Fablab et DIY : de nouvelles voies pour l'enseignement de l'électronique?

Fabien Mieyeville
fabien.mieyeville@ec-lyon.fr
Ecole Centrale Lyon,
INL UMR CNRS 5270
Programme I.D.E.A.
17, Chemin du Petit Bois
69130 ECULLY

David Navarro, Laurent Carrel
david.navarro@ec-lyon.fr
laurent.carrel@ec-lyon.fr
Ecole Centrale Lyon,
INL UMR CNRS 5270
36, Avenue Guy de Collongue
69130 ECULLY

Abstract—L'enseignement de l'électronique est devenu aujourd'hui, en université, un enseignement de spécialité et d'experts. Pourtant, avec l'avènement de l'ère de l'Internet des Objets, jamais l'électronique n'a été aussi pervasive et imbriquée dans la vie quotidienne de l'homme. En parallèle à cette tendance, de nouveaux mouvements tels que celui des makers et en particulier des fablabs rendent la technologie accessible à tout un chacun. Ce papier va présenter comment, inspire de cette approche des makers, l'enseignement d'électronique peut être prodigué et rendre en peu de temps les étudiants autonomes et désireux d'aller plus loin dans le domaine.

Keywords—enseignement électronique, Arduino, Fablab, project-based learning

1. INTRODUCTION

L'enseignement d'électronique en université est actuellement un enseignement théorique visant à enseigner l'électronique de la physique du semi-conducteur à la création de Systèmes sur Puces (ou SOC – System On Chip). Dans l'enseignement de l'électronique, trois grands secteurs peuvent être actuellement identifiés : l'enseignement lié à l'aspect de la physique du semiconducteur qui s'intéresse à l'aspect matériaux et physique du composant élémentaire qu'est l'électronique (domaine à la frontière de la physique que les anglo-saxons appellent "Electronics"), la conception de circuits intégrés et discrets (domaine regroupé sous l'appellation anglo-saxonne "Electrical Engineering") et enfin le domaine des systèmes embarqués (parfois appelé informatique industrielle), domaine à la limite des sciences de l'informatique. Cet enseignement aujourd'hui, qui selon les approches pédagogiques adoptées peut nécessiter un minimum de maîtrise des trois domaines avant de spécialiser dans un seul domaine est amené à être remis en question à une époque où par le biais de l'Internet des Objets, la technologie électronique n'a jamais été aussi proche de l'être humain. En parallèle, les progrès technologiques de miniaturisation rendent accessibles à faible coût des systèmes électroniques performants tels que Raspberry Pi ou Arduino pour ne citer que les plus connus. Ces dispositifs ont su créer une adhésion et des communautés d'utilisateurs s'entraînant de par le monde. Ainsi Internet regorge de Success Story d'individus avec peu de bagage académique en électronique ayant réalisé des systèmes avancés

et fonctionnels. Au vu de ces résultats, on peut se demander dans quelle mesure l'enseignement supérieur pourrait s'inspirer de ces pratiques pour repenser l'enseignement de l'électronique et en particulier celui des systèmes embarqués.

Cet article va présenter l'expérience menée à l'Ecole Centrale de Lyon autour cette approche. Après une présentation rapide de ce qu'est le mouvement des makers et des fablabs, nous introduirons la plateforme Arduino. Puis nous présenterons deux enseignements menés au sein de Centrale Lyon, le premier conduit auprès de deuxième année du cursus ingénieur et le second conduit auprès d'un public de post-licence au sein d'un programme de formation conjointement porté par Centrale Lyon et EMLyon Business School. Enfin, une synthèse et un bilan sera tiré des ces enseignements afin d'en proposer des conclusions et perspectives.

2. MOUVEMENT DES MAKERS ET DES FABLABS

2.1. Mouvement des Makers

La culture Maker est un mouvement contemporain qui s'inscrit dans la culture du Do-It-Yourself (DIY) dont il représente une extension fondée sur la technologie. Les champs d'intérêt typiques de cette culture s'inscrivent dans une vision ingénierie du DIY et sont l'électronique, la robotique, l'impression 3D et l'utilisation de machines numériques telles que les fraiseuses mais aussi des machines outils plus traditionnelles telles que celles permettant le travail du métal et du bois. Cette sous-culture met l'accent sur l'utilisation des nouvelles technologies et encourage le prototypage et les inventions. Il y a enfin un fort attachement à l'apprentissage et l'utilisation de compétences pratiques et leur mise en œuvre dans une démarche créative.

2.2. Fabrication Laboratory ou Fablab

La montée de ce mouvement des Makers est associé à l'augmentation de tiers-lieux permettant la mise en œuvre tels que les "hackerspaces", "maker spaces" ou autres Fablabs. Les fablabs sont des lieux regroupant un certain nombre d'outils de prototypage rapide, principalement centré autour de la production numérique, comme le montre la Figure 1. Ils sont une émanation du dispositif pédagogique élaboré par Neil Gerschenfeld (directeur du Center for Bits and Atoms au

Massachusetts Institute of Technology – MIT–) pour son cours “How to Make (Almost) Anything” en 2001. Ce cours est né du constat de Neil Gershenfeld selon lequel, si le MIT produisait des étudiants aux connaissances académiques, théoriques étendues, ces derniers étaient assez vite en difficulté au moment de passer à la réalisation pratique. Le mouvement des fablabs est maintenant un mouvement autonome qui s’est détaché du CBA et s’est structuré en réseau indépendant animé par une communauté de bénévoles.

Figure 1 : Fablab - Un lieu de production numérique

Au-delà de l’aspect purement prototypage que permettent naturellement de tels lieux, les fablabs sont avant tout des lieux d’échange de compétences et de partage (cf. Figure 2) et en tant que tel se prêtent particulièrement bien à une intégration dans des cursus pédagogiques existants ou à la mise en œuvre de nouveaux parcours pédagogiques. Dans cet article, les deux cas de figures vont se retrouver : d’une part l’intégration d’une approche fablab dans le cadre de cours optionnels du cursus de deuxième année de l’École Centrale Lyon et d’autre part l’utilisation du fablab comme cheville ouvrière d’un programme de formation de deux post-licence sur l’innovation et l’entrepreneuriat.

Figure 2 - Fablabs : un lieu de partage et d’échanges des compétences et savoirs

3. ARDUINO – UNE PLATEFORME DE PROTOTYPAGE RAPIDE

L’approche des Makers n’aurait pas eu autant d’impact qu’elle en a eu s’il n’y avait pas eu des plateformes de prototypage rapide telle qu’Arduino dont la simplicité d’utilisation réduit très fortement le fossé technique nécessaire à l’apprentissage des systèmes embarqués tout en ne sacrifiant rien des possibilités offertes et des performances envisageables.

3.1. Arduino : le couteau suisse de l’électronique embarquée moderne

Sorti en 2005 comme un modeste outil pour les étudiants de Massimo Banzi à l’Interaction Design Institute d’Ivrea (IDII) dans le Piémont (Italie), Arduino a initié une révolution du DIY dans l’électronique à l’échelle mondiale : Arduino est devenu le projet le plus influent de son époque dans le monde du matériel libre.

Cette petite carte est actuellement perçue comme le couteau suisse de nombreux artistes, passionnés et étudiants. En tant que telle, on la retrouve dans de nombreuses expériences pédagogiques académiques [1]. Au-delà de sa facilité d’accès, c’est un outil performant utilisé dans des grands centres de recherche internationaux tels que le CEA-LETI pour les structurations préliminaires de cahier des charges, utilisé pour piloter la grande majorité des imprimantes 3D commercialisées actuellement. La communauté qui s’est constituée autour cette plateforme est telle qu’Intel en Octobre 2013 a annoncé la sortie d’une carte Arduino nommée Galileo intégrant leur SOC QUARK X1000. Ce circuit, véritable carte mère dont les plans sont disponibles en open source, se place certes en concurrent des systèmes type Raspberry Pi mais une telle initiative relève avant tout du fait que même si Intel n’est pas en mesure d’évaluer tout le potentiel de l’approche Arduino et sa communauté, c’est un mouvement dans lequel il se doit de s’inscrire et de développer des actions.

3.2. Arduino : bref historique

Arduino est né d’un défi somme toute assez classique : comment apprendre à des étudiants à créer rapidement des

systèmes électroniques. En 2002, Massimo Banzi, professeur associé à IDII après une carrière dans l'industrie d'architecte logiciel avait besoin d'une plateforme pour promouvoir de nouvelles approches pour la conception interactive — un champ naissant connu sous le nom d'informatique physique. Cependant avec un budget et un temps d'enseignement limités, ses options de choix d'outils étaient rares. La plateforme couramment utilisée à l'époque dans de tels contextes, le BASIC Stamp (microcontrôleur de l'entreprise californienne Parallax) présentait trois inconvénients : un coût unitaire de carte un peu élevé, une puissance de calcul trop limitative et un portage logiciel excluant les plateformes d'Apple (couramment utilisées dans le monde du design).

S'inspirant de Processing, langage de programmation intuitif développé au MIT qui permet aux programmeurs sans expérience de créer des infographies complexes, Massimo Banzi décida de créer un interface similaire à l'environnement de développement très facile à utiliser de Processing pour programmer un microcontrôleur. A partir d'une contrainte budgétaire d'un coût de 30\$ et d'une utilisation "plug'n'play", décision clé qui permet une utilisation immédiate de la carte sans ajout de composants ou de matériels supplémentaires sans compter l'impératif, Massimo Banzi et son équipe sont parvenus à élaborer une plateforme de développement de systèmes électroniques dont les préceptes sont les suivants :

- Apprentissage de l'électronique passe par la pratique dès le premier jour au lieu de commencer par en apprendre l'algèbre
- Démocratisation de l'ingénierie
- Mise à disposition d'une plateforme ouverte aux artistes et concepteurs et permettant d'être le support médiateur pour un dialogue aisé avec les ingénieurs

3.3. Architecture matérielle de la carte Uno d'Arduino

Il existe actuellement 17 déclinaisons de cartes Arduino : Uno, Leonardo, Due, Mega, Micro, Nano, LilyPad Arduino, Esplora, Yun,...

Figure 3 - Fonctions principales de l'ATMEGA328P [2]

La plus utilisée est sans nul doute la carte Uno qui en est à sa troisième révision. Le microcontrôleur ATMEGA328P de la société ATMEL (cf Figure 3) qu'elle embarque a 32ko de

Flash, 1ko d'EEPROM, 2ko de SRAM, dispose de 3 ports entrées/sorties numériques (ports B, C et D) de deux compteurs 8 bits (Timer0 et Timer2) et un compteur 16 bits (Timer1), 6 broches PWM (réparties sur les ports B et D), 6 entrées multiplexées permettant la conversion Analogique-Numérique sur une résolution de 10 bits (port C) et intègre un comparateur analogique. Il gère le bus I2C, le port série (USART), intègre un compteur Watchdog programmable et dispose de 24 sources possibles d'interruption.

La carte Uno est ainsi dotée de 14 broches d'entrées/sorties numériques, 6 broches permettant de générer un PWM (utilisation des compteurs du 328P) et 6 entrées dites analogiques intégrant le Convertisseur Analogique-Numérique. Cette carte qui mesure 68,6mm par 53,3 mm est dotée d'un port USB (interfaçage via une puce ATmega8U2) pour la programmation. Pour plus de détails, se reporter à la Figure 4.

Figure 4 - Carte Uno d'Arduino : architecture principale [2]

3.4. Arduino : aspect logiciel

La force d'Arduino réside dans son interface est dans la réécriture qui a été faite des fonctions qui masquent un certain nombre d'aspects de la programmation des microcontrôleurs (configuration des registres, manipulation directe des ports, pas de debugging). Deux éléments concourent au succès d'Arduino : l'environnement de développement et la librairie Arduino.

3.4.1. L'environnement de programmation

Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateforme, servant à la fois d'éditeur de code et de compilateur. Firmware et programmes sont transférés au travers de la liaison série réalisée par une connexion RS-232, bluetooth ou USB selon les modules. Il est également possible de ne pas utiliser l'interface Arduino et de compiler et uploader les programmes via l'interface en ligne. Du fait de la présence du port ICSP, il est enfin possible de programmer le microcontrôleur de manière classique à l'aide des outils d'ATMEL

Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend

aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++.

3.4.2. Des commandes simplifiées

En masquant les configurations des registres et en réécrivant les fonctions d'utilisation des ports, l'équipe d'Arduino a redéfini les fonctions élémentaires que peut réaliser une carte Arduino et on se retrouve ainsi avec un jeu réduit de possibilités permettant de traiter la quasi totalité des nécessités d'un système embarqué :

1. Écriture/Lecture de signaux binaires : commande `arduino digitalRead()` et `digitalWrite()`
2. Écriture et lecture de signaux analogiques : commande `AnalogRead()` et `AnalogWrite()`. La première commande utilise les convertisseurs analogique-numérique du microcontrôleur la seconde utilise les ports PWM.
3. Utilisation du port RS-232 pour une communication série : commande `Serial.print()` ou `Serial.read()`.

A ceci s'associent un ensemble de commande permettant de contrôler la structure du programme (boucles,...), divers opérateurs (arithmétiques), booléens, comparaison, pointeurs), la gestion des constantes et des variables (différents types), la gestion simplifiée des interruptions,... Avec ce jeu réduit de fonctions élémentaires, on est ainsi capable de créer un grand nombre de systèmes plus ou moins complexes, d'autant plus que la communauté a développé au fil des ans un certain nombre de modules qui s'adaptent sur les cartes Arduino pour réaliser des fonctions spécifiques.

3.5. Arduino et ses cartes filles

La richesse d'Arduino tient beaucoup à ses "Shields", nom donné à ses cartes filles qui peuvent se connecter et s'empiler au-dessus de la carte principale comme montré dans l'exemple de la Figure 5.

Figure 5 - Exemple d'une carte Arduino équipée de cartes filles ou "Shields"

Chaque carte fille a des spécificités : lecture MP3, contrôleur DMX, carte Wifi/Bluetooth/Zigbee, Contrôleur moteurs, module GPS, NFC,... Chacune de ces cartes filles est fournie avec une librairie qui permet son utilisation à partir du recours à un certain nombre de commandes et mots clés traditionnellement délivrés à la carte fille par le biais d'une commande `Serial.print()` issue de la carte Arduino principale.

En délocalisant le traitement de fonctions spécifiques dans les cartes filles et en simplifiant l'utilisation par le recours à des mots clés clairs par la construction d'une librairie adhoc, il est donc possible à tout néophyte de construire très rapidement des systèmes complexes.

4. INTEGRATION D'UNE APPROCHE DIY POUR L'APPRENTISSAGE DE L'ELECTRONIQUE

En se basant sur le kit Arduino développé conjointement par Radiospares et Massimo Banzi [3] et sur son ouvrage de référence sur Arduino [4], nous avons développé une approche pédagogique fondée sur l'expérimentation directe. Aucun cours n'est donné sur l'architecture matérielle ou logicielle de la plateforme, ne comptent que la programmation du système embarqué et l'implémentation de programme plus ou moins complexe utilisant parfois des cartes filles. L'objectif est de rendre les étudiants autonomes sur la programmation de telles cartes. La carte est vue comme une boîte noire dont le fonctionnement interne est en grande partie occulté. Au travers d'un certain nombre d'exemples les étudiants apprennent à programmer la carte Arduino et à structurer un programme informatique. L'enjeu de ce cours est de rendre les étudiants capable d'écrire des briques élémentaires de code, de comprendre un code qui n'est pas le leur et de s'approprier toute ou partie d'un code. Ainsi, sur une pratique dérivée du hacking, les étudiants sont amenés à développer leurs propres applications à partir d'un assemblage de codes récupérés et de code personnel. L'objectif final n'est pas de produire un code optimal mais de produire un code fonctionnel.

Cette approche a été développée dans le cadre de deux milieu académiques complètement différents : le premier s'inscrit dans le cursus d'une école d'ingénieur généraliste l'École Centrale Lyon dont le public est majoritairement issu des classes préparatoires. Le second s'inscrit dans le cadre d'un programme post-licence de 2A développé par École Centrale Lyon et EMLyon Business School centré sur l'innovation et l'entreprenariat dont le public est très hétérogène puisque les étudiants sont issus de tous les domaines : sciences de gestion, sciences de l'ingénieur, sciences humaines et sociales, droit, design, arts,...

Nous allons présenter comment cette approche a pu être déployé dans ces environnements après en avoir fait une brève description.

4.1. Intégration dans le cursus ingénieur de l'École Centrale Lyon

Le public étudiant de l'École Centrale Lyon est majoritairement issu des classes préparatoires. Cette école d'ingénieurs généralistes propose un cursus de tronc commun généraliste sur 3 semestres, suivi d'un parcours électif constitué d'un semestre d'option et de deux semestres (troisième année d'école d'ingénieur) de spécialisation sur un domaine sectoriel (trafic et transports, aéronautique, énergie, bâtiment, micronana technologies,...) et un métier (finances, conception, développement, recherche,...).

4.1.1. La place de l'enseignement d'électronique à l'École Centrale Lyon

Il n'y a plus, depuis quelques années déjà, d'option d'électronique à l'École Lyon. Les filières un peu spécialisées en électronique traitant des systèmes embarqués en Trafic et Transports ou bien en Aéronautique ont été fermées faute d'inscription. L'enseignement d'électronique se décompose ainsi

✓ Cours d'électronique générale en première année : de la physique du semi-conducteur au fonctionnement du microprocesseur ; 15h CM, 14h TD et 8hTP

✓ Un cours optionnel au troisième semestre du tronc commun (3 modules proposés en électronique parmi un choix de 32 cours) : Architecture Embarquées et Informatique Industrielle, Architectures Embarquées de calcul et de traitement de l'information et Capteurs intelligents communicants : systèmes d'interface. Ces cours ont le volume suivant : 16h de CM, 10h de TD/BE et 8h de TP

✓ Un cours à choisir en parcours électif sur le quatrième semestre du tronc commun :

- Filtrage adaptatif : application au contrôle actif de bruit,
- Capteurs et traitement d'images
- Design de l'interaction et prototypage rapide par Fablab.

Ces cours ont un volume de 24h présentiel élève.

4.1.2. Introduction d'Arduino dans le cursus de l'élève ingénieur Centrale Lyon

L'apprentissage des systèmes embarqués au travers d'une plateforme Arduino a lieu dans le cadre du premier semestre du parcours électif (soit le second semestre de l'année 2 du cycle d'ingénieur) dans un cours intitulé : "Design de l'interaction et prototypage rapide par Fablab". Ainsi les étudiants (l'effectif du cours est fixé à 24) susceptibles d'assister à ce cours ont tous suivis le cours de première année d'électronique générale et ont des solides notions de programme informatique. Par contre, du fait de la latitude laissée par les choix d'option dans le début de la deuxième année (troisième semestre du tronc commun), certains étudiants peuvent avoir des connaissances et compétences dans la programmation de systèmes embarqués, d'architecture de microprocesseurs, de programmation de FPGA et d'interfaçage de capteurs tandis que d'autres n'ont rien de plus comme bagage de connaissance que celui de la première année.

L'objectif de ce cours est d'initier les étudiants à la pratique d'un fablab comme plateforme de prototypage rapide mais surtout en tant que laboratoire de création d'activité et centre de ressources. Au travers de ce cours est introduite également la notion de Design Thinking [5] et son apport à un cursus d'ingénieur. Parmi les pratiques fablab abordées, sont développées les outils de prototypage électroniques dans un contexte d'Internet des Objets omniprésent.

A l'issue de ce cours, deux constats émergent : le premier est l'enthousiasme général des étudiants vis à vis de cette

approche de l'électronique, certains demandant même à emprunter des kits de développement pour aller plus loin dans la pratique à titre personnel. Le second constat est que les étudiants ont le sentiment que développer des systèmes embarqués n'est pas une affaire de spécialiste et que cela est à la portée de tout le monde. Ainsi des étudiants rebutés par les cours de première année et la vision académique qui en est donnée, sortent de ce cours avec une vision plutôt positive de l'électronique embarquée voire un intérêt profond pour celle-ci. D'autres étudiants, généralement ceux qui ont fait les cours d'option de début de deuxième année, voient en Arduino une plateforme extrêmement performante pour le déploiement rapide et efficace de solutions à base de systèmes embarqués.

4.2. Intégration dans le Programme I.D.E.A.

Le Programme I.D.E.A. (Innovation Design Entrepreneuriat et Arts) a pour objectif de former des entrepreneurs et intrapreneurs de l'innovation dans un enseignement transdisciplinaire tel que prescrit par Dunne and Martin [6]. Ce modèle prend ses racines dans l'approche du Design Thinking telle que développée par Tim Brown [7]. Le Design Thinking est enseigné comme une méthode opérationnelle de gestion des projets. Ce cursus a intégré un fablab au coeur de son approche pédagogique développée autour de 4 unités d'enseignement : Entrepreneuriat, Design Thinking, Sciences Technologies et Enejeux Sociétaux et Sciences Humaines et Sociales. Ce cursus de 2 ans est de plus basé sur une approche d'apprentissage par le projet [8] intégrant l'apprentissage par le problème [9]. Ainsi, plus de 800 heures par an sont consacrées aux projets.

Une des grandes difficultés d'enseignement dans ce programme est la très forte hétérogénéité de niveau des étudiants dans des domaines spécifiques. Par exemple, certains étudiants provenant d'école d'art peuvent ne pas avoir eu d'enseignement mathématiques et scientifiques depuis la fin du lycée voire parfois depuis la troisième.

Ainsi pour l'apprentissage des systèmes embarqués, une approche par le "Faire" (Do It Yourself – DIY-) est naturellement appropriée. Dans le cadre de ce cours, un enseignement d'Arduino de 15 heures est développé pour leur fournir les bases de développements de systèmes embarqués communicants qu'ils mettront en œuvre lors de leurs projets à savoir celui associé à la Fête des Lumières [10] dont l'enjeu est la réalisation d'une installation lumineuse et interactive intégrée dans les manifestations liées à cet événement (4 jours d'exposition et de médiation avec le public, cf. Figure 6) et un autre projet de plus grande envergure qui les amène à développer des produits et services associés à une thématique particulière, produit et services qui sont ensuite exposés pendant une semaine lors d'une manifestation faisant écho à la Biennale Internationale de Design de Saint Étienne.

Figure 6 - Fête des Lumières IDEA 2013 dans la cours de l'ESPE de Lyon

A l'issue de ce cours et son application dans les projets, on est ainsi capables d'amener la quasi-totalité de la promotion (30 étudiants) à une familiarité avec les systèmes embarqués. Environ un tiers des étudiants ne sera pas capable d'aller au-delà des programmes basiques sous Arduino (généralement par manque d'appétence pour le domaine). Sur les deux tiers restants, un quart est capable de programmer des applications complexes, le restant capable de développer des applications standard à base des exemples basiques qu'ils maîtrisent et leur capacité à réutiliser du code existant. Parmi les exemples d'application développés sur l'année 2013-2014, on pourra citer l'animation de lampes interactives et animées comme le montrent la Figure 7 ou encore le produit Nymphaea conçu pour l'animation interactive et contextuelle de plans d'eau (Figure 8).

Figure 7 - Préparation des installations lumineuses et interactives du 8 décembre 2013

5. LIMITES DE L'APPROCHE PAR LE "FAIRE"

Ces deux enseignements ont été menés sur deux populations extrêmement différentes avec un objectif commun d'apprendre par le "Faire" (Learning By Doing – LBD-).

L'approche utilisée dans les cas a été la même : à partir du kit de démarrage développé par Massimo Banzi en partenariat avec Radiospares [4], en suivant le livre d'apprentissage inclus dans le kit de développement les étudiants apprennent à utiliser la carte Arduino dans une approche pas à pas, du traditionnel "blink" à l'utilisation des ports analogiques et de la communication entre une carte Arduino et un ordinateur via le port USB.

Si cette approche se révèle efficace dans les deux, la population du Programme I.D.E.A. se heurte assez vite (dès que les étudiants commencent à vouloir réutiliser et/ou détourner des morceaux de codes existants) à un problème de compréhension de la structure d'un code, de la notion de variables, de boucles, de fonctions et donc de paramètres. L'approche par le "Faire" en tant que telle pourrait suffire si le temps imparti à l'apprentissage était plus long et si tous les étudiants étaient d'une égale motivation. On arrive dans le cas présent aux limites de l'approche dite des "Makers" dans laquelle la motivation individuelle soutient l'effort d'apprentissage et accompagne les individus dans un apprentissage autonome des bases "théoriques" de la programmation quand le besoin s'en fait ressentir. Dans le cas d'un enseignement universitaire, il faut donc compléter l'approche du DIY par un accompagnement pédagogique afin d'amener les étudiants au niveau souhaité, en intégrant des compléments théoriques réguliers.

Figure 8 - Objet flottant interactif et contextuel

Au contraire, la démarche de l'apprentissage uniquement par la pratique n'a posé aucune difficulté pédagogique sur la population centralienne pour deux raisons principales. La première est que les fondements de programmation et d'algorithmie leur sont enseignés en première année du cursus d'élève ingénieur : ils disposent donc de ce bagage théorique. La deuxième raison est que cet enseignement leur permet de

mettre en pratique les notions théoriques qu'ils ont pu apprendre et ce de manière très opérationnelle puisqu'en moins de 4 séances ils sont capables de créer un réel système électronique embarqué moyennement complexe répondant à un cahier des charges précis.

La plateforme Arduino est donc un outil qui se prête aussi bien à l'enseignement des systèmes embarqués en cursus d'ingénieur à un assez bon niveau de technicité qu'à l'apprentissage de l'électronique par le biais des systèmes embarqués pour un public de néophyte à cela près que dans ce dernier cas, l'approche de l'apprentissage par le faire ne doit pas être exclusive et accompagné par un cadrage théorique à dispenser de manière ponctuelle et adhoc.

6. CONCLUSION ET PERSPECTIVES

Dans ce travail nous avons montré comment une approche alternative à l'enseignement de l'électronique traditionnel pouvait permettre d'atteindre des résultats convaincants dans le développement de systèmes embarqués. L'expérimentation a été menée sur un public académiquement néophyte dans le domaine et sur un public initié à l'électronique. Dans les deux cas, la rapidité de mise en action et d'obtention de solutions viables a été favorablement perçue et a dans certains cas été le déclencheur d'un intérêt pour la matière.

Le recours à cette plateforme offre donc des perspectives intéressantes dans la manière dont l'enseignement de l'électronique peut être repensé, aussi bien dans un cycle d'ingénieur généraliste que pour un public de niveau master sans pré-requis de formation scientifique.

Cependant, la facilité d'utilisation des solutions Arduino ainsi que le support offert par sa communauté ne doit pas faire oublier que le développement de solutions complexes nécessite souvent le recours à du debuggage en ligne pour lequel la solution logicielle offerte par Arduino n'offre pas de solution de debug in-situ (absence de mode pas-à-pas) ce qui peut rendre certaines conception relativement complexes et fastidieuses.

Arduino est actuellement en phase de déploiement sur certains cours spécialisés de troisième année et fait l'objet d'une formalisation dans son usage au sein du fablab et des enseignements qu'il peut générer.

7. REMERCIEMENTS

Ces travaux ont pu être mis en place grâce à l'Aide financée sur le programme d'Investissements d'Avenir lancé par l'Etat et mis en œuvre par l'ANR.

BIOGRAPHIES

Fabien Mieyeville a obtenu son doctorat en électronique intégrée en 2001. Il est actuellement Maître de Conférences HDR à l'École Centrale Lyon et chercheur à l'Institut des Nanotechnologies de Lyon (UMR CNRS 5270). Ses thématiques de recherche englobent les systèmes embarqués distribués communicants, les méthodologies de conception hiérarchique de systèmes hétérogènes, les living labs et la conduite de l'innovation par le Design Thinking et l'effectuation. Il est par ailleurs co-responsable du Programme I.D.E.A. et responsable du groupe de recherche et du fablab qui y sont associés.

David Navarro a obtenu un doctorat en Systèmes Electroniques en 2003. Il est actuellement Maître de conférences HDR à l'Ecole Centrale de Lyon. Il effectue sa recherche à l'Institut des Nanotechnologies de Lyon (UMR CNRS 5270) dans le domaine de la simulation et de la conception de systèmes électroniques intelligents et communicants.

Laurent Carrel est Ingénieur Systèmes et Réseau. Il est actuellement Administrateur réseau, ressources informatiques et systèmes d'information à l'Ecole Centrale de Lyon depuis 1999. Il partage son temps de travail entre la DSI de l'Ecole Centrale et l'Institut des Nanotechnologies de Lyon (UMR CNRS 5270).

REFERENCES

- [1] J. Sarik and I. Kymissis, "Lab kits using the Arduino prototyping platform," in *Frontiers in Education Conference (FIE)*, 2010 IEEE, 2010, p. T3C-1.
- [2] <http://www.pighixxx.com/>
- [3] <http://fr.rs-online.com/web/generalDisplay.html?id=arduino-starterkit>
- [4] Banzì, Massimo. *Démarrez avec Arduino - 2e édition: Principes de base et premiers montages*. Édition : 2e édition. Paris: Dunod, 2013.
- [5] Dym, Clive L., et al. "Engineering design thinking, teaching, and learning." *Journal of Engineering Education* 94.1 (2005): 103-120.
- [6] Dunne, David, and Roger Martin. "Design Thinking and How It Will Change Management Education: An Interview and Discussion." *Academy of Management Learning & Education* 5, no. 4 (2006): 512-23.
- [7] Brown, T. *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. NY: Harper Business, 2009.
- [8] Mills, Julie E., and David F. Treagust. "Engineering education—Is problem-based or project-based learning the answer?." *Australasian Journal of Engineering Education* 3.2 (2003).
- [9] Savery, J. R. Overview of Problem-based Learning: Definitions and Distinctions. *Interdisciplinary Journal of Problem-based Learning*, 1(1), 2006.
- [10] <http://programme-idea-lumieres2013.tumblr.com/>