

# Projets de robotique mobile à bac+2 : des Lego Mindstorm au robot fabriqué par soi-même

Xavier Redon<sup>1</sup>, Thomas Vantrois<sup>1,2</sup>, Alexandre Boé<sup>1,3</sup>

<sup>1</sup> Univ. Lille, Polytech Lille, F-59000 Lille, France  
rex@polytech-lille.fr

<sup>2</sup> Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRISTAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France  
thomas.vantrois@univ-lille1.fr

<sup>3</sup> Univ. Lille, CNRS, ISEN, Univ. Valenciennes, UMR 8520 - IEMN, F-59000 Lille, France  
alexandre.boe@univ-lille1.fr

**Résumé.** Cet article présente l'évolution des projets de robotiques mobiles dispensés à bac +2 pour préparer l'entrée à Polytech Lille. Ces projets permettent aux étudiants de découvrir les problématiques liées aux la conception des objets embarqués. Afin de rendre les projets attractifs, nous avons choisi de travailler sur des robots mobiles. Pour plus de simplicité, nous avons mis à disposition des robots Lego Mindstorm avec un logiciel de de programmation graphique. Une carte mini-ordinateur FoxBoard a été ajoutée pour augmenter la connectivité. Rapidement, des limitations en termes de développement et d'apprentissage ont été mises à jour. Le projet a donc évolué vers la conception intégrale d'un robot mobile autour d'un microcontrôleur.

**Mots-clés.** Informatique, robotique mobile, électronique, projet étudiant

**Abstract.** This paper presents the evolution of mobile robotic project for graduated students applying to Polytech Lille School. These projects are dedicated to students to discover the issues encountered when we design embedded objects. To make them interesting, we have chosen to work on mobile robots. For simplicity reasons, Lego Mindstorm boxes with dedicated graphic programming language have been used. A mini-computer Foxboard has been added to enhance the connectivity. Quickly, some limitations in terms of design and learning have been discovered. The project has evolved to a complete design of a mobile robot with a microcontroller.

**Keywords.** Computer science, mobile robotics, electronics, student project

## 1 Introduction

L'école d'ingénieurs Polytech Lille [1] fait partie du réseau Polytech. Le recrutement se fait majoritairement à bac+2 après un DUT (Diplôme Universitaire de Technologie), un parcours en License 2 ou CPGE (Classe Préparatoire aux Grandes Ecoles). Une nouvelle voie originale inspirée des classes préparatoires intégrées a été créée afin de diversifier le recrutement : le parcours PEIP (Parcours des écoles d'ingénieurs Polytech). Ces deux années permettent de découvrir les différents

domaines d'ingénierie et de choisir sa spécialité parmi plus de 80 proposées par les 13 écoles du réseau. À l'issue de ces 2 années, les étudiants ayant validé leur parcours ont un accès direct, et de droit, à l'une des 13 écoles du réseau Polytech. Ce parcours préparatoire se déroule conjointement à l'Université Lille 1 et à Polytech Lille. En plus des enseignements de licence, les étudiants suivent des modules complémentaires d'anglais et de culture de l'ingénieur. Parmi ces modules complémentaires, les étudiants choisissent un projet tutoré appelé Bureau d'Études (BE) leur permettant de découvrir des aspects des différentes spécialités offertes. Ce BE a lieu en semestre 4 pour un volume horaire étudiant de 40 h (20 x 2 h).

La formation IMA (Informatique Microélectronique et Automatique) est centrée sur la conception des systèmes embarqués avec notamment la découverte des interactions entre le matériel et le logiciel embarqué. Afin de présenter cette formation et de préparer les étudiants aux travaux sous forme de projet, nous proposons un BE centré sur les systèmes embarqués. Pour rendre le bureau plus attractif, les systèmes embarqués considérés sont des robots.

La section 2 présente plus en détail les objectifs et attendus du bureau d'étude. Après avoir testé la version initiale quelques années, nous avons changé le bureau d'étude. Les évolutions seront présentées en section 3.

## **2 Présentation du bureau d'étude initial**

Le bureau d'étude a pour but de présenter les problématiques que les étudiants pourront rencontrer s'ils intègrent la formation IMA par la suite. Afin de répondre aux questionnements des étudiants et d'ancrer le BE à l'interface entre matériel et logiciel, les projets sont encadrés par des enseignants en informatique et en électronique. Le travail se fait en binôme sur des sujets différents ayant chacun une base commune.

Les étudiants sont pour la plupart novices (bac +2) dans les domaines traités lors du bureau d'étude. De plus, leur niveau est hétérogène, notamment en informatique ; le problème se pose moins pour la partie électronique, les connaissances étant essentiellement théoriques (électrocinétique, électromagnétisme, ...). En informatique, deux facteurs peuvent influencer leur niveau réel ou ressenti (par l'étudiant) : leur parcours préalable peut être variable en fonction des options choisies (initiation au langage C, Python, Caml, ...) ou en fonction de leur affinité personnelle.

### **2.1 Objectifs**

Le bureau d'étude poursuit plusieurs objectifs tant pour les étudiants que pour la formation. Le premier objectif est d'attirer des étudiants vers la formation en leur présentant les problématiques des systèmes embarqués. Le second objectif est pédagogique : amener les étudiants à un niveau « raisonnable » en informatique embarqué et électronique. Afin de rendre attractif cette étude pour des étudiants possiblement débutants, nous avons choisi de nous appuyer sur des robots.

Le troisième objectif est d'initier une réflexion sur les problématiques rencontrées dans les systèmes embarqués et d'appliquer des méthodes scientifiques pour résoudre les problèmes qui apparaissent lors des expérimentations. Par ailleurs, une première

initiation à la gestion de projet est proposée, notamment car les étudiants travaillent en autonomie et doivent gérer leur temps et leurs interactions avec leur binôme ou les autres binômes

## 2.2 Déroulé du bureau d'études et attendus

Le bureau d'étude est organisé autour du travail des binômes d'étudiants sous la forme d'un projet. Le cadre et les objectifs généraux sont proposés par les enseignants et présentés lors de la première séance.

Par la suite, les étudiants se répartissent en binômes et choisissent de façon concertée un des « sous-projets » proposés. Ils peuvent alors redéfinir les objectifs à atteindre, amender le sujet en ajoutant des fonctionnalités. Dans l'idéal, une fois les objectifs fixés, les étudiants analysent les problèmes. Dans la réalité, ils commencent à réaliser des expérimentations pour mieux cerner les dits problèmes. Par la suite, ils résolvent petit à petit les problèmes rencontrés. Lors de cette résolution, ils reviennent à l'analyse du sujet, ce qui permet de les amener pas-à-pas vers une démarche scientifique plus construite.

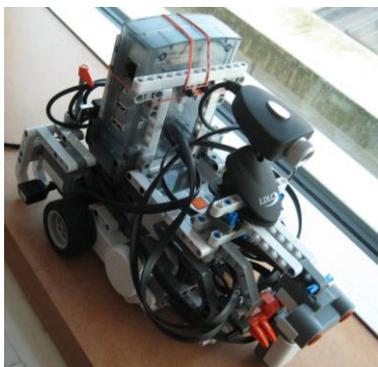
Tout au long des séances, les étudiants tiennent un journal d'expérimentation dans lequel ils consignent leur travail, leurs éventuelles difficultés et résultats obtenus. Pour garder la mémoire des travaux précédents et permettre la réutilisation des résultats d'autres binômes, ce journal est public et organisé sous la forme d'une page d'un Wiki [2]. En fin de projet, les étudiants présentent leur projet et leur travail sous la forme d'une vidéo (disponibles sur le Wiki). Elle est constituée d'une description du robot réalisé et d'une démonstration du fonctionnement réel.

Les robots sont basés sur des kits Lego Mindstorm NXT [3] comportant, outre des pièces de Lego de construction, des servo-moteurs, des capteurs (ultrason, couleur, boussole, RFID, ...) et une brique « intelligente » embarquant un processeur, un écran et de la mémoire. Pour permettre des interactions avec un PC distant, des systèmes embarqués FoxBoard sont ajoutés. Ils permettent d'embarquer un serveur Web, une caméra et sont accessibles via connexion Wifi. La liaison avec le robot se fait au travers du protocole Bluetooth.

## 2.3 Exemples de projets

Afin de clarifier le déroulé des projets, voici un exemple représentatif entre les années 2010/2011 (première édition) et 2014/2015 (dernière édition dans la version Lego).

En 2011/2012, le but général du projet était la réalisation de robots patrouilleurs devant parcourir de façon semi-autonome un espace, tout en recueillant des données. Le parcours est balisé par des bandes de couleur au sol et par des tags RFID. La figure 1 présente une vue d'un robot construit par les étudiants. La construction n'est pas linéaire, le modèle présenté ici résulte d'aller et retour entre conception mécanique, installation des capteurs, de la FoxBoard [4], de la caméra, ...



**Fig. 1.** Exemple de robot construit par les étudiants en éléments Lego avec une caméra et un système FoxBoard (tenu par les élastiques orange).

Les tâches prévues pour les binômes sont résumées à la Table 1. Les robots devaient – à terme – embarquer toutes les fonctionnalités développées par les différents binômes.

**Table 1.** Résumé des taches développées par les binômes.

Type de robot	Objectifs	Matériel	Binômes impliqués
Suiveur de lignes	Suivre une courbe discontinue au sol	Capteurs de couleur, sonar.	1
Robots synchronisés	Synchroniser deux robots pour la détection conjointe d'obstacles et leur évitement	Sonar	2
Suiveur RFID	Détecter un tag RFID et suivre le cap indiqué par le tag	Lecteur RFID, boussole	1
Navigation aux instruments	Navigation précise avec une boussole et transmission de l'orientation	Boussole	1
<b>Robot téléguidés</b>	<b>Permettre de diriger le robot avec une télécommande avec un retour visuel</b>	<b>FoxBoard et clé Bluetooth, webcam, téléphone sous Android</b>	<b>3</b>
Intégration des fonctionnalités	Intégrer toutes les fonctionnalités développées dans un seul robot	-	tous

Dans un premier temps les binômes font la conception et la réalisation de leur tâche. Ils sont dès lors confrontés à un premier choix : le langage de programmation à utiliser. Soit ils s'orientent vers le langage graphique fourni avec les boîtes Lego, soit ils utilisent un langage proche du C (NXC, Not eXactly C [5]). Le langage graphique permet une mise en œuvre visuelle et adaptée aux débutants des algorithmes à implanter. Rapidement, ils se rendent compte de deux types d'interactions très importantes. Tout d'abord, les différents binômes doivent coordonner leurs efforts car ils sont vite dépendants du travail des autres. La coordination facilite l'intégration de parties développée par les autres binômes et fait donc gagner du temps de développement. Deuxièmement, les étudiants prennent conscience du lien entre

matériel et logiciel. Ce lien est au départ basique, tel que la prise en compte du port de liaison des capteurs ou actionneurs sur la brique de commande. Par la suite des interactions plus subtiles apparaissent comme la prise en compte de l'état de charge des batteries qui conditionne les déplacements et les erreurs de trajectoire associées.

Les binômes ayant travaillé sur les robots téléguidés est représentative de l'hétérogénéité des parcours et connaissance, mais pour obtenir au final des résultats assez similaires. Le bloc contrôleur NXT permet de brancher les capteurs (quatre au maximum) et les actionneurs (3 servo-moteurs), mais ne possède qu'une liaison Bluetooth pour communiquer avec l'extérieur. Pour permettre de transmettre des informations, nous avons conçu sur la FoxBoard un serveur faisant office de passerelle entre la connexion WiFi et la connexion Bluetooth. Sur le contrôleur NXT, les étudiants peuvent – par l'intermédiaire de boîtes aux lettres – envoyer ou recevoir des messages (limités à un octet) et exécuter en conséquence les actions associées. La figure 2 résume le système d'envoi des données.



**Fig. 2.** Représentation du système d'envoi de commande entre un client (téléphone Android (a)) et l'actionneur du robot (d), via un système FoxBoard (b) connectée par Bluetooth à la brique intelligente (c).

Un algorithme de traitement de l'information est proposé par les groupes d'étudiants. Il est exprimé de façon libre en pseudo-langage :

```

Afficher "NXT"
Tant que VRAI
  Imprimer dans Textel la valeur du signal reçu
  Si signal reçu
    Alors   afficher "Smiley"
           Comparer Textel:
             égal 0:Moteur A & C => Arrêter
             égal 1:Moteur A & C => Avancer
             égal 2:Moteur A & C => Reculer
             égal 3:MoteurA=>Avancer ; MoteurC=>Reculer
             égal 4 : MoteurA=>Reculer ; MoteurC=>Avancer
  Envoi signal "Tout s'est bien passé"

```

Fin si

Fin tant que

Par la suite, il est mis en œuvre soit en langage NXC soit en langage graphique par blocs (cf. figure 3). Il faut aussi noter que, pour certaines parties fixes de la piste des robots, les étudiants ont à disposition des Arduino, programmé en C/C++ avec l'IDE Arduino.

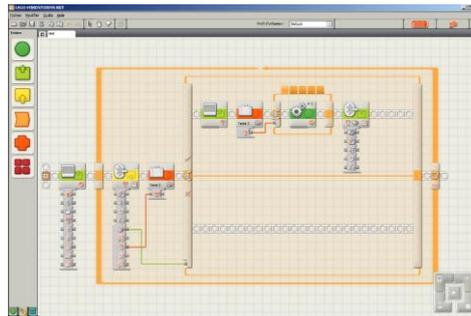


Fig. 3. Implémentation de l'algorithme de télécommande à l'aide du logiciel graphique.

### 2.3 Choix du langage et impact sur les projets

La figure 4 résume le choix des étudiants quant au langage de programmation pour le robot. Ce choix est libre pour les étudiants. Ils ont tendance au départ à privilégier le langage graphique qui paraît plus intuitif et simple à prendre en main.

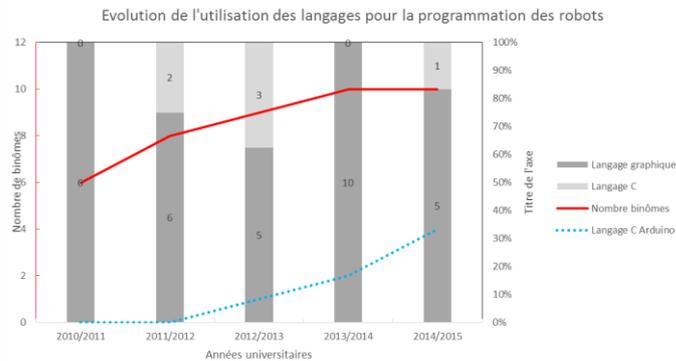


Fig. 4. Évolution du choix du langage de programmation.

Dès que les fonctionnalités deviennent plus complexes, la structuration graphique perd de sa simplicité et les étudiants passent énormément de temps au débogage du code et à sa correction. Cette tâche est complexe car les modifications sur les blocs sont malaisées et difficilement maîtrisables.

« Au départ, l'utilisation du logiciel livré avec la boîte NXT s'était révélé la meilleure solution, en effet ce logiciel permet d'assembler intuitivement un programme simple

*sous forme graphique et n'oblige pas à apprendre un langage comme le C. Cependant, le logiciel a vite montré ses limites dès que le programme devenait un peu plus complexe et contenait beaucoup d'instructions.* » (Conclusion d'un binôme).

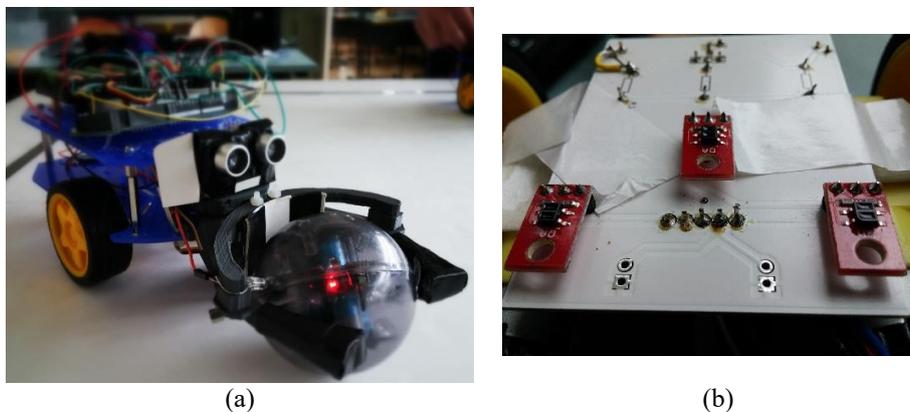
De plus le langage et le matériel étant propriétaire, les modifications matérielles ou les ajouts de capteurs non prévus sont presque impossibles. Nous avons donc décidé de faire évoluer le format du bureau d'étude vers un système à fabriquer à base de microcontrôleurs ATMEL ATmega328P (carte réalisée par les étudiants ou à base d'Arduino).

### **3 Évolution vers la conception totale du robot**

Les expérimentations avec les robots Lego ont montré leurs limites en termes d'apprentissage pour les étudiants. En effet, après quelques séances de montage, quelques séances d'initiation (programmation d'algorithmes simplistes), ils passaient énormément de temps à essayer de modifier leurs programmes ou à ajouter des fonctionnalités. Ce temps consommé était relativement improductif car les outils ne sont pas adaptés. Par ailleurs, les évolutions du matériel et logiciel associés étaient impossibles sans de très long développement. Pour remédier à ces différentes limites et pour redonner un intérêt plus centré sur le développement de logiciel embarqué, nous avons décidé en 2015/2016 de faire évoluer le bureau d'étude vers une conception et fabrication d'un robot organisé autour de microcontrôleurs ou de plateformes Arduino. Les étudiants disposent de châssis en plexiglass sur lesquels sont fixés des moteurs à courant continu. Le contrôle se fait à l'aide de contrôleurs dédiés (TB6612FNG) et un Arduino programmé à l'aide de l'IDE fourni. Différents capteurs sont disponibles et remplacent avantageusement ceux des Lego par leur mise en œuvre plus ouverte et par leur prix.

Les étudiants suivent un plan de travail plus ou moins identique à celui présenté auparavant, en ajoutant une dimension de conception mécanique (assemblage avec vis et écrous, perçages, ajout d'engrenages, boîtiers 3D ...). Par ailleurs, ils doivent réaliser des cartes électroniques permettant de relier les différents composants électroniques utilisés. Dans un second temps, il convient d'implémenter les algorithmes permettant aux robots de remplir leur tâche. L'accent est donc mis sur les systèmes embarqués mêlant intimement conceptions logicielle et matérielle. La figure 5 présente un des robots conçu l'année dernière.

Cette évolution amène des difficultés pour la réalisation des objectifs de début de projet. En effet, le travail demandé est plus varié et peu de briques (tant logicielles que matérielles) sont disponibles. Par ailleurs, c'est aussi un défi pour les encadrants qui doivent résoudre des problèmes très variables et parfois chronophages. Cependant, ce type de projet permet aux étudiants d'aborder de façon relativement ludique les problématiques de la conception d'objets embarqués.



**Fig. 5.** Robot entièrement (mécanique, électronique et logiciel embarqué) conçu par les étudiants (a) et zoom sur une carte électronique conçue et réalisée à Polytech Lille (b).

## 4 Conclusion

L'apprentissage des contraintes liées aux systèmes embarqués peut être fait aux travers de projets tutorés sur des robots mobiles. Cette approche permet de rendre attractif et démonstratif l'approche de concepts de programmation embarquée ou de liens entre matériel et logiciel.

Les robots LEGO Mindstorm permettent aux étudiants de rapidement prendre en main la conception matérielle (mécanique et électronique) et de s'attacher à la programmation. Cependant, les programmes complexes sont difficiles à programmer au travers du langage graphique fourni. Par ailleurs, les modifications sur le matériel sont ardues. Il apparaît donc qu'il est plus opportun de s'orienter vers la conception complète du robot. Cette conception nécessite de prendre en compte la mécanique du robot, les éléments électroniques et bien entendu la programmation associée. La liaison intime entre ces domaines reflète ainsi mieux les problématiques rencontrées lors de la conception de systèmes embarqués.

**Remerciements.** Les auteurs remercient les différents étudiants ayant involontairement participé à cet article ainsi que Laurent Engels pour le tournage et le montage des vidéos.

## Bibliographie

1. [www.polytech-lille.fr](http://www.polytech-lille.fr)
2. <https://peip-ima.polytech-lille.net:40001/mediawiki/index.php?title=Accueil>
3. <https://www.lego.com/fr-be/mindstorms/>
4. <https://www.acmesystems.it/FOXG20>
5. <http://briexcc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html>