecture générale d'un objet connecté et les méthodes de conception mécanique (sous catia) et électronique (sous Orcad). Dans CROC2, les capteurs, actionneurs et la gestion d'énergie sont abordés sous forme de cours classiques. Une partie technologie des circuits permet la réalisation et le test de l'ObCP en séance de travaux pratiques. Dans CROC3, la programmation microcontrôleur C/C++ est introduite ainsi que les problématiques des réseaux et systèmes d'exploitation embarqués sous forme de cours, TD et TP. Enfin dans CROC4, une initiation à la programmation d'application mobile sous Android Studio permet de faire communiquer l'ObCP avec une tablette ou le smartphone de l'étudiant.

Dans la suite de cet article, une description exhaustive de l'ObCP sera proposée. La mise en œuvre auprès des étudiants pendant ces deux dernières années permettra également de faire un retour étudiant et enseignant de cette activité pédagogique.

2 CHRONOLOGIE DE L'ObCP

Les différentes parties présentées ci-après sont réalisées chronologiquement dans l'ordre de présentation sauf contraintes fortes liées à l'emploi du temps. La partie réalisation de l'ObCP intervient en milieu de l'enseignement de programmation microcontrôleur alors que la conception de circuit imprimé intervient en dernier.

2.1 Conception électronique

Cette partie aborde le cahier des charges de l'ObCP au travers des fonctionnalités voulues sur la carte finale. Elle est exécutée sous la forme de TD interactifs, guidés par l'enseignant. Le travail sur le schéma et les simulations sont effectués sous la suite logicielle Orcad [2] qui servira aussi au design des PCB par la suite. Le schéma complet de l'ObCP et l'ensemble des documentations des composants sont fournis aux étudiants, il leur est demandé d'analyser les documentations de chaque fonction électronique pour déterminer la cartographie complète des alimentations, les valeurs des composants d'ajustement (exemple fig. 1) et la cartographie des ports d'entrées/sorties du microcontrôleur et de leurs fonctions associées (SPI, I2C, UART...).

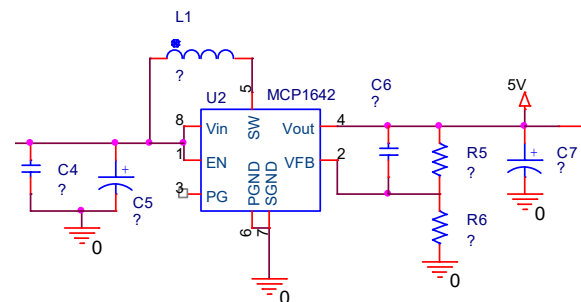


fig 1 : Exemple de schéma à trous où les étudiants doivent déterminer les valeurs manquantes.

En se basant sur l'analyse des alimentations (courants, tensions...), les étudiants doivent déterminer la consommation de l'ObCP en fonction de son utilisation (veille, bluetooth, sortie PWM...) et au moins évaluer la durée de fonctionnement de l'ensemble sur une batterie donnée (lithium, 400 mAh [3]). Ils doivent aussi dimensionner par simulation la partie driver LED trico-

lores et driver PWM. L'architecture de l'ObCP est décrite fig. 2.

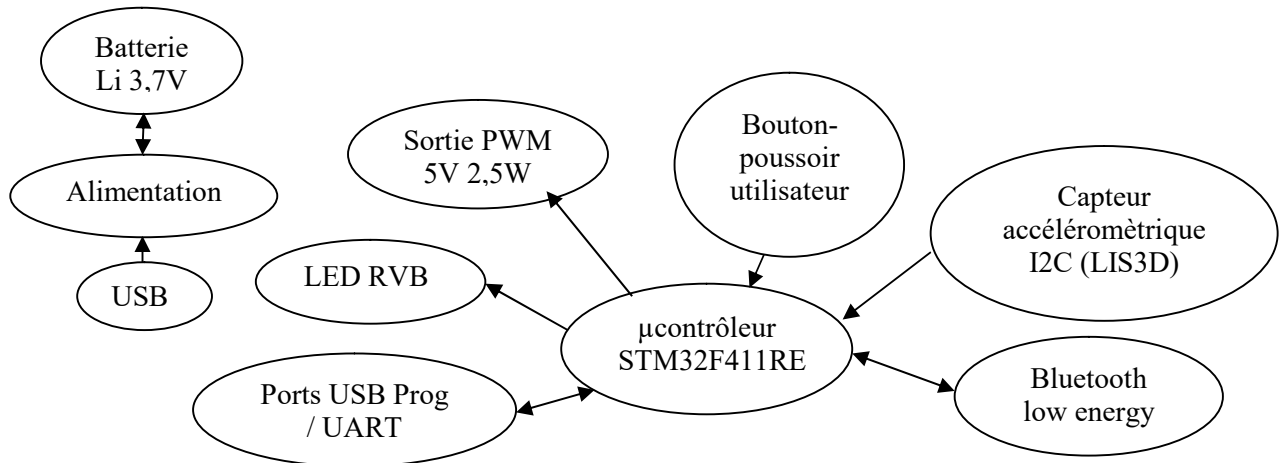


fig 2 : Architecture de l'ObCP

La conception de la partie microcontrôleur STM32F411 et de la partie bluetooth low energy (BLE) se base sur les schémas des cartes de développement ST de la Nucleo F411RE [4] et de la carte shield IDB05A1 [5].

2.2 Programmation microcontrôleur

Dans cette partie, les étudiants abordent la programmation C/C++ embarquée à l'aide de l'environnement de programmation en ligne MBED [6] fig. 3.

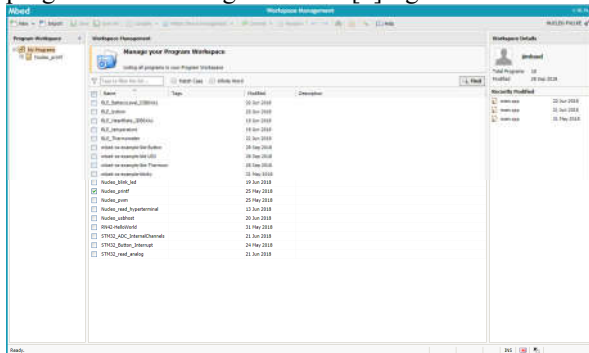


fig 3 : Environnement de programmation MBED

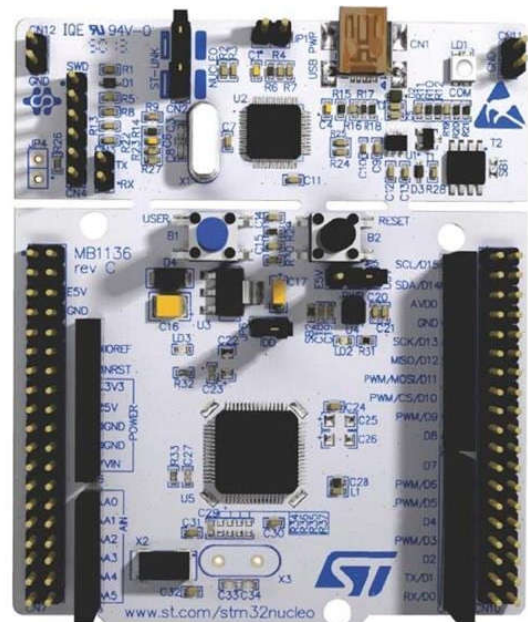


fig 4 : Carte Nucleo F411RE

Ils débutent avec la carte de développement Nucleo F411RE de chez ST, fig. 4, qui est la même base que l'ObCP en termes de connectivités-fonctions.

Après un cours d'introduction, l'enchaînement TP/TD se fait en autonomie avec l'aide ponctuelle de l'enseignant en suivant un texte d'expérimentations traitant le contrôle d'entrées/sorties classiques, la communication UART, les timers, les interruptions, l'horloge temps réel RTC, les sorties PWM, la communication avec un capteur I2C, la réalisation d'un PID sur une régulation thermique (capteur de température LM35 + résistance chauffante + transistor MOS sur sortie PWM voir fig. 5) et enfin la mise en œuvre du BLE avec le module IDB05A1 (fig. 6) identique à celui de l'ObCP.

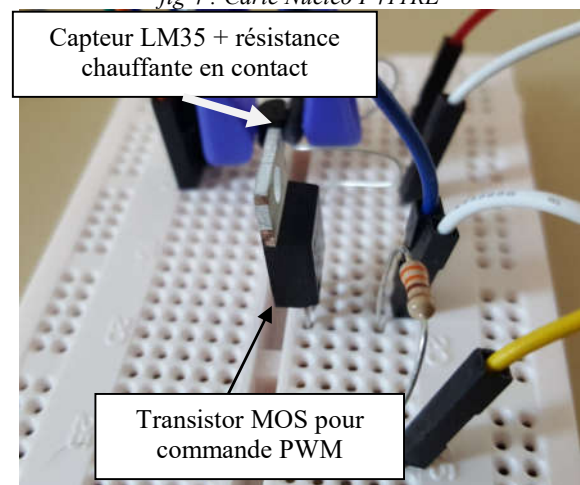


fig 5 : Mise en œuvre d'une régulation de température type PID.



fig 6 : Module Bluetooth Low Energy IDB05A1.

La fin de cet enseignement consiste en la mise en œuvre de l'ObCP qui a été réalisé en parallèle. Une procédure de programmation est mise en place pour son utilisation. En effet, la carte de développement comporte un programmeur/debugger ST du type ST-Link [7] qui n'est pas présent sur l'ObCP. L'opération consiste à programmer l'ObCP en utilisant le DFU USB (Device Firmware Update) [8] du microcontrôleur. Pour cela, il suffit de transformer le fichier de programmation BIN fourni par MBED en un fichier DFU en utilisant le logiciel gratuit DfuSe Demo fourni par ST (fig. 7).

Le prototypage de programmation est moins aisé qu'avec le ST-Link puisqu'il y a une étape supplémentaire pour la programmation, néanmoins l'étudiant repart avec une carte autonome qu'il peut utiliser chez lui sans programmeur dédié.

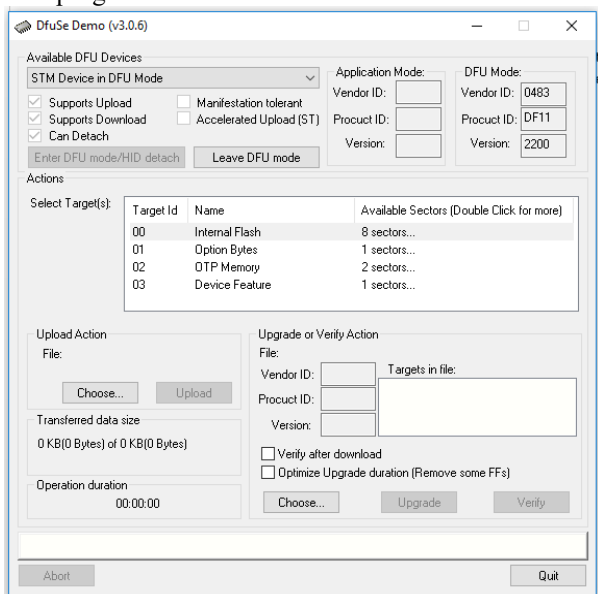


fig 7 : Mise à jour firmware par USB DfuSe Demo fourni par ST.

2.3 Technologie des circuits

Dans cette partie, les étudiants abordent les technologies industrielles de réalisation de circuits imprimés ainsi que les règles élémentaires de conception des cartes électroniques durant un cours de deux heures.

A ce cours s'ajoute un TP qui consiste au câblage de l'ObCP suivant un processus de brasure par refusion. La carte est fournie, elle est de type professionnel, avec sérigraphie et vernis épargne, elle est réalisée par un prestataire extérieur. L'objectif étudiant est de prendre en main l'équipement de prototypage électronique : dépôt de pâte à braser, placement de composants, four à refusion, station de brasure, station à air chaud et test électrique (fig. 8 à 12).



fig 8 : Station dépôt de pâte à braser pneumatique.

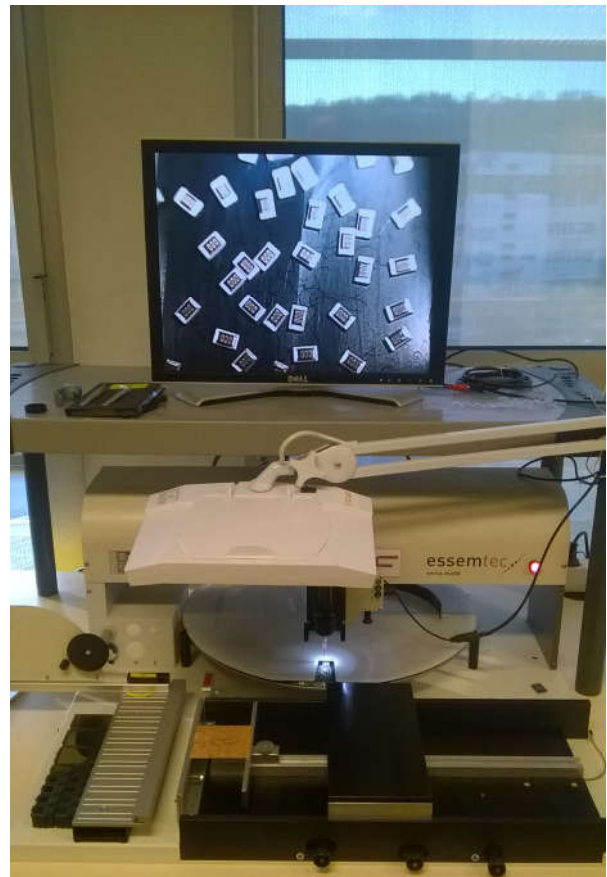


fig 9 : Station placement des composants CMS.



fig 10 : Station four à refusion.



fig 11 : Station brasure classique, air chaud.



fig 12 : Station tests électriques (continuité/court-circuit).

La réalisation de l'ObCP s'appuie sur la modélisation sur PCB Editor de l'ensemble de la carte (fig. 13).

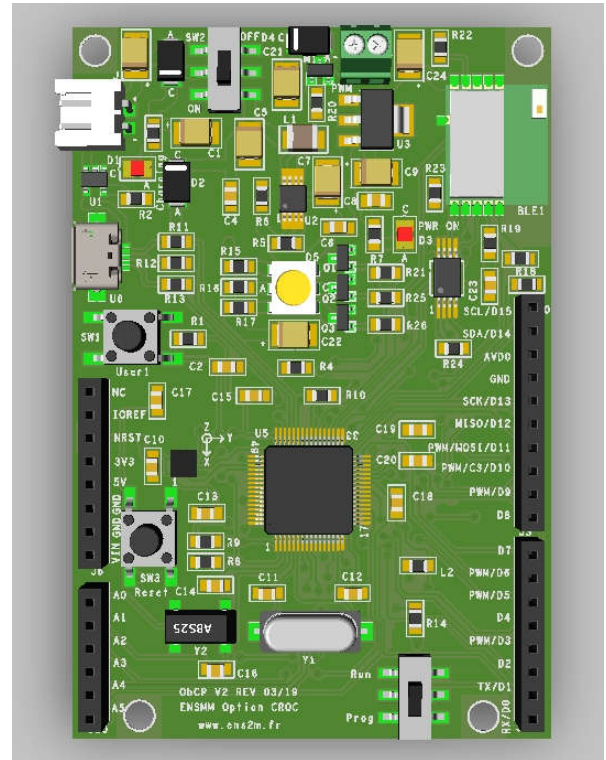
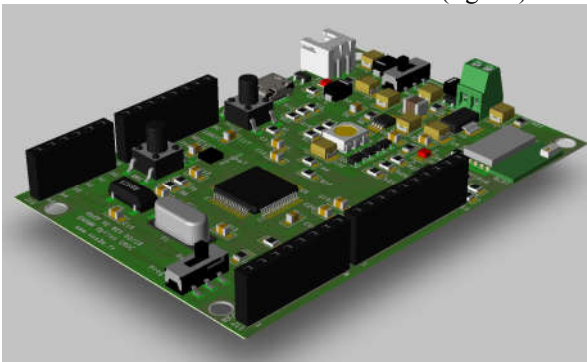


fig 13 : Modélisation 3D de la carte de l'ObCP.

La réalisation complète de l'ObCP est visible fig. 14.

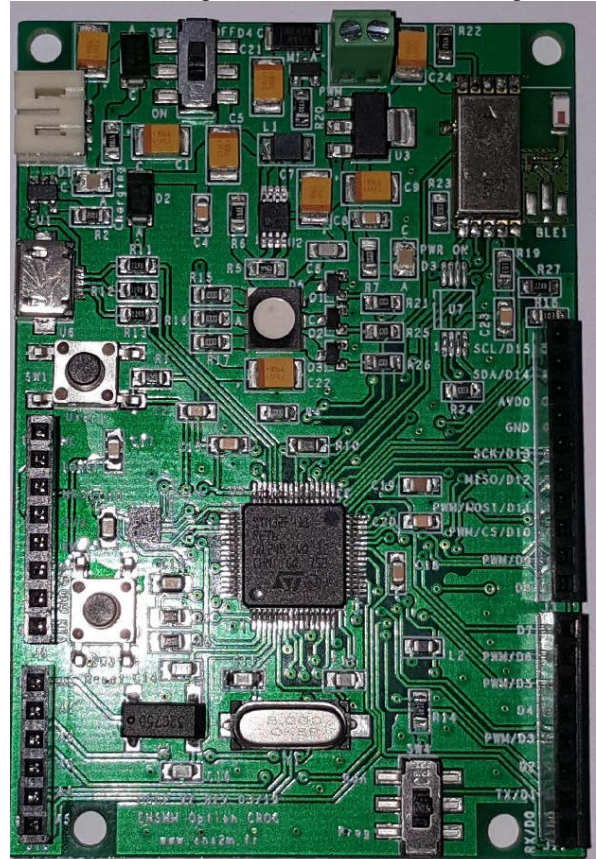


fig 14 : ObCP complet câblé.

Cette réalisation est ensuite testée avec deux programmes :

- un premier qui fait varier la luminosité de la LED tricolore en fonction des accélérations sur les axes x , y et z , permettant ainsi de tester la brasure correcte du capteur d'accélération. Ce dernier étant le plus petit composant qui est aussi le plus complexe à braser.

- Un second programme associé à une application Android développée sous MIT App Inventor [9], fig.15, permettant de tester la connectivité Bluetooth. Les sources sont données aux étudiants pour servir de base éventuelle pour les projets.

```

when BluetoothLE1 - Connected
do
  set LabelStatus - Text - to "Status: connected"
  set ListBLE - Visible - to false
  set PWMarr - Visible - to true
  set Acceleration - Visible - to true
  set REDarr - Visible - to true
  set GREENarr - Visible - to true
  set BLUEarr - Visible - to true
  set LogoENSMM - Visible - to false
  call BluetoothLE1 - RegisterForFloats
  serviceUuid get global SUUID2
  characteristicUuid get global AccX
  shortFloat false
  call BluetoothLE1 - RegisterForFloats
  serviceUuid get global SUUID2
  characteristicUuid get global AccY
  shortFloat false
  call BluetoothLE1 - RegisterForFloats
  serviceUuid get global SUUID2
  characteristicUuid get global AccZ
  shortFloat false
  
```

fig 15 : Programmation type scratch MIT App Inventor.

La suite de l'enseignement consiste au design d'une carte fille pour l'ObCP (shield) en lien avec le projet. Le format de la carte avec les connecteurs est fourni, l'étudiant doit placer les composants nécessaires à son projet. Cette carte est ensuite réalisée en prototypage rapide au sein de l'école à l'aide d'une graveuse anglaise Protomat S103 fig. 16 [10] et d'un process de via chimique Contact S4 fig. 17 [11]. Un exemple de prototype étudiant est visible basé sur le Template fourni par l'enseignant fig. 18.



fig 16 : LPKF - ProtoMat S103



fig 17 : LPKF - Contact S4.

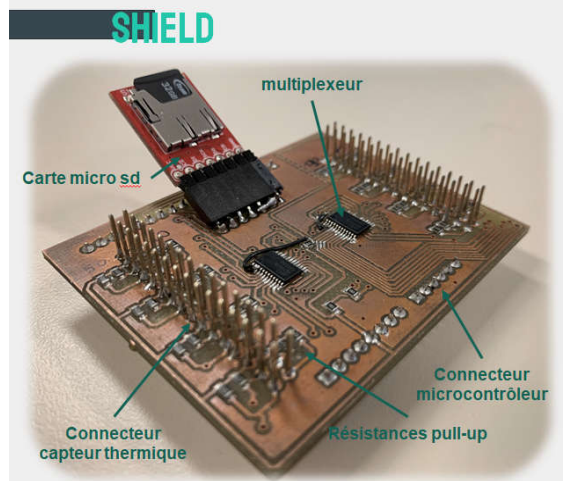
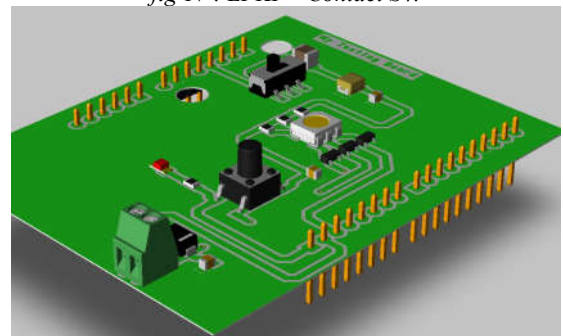


fig 18 : Exemple de réalisation de shield étudiant pour le projet basé sur un Template.

3 RETOUR ÉTUDIANT - ENSEIGNANTS

3.1 Étudiants

Le retour étudiant est très positif, ils ont vraiment apprécié le lien fort entre enseignements, réalisations et projet. Les étudiants ont totalement adhéré au principe de l'ObCP, ils ont finalement tous choisi cette plateforme pour leurs projets sans que celle-ci soit imposée. La conception électronique abordée sous une approche de type système leur a semblé pertinente et beaucoup plus proche de leurs préoccupations projets. Le cadencement de réalisation des cartes a été très apprécié ; pour les étudiants, la réalisation du câblage de leur ObCP en amont a suscité plus d'attention lors de la

phase dédiée au design des shields pour les projets. Ce cadencement leur a permis de bien appréhender la difficulté d'assemblage des composants et leurs faibles tailles. La réalisation d'une carte prototype qui débute par le dessin du schéma, qui se poursuit par le routage de la carte et qui se finalise par la réalisation du PCB est aussi très appréciée.

Les points d'améliorations soulevés par les étudiants seraient d'avoir un cours de C/C++ embarqué avant de commencer la programmation microcontrôleur. Actuellement, l'enseignement se fait sur les bases de programmation des années précédentes qui ne sont pas forcément du C/C++. Globalement, les étudiants apprécieraient une augmentation du volume horaire de ces enseignements au détriment d'autres enseignements moins appliqués.

3.2 Enseignants

Côté enseignant, la motivation des étudiants pour le projet est un réel point positif. Le programme de réalisation de l'ObCP étant ambitieux sur le nombre d'heures dédiées dans l'emploi du temps, il est nécessaire de déborder sur les heures dédiées au projet pour pouvoir finaliser l'ObCP. Ce qui nécessite obligatoirement une implication forte des enseignants et du personnel technique impliqués dans la réalisation (à titre indicatif, pour 8 étudiants, le câblage du premier groupe débute à 8H00, les tests du dernier groupe se terminent à 20H00 !).

Le coût de la réalisation est aussi à mettre en lumière, l'ObCP comporte pour approximativement 20€ HT de composants pour sa réalisation avec une carte commandée chez SEEED studio en Asie pour 30€ TTC comptant les 20 pièces, frais de port compris, le PCB revient donc à 1,5 € TTC/pièce. Soit un coût OBCP de l'ordre de 30€ TTC / étudiant en prenant en compte les éventuels ratés et les consommables. Il est à noter que la batterie n'est pas fournie aux étudiants pour des raisons de sécurité, elle est uniquement prêtée en cas de besoin pour les projets.

En ce qui concerne les perspectives, l'ajout d'une partie cours de remise à niveau sur la programmation C/C++ permettrait d'aborder plus sereinement la partie microcontrôleur et surtout d'aller plus loin dans les projets. L'introduction d'une partie sur la programmation du STM32 à l'aide de l'outil constructeur CUBEIDE de ST permettrait l'ouverture sur le développement avancé d'application embarquée tel que le réseau mesh BLE ou encore l'intelligence artificielle qui ne sont pas encore implantés dans l'outil en ligne MBED. En cas de coupure du réseau internet ou de défaillance de l'outil en ligne MBED, c'est une solution alternative de programmation locale. Une modification de la carte de l'ObCP visant à remplacer le capteur accélérométrique par une centrale inertielle, l'ajout d'un module WIFI et d'un connecteur pour debugger ST-LINK augmenteraient de façon significative la polyvalence de la réalisation et sa facilité de mise en œuvre.

4 CONCLUSION

Le retour sur les deux premières années de ces enseignements est très concluant, les personnels enseignants et personnels techniques impliqués dans la réalisation de l'ObCP et des projets de l'option ont suscité un réel intérêt de la part des étudiants. Par ailleurs, ces derniers ont manifesté un réel engagement dans les aspects pratiques de la conception électronique et plus particulièrement dans la réalisation de l'ObCP.

Les exemples d'utilisation de cet outil sont vivants et vont s'étoffer au fil du temps avec les nouvelles fonctionnalités apportées par les projets. Néanmoins, cet enseignement nécessite ou nécessitera de porter une vigilance particulière quant à l'évolution des environnements/platformes de programmation, ainsi que l'obsolescence des composants utilisés dans l'ObCP pour intégrer les évolutions technologiques futures.

L'ouverture de ce type de formation aux autres établissements, à l'image de ce qui est aménagé par exemple avec les TP de salle blanche AIME [12] sous l'égide du CNFM [13], pourrait également s'envisager.

Bibliographie

- [1] ENSMM : <https://www.ens2m.fr/>, consulté le 7 février 2020.
- [2] Suite Orcad : www.artedas.fr/OrCAD/OrCAD.php, consulté le 12 février 2020.
- [3] Batterie LiPo : www.gotronic.fr/art-accu-lipo-3-7-v-400-mah-pr502535-5812.htm, consulté le 12 février 2020.
- [4] Nucleo F411RE : www.st.com/en/evaluation-tools/nucleo-f411re.html, consulté le 12 février 2020.
- [5] Shield IDB05A1 : www.st.com/content/st_com/en/products/ecosystems/stm32-open-development-environment/stm32-nucleo-expansion-boards/stm32-ode-connect-hw/x-nucleo-idb05a1.html, consulté le 12 février 2020.
- [6] MBED : www.mbed.com, consulté le 12 février 2020.
- [7] ST-LINK : www.st.com/content/st_com/en/products/development-tools/hardware-development-tools/hardware-development-tools-for-stm32/st-link-v2.html, consulté le 12 février 2020.
- [8] DfuSe demo : www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/stsw-stm32080.html, consulté le 12 février 2020.
- [9] MIT APP INVENTOR : appinventor.mit.edu, consulté le 12 février 2020.
- [10] ProtoMat S103 : www.youtube.com/watch?v=YR9Cdg9ksNI, consulté le 12 février 2020.
- [11] Contact S4 : www.lpkf.com/en/industries-technologies/research-in-house-pcb-prototyping/produkte/lpkf-contact-s4, consulté le 12 février 2020.
- [12] P. Arguel & al, *Projet LUMELEC : mariage de l'optique et de l'électronique*, Proc. 9^{ème} CETSIS, Trois Rivières, Québec, Canada, 2011.
- [13] CNFM : <http://web-pcm.cnfm.fr/>, consulté le 7 février 2020.