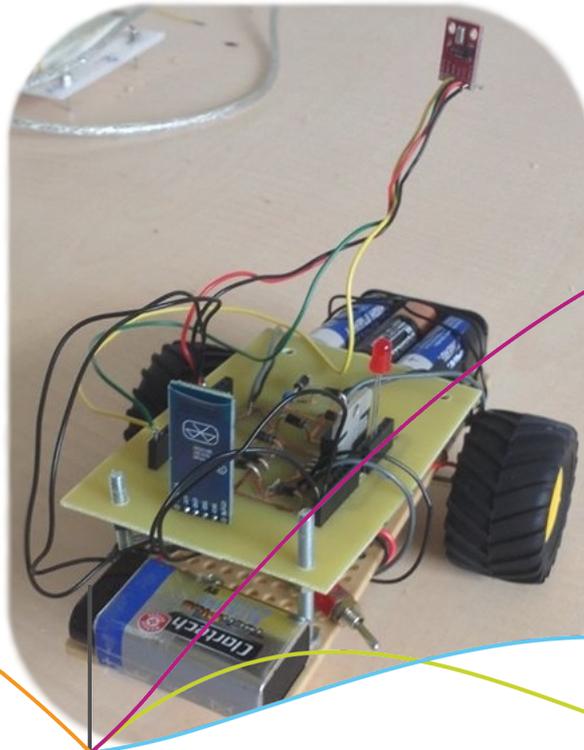


Robot guidé par Bluetooth



Enseignant responsable
Mounir MESSEDI

Étudiants :
Clément BRUNET
Guillaume CONTI
Pierre MONGOURT

Geoffrey COLINET
Etienne JULES

Date de remise du rapport : 16/06/14

Référence du projet : STPI¹/P6/2014 – 51

Intitulé du projet : Robot guidé par Bluetooth

Type de projet : *experimental*

Objectifs du projet : *Notre projet a pour principal objectif de rassembler nos compétences pour réaliser un robot guidé par un smartphone via Bluetooth.*

Mot-clé du projet :

- Robotique
- Programmation
- Mécanique

Table des matières

Introduction	4
1 Méthodologie, organisation du travail	5
1.1 Première approche	5
1.2 Répartition du travail au sein du groupe	6
1.3 Emploi du temps	7
2 Travail réalisé	8
2.1 Le robot et ses composants	8
2.1.1 Partie mécanique	8
2.1.2 Partie électronique	8
2.2 Réalisation du circuit imprimé	12
2.3 Programmation	14
2.3.1 Conception programme	14
2.3.2 Code BASIC du robot	15
2.3.3 Conception de l'application	16
2.3.4 Evolution possible	18
3 Retours et évolution du projet	21
3.1 Résultats	21
3.2 Problèmes rencontrés	21
Conclusions et Perspectives	23

Introduction

Dans le cadre de nos études d'ingénieur à l'INSA de Rouen, il nous a été demandé, ainsi qu'à tous les élèves, de réaliser un projet de physique, qui devra être présenté aux autres élèves à l'issue du troisième semestre. Les sujets sont très variés mais chaque projet nécessite un travail de groupe sérieux et régulier tout au long du semestre avec une bonne répartition des tâches.

Seulement quelques élèves avaient déjà eu l'occasion en terminale, de réaliser un travail similaire lors des Projets Personnalisés Encadrés, dans la filière scientifique, sciences de l'ingénieur. En revanche, pour la plupart des étudiants, cela représente un nouveau défi car il est nécessaire de réaliser le projet entièrement, du début à la fin, de façon efficace et disciplinée au sein du groupe.

Nous pensons que ce travail est vraiment intéressant dans le cadre de nos études car il nous oblige à travailler ensemble, à partager nos connaissances, à ne pas reproduire des erreurs ayant été réalisées par certains dans le passé : il est très important d'être à l'écoute de ses partenaires.

Ainsi, nous avons décidé de travailler sur la réalisation d'un robot, qui, d'après notre décision, sera contrôlé à distance par une application sur smartphone. C'est un projet très conséquent que nous avons tenté de mener à bien tout au long de ce semestre car nous ne disposions que de peu de connaissances en robotique.

Ce dossier retrace le parcours de nos recherches et fait le bilan de nos problèmes rencontrés et de nos résultats.

Chapitre 1

Méthodologie, organisation du travail

1.1 Première approche

Pour commencer, les premières séances étaient pleines d'appréhensions, mais heureusement nous nous connaissions déjà bien dans notre groupe et cela a évidemment facilité la communication. Nous n'avions aucune idée ni du travail qui nous serait demandé ni des moyens mis à notre disposition pour y parvenir.

Après que le professeur M. Messedi nous ai brièvement expliqué le projet à réaliser, nous avons constaté que celui-ci était assez libre ce qui peut être assez angoissant au début dans la mesure où nous n'avions aucune expérience dans le milieu de la robotique donc nous ne savions pas par où commencer. Mais dans un deuxième temps, cette liberté sur ce projet implique aussi de réaliser quelque chose de personnel et qui nous ressemble.

Ensuite pour nous aider à commencer notre projet dans le bon sens M. Messedi nous a montré les projets des années précédentes cela nous a permis de nous faire une idée beaucoup plus précise de ce qui pouvait nous attendre en termes de complexité du robot et des moyens mis à notre dispositions.

Enfin nous avons pu commencer à réfléchir aux fonctions du robot qu'on allait concevoir : pour cela nous avons rassemblé les idées de tout le monde pour se mettre d'accord sur un projet qui nous concerne tous et qui réponde aux critères imposés.

C'est ainsi qu'après plusieurs hésitations une certitude est tout de suite sorti de nos débats : « un robot commandé par Bluetooth ». En effet c'était quelque chose qui n'avait jamais été fait par les groupes des années précédentes de plus c'est un concept totalement dans l'air du temps. Nous avons donc pu finalement commander les pièces nécessaires à la réalisation de ce robot.

1.2 Répartition du travail au sein du groupe

Après s'être engagé dans ce projet, nous sommes rapidement tombés d'accord sur un point : pour avancer, il fallait se répartir les tâches. Au cours des premières séances, tout le monde travaillait sur tout sans vraiment d'organisation précise. Puis progressivement, après s'être rendu compte de la quantité de travail qu'il nous faudrait fournir dans chacune des parties du robot, Pierre et Clément sont partis sur la partie mécanique, tandis que Guillaume dans un premier temps s'est chargé de concevoir le(s) application(s) puis a aidé Geoffrey et Etienne qui se sont penchés sur la partie programmation.

Immédiatement après la première séance, nous nous sommes regroupés sur les réseaux sociaux afin de partager nos idées et nos envies.

1.3 Emploi du temps

Séance 1 (03/02)	<ul style="list-style-type: none"> – Formation du groupe – Recherche d'idées respectant la consigne (« mini-tank » avec canon à eau, lanceur de balles de tennis de table,...). – Création d'une page Facebook pour optimiser le travail en équipe et la communication.
Séance 2 (10/02)	<ul style="list-style-type: none"> – Sélection d'une idée finale. – Réalisation du cahier des charges. – Commande des pièces manquantes (capteurs, bluetooth,...).
Séance 3 (17/02)	<ul style="list-style-type: none"> – Recherche du logiciel pour coder l'application smartphone (sous android). – Recherche d'idées pour réaliser la partie mécanique et test de différentes structures. – Installation des logiciels pour la simulation électronique (Fritzing) et pour coder l'application (App Inventor).
Séance 4 (10/03)	<ul style="list-style-type: none"> – Arrivée des pièces commandées. – Recherche sur le branchement et le fonctionnement des nouvelles pièces. – Codage d'une application test.
Séance 5 (17/03)	<ul style="list-style-type: none"> – Création du schéma électronique. – Début de la programmation. – Début de la création de l'application.
Séance 6 (24/03)	<ul style="list-style-type: none"> – Début de la réalisation du schéma électronique sur le logiciel. – Suite de la programmation. – Suite de la création de l'application.
Séance 7 (07/04)	<ul style="list-style-type: none"> – Suite du schéma électronique sur le logiciel. – Suite de la programmation. – Fin de l'application.
Séance 8 (14/04)	<ul style="list-style-type: none"> – Fin du schéma électronique et installation des composants sur la carte (virtuellement). – Suite de la programmation et fin de la structure générale du code. – Mise en place du plan pour la rédaction du rapport de projet.
Séance 9 (12/05)	<ul style="list-style-type: none"> – Rectifications du schéma électronique. – Tests du programme avec seulement les composants.
Séance 10 (19/05)	<ul style="list-style-type: none"> – Fin du schéma électronique et impression du circuit imprimé.
Séance 11 (26/05)	<ul style="list-style-type: none"> – Perçage et soudure de la carte électronique.
Séance 12 (02/06)	

Chapitre 2

Travail réalisé

2.1 Le robot et ses composants

2.1.1 Partie mécanique

Cette partie ne fut que très rapidement abordée. En effet, au vu des composants disponibles ainsi que la petite taille des composants électroniques, notre choix se porta sur un véhicule basique, de forme rectangulaire avec d'un côté les deux moteurs assemblés en une pièce et de l'autre une simple roue pouvant tourner à 360 degré, laissant ainsi la direction aux deux moteurs. L'Arduino, fixée au circuit électronique, est ensuite attachée à ce rectangle à l'aide de vis et boulons et le tour est joué. Nous avons donc privilégié la fonctionnalité du robot avant son esthétique qui n'est qu'un détail.

2.1.2 Partie électronique

Tout d'abord, la pièce maîtresse du robot est l'Arduino Uno.



FIGURE 2.1 – *Photo d'une Arduino Uno.*

Ce dernier est un circuit imprimé intégré d'un microcontrôleur programmable afin de produire des signaux électriques. Son utilisation est simple et les résultats très satisfaisants. De plus, son faible coût en fait une pièce phare de la robotique de nos jours. Lorsque celui-ci est mis en tension, une LED verte s'allume avec pour intituler « On ».

Ensuite, afin de calculer l'altitude et la température, nous avons utilisé le capteur MPL31152.



FIGURE 2.2 – Photo d'un capteur MPL31152.

De même que précédemment, ce composant est parfait au vu de sa taille, sa facilité d'utilisation, son efficacité et son faible coût. Le branchement avec l'Arduino s'effectue de cette manière :

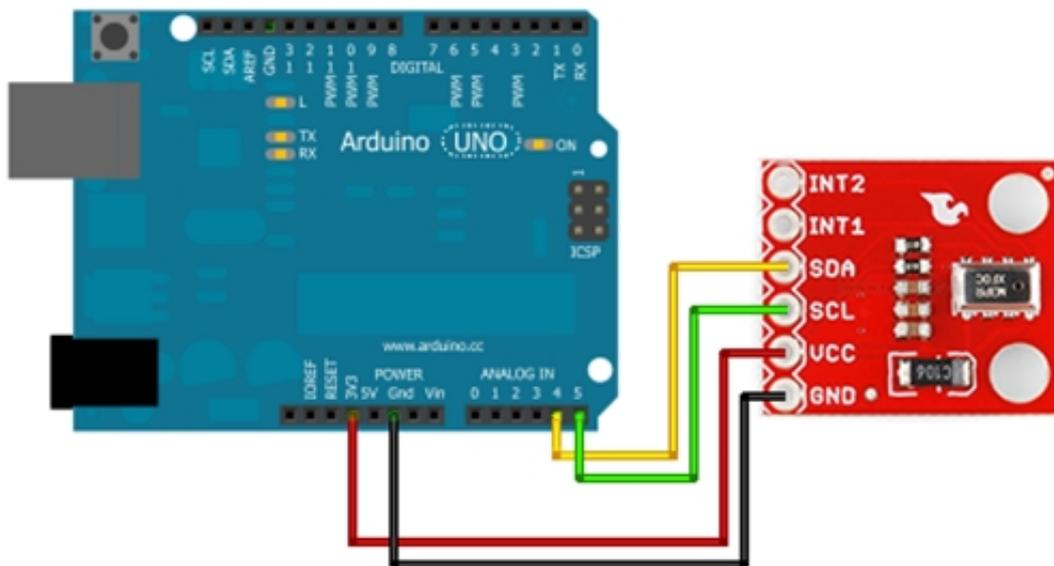


FIGURE 2.3 – Capture de notre réalisation sous le logiciel Fritzing.

Le VCC est toujours branché à l'alimentation et le GND (Ground) au GND. Le SDA et SCL sont branchés à des ports analogiques de l'Arduino, utilisés pour ce type de composant.

Cependant, à l'inverse des câbles rentrant dans l'Arduino qu'il suffit de faire rentrer dans le connecteur, nous avons dû souder les fils au capteur et ce qui fût une de ses forces se transforma en problème : sa taille. En effet, il ne faut surtout pas que l'étain soudé sur une pte soit en contact avec celui d'une autre. Le rajout d'une LED sur ce composant aurait été un plus afin de voir que celui-ci était bien en fonctionnement.

Afin de contrôler le robot via le Bluetooth, nous avons utilisé le module free2move F2M03GLA.

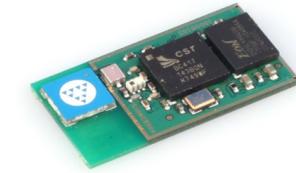


FIGURE 2.4 – Photo d'un module free2move F2M03GLA.

Notre module était déjà soudé à un petit circuit imprimé où se trouvaient des connecteurs afin de faciliter l'utilisation du module. Cependant, avec ce module est arrivé un des plus gros problèmes du projet. Nous n'avons eu aucune formation en électronique au cours de notre cursus, ainsi les données du fournisseur étaient très complexes. De plus, nous n'avons pas eu d'aide extérieur afin de mieux nous faire comprendre de quelle manière se brancher ce type de composant.

Nous avons réussi, après bons nombres d'heures, à brancher le module de manière à ce qu'il soit bien alimenté. Aucune LED ne le confirmait et ce fut assez gênant mais l'appareil était bien présent en source Bluetooth via le smartphone. Après programmation de la carte Arduino, le module arrivait à envoyer des informations comme un texte, même si celui-ci n'a jamais correspondu à celui initial. A titre d'exemple, un simple phrase du type «Connexion établie », à afficher toutes les 3 secondes, se traduisait par un simple caractère changeant à chaque fois. A cela s'est ajouté le fait que même si le module envoyait des données, il n'en recevait aucune.

Nous en venons donc au second problème qui est l'absence totale d'informations concernant ce module sur internet. En effet, en dehors du site du constructeur, il n'existe ni exemple d'utilisations ou question/réponse à son sujet sur des forums. A cela, s'ajouta que notre professeur lui-même ne savait pas pourquoi cela ne marchait pas. Nous avons décidé alors après une perte de temps très conséquente sur les heures de cours comme de travail personnel, de commander un nouveau module Bluetooth. Le choix fut simple : prendre un module populaire afin d'avoir de l'aide en cas de problème. Et notre choix se porta sur le HC-60.

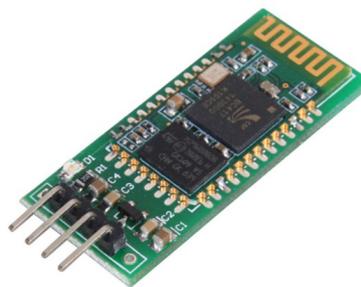


FIGURE 2.5 – Photo d'un module bluetooth HC-60.

Comme le suggère la photo, il n'y a que 4 ports à cette carte et les branchements sont effectués de cette manière :

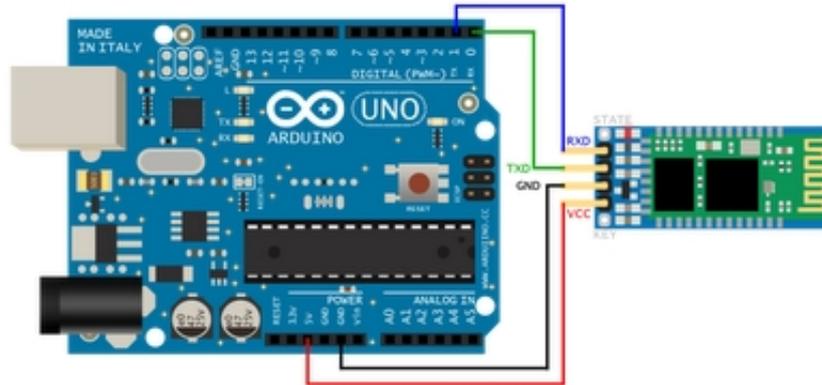


FIGURE 2.6 – Capture de notre réalisation sous le logiciel Fritzing.

De la même manière que le capteur pour le Vcc et le GND. Il faut faire attention au branchement des deux autres sorties, en effet, le Rx du module Bluetooth se branche sur le Tx de l'Arduino et vice versa.

Le gros avantage de ce module, en plus d'être très simple d'utilisation en comparaison du précédent, est qu'une LED s'allume lorsque celui-ci est mis en tension et clignote. Le clignotement s'arrête lorsque le module est connecté en Bluetooth à un appareil. La LED est à ce moment allumée constamment. Aucun problème ne fut rencontré avec ce module et nous purent l'utiliser au bout de seulement quelques minutes, frustrant au vu du nombre d'heures passées sur le précédent.

Pour finir, les moteurs mais aussi ce qui permet de réguler ces derniers, le pont H. Celui-ci contrôle la polarité des dipôles et permet ainsi d'inverser le sens de rotation d'un moteur lorsqu'on lui demande. Le nôtre est réalisé à l'aide d'un circuit imprimé :

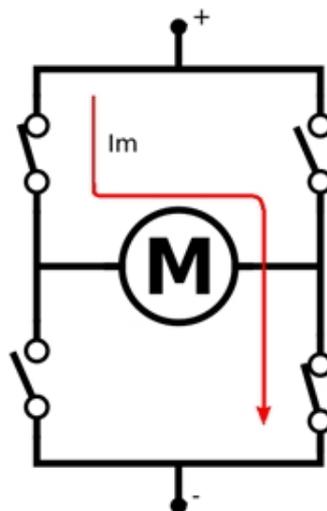


FIGURE 2.7 – Schéma d'un pont H.

2.2 Réalisation du circuit imprimé

Ce travail de conception du circuit imprimé a été réalisé par Pierre et Clément. Il fut tout d'abord nécessaire de sélectionner le logiciel qui servirait à modéliser la carte électronique avec le routage et le schéma électrique. Nous avons choisi Fritzing. En effet, ce logiciel correspondait exactement à nos besoins car il contient une grande bibliothèque de composants et qu'il est entièrement gratuit.

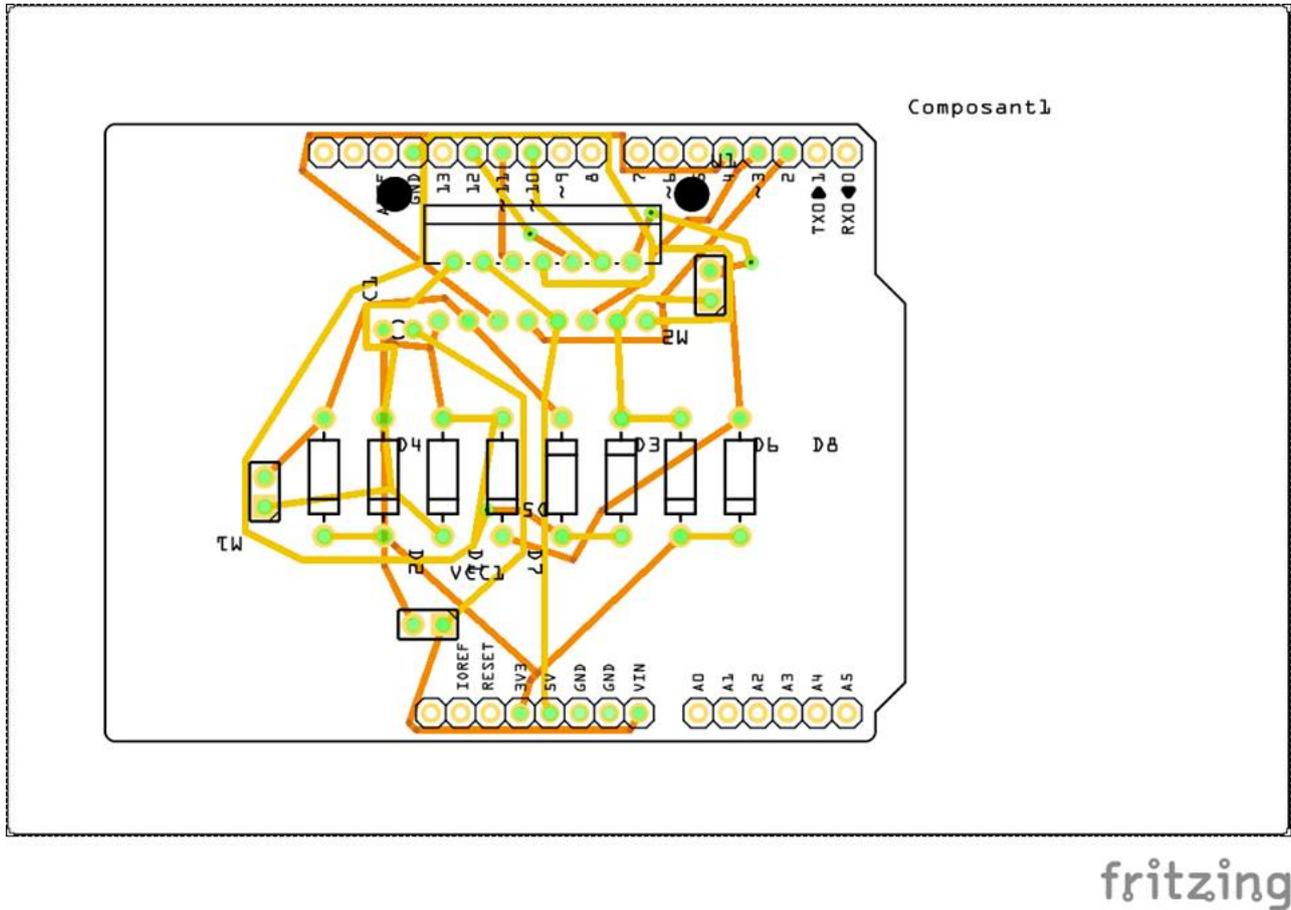


FIGURE 2.8 – Capture de notre réalisation sous le logiciel Fritzing.

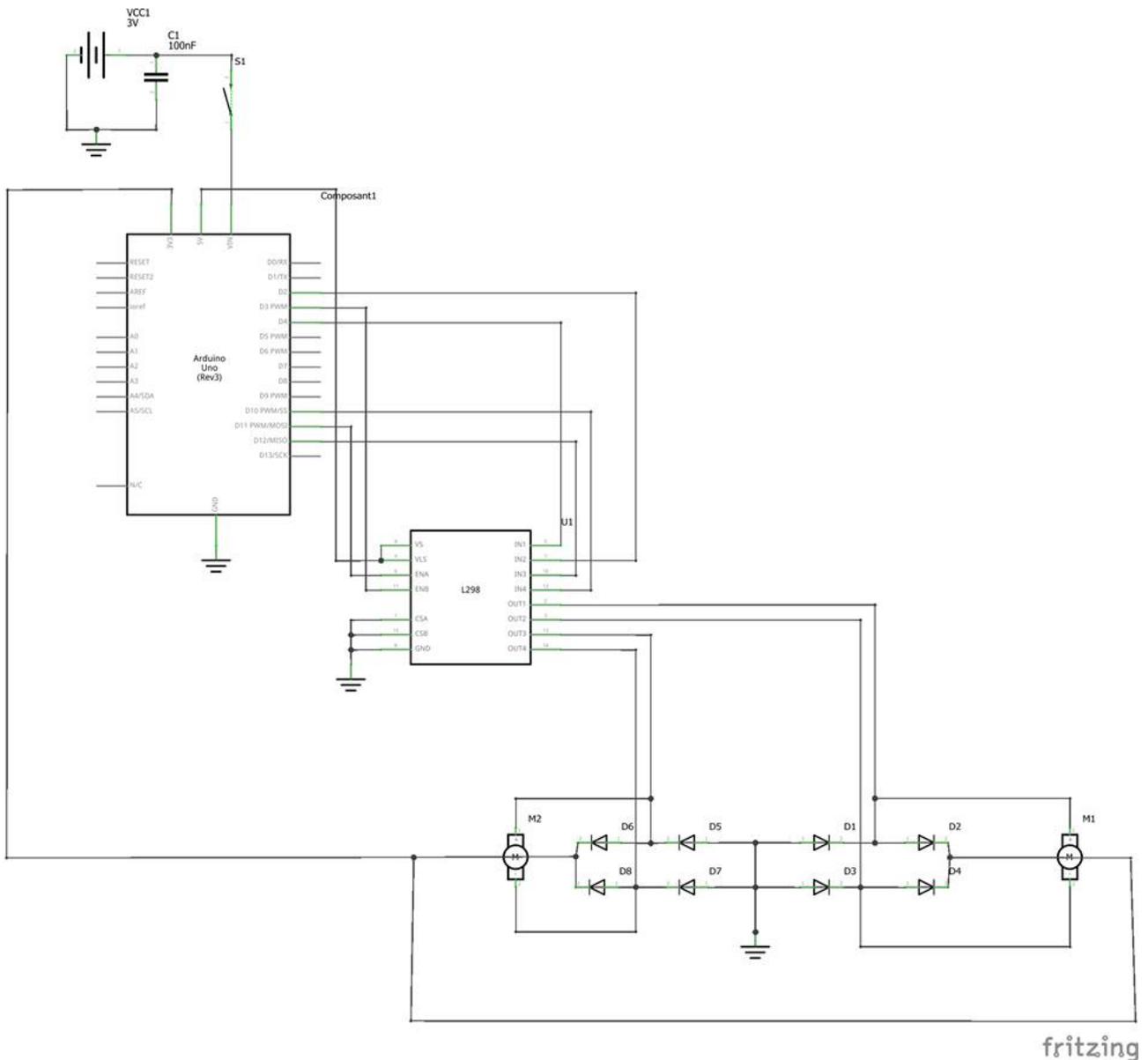


FIGURE 2.9 – Capture de notre réalisation sous le logiciel Fritzing.

Il fallait tout d'abord commencer par réaliser le schéma électronique. Nos connaissances ne nous permettant pas de le réaliser directement, nous avons dû faire des recherches et étudier les schémas réalisés les années précédentes. Une fois les éléments choisis, il fallait les trouver dans la bibliothèque. Une fois tous les composants téléchargés, nous avons dessiné le schéma électrique de la carte en reliant les différents composants. Il fallait aussi ajouter le module arduino et relier correctement les différentes broches en fonction de la programmation. Nous devons placer tous les composants dans un cadre que nous devons définir en fonction de la taille réelle de la future carte. Nous avons ensuite réalisé le routage des composants, automatique avec Fritzing.

Lorsque nous avons reçu la carte, imprimée par notre enseignant, nous avons pu commencer le perçage pour les broches des composants et le soudage. Il fallait faire attention aux courts circuits. Pour cela, chaque endroit où les pistes étaient trop rapprochées et se touchaient, a été « gratté » afin d'enlever le cuivre entre les deux pistes. Après avoir revu les différentes modifications et corrigé les problèmes occasionnés, nous nous sommes occupés d'ajouter la pile et les moteurs, extérieurs à la carte.

2.3 Programmation

2.3.1 Conception programme

Pour commencer, pour programmer notre carte Arduino, nous avons utilisé le logiciel développé par Arduino et prévu à cette effet.

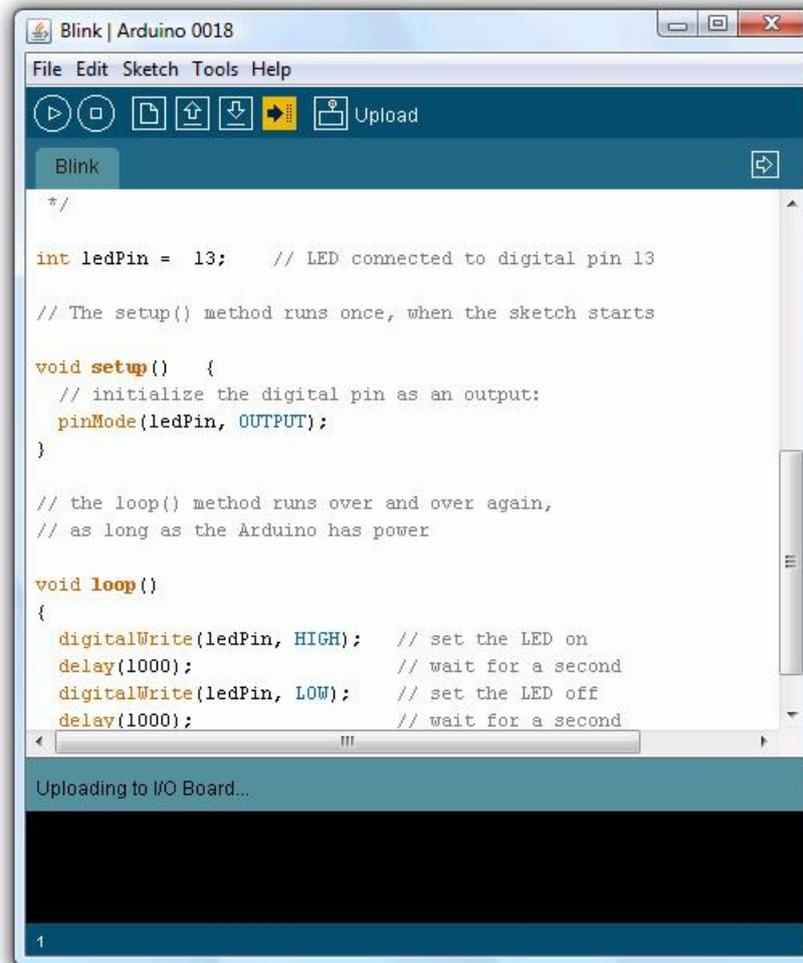


FIGURE 2.10 – Capture du logiciel de programmation sur Arduino.

Ce logiciel est disponible sur différents systèmes d'exploitation et, tout comme sa carte, est simple d'utilisation et facile à comprendre malgré que le codage se fasse en C et qu'aucune formation ne nous avait été donné dans ce langage. Pour « discuter » avec la carte, il suffit de la relier via USB à l'ordinateur. Ensuite, nous pouvons mettre un programme sur la carte via la commande « téléverser ». Il faut cependant savoir qu'un seul programme peut être mis sur la carte. Ainsi, à chaque téléversement, le programme mis précédemment sur la carte est effacé.

Pour tester la carte, nous avons utilisé un programme contenu dans la grande bibliothèque fournie avec le logiciel (présent sur la photo précédente) qui permet d'allumer une LED lorsque l'on appuie sur « 1 » ou « 0 ». Le moniteur série du logiciel permet de tester les programmes directement.

2.3.2 Code BASIC du robot

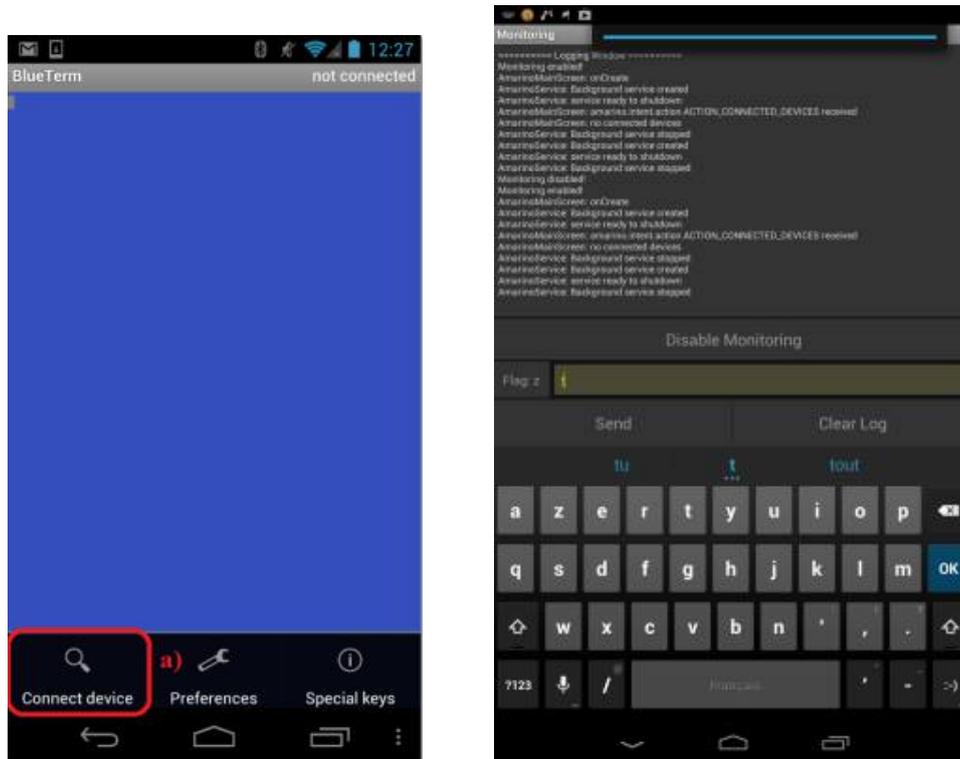


FIGURE 2.11 – Captures de l'application téléphonique *BlueTerm*.

Premièrement, il faut savoir que le branchement du module Bluetooth ne change rien au programme, ainsi, « Blink » marchait toujours aussi bien, à la différence que les ordres étaient donnés par un smartphone grâce à des applications comme *Amarino* ou *Blueterm*.

Deuxièmement, au vu de difficultés rencontrées avec le module Bluetooth, nous avons commencé à programmer le capteur thermique.

Pour finir, le programme de pilotage du robot. Celui-ci est assez basique dans l'ensemble. Nous commençons par définir les pins de l'arduino utilisés par le moteur via `#define X 6` où le X représente un nom voulu par l'utilisateur. Ensuite, nous les activons en tant que sortie via la commande « `pinMode(X , OUTPUT)` ».

Le reste est très simple, nous formons une boucle avec un « case ». Si la carte reçoit un chiffre de 1 à 5, une des 5 fonctions attribuées se déclenche. Celles-ci correspondent à faire avancer, reculer, tourner à gauche et à droite et arrêter le robot.

Il ne reste plus qu'à assembler le programme du capteur et celui de contrôle du robot.

2.3.3 Conception de l'application

Pour pouvoir créer l'application, AppInventor2 a été utilisé. Cette application permet à la fois aux débutants de commencer à créer leur propre application pour smartphone, mais permet aussi aux plus confirmés de manipuler des fonctionnalités plus complexe.

Dans un premier temps, il faut créer son application en ajoutant des boutons, des composants,... sur l'écran d'accueil.



FIGURE 2.12 – Vue de l'interface de création d'application.

Après cette étape, il faut connecter tous les éléments afin d'obtenir un programme fiable et opérationnel. Grâce à AppInventor, pas besoin d'avoir de solides connaissances en Java : les fonctions sont déjà préparées et l'utilisateur n'a besoin que de les rassembler pour créer son application. Voici quelques étapes de cette conception :

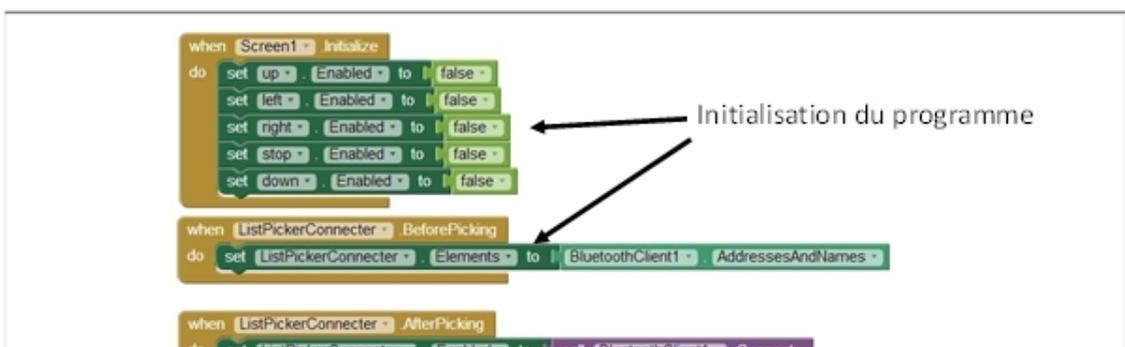


FIGURE 2.13 – Capture de l'interface de création d'application.

Ici on initialise toutes nos fonctions et on permet à l'utilisateur de voir toutes les adresses Bluetooth disponibles.



FIGURE 2.14 – Capture de l'interface de création d'application.

Dans cette deuxième étape, l'utilisateur choisit l'appareil avec lequel il désire se connecter et dès que la connexion est établie, un bandeau de notification affichera « connecté » et les commandes de déplacements seront déverrouillées (grâce aux « true »).



FIGURE 2.15 – Capture de l'interface de création d'application.

Maintenant, il est temps d'envoyer des instructions à notre carte Arduino : de manière très simpliste, seul des chiffres sont envoyés, et la carte réagit spécifiquement à chaque donnée.

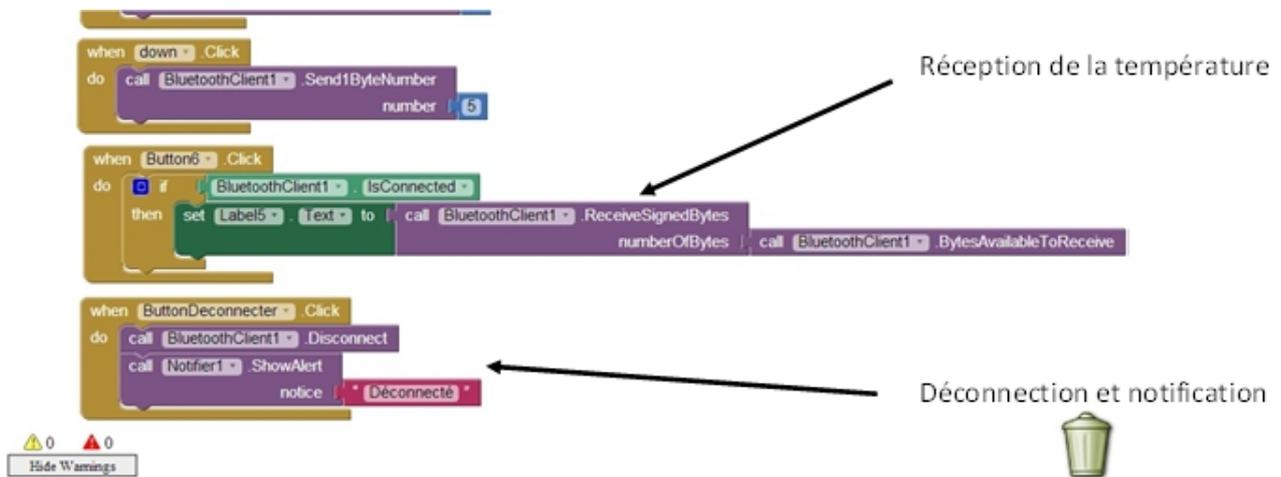


FIGURE 2.16 – Capture de l'interface de création d'application.

Dans cette dernière partie, grâce à « `ReceiveSignedBytes` », le programme lit les nombres envoyés par l'Arduino, nous avons choisi `SignedNumber` car cela permet de recevoir des températures négatives (au contraire de « `UnsignedBytes` ». Il suffit juste alors d'afficher les valeurs obtenues dans une petite case « `label5` ». Pour finir, nous avons incorporé une procédure de déconnexion qui désactive les commandes de contrôle et affiche une notification.

2.3.4 Evolution possible

Nous avons aussi créé une autre application pour notre robot : au lieu de commander le robot par des pressions sur des touches « haut », « bas », « gauche », « droite » l'autre option était de le commander grâce à l'accéléromètre du smartphone. Le fonctionnement de cette fonctionnalité est assez simple : lors de test sur le smartphone, l'accéléromètre du téléphone renvoie 2 variables : X et Y :

- Quand X est positif, l'appareil est penché vers l'avant.
- Quand X est négatif, l'appareil est au contraire penché vers l'arrière.
- Quand Y est positif, l'appareil est tourné vers la droite.
- Quand Y est négatif, l'appareil est dirigé vers la gauche.

Dans ces conditions nous avons créé l'application suivante :

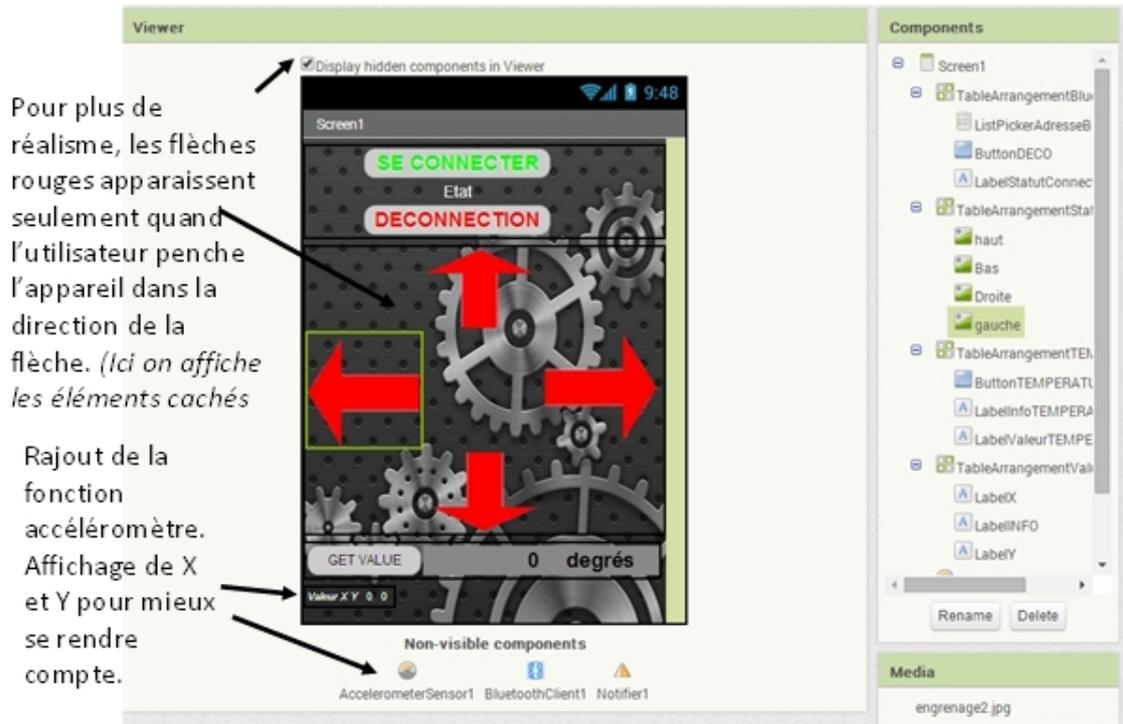


FIGURE 2.17 – Capture de notre application.

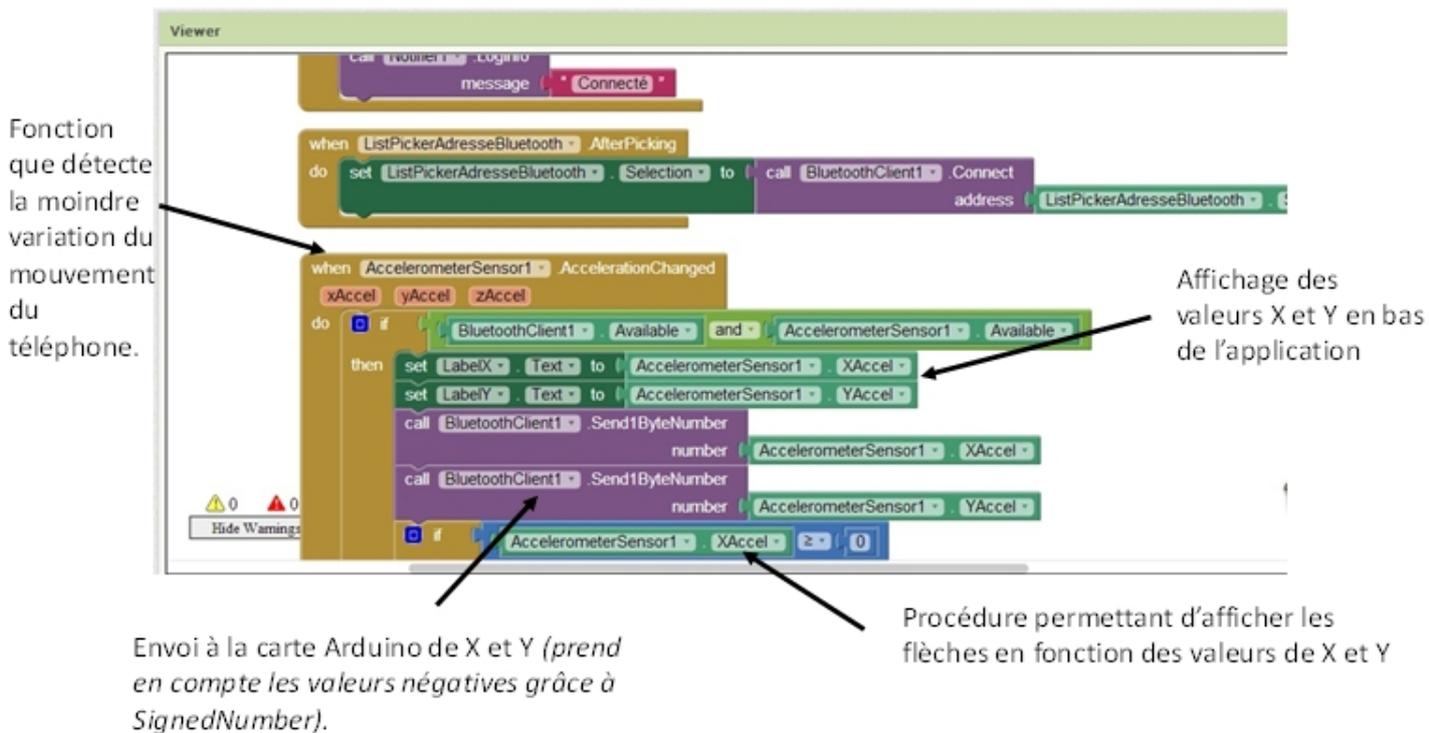


FIGURE 2.18 – Capture de l'interface de création d'application.

Pour finir : nous considérons cela comme seulement une évolution possible de notre robot, puisque nous sommes rapidement tombés sur deux problèmes : comment différencier les deux informations qu'envoie l'application ? Nous avons vainement essayé de nombreuses solutions, mais aucune n'a été satisfaisante jusqu'ici.

Secondement, notre robot pouvait se déplacer dans 4 directions uniquement, mais avec l'ajout de l'accéléromètre, en fonction de comment l'utilisateur penchait, le robot devait se déplacer dans toutes les directions possible. Or nous avons fait le choix au début de notre projet de faire varier la vitesse des moteurs pour faire tourner le robot. Or nous n'avons pas réussi à lier toutes les directions avec toutes les vitesses de moteur possible.

Chapitre 3

Retours et évolution du projet

3.1 Résultats

Le robot fonctionne en partie, la communication entre le téléphone portable, la carte bluetooth et la carte arduino est bonne. Des tests ont montrés que le problème restant se situait dans le circuit imprimé, mais, même après de nombreuses vérifications nous n'avons pas compris pourquoi la connexion entre la carte arduino et les moteurs ne fonctionnent pas. Nous pensons que le problème vient soit du Pont H soit du système d'alimentation des moteurs, mais, nos connaissances en électronique étant limitées, il nous aurait fallu plus de temps pour le résoudre, puisqu'il aurait fallu redessiner un circuit imprimé. Il aurait été ensuite nécessaire de ressouder tous les composants, effectuer tous les perçages et réintégrer le circuit sur le robot, ce qui aurait pris beaucoup de temps.

3.2 Problèmes rencontrés

Dès le début du projet nous avons dû choisir parmi une large gamme de programmes et de logiciels, des outils qui nous correspondait et nous permettait de mettre en application nos connaissances. Grâce à cela, ce projet fut une expérience particulièrement enrichissante.

Concernant la partie « mécanique » du robot, comme dis précédemment, nous avons utilisé le logiciel Fritzing pour le traçage du circuit imprimé. En effet, au début on nous avait conseillé d'utiliser Kidcad. Mais ce dernier était trop complexe, et de nombreuses bibliothèques n'étaient pas présentes. Alors qu'avec Fritzing, le logiciel était gratuit et les bibliothèques présentes. Par ailleurs la prise en main du logiciel était plus simple.

Concernant la programmation de la carte Arduino, nous nous sommes naturellement tournés vers le logiciel Arduino. Celui-ci permettait de faire le lien entre l'ordinateur et la carte principale du robot. Malheureusement l'un des principaux problèmes de cette application fut le manque de compatibilité entre le logiciel et les ordinateurs de l'INSA et les nôtres. Il a fallu donc effectuer de nombreuses manipulations pour pouvoir l'utiliser. Une autre difficulté s'est présentée à nous : un nouveau langage de programmation. Pour comprendre les bases de celui-ci et les mettre en application, il nous a fallu beaucoup de temps, mais ça nous a permis d'enrichir nos connaissances en informatique.

Après avoir essayé de nombreux logiciels permettant de développer des applications sous Android, AppInventor2 s'est imposé comme l'outil le plus simple et le plus pratique. Car nous avons découvert que le langage prédominant sur les smartphones était le Java. Et c'était encore un nouveau langage à apprendre. Mais AppInventor2 est un logiciel destiné aux débutants désireux créer des applications, qui facilite grandement l'apprentissage du Java quand on connaît déjà le principe général de programmation sans connaître le nom et la syntaxe de celui-ci.

Conclusions et Perspectives

Conclusion générale

Pour conclure, ce projet nous a permis, d'une part, d'améliorer grandement nos compétences scientifiques, que ce soit du côté programmation par l'apprentissage de nouveaux langages, ou du côté électronique par la conception d'un circuit imprimé. D'autre part ce projet nous a permis de nous mettre en situation en tant qu'ingénieur puisque nous avons dû travailler en groupe que ce soit pour la répartition du travail, la conception du robot ou résoudre les nombreux problèmes rencontrés.

Malgré les lacunes restantes de notre robot nous sommes satisfaits du résultat, nous sommes conscients que de nombreuses améliorations pourraient être apportées, notamment sur le plan esthétique, mais nous sommes parvenus à résoudre de nombreux problèmes en travaillant de concert.

Conclusions personnelles

Pierre

Personnellement, ce projet m'a beaucoup apporté car il m'a permis de découvrir quelque chose de différent. En effet, en terminale j'ai eu l'occasion de réaliser les PPE (projets personnalisés encadrés, seulement pour la filière scientifique - sciences de l'ingénieur). Il était là aussi question de réaliser un travail similaire (partie mécanique, partie électronique et partie programmation) mais cela restait théorique et beaucoup de problèmes ont été négligés car ils n'empêchaient pas la réalisation virtuelle du projet. Avec ce projet de physique, j'ai pu m'apercevoir de la difficulté du travail de groupe et de tous les problèmes que nous pouvons rencontrer dans certaines situations (lors de la réalisation de la carte, de la programmation,...). De plus, j'ai pu découvrir de nouveaux logiciels et acquérir de nouvelles connaissances.

Clément

Ce projet fut une expérience enrichissante, j'ai pu améliorer mes compétences en latex et en électronique. La conception d'un robot de A à Z fut très intéressante, malgré un manque de connaissance évident qui nous a fait perdre énormément de temps. J'aurais beaucoup aimé continuer ce projet qui offre de très nombreuses possibilités d'amélioration, contrairement à d'autres projets menés dans des années précédentes. J'ai beaucoup apprécié la grande liberté qui nous a été donnée, et qui nous a vraiment permis de travailler en tant qu'équipe de futurs ingénieurs où chacun a pu apporter ses connaissances propres

Guillaume

Ce projet fut pour moi un réel apport de connaissances et de méthodes : comme spécifié dans le rapport, nous avons fait face à un très grands nombre de problèmes : sitôt un de résolu, un nouveau apparaissait. Mais grâce à cela, j'ai appris à gérer ces situations parfois déprimantes : car à force de travail, les solutions apparaissaient toujours, ce qui était à chaque fois une réelle satisfaction. Et en dépit des difficultés, ce projet m'a énormément appris sur la création d'applications, la gestion d'une communication sans fil et sur l'utilisation d'une carte microcontrôleur. Par ailleurs, le côté humain de cette aventure fut particulièrement intéressant : comme personne n'avait de connaissance, l'aide mutuelle fut importante et nous permit d'avancer.

Etienne

Personnellement ce projet m'a beaucoup apporté surtout dans l'organisation d'un travail en groupe, en effet dans notre groupe tout le monde n'avait pas les mêmes qualités ou points forts, c'est pourquoi il était important d'exploiter le potentiel de chacun pour mener a bien chaque partie de notre projet. En plus de cela j'ai pu apprendre à faire face à des situations très compliqués qui ont étaient très nombreuses durant ce projet. J'ai aussi découvert le domaine de la robotique et plus particulièrement en Arduino. De plus, voulant me diriger vers le département ASI, j'ai pu apprendre à coder en C pour faire fonctionner le robot et cela va donc sûrement me servir dans mes années futures. Pour conclure je dirai que ce projet fut bénéfique pour moi que ce soit du côté de l'organisation ou de la technique.

Geoffrey

Mon choix du projet s'est très rapidement tourné vers la robotique. Univers que je ne connaissais pas et qui m'attrayais. En effet, ce domaine en mêle d'autre à savoir la mécanique, la programmation et l'électronique. Ce dernier m'intéressant particulièrement et je ne fus pas déçu. L'arduino offre d'innombrables possibilités, et l'on peut facilement réinvestir les connaissances acquises dans sa vie personnelle. De plus, le travail de groupe est toujours appréciable et permet de se familiariser avec une méthode de travail qui sera majoritaire dans notre avenir d'ingénieur. A ceci s'implante un travail personnel d'autant plus facile quand l'envie y est, par le fait supplémentaire que ce n'est pas un enseignement théorique comme la majorité des cours. Pour résumé, ce projet est donc très satisfaisant.