

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Pour l'obtenir du diplôme master en Électronique
OPTION : SYSTÈMES DE VISION ET ROBOTIQUE (SVR)

présenté par

ROUBEHIE MOHAMED NABIL

&

HASSENI MOHAMED

Commande d'un robot mobile via une application Android

Proposé par : DAHMANI SAMIR

Année Universitaire 2015-2016

Remerciements

Nos premiers remerciements sont adressés à nos enseignants et enseignantes à Université Saad Dahleb Blida 1 département d'électronique spécialité « SYSTÈMES DE VISION ET ROBOTIQUE » pour leurs contributions à notre formation en master

Ensuite, bien sûr, Nous remercions notre encadreur Monsieur « DAHMANI SAMIR », Enseignant au sein de L'université Saad Dahleb Blida 1, pour son encadrement, ses Recommandations précieuses et sa disponibilité, ainsi que tous ceux qui ont participé de près ou de loin à la réalisation de ce mémoire

Nous remercions l'ensemble de jury qui a bien voulu examiner et évalué ce mémoire.

ملخص: أصبحت **الإلكترونيات** تحيط بنا من كل جانب, ستجدها ترافقك أينما ذهبت في السيارة في بيتك في العمل أو حتى في جيبك الصغير (الهاتف...), هناك دائما أداة إلكترونية ذكية تتفاعل معك في أي مكان تذهب إليه, **أردوينو** هو مشروع كبير مفتوح المصدر يهدف إلى توفير برمجيات مجانية و لوحة تطوير تفاعلية مفتوحة المصدر تستخدم في بناء دوائر إلكترونية ذكية و تستطيع التفاعل مع البشر بسهولة و يمكن لأي شخص إستخدامها لعمل مشاريعه الخاصة دون الحاجة لمعرفة مسبقة بعلم الإلكترونيات و دون دراسة تعقيدات الدوائر الإلكترونية, و تتكون هذه اللوحة من دائرة إلكترونية تحتوى على **متحكم دقيق** قابل للبرمجة عن طريق برنامج بيئة التطوير المتكاملة للأردوينو و الذي يمكن تحميله مجانا لجميع أنظمة التشغيل, و بفضل أردوينو توصلنا في مشروعنا إلى ابتكار "روبوت" يستطيع تجنب العقبات ويمكن التحكم فيه بواسطة تطبيق في **نظام الأندرويد** .

كلمات المفاتيح: الإلكترونيات؛ أردوينو؛ نظام الأندرويد؛ متحكم دقيق؛ روبوت؛ العقبات.

Résumé : Aujourd’hui **l’électroniques** nous entoure de tous les côtés, tu trouveras qu'elles t'accompagnent là où tu vas dans ta voiture, dans ta maison, au travail et même dans ta petite poche (smartphone...), il y’a toujours un outil électronique intelligent qui réagit avec toi dans n’importe quel endroit, **Arduino** est un grand projet open source et une plateforme de développement interactive open source ,constitué d’un **microcontrôleurs** d’une commande précise pour la programmation à travers IDE (environnement de développement intégré) ,grâce à ARDUINO nous somme arrivé dans notre projet a la création d’un "**robot**" qui peut éviter tout **obstacle** et est commandé à travers une application du **système androïde**.

Mots clés : électroniques ; arduino ; obstacle ; robot ; microcontrôleurs ; système androïde.

Abstract : **Electronics** become interactive surround us from every side, accompany you'll find wherever you go in your car at home in work or even in your small pocket(phone...), there is always a smart electronic tool interact with you anywhere you go, **Arduino** is a great open source project and Plate interactive development of open source, and the painting is made up of the cycle of email with **microcontroller** accurate programmable through the Arduino IDE program for the Arduino and which can be downloaded free of charge to all operating systems, in our project and with merit Arduino we try to build machine "**robot**" ,it can avoiding **obstacles** and can be controlled by the application in **Android system**.

Keywords: Electronics; Arduino; obstacles; robot; microcontroller; system android.

Introduction générale

Introduction générale

La forte augmentation des ventes de smartphone et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies ainsi que l'autopilotage. Au fond, le smartphone, avec sa connectivité Bluetooth intégrée, devient une télécommande universelle pour les équipements électriques et électronique. Les utilisateurs pourront à terme contrôler à distance un très grand nombre de fonctions sans avoir à tenir compte de la marque ou de l'origine du produit qu'ils pilotent. Pour répondre à cette évolution majeure, nous avons créé une application androïde qui permet de contrôler un robot mobile sa mobilité est intelligente car il peut éviter les obstacles le contrôle de ce robot s'effectuera à l'aide d'un smartphone ou à une tablette mais aussi un point majeur le contrôle de ce robot peut s'effectuer a une grande distance.

Dans la vie moderne, on utilise pas mal d'outils et d'accessoires de commande à distance afin de simplifier notre mode de vie, donc nous chercherons toujours à se concentrer sur la souplesse de la commande et de contrôler sur une zone bien définie le plus grand nombre possible d'accessoires.

Le smartphone occupe la première place d'objets qui ne nous quittent pas donc notre travail se concentre sur l'utilisation de ce dernier avec bien sur sa liaison avec un système tel que l'Arduino pour réaliser notre projet commander un robot mobile qui détecte les obstacles et peut les éviter par un smartphone de system android.

1.1 Introduction

La robotique peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles.

Pour concevoir, simuler ou commander un robot, il est nécessaire, entre autres, de disposer de modèles du mécanisme. Plusieurs niveaux de modélisation sont possibles. Ils dépendent des spécifications du cahier des charges de l'application envisagée : il en découle des modèles géométriques, cinématiques et dynamiques à partir desquels peuvent être engendrés les mouvements du robot, ou bien des modèles statiques qui décrivent les interactions du mécanisme avec son environnement.

L'obtention de ces différents modèles n'est pas aisée, la difficulté variant selon la complexité de la cinématique de la chaîne articulée. Entrent en ligne de compte le nombre de degrés de liberté, le type des articulations mais aussi le fait que la chaîne peut être ouverte simple, arborescente ou fermée. [1]

1.2 Définitions

Le Petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable. En fait, l'image que chacun se fait d'un robot est généralement vague, souvent. [2]

un robot est défini comme un manipulateur automatique à cycles programmables. Pour « mériter » le nom de robot, un système doit posséder une certaine flexibilité, caractérisée par les propriétés suivantes :

- La versatilité : Un robot doit avoir la capacité de pouvoir exécuter une variété de tâches, ou la même tâche de différente manière ;
- L'auto-adaptativité : Un robot doit pouvoir s'adapter à un environnement changeant au cours de l'exécution de ses tâches.

L'association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant

un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs, bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel.

1.3 Historique de la robotique

Le terme « robot » est introduit en 1920 par l'écrivain tchèque Karel Čapek dans sa pièce de théâtre RUR (Rossum's Universal Robots). Ce terme, provenant du tchèque robot, « travail forcé », désigne à l'origine une machine androïde capable de remplacer l'homme dans toutes ses tâches. Dans les années 1940 et 1950, les progrès de l'électronique permettent de miniaturiser les circuits électriques (inventions du transistor et du circuit intégré), ouvrant ainsi de nouvelles voies à la fabrication de robots. Dans les premiers temps de la robotique, le robot est considéré comme une imitation de l'homme, aussi bien fonctionnelle que physique. [3]

La robotique est passée par plusieurs générations comme suit :

- 1947 : Premier manipulateur électrique télé-opéré.
- 1954 : Premier robot programmable.
- 1961 : Utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une chaîne de montage de General Motors.
- 1961 : Premier robot avec contrôle en effort.

Chapitre 1 Les robots

- 1963 : Utilisation de la vision pour commander un robot.
- 1967 : Naissance du premier robot mobile électrique "Shakey" par l'Institut de Stanford. Il est très encombrant et calcule longuement avant d'opérer un mouvement (une heure environ !).
- 1973 : Wabot-1, premier robot humanoïde bipède est créé par les chercheurs de l'université de Waseda à Tokyo. Il peut voir, parler, marcher toucher et prendre des objets. Son intelligence équivaut à un enfant de un an et demi.
- 1978 : Le robot ARGOS. Développé à l'Université Paul Sabatier de Toulouse (France). Le robot ARGOS simule la navigation d'un robot mobile équipé d'un système de vision au fur et à mesure de ses déplacements.
- 1979 : Le robot HILARE. Les chercheurs du L.A.A.S. de Toulouse (France) étudièrent la planification des trajectoires d'un robot mobile ponctuel, dans un environnement totalement connu.
- 1981 : Le robot VESA. Ce robot, construit à l'I.N.S.A (France). De Rennes, est équipée d'un arceau de sécurité pour réaliser la détection d'obstacles dans un environnement totalement inconnu.
- 1984 : Le robot FLAKEY. Ce robot, conçu et construit au Stanford Research Institute et le reflet des améliorations apportées par 14 années de développement. Le robot FLAKEY est équipé de deux roues motrices avec encodeurs, mais sa vitesse maximale est de 66 cm/s au lieu de quelques centimètres par seconde. Ce robot est capable de naviguer dans des environnements réels.
- 1993 : Les robots ERRATIC et PIONNER. Le robot ERRATIC a été conçu par Kurt Konolige, au Stanford Research Institute, comme un robot mobile de faible coût pour ses cours de robotique.

Chapitre 1 Les robots

- 1999 : Aibo, le chien-robot voit le jour. Il est capable de se déplacer, reconnaître son environnement, les commandes vocales. Son propriétaire peut lui apprendre des tours, son intelligence grandit au fil du temps.
- 2003 : Le robot Spirit est envoyé sur Mars par la NASA. Équipé de panneaux solaires, il a étudié le sol et les roches martiennes pendant six ans. Ce robot a parcouru 7,7 km au lieu des 600m prévus.
- 2005 : Financé par le ministère de la Défense des États-Unis, des chercheurs de Boston (entre autres) créent un robot à quatre pattes appelé BigDog.

Il est destiné à aider les soldats pour transporter du matériel (jusqu'à 150 kg) et faire des opérations d'observations. BigDog est capable de s'adapter aux terrains les plus irréguliers et se redresser en cas de chute. Il a déjà été utilisé en Afghanistan en 2009.

- La *Figure 1.1.* montre quelques robots d'ancienne génération :

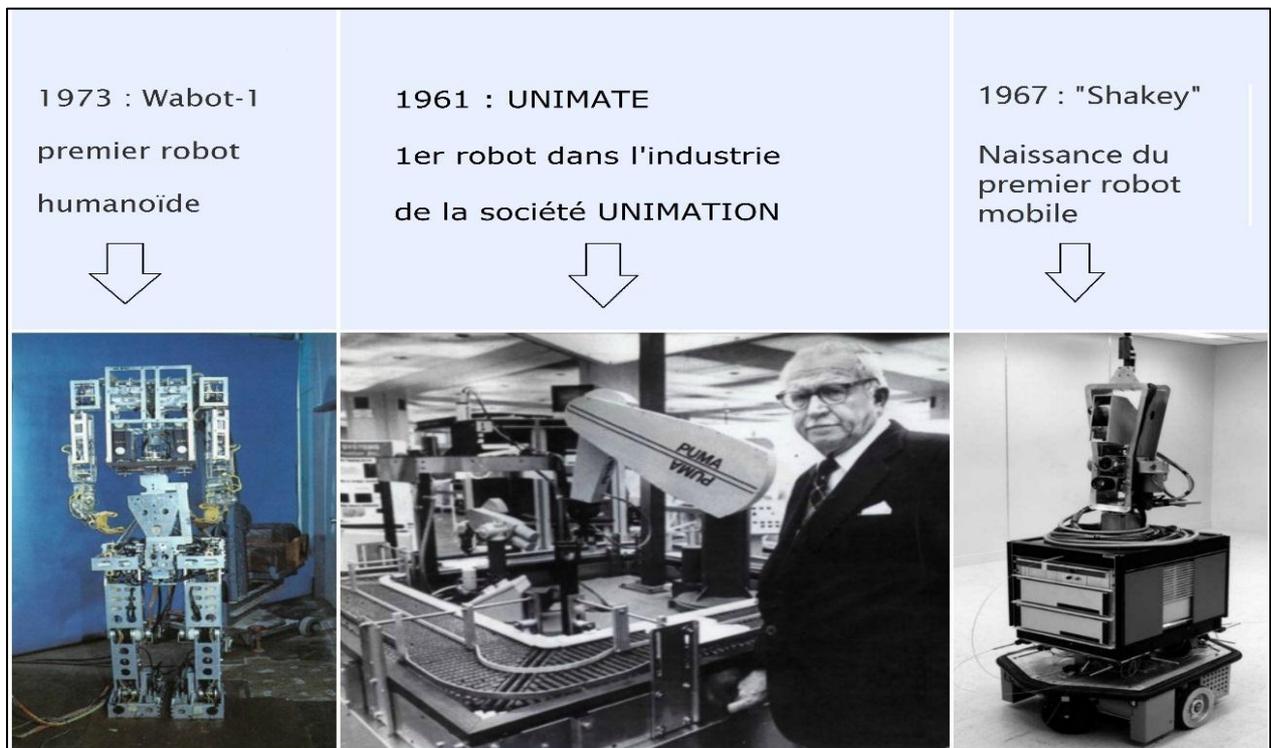
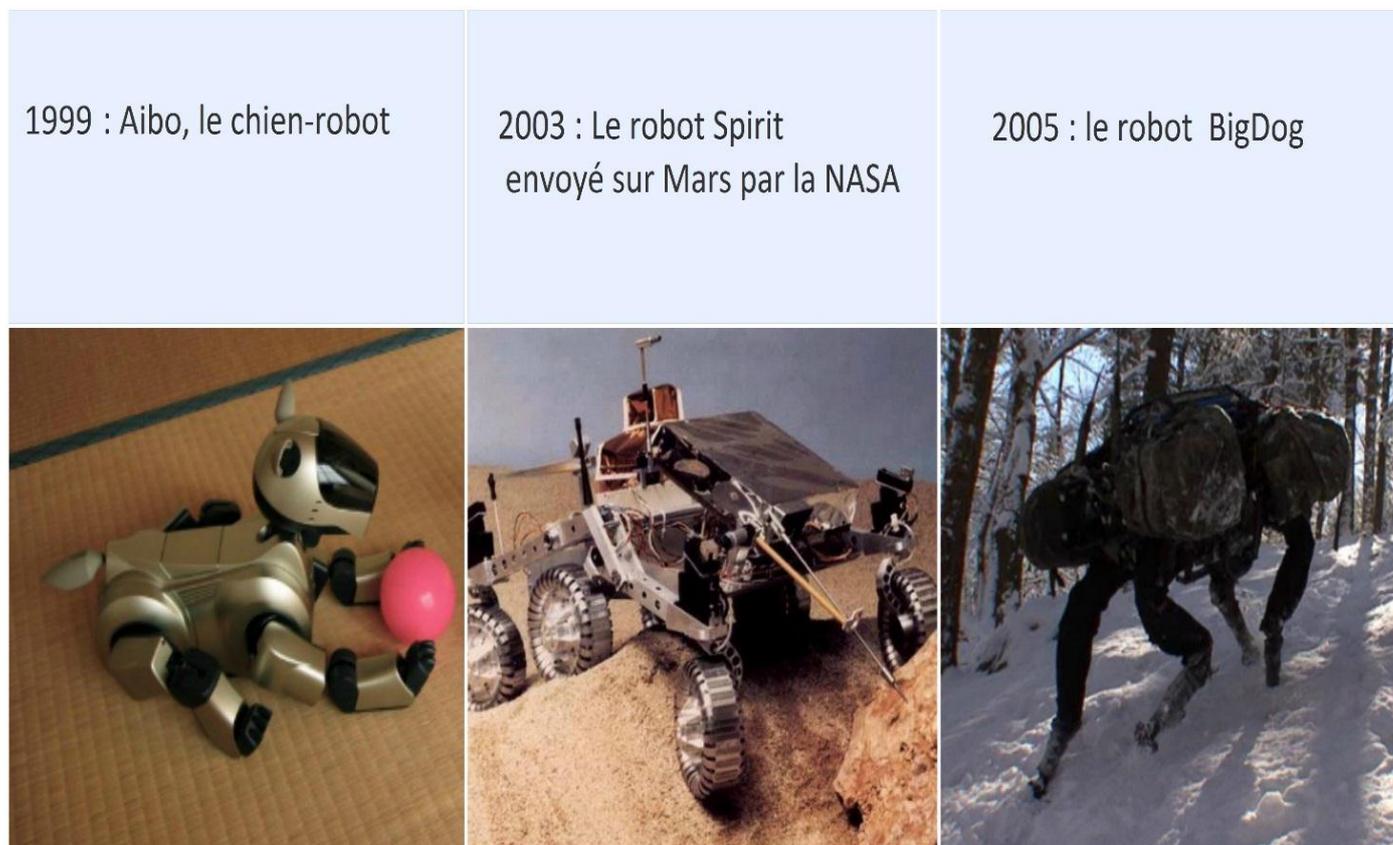


Figure 1.1. Robot wabot-1, robot unimate et robot shakey.

Chapitre 1 Les robots

La **Figure 1.2**.montre quelques robots de la nouvelle génération :



*Figure 1.2.*Robot aibo, robot spirit et robot bigdog.

1.4 Les trois lois de la robotique :

- **Première loi** : « Un robot ne peut porter atteinte à un être humain ni, restant passif, laisser cet être humain exposé au danger.»
- **Deuxième loi** : « Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la première loi. »
- **Troisième loi** : « Un robot doit protéger son existence dans la mesure où cette protection n'est pas en contradiction avec la première ou la deuxième loi.»

1.5 Structure d'un robot :

Un robot est un dispositif susceptible d'accomplir une ou plusieurs fonctions. Il se compose de sept parties principales : le socle muni d'un ou de plusieurs bras articulés, équipés de systèmes de préhension (pinces, ventouses, etc.), la source d'énergie, les actionneurs, les systèmes de transmission, les capteurs internes, les capteurs externes (de position, de température, etc.), et le système de traitement de l'information.

L'énergie peut être utilisée sous forme pneumatique, hydraulique ou électrique, cette dernière étant préférée par les utilisateurs et les constructeurs de robots déplaçant des charges inférieures à 1 000 kg. Les actionneurs sont des dispositifs capables de modifier le fonctionnement du robot, en gérant les transferts d'énergie à partir des signaux émis par le système d'automatisation. Ils se présentent généralement sous la forme de systèmes d'asservissement (voir servomécanisme).

Les capteurs internes et externes permettent de mesurer des grandeurs mécaniques, thermiques, électriques ou chimiques (température, position, concentration chimique, etc.). Les valeurs ainsi mesurées sont converties en signal électrique, et transmises à un autre dispositif — électronique — pouvant lire, enregistrer ou contrôler les grandeurs mesurées.

1.6 Types de robots :

On peut classer les robots en quatre catégories, par ordre de complexité croissante. Les robots les plus simples et les plus courants se contentent de répéter les opérations inscrites dans leurs programmes. Les robots appartenant au deuxième type sont capables de reproduire certains mouvements humains enregistrés sur bande magnétique, tandis que les robots de troisième génération sont des dispositifs à commande numérique, dont les mouvements sont enregistrés sur une unité de stockage et qui peuvent accomplir plusieurs opérations d'usinage différentes. Enfin, les robots évolués — les plus complexes —, faisant appel à des capteurs d'environnement et aux techniques d'intelligence artificielle,

Sont dotés d'une grande capacité de traitement de l'information. Certains d'entre eux peuvent ainsi reconnaître la forme d'objets sur un écran de télévision, ou encore réagir à des paroles employant un vocabulaire spécifique (voir reconnaissance vocale).

1.7 Fonctionnement des robots évolués :

On peut définir un robot évolué comme une machine agissant physiquement sur son environnement en vue d'atteindre un but qui lui a été fixé. Cette machine doit être capable de percevoir son environnement et le cas échéant de s'adapter à certaines variations de celui-ci (capacité d'autonomie). Par exemple, si le but du robot est de se déplacer d'un point à un autre dans une pièce, il doit être capable de contourner les obstacles qui y sont présents. Il lui faut donc assurer des fonctions de perception, de décision et d'action. Pour ce faire,

Un robot est un mélange de mécanique (bras articulé, système hydraulique, système de déplacement...), d'électronique (capteurs en tout genre, caméras...), d'informatique (micro-ordinateurs, microprocesseurs) et de logiciels. Les robots les plus anciens contiennent beaucoup de mécanique et d'électronique et peu d'informatique et de logiciels. [5]

La part de l'informatique tend à augmenter car c'est elle qui influe sur les capacités de perception et de décision. Ainsi, le cœur du robot peut être vu comme une boucle perception décision action.

1.7.1 Perception :

La perception repose sur un ensemble de capteurs plus ou moins sophistiqués. Les robots sont de plus en plus équipés de caméras CCD, qui font office d'interfaces visuelles. Ces capteurs fournissent un ensemble de mesures de bas niveau que le robot va devoir être capable d'interpréter à plus haut niveau pour préparer la prise de décision. L'interprétation est très complexe et met en jeu des outils mathématique

Somatiques de haut niveau (c'est le cas par exemple de la vision par ordinateur). La qualité de perception détermine ainsi les possibilités d'évolution d'un robot.

1.7.2 Décision :

Le processus de décision d'un robot prend en compte d'une part, le but qu'il doit atteindre et d'autre part, la perception de son environnement. Cette tâche est directement dépendante de la qualité de la perception. Les modèles de décision peuvent être très complexes et sont fondés sur des outils mathématiques et informatiques très évolués, souvent issus de l'intelligence artificielle. La difficulté majeure de cette opération vient du nombre de l'ensemble des décisions possibles qui peut être très grand et empêche une énumération complète.

La prise de décision s'appuie ainsi sur des heuristiques, qui permettent de déterminer les choix les plus efficaces en fonction de l'objectif visé.

1.7.3 Action

L'action consiste à mettre en application la décision prise. Cela peut être très simple ou complexe en fonction du domaine considéré. Cette fonction reste très liée à la mécanique et à l'électronique, et beaucoup moins à l'informatique.

Interaction des robots et robotique évolutionniste :

De plus en plus, les roboticiens cherchent à concevoir des robots doués de capacités d'interaction. Cette interaction peut se faire avec son environnement physique (perception par le robot et réaction/adaptation), mais aussi avec des humains (robot qui perçoit des intentions humaines, comme par exemple le fait qu'une personne va s'approcher de lui déclenchera les salutations du robot), voire avec d'autres robots (plusieurs robots peuvent coopérer pour accomplir une tâche ou au contraire être en compétition). Ces capacités d'interaction constituent les fonctions les plus complexes d'un robot, car elles nécessitent de fortes capacités de perception (anticipation de mouvements, reconnaissance d'expressions sur un visage ou plus généralement

D'attitudes du corps humain), ainsi que de décision (il faut pouvoir modéliser à la fois le robot, son environnement et les évolutions possibles de chacun d'entre eux).

Parmi les nombreux travaux entrepris dans ces domaines de l'interaction, un axe de recherche majeur se dégage : celui de la sociologie de ce que l'on appelle les « communautés de robots » — groupes de robots que l'on va laisser interagir sur des grandes échelles de temps (plusieurs jours ou semaines) entre eux et/ou avec des humains.

Cette discipline émergente, baptisée robotique évolutionniste, qui tente d'appliquer aux robots les principes d'évolution et de sélection naturelle de Charles Darwin, pose notamment des questions sur la nature même de l'intelligence (voir sciences cognitives) et ouvre la voie à des applications révolutionnaires — principalement dans le secteur des jeux vidéo pour la création de personnages secondaires qui mèneraient une vie autonome.

1.8 Les robots mobiles :

Bien souvent, quand on parle de robotique mobile, on sous-entend robots mobiles à roues. Ce sont en effet les systèmes les plus étudiés, parce qu'ils sont plus simples à réaliser que les autres types de robots mobiles, ce qui permet d'en venir plus rapidement à l'étude de leur navigation. Ce type de robots est notamment très souvent utilisé pour l'étude des systèmes autonomes.

1.8.1 Définition robot mobile :

L'appellation "robot mobile" désigne généralement un véhicule équipé de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome ou semi-autonome dans un environnement complexe, parfois évolutif, partiellement connu ou inconnu, et d'exécuter les tâches programmées sans intervention humaine ou avec une intervention réduite.

Il existe deux principaux modes de fonctionnement pour un robot mobile :

Chapitre 1 Les robots

télé- opéré et autonome. En mode télé-opéré, une personne pilote le robot à distance. Elle donne ses ordres via une interface de commande (joystick, clavier...), et ceux-ci sont envoyés au robot via un lien de communication (internet, satellite ...).

Le robot doit donc obéir aux ordres de l'opérateur qui perçoit l'environnement autour du robot, par différents moyens (camera, radar...), de manière à donner des ordres adaptés au robot. A l'inverse, en mode autonome le robot doit prendre ses propres décisions. Cela signifie qu'il doit être capable à la fois de percevoir correctement son environnement, mais également de savoir comment réagir en conséquence, suivant le niveau d'autonomie. C'est à lui d'envisager son parcours et de déterminer avec quels mouvements il va atteindre son objectif.

Cette notion d'autonomie prise en exemple ci-dessus, que nous pourrions qualifier de décisionnelle, ne doit pas être confondue avec celle d'autonomie énergétique (capacité du robot à gérer efficacement son énergie, à la préserver, voire à se ravitailler), même si ces deux notions sont étroitement liées : idéalement une des préoccupations principales d'un robot mobile totalement Autonome (du point de vue décisionnel), serait en effet de pouvoir gérer de lui-même ses réserves d'énergie. [5]

Cette définition s'illustre par la *Figure 1.3*. classique des interactions d'un robot avec son environnement :

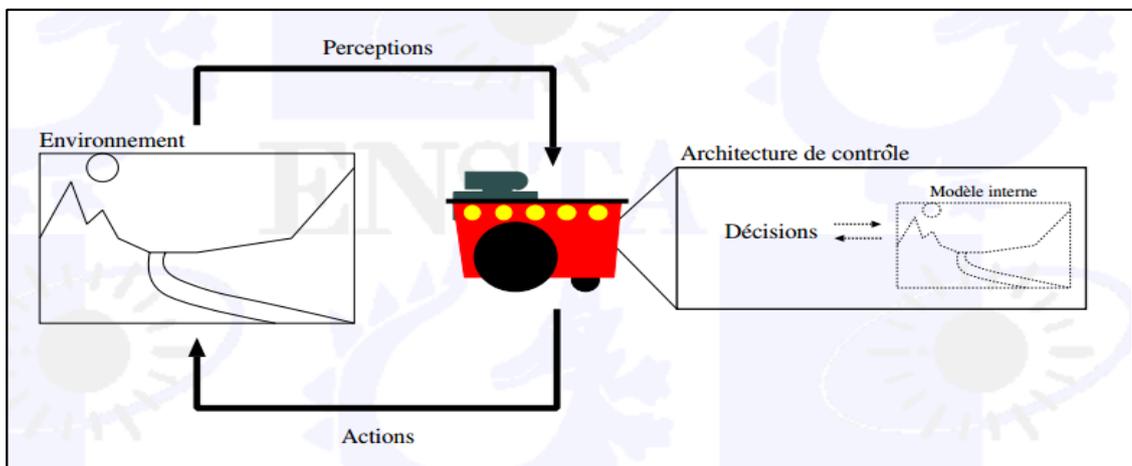


Figure 1.3. Classique des interactions d'un robot avec son environnement.4

1.8.2 Les Classes des robots mobiles à roues :

Pour déplacer un robot mobile sur une surface, il faut au moins deux degrés de liberté, donc deux moteurs. Et c'est aussi la combinaison du choix des roues et de leur disposition qui confère à un robot son mode de locomotion propre, on rencontre principalement quatre types de robot [6] :

- **Robot mobile de type uni-cycle** : Actionné par deux roues indépendantes et possédant éventuellement un certain nombre de roues folles assurant sa stabilité.
- **Robot mobile de type tricycle** : Constitué de deux roues fixes de même axe et d'une roue centrée orientable placée sur l'axe longitudinal du robot. Le mouvement est conféré au robot par deux actions (la vitesse longitudinale et l'orientation de la roue orientable).
- **Robot mobile de type voiture** : Un robot mobile de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe. Le robot mobile de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé.
- **Robot mobile de type omnidirectionnel** : Un robot mobile omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions.

Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral. L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande.

Chapitre 1 Les robots

Ces classes sont représentées dans ce *Tableau 1.1* :

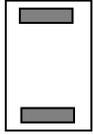
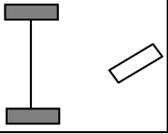
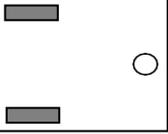
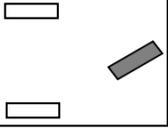
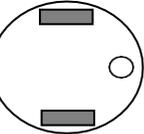
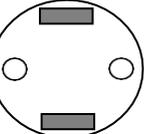
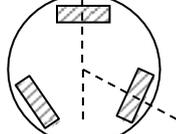
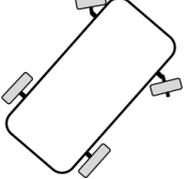
	<p>1. Uni-cycle, avec deux roues motrices différentielles exemple : Cye personale robot.</p>
	<p>2. Tricycle, deux roues motrices en arrière, et une roue dirigée à l'avant exemple : Piaggio minitrucks.</p>
	<p>3. Uni-cycle, deux roues motrices différentielles en arrière, et une roue libre à l'avant exemple : Pygmalion et Alice.</p>
	<p>4. Tricycle, deux roues libres en arrière, une roue de traction dirigée à l'avant exemple : Neptune, Hero-1.</p>
	<p>5. Uni-cycle, deux roues motrices différentielles centrées, et un point de contact a l'avant exemple : Nomad Scout, smartRob.</p>
	<p>6. Uni-cycle, deux roues motrices différentielles centrées, et deux points de contact avant et arrière exemple : EPFL Khepera, Hyperbot Chip.</p>
	<p>7. Omnidirectionnel, trois roues suédoises motorisées exemple : Tribolo EPFL, Palm Pilot Robot Kit.</p>
	<p>8. Voiture, - stable et complexité - mécanique modérée</p>

Tableau 1.1. Classes des robots mobiles à roue.

1.8.3 Application des robots mobiles :

Aujourd'hui, le marché commercial de la robotique mobile est toujours relativement restreint, mais il existe de nombreuses perspectives de développement qui en feront probablement un domaine important dans le futur. Les applications des robots peuvent se trouver dans de nombreuses activités "ennuyeuses, salissantes ou dangereuses, mais également pour des applications ludiques ou de service, comme l'assistance aux personnes âgées ou handicapées. Parmi les domaines concernés, citons :

- ❖ La robotique de service (hôpital, bureaux)
- ❖ La robotique de loisir (aibo, robot 'compagnon')
- ❖ La robotique industrielle ou agricole (entrepôts, récolte de productions agricoles)
- ❖ La robotique en environnement dangereux, (robots exploitation spatial, industries, militaires)

La *Figure 1.4.* montre quelques exemples de robot réels :



Figure 1.4. Exemples de robots utilisés dans différentes applications.

1.9 Conclusion :

La connaissance de ces informations à propos des robots en cas générale et le robot mobile en cas spéciale est importante pour réaliser ce projet. Ce chapitre a réussi de fournir des informations à propos la robotique

Chapitre 2 Moteur à courant continu

2.1 Introduction

Une machine à courant continu est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique. Elle est aussi appelée dynamo

- En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique. La machine se comporte comme un frein.

Inventée par Zénobe Gramme, c'était au départ un simple générateur de courant continu (pour applications galvanoplastiques, par exemple, les accumulateurs étant onéreux). Le moteur sans balais, qui l'a supplanté, est un perfectionnement de cette machine. [7]

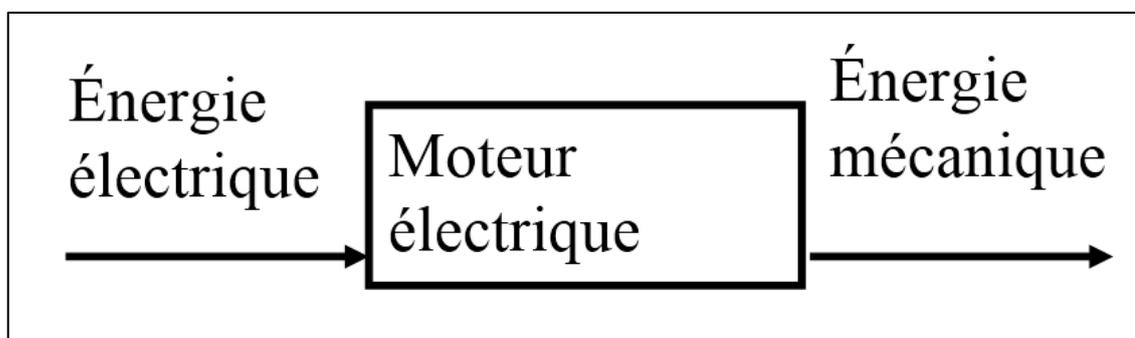


Figure 2.1. Principe de fonctionnement moteur courant continu.

2.2 Description de la machine à courant continu

Une machine électrique à courant continu est constituée : D'un stator qui est à l'origine de la circulation d'un flux magnétique longitudinal fixe créé soit par des enroulements statoriques (bobinage) soit par des permanents. il est aussi appelé « inducteur » en référence au fonctionnement en génératrice de cette machine.

Chapitre 2 Moteur à courant continu

- D'un rotor bobiné relié à un collecteur rotatif inversant la polarité dans chaque enroulement rotorique au moins une fois par tour de façon à faire circuler un flux magnétique transversal en quadrature avec le flux statorique. Les enroulements rotoriques sont aussi appelés enroulements d'induits, ou communément « induit » en référence au fonctionnement en génératrice de cette machine.

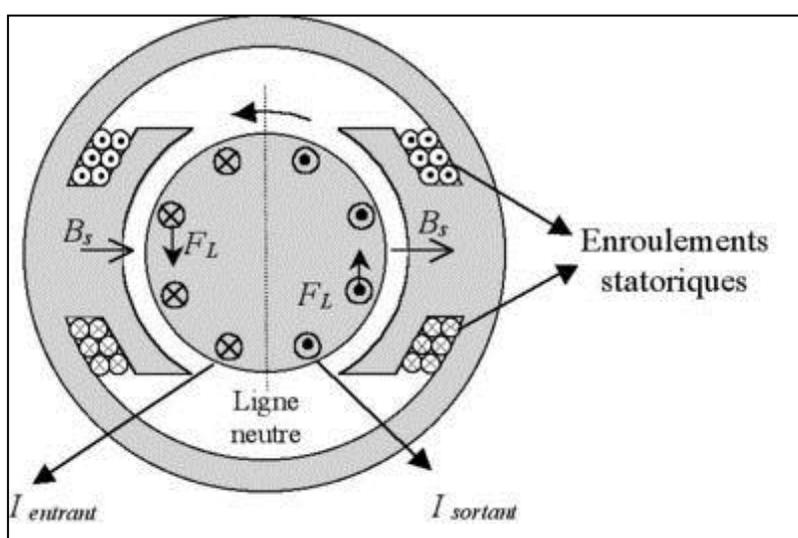


Figure 2.2. Schéma de continu vu en coupe latérale. La partie ronde en gris au centre est le rotor, l'autre partie en gris constitue le stator.

- Le courant I , injecté via les balais au collecteur, traverse un conducteur rotorique (une spire rotorique) et change de sens (commutation) au droit des balais. Ceci permet de maintenir la magnétisation du rotor perpendiculaire à celle du stator.

La disposition des balais sur la « ligne neutre » (c'est machine à courant c-à-dire la zone où la densité de flux est nulle), permet d'obtenir la force contre-électromotrice (FCEM) maximum. Cette ligne peut néanmoins se déplacer par la réaction magnétique d'induit (influence du flux rotorique sur le champ inducteur) selon que la machine travaille à forte ou à faible charge. Une surtension, due en partie à la mauvaise répartition de la tension entre lames du

Chapitre 2 Moteur à courant continu

collecteur et en partie à l'inversion rapide du courant dans les sections de conducteur lors du passage de ces lames sous les balais, risque alors d'apparaître aux bornes de la spire qui commute et de provoquer la destruction progressive du collecteur. Pour pallier cela, c'est-à-dire compenser la réaction d'induit, et aussi améliorer la commutation, on utilise des pôles auxiliaires de compensation/commutation.

- L'existence du couple s'explique par l'interaction magnétique entre stator et rotor :
 - Le champ statorique (B_s sur le schéma) est pratiquement nul sur les conducteurs logés dans des encoches et n'agit donc pas sur eux. L'origine du couple reste la magnétisation transversale du rotor, inchangée au cours de sa rotation (rôle du collecteur). Un pôle statorique agit sur un pôle rotorique et le moteur tourne.
 - Une manière classique mais simpliste de calculer le couple est de s'appuyer sur l'existence d'une force de Laplace (fictive) créée par le champ statorique (B_s sur le schéma) et agissant sur les conducteurs rotoriques traversés par le courant d'intensité.
- La force ainsi créée est proportionnelle à I et à B_s . Le couple moteur T est donc lui aussi proportionnel à ces deux grandeurs.
- par le La tige conductrice courant I se déplace soumise au champ statorique B_s . Elle est donc le siège d'une FCEM induite (loi de Lenz) proportionnelle à B_s et à sa vitesse traversée au rotor de déplacement, donc à la fréquence de rotation. L'ensemble de ces forces contre-électromotrices a pour conséquence l'apparition d'une FCEM globale E aux bornes de l'enroulement rotorique qui est proportionnelle à B_s et à la vitesse de rotation du moteur.
- Pour permettre au courant I de continuer à circuler, il faudra que l'alimentation électrique du moteur délivre une tension supérieure à la force contre-électromotrice E induite au rotor.

Chapitre 2 Moteur à courant continu

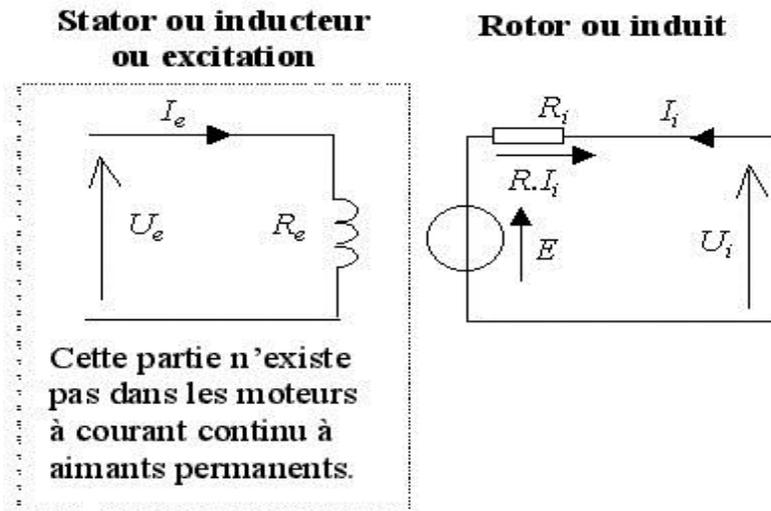


Figure 2.3. Schéma électrique idéalisé.

Ce schéma rudimentaire n'est pas valable en régime transitoire.

R_i et R_e sont respectivement les résistances du rotor et du stator

Ce schéma correspond aux équations électriques suivantes :

- au stator : $U_e = R_e \cdot I_e$ (loi d'ohm) et le champ statorique vaut $B_s = k_e \cdot I_e$
- au rotor : $U_i = E + R_i \cdot I_i$

D'autre part on a deux équations électromécaniques :

- La force contre électromotrice : $E = Cte \cdot B_s \cdot \Omega$ (Ω = fréquence de rotation en rad/s).
- Le couple électromécanique (moteur ou résistant) : $T = Cte \cdot B_s \cdot I_i$

On peut montrer que les constantes sont les mêmes pour les deux lignes, ce qui implique :

- $E \cdot I_i = T \cdot \Omega$ ou « Puissance électrique utile » = « Puissance mécanique ».

Chapitre 2 Moteur à courant continu

2.3 Descriptif du fonctionnement :

Imaginons une machine électrique alimentée par une source de tension U constante. Lorsque le moteur tourne à vide (il ne fait pas d'effort) il n'y a pas besoin de fournir de couple, I_i est très faible et $U \approx E$. La vitesse de rotation est proportionnelle à U .

2.3.1 Fonctionnement en moteur :

Lorsque l'on veut la faire travailler, en appliquant un couple résistant sur son axe, cela la freine donc E diminue.

Comme U reste constante, le produit $R_i \cdot I_i$ augmente donc I_i augmente, donc le couple T augmente lui aussi et lutte contre la diminution de vitesse : c'est un couple moteur.

Plus on le freine, plus le courant augmente pour lutter contre la diminution de vitesse. C'est pourquoi les moteurs à courant continu peuvent « griller » lorsque le rotor est bloqué, si le courant de la source n'est pas limité à une valeur correcte.

2.3.2 Fonctionnement en génératrice :

Si une source d'énergie mécanique essaie d'augmenter la vitesse de machine, (la charge est entraînée : ascenseur par exemple), Ω augmente donc E augmente.

Comme U reste constante, le produit $R_i \cdot I_i$ devient négatif et augmente en valeur absolue, donc I_i augmente, donc le couple T augmente lui aussi et lutte contre l'augmentation de vitesse : c'est un couple frein.

Le signe du courant ayant changé, le signe de la puissance consommée change lui aussi.

La machine consomme une puissance négative, donc elle fournit de la puissance au circuit. Elle est devenue génératrice. Ces deux modes de fonctionnement existent pour les deux sens de rotation de la machine.

Chapitre 2 Moteur à courant continu

Celle-ci pouvant passer sans discontinuité d'un sens de rotation ou de couple à l'autre. On dit alors qu'elle fonctionne dans les quatre quadrants du plan couple-vitesse.

2.4 Les réducteurs :

2.4.1 Introduction :

Un moteur électrique est bien souvent très rapide en rotation. Hors si vous avez besoin de faire un robot qui ne va pas trop vite, il va falloir faire en sorte de réduire sa vitesse de rotation. On peut très bien mettre un "frein" qui va empêcher le moteur de tourner vite, ou bien le piloter (on va voir ça toute à l'heure). Cela dit, même si on réduit sa vitesse de rotation, le moteur ne va pas pouvoir supporter des charges lourdes.

Autrement dit, votre robot ne pourra même pas se supporter lui-même ! Nous avons donc besoin de couple. Et pour avoir du couple, tout en réduisant la vitesse de rotation, on va utiliser ce que l'on appelle un réducteur.

2.4.2 Définition :

Un réducteur est un ensemble composé d'engrenages qui permet de réduire la vitesse de rotation de l'axe du moteur tout en augmentant le couple de sortie. Sur l'image suivante, on peut observer un ensemble moteur + réducteur + roue [8] :

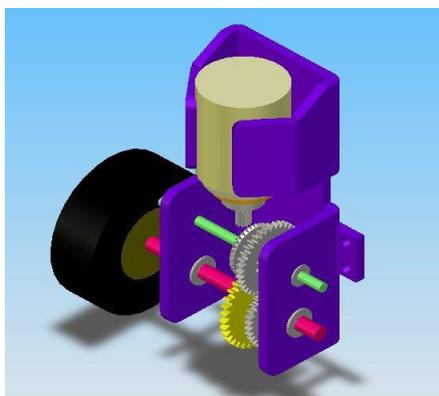


Figure 2.4. Les réducteurs.

Chapitre 2 Moteur à courant continu

2.4.3 le rapport de réduction

$$R = \omega_{\text{entrée}} / \omega_{\text{sortie}} = C_{\text{sortie}} / C_{\text{entrée}}$$

Avec :

R : le rapport de réduction du réducteur.

$\omega_{\text{entrée}}$: la vitesse de rotation de l'axe du moteur en entrée du réducteur.

ω_{sortie} : la vitesse de rotation de l'axe du moteur en sortie du réducteur.

C_{sortie} : couple exercé par l'axe de sortie du réducteur.

$C_{\text{entrée}}$: couple exercé par l'axe du moteur, en entrée du réducteur.

Un réducteur s'apparente donc à un système qui modifie deux grandeurs qui sont liées : le couple et la vitesse. On peut schématiser le fonctionnement d'un réducteur de la manière suivante :

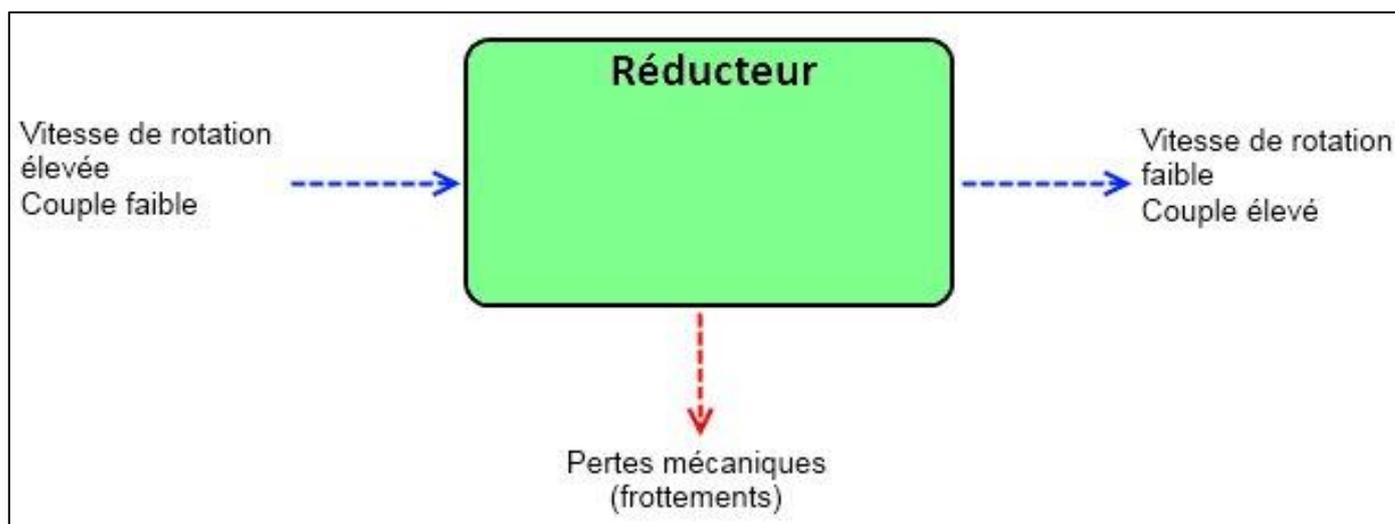


Figure 2.5. Schéma d'un réducteur.

Chapitre 2 Moteur à courant continu

2.5 Conclusion

La connaissance de ces informations à propos les moteur à courant continu est importante pour réaliser ce projet, Dans notre projet on va utiliser deux moteurs à courant continu (motoréducteur 6V).

3.1 Introduction

L'Android est parmi les derniers systèmes d'exploitation qui développent les exigences des téléphones intelligents. La plateforme Android de smartphone devient de plus en plus importante pour les réalisateurs de logiciel, en raison de ses puissantes possibilités et open source.

Lors des années précédentes, le traitement des données informatiques se fait par des ordinateurs, en revanche le smartphone a des avantages qui ont les mêmes fonctions que l'outil informatique, ce dernier porte l'intérêt de l'ordinateur grâce à l'Android.

La téléphonie mobile a connu une explosion dans les années 2000 mais aucune révolution n'a semblé arriver depuis que les appareils se ressemblent. Les innovations n'avaient plus vraiment de saveur, les applications étaient difficiles d'accès de par leur mode de distribution et souvent peu performantes à cause des faibles capacités des appareils.

Depuis quelques mois, les smartphones sont dotés d'une puissance plus importante et d'espaces de stockages conséquents. Les téléphones tendent à devenir des objets artistiques, presque de reconnaissance sociale, et possèdent des fonctionnalités qu'aucun téléphone ne pouvait espérer auparavant : connexion haut débit, localisation GPS, boussole, accéléromètre, écran tactile souvent multipoint, marché d'applications en ligne. Autant de qualités permettant de créer des applications innovantes et de les distribuer en toute simplicité.

La plate-forme Android apporte tout cela au consommateur, mais surtout, elle affranchit le développeur de nombreuses contraintes. Par son ouverture, elle permet à n'importe quel développeur de créer ses applications avec un ticket d'entrée quasi nul. Le Framework et le système d'exploitation et outils associés ont un code source ouvert, leur accès est gratuit et illimité. Plus besoin de négocier avec le constructeur du téléphone pour qu'il vous laisse développer sur sa plate-forme.

Chapitre 3 Les applications Android

Tous les développeurs sont ainsi sur un même pied d'égalité, tous peuvent ajouter de la mobilité à des applications existantes.

Cette partie de notre étude ne nous donnera pas de bons résultats si on néglige certains paramètres, donc le bon fonctionnement de notre système se base essentiellement sur une bonne démarche et une bonne réflexion de notre programme.

3.2 Présentation Android



Android est un système d'exploitation libre qui utilise les téléphones mobiles modernes nombre (smartphones = smartphones), est basé sur Linux. Android est détenue par Google.

1.5 (Cupcake), 1.6 (Donut), 2.0 (Eclair), 2.2 (Froyo), 2.3 (Gingerbread), 3.2 (Honeycomb), 4.0 (Ice Cream Sandwich), 4.1 (Jelly Bean) : Chaque semble si souvent de nouvelles versions 4.2 Jelly Bean (Gummy Bear), 4.3 (Jelly Bean), 4.4 (KitKat) 4,5 (Kit Kat), 5.0 (Lollipop) ...

Il existe d'autres systèmes d'exploitation de téléphone comme iOS d'Apple (20%) et Microsoft Windows Phone (2%), Android (78%).

L'Android vous pouvez facilement programmer et effectuer des programmes de formules mathématiques, des jeux, emplacement sur Google Maps, le déplacement d'un robot via Bluetooth, réponse automatique de SMS ... [9]

3.3 Présentation d'App inventor



APP INVENTOR est un IDE (environnement de développement intégré) qui permet la création d'applications destinées à des systèmes équipés de plateformes Android tels que les téléphones (tactiles ou non) et les tablettes (ACER, SAMSUNG ...).

A l'origine le projet APP INVENTOR a été créé par le Google Labs puis abandonné. Aujourd'hui ce concept est repris par le MIT (Massachusetts Institute of Technology). Le téléchargement et l'utilisation de APP Inventor nécessite un compte google (gmail). [10]

Cet environnement de programmation permet une programmation graphique, basée sur l'assemblage de blocs (langage Scratch). Des connaissances en programmation orientée objet sont toutefois nécessaires. Cependant, on pourra s'affranchir complètement de la connaissance des noms des propriétés et méthodes liées aux objets. En effet des blocs de propriétés, méthodes et évènements seront directement proposés dès la création d'un objet.

L'IDE est formé de deux composantes logicielles :

Un site Web permettant de créer l'interface homme machine (aspect graphique) de la future application ;

Un programme résidant sur l'ordinateur, qui permet par association des blocs liés aux objets précédemment définis dans l'espace web, de créer le comportement de l'application.

Chapitre 3 Les applications Android

Il permet également le test de l'application sur un émulateur ou le téléchargement vers le média désiré (tablette, téléphone...).

3.4 Créé un projet :

Depuis un navigateur internet ouvrir le compte google.

Aller sur le site du MIT App Inventor : <http://appinventor.mit.edu/>.

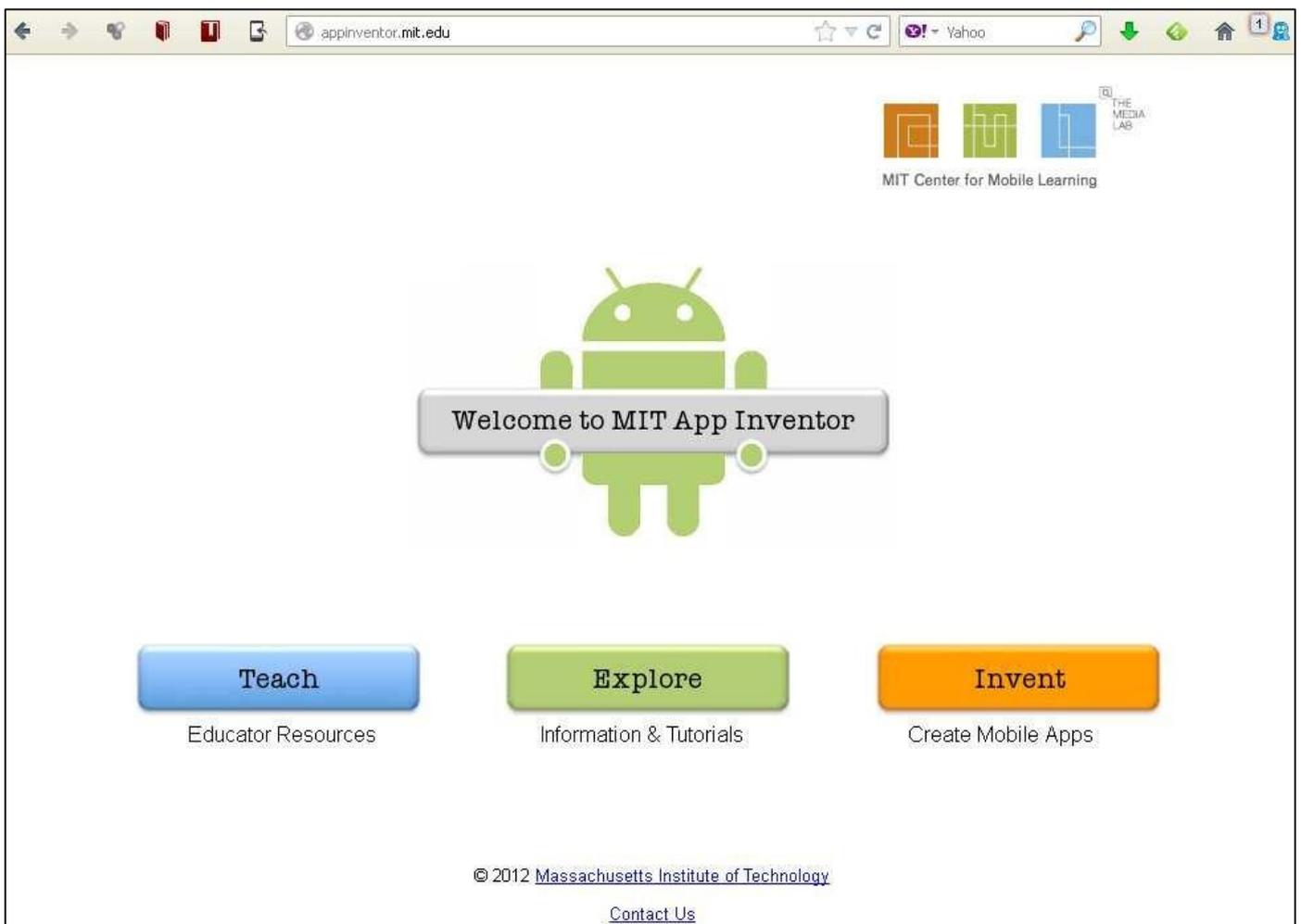
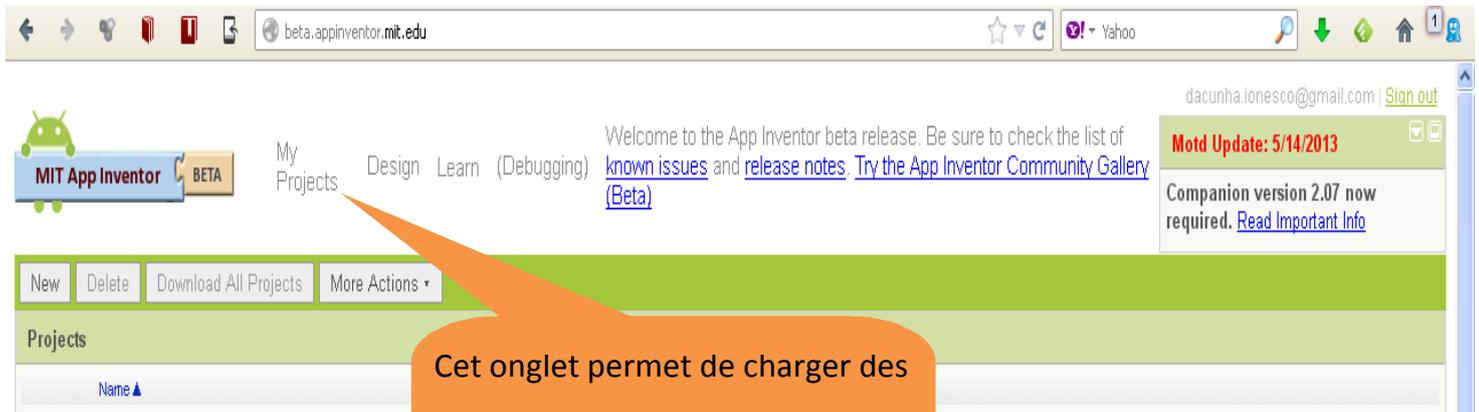


Figure 3.1. Site mit app inventor.

Chapitre 3 Les applications Android

- Cliquer sur



- Cliquer sur **New** . **Donner** un nom au nouveau projet.

New App Inventor for Android Project...

Project name:

Chapitre 3 Les applications Android

Et voilà Interface graphique d'utilisateur :

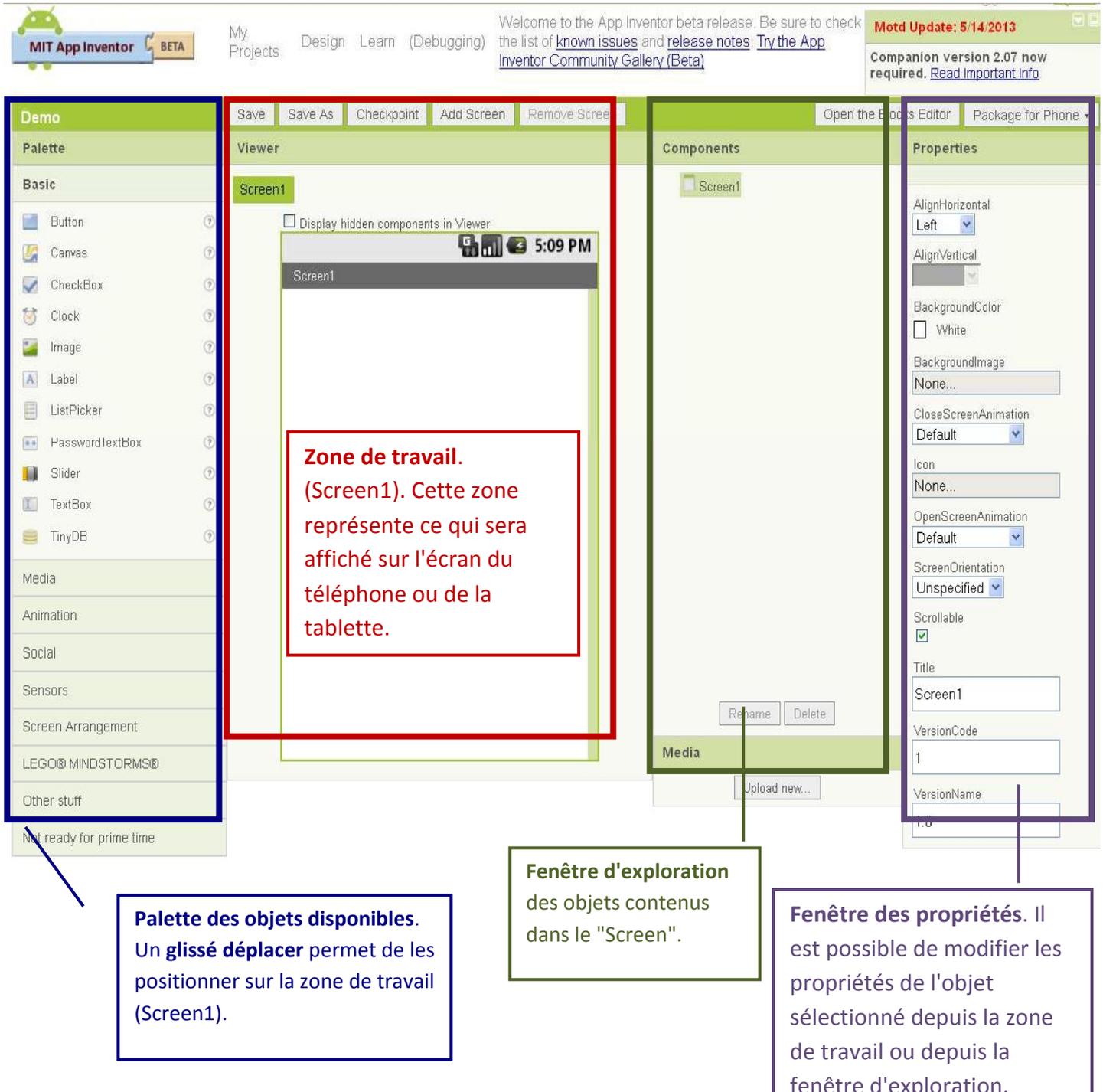


Figure 3.2. Première interface de la création App Inventor.

Chapitre 3 Les applications Android

3.4.1 Application permettant la commande du robot

L'application « **svr application robotique** » permet la commande des moteurs du robot. On désire commander le robot à partir de l'application pour smartphone.

L'aspect de l'application doit ressembler à la présentation suivante :

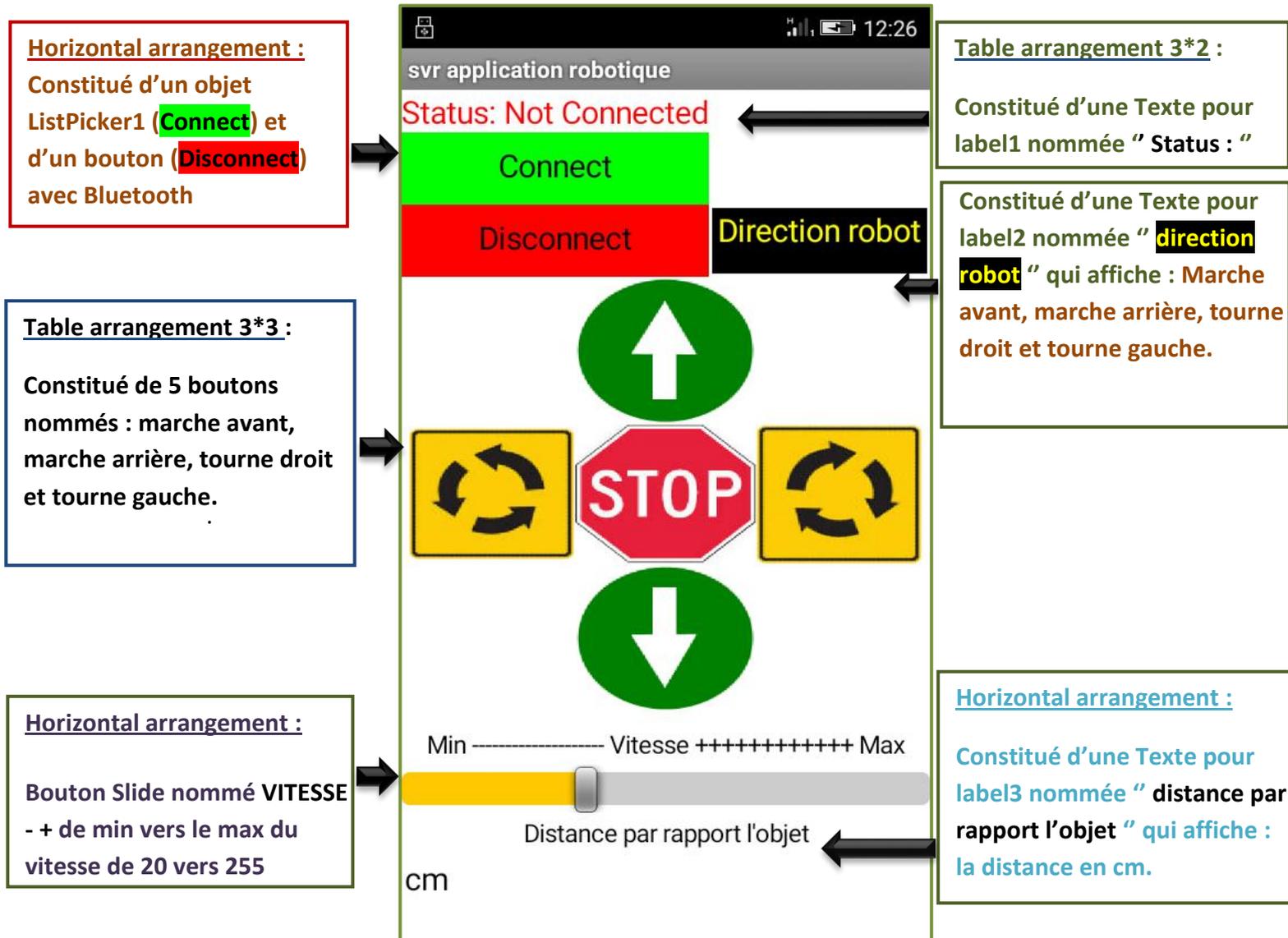


Figure 3.3. L'interface de l'application.

Chapitre 3 Les applications Android

3.4.2 Connexion de la smartphone avec le robot par Bluetooth :

Voici comment réalisée le programme de la connexion Bluetooth entre l'application et le module de Bluetooth hc-06 liées avec le robot :

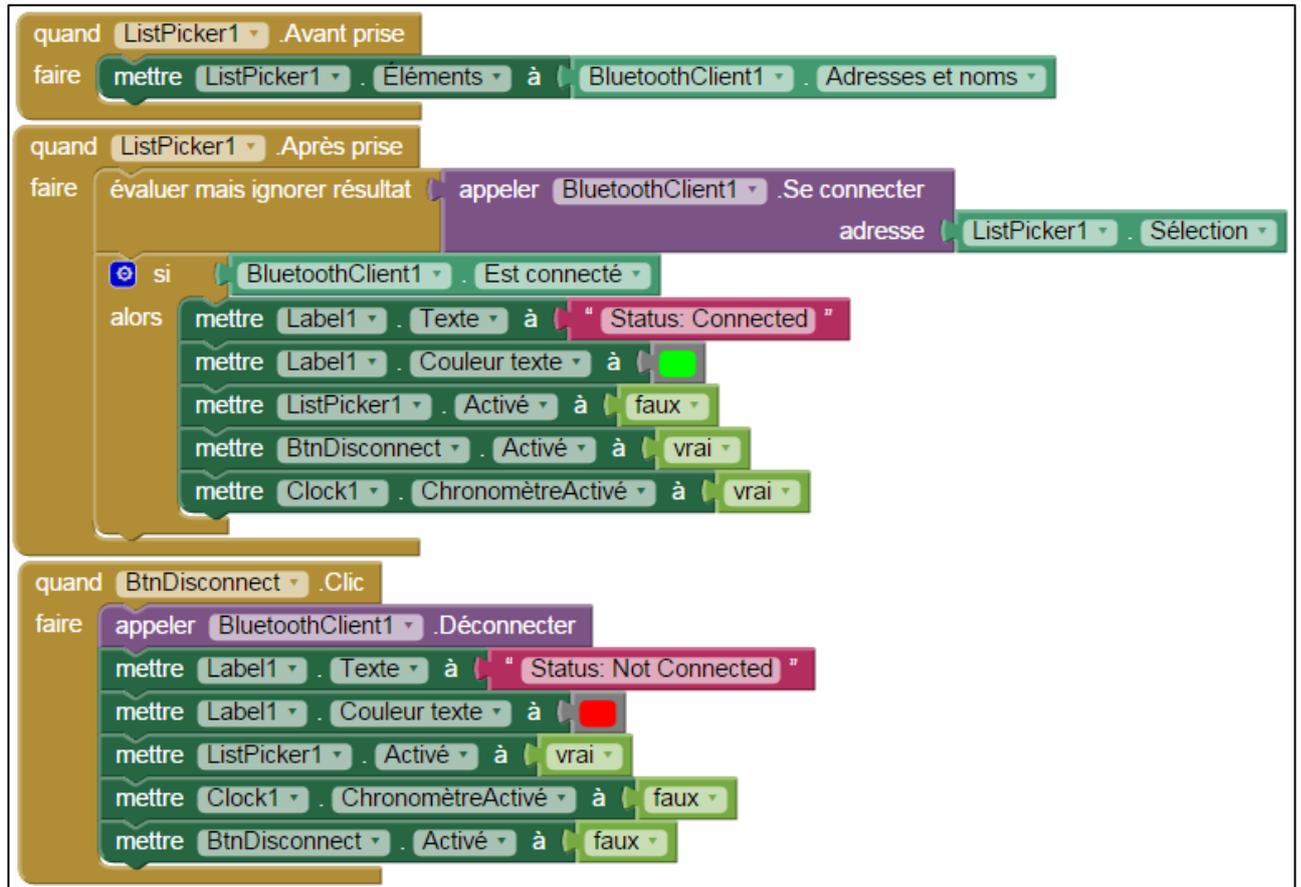


Figure 3.4. Code source d'application 1.

- L'objet **ListPicker1** permettra de choisir dans une liste, un robot parmi ceux appariés en Bluetooth avec le smartphone
- L'objet **texte pour label1** permettra l'affichage du fonctionnement de la connexion avec Bluetooth du robot si elle est connecté ou pas.
- le bouton **btndisconnect** est pour déconnecter BluetoothClient1

Chapitre 3 Les applications Android

Lorsque le ListPicker « connected » est appuyé, une nouvelle s'ouvre avec la listes des Bluetooth des robots appariés. Lorsque l'un robot a été sélectionné dans la liste, une connexion Bluetooth s'établie. Si le smartphone est connectée avec le robot sélectionné, le message **texte pour label1 «connected »** en couleur vert est notifié, sinon le message « **not connected**» en couleur rouge est notifié.

3.4.3 mise en fonctionnement du robot :

Le robot doit pouvoir être commandé en marche avant, marche arrière, virage sur la gauche ou le droit ou bien mis à l'arrêt par l'intermédiaire de l'appui sur les 5 boutons correspondant. La commande du robot en marche avant, marche arrière, virage sur la gauche ou le droit ou bien mis à l'arrêt est réalisée par le programme suivant :

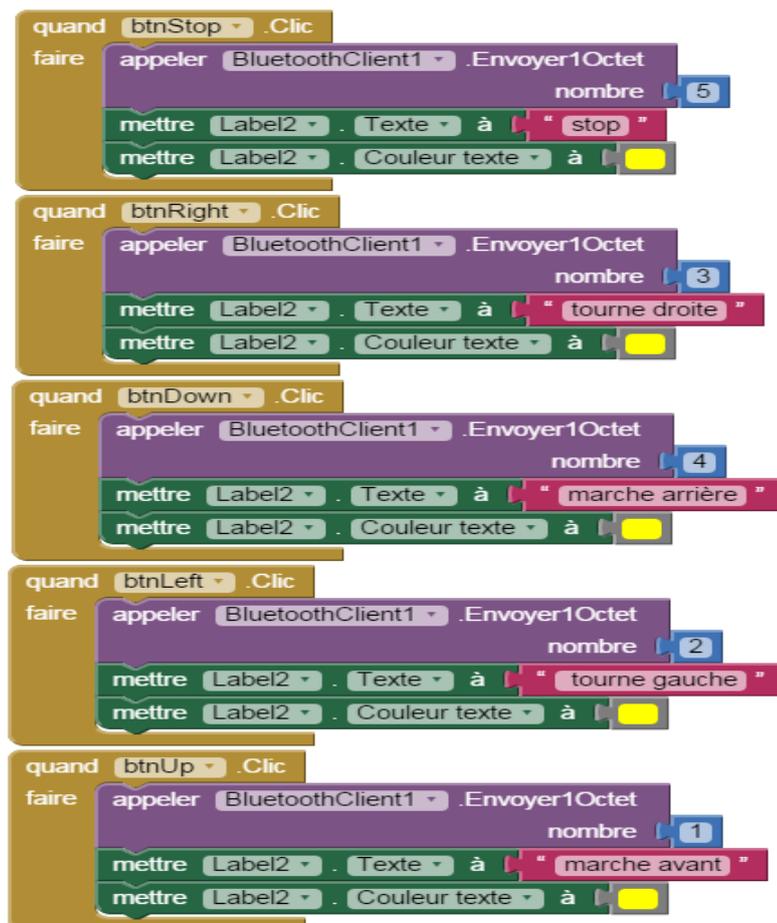


Figure 3.5.Code source d'application 2.

Chapitre 3 Les applications Android

- Quand quelqu'un de 5 boutons de commande est cliqué alors il envoie 1 octet numéro au module Bluetooth hc-06
- L'objet **Texte pour label2** permettra l'affichage du fonctionnement du robot qu'il est en cas d'exécution soit (marche avant, marche arrière, tourne droit et gauche.) en couleur jaune.

Le bouton **slider1** c'est pour commander la vitesse du robot on a choisi la valeur minimale égale à 20 et la valeur maximale égale à 255. Lorsque la position du curseur se change elle a envoyé la position actuelle au récepteur.

La réalisation est par le programme suivant :

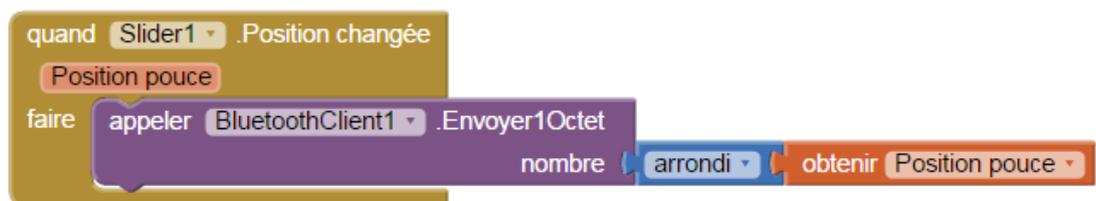


Figure 3.6. Code source d'application 3.

Et on ajoute un **Texte pour label3** pour afficher la distance entre l'objet (obstacle) et notre robot en centimètre est réalisé par le programme suivant :

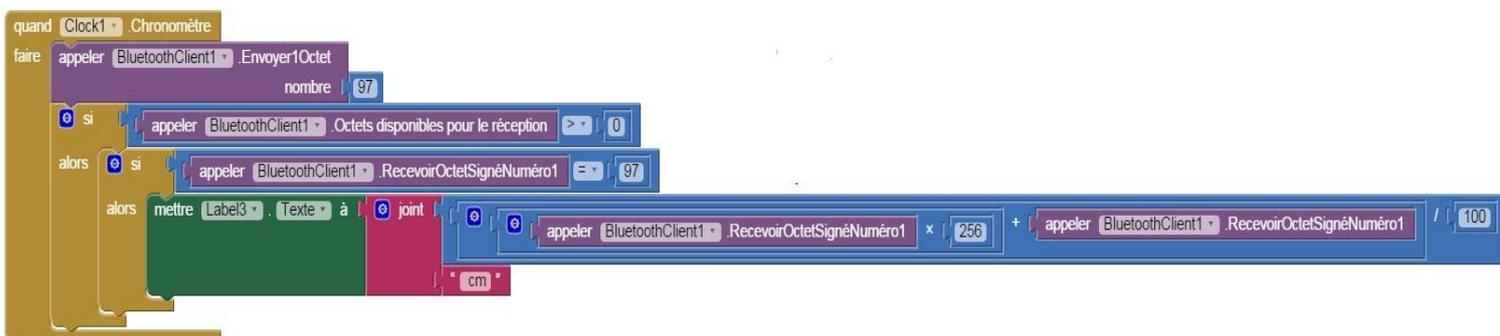


Figure 3.7. Code source d'application 4.

On a utilisé pour ça le capteur ultrason.

Chapitre 3 Les applications Android

- Quand un objet proche ou plus loin vers l'avant depuis le voisinage du capteur à ultrasons, il mesure la distance en centimètres sur l'objet et l'envoie au Bluetooth l'application android reçoit cette information de distance et d'écrire sur l'écran.

3.4.4 Création de l'application «svr application robotique » :

Notre interface contient une touche d'activation et désactivation du Bluetooth, donc avant d'envoyer n'importe quel caractère à travers les 5 boutons de direction ou le bouton slide de vitesse il faut d'abord s'assurer que le Bluetooth est allumé. Cette figure montre que notre application nommée «svr application robotique» peut être installée sur smartphone, sa taille est de 1.4 mégas.

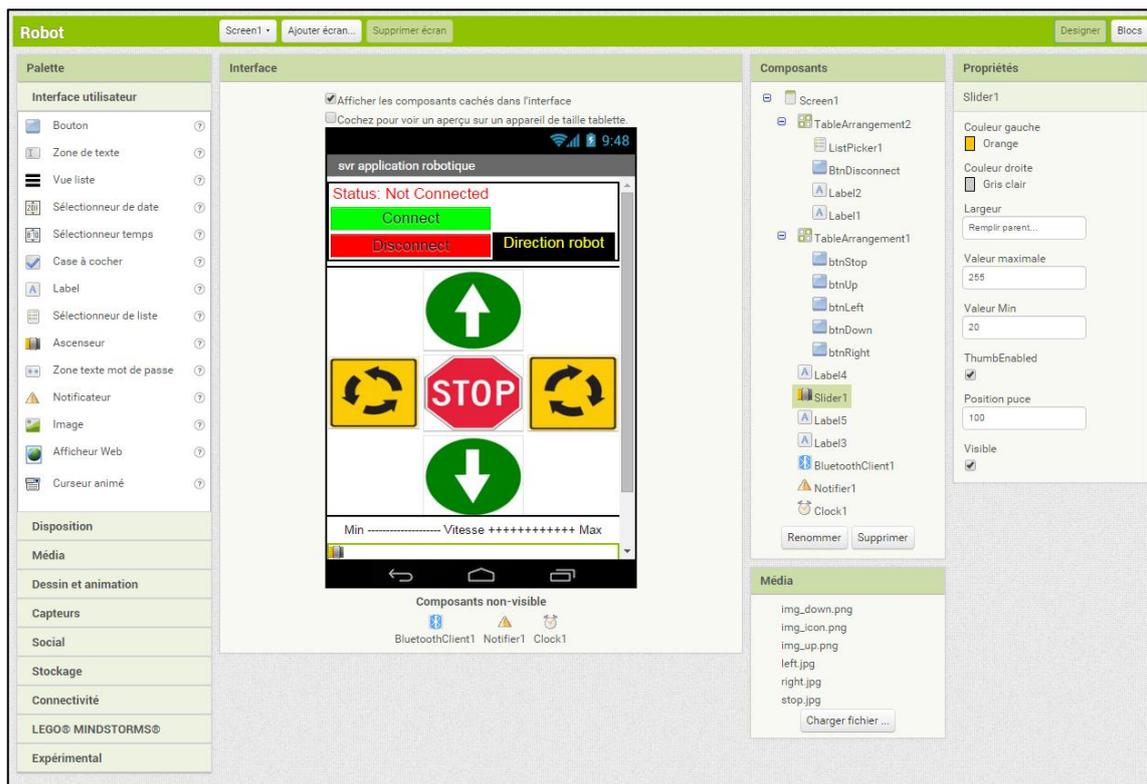


Figure 3.8. Photo de notre application de contrôle.

3.5 Conclusion :

En fin une simple élaboration de l'environnement App Inventor nous pousse à réaliser une application « *.apk » nommée « svr application robotique » sous smartphone capable de lier une carte Arduino et un smartphone via Bluetooth afin d'exécuter des ordres bien définis. On peut conclure que les résultats obtenus sont satisfaisants compte tenu des limitations du matériel et des moyens dont nous disposons.

Chapitre 1

Chapitre 2

BIBLIOGRAPHIE

ANNEXE

Chapitre 3

Introduction Générale

Conclusion Générale

Chapitre 4

Conclusion générale

Conclusion générale

Nous pouvons dire à la fin de notre projet que nous avons réalisé un robot mobile autonome qui peut éviter les obstacles (en utilisant des capteurs ultrason), et il est télécommandé à distance grâce à une application dans le système Android de ce pat nous pouvons dire que nous avons atteint nos objectifs discuter en 4 chapitre et qui nous ont permis d'obtenir le résultat attendu organisé comme suit :

- Dans le premier chapitre nous avons discuté le principe des robots en général est en précisent les robots mobile car notre travail est basé sur ce type des robots

- Dans le second chapitre nous avons abordé le sujet des moteurs courant continu, définition principe de fonctionnement, est surtout nous avons projeté la lumière sur les moteur réducteurs qu'on a utilisé dans notre projet.

- Dans le troisième chapitre-t-il s'agit de système Android et toutes les étapes de création de notre application qui commande le robot mobile à travers la carte arduino, en utilisant la communication Bluetooth.

- Et enfin dans le dernier chapitre nous avons parlé du composant électronique de robot et surtout de l'arduino, Le système Arduino est un outil pour fabriquer des dispositifs qui peuvent capter et contrôler davantage de choses du monde matériel que votre ordinateur. C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

A la fin de ce travail on peut dire qu'Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie réalisation de la carte.

Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études. Nous souhaitons vivement que ce projet puisse servir comme élément de base pour d'autres études plus approfondies pour le faire intégrer sous des systèmes plus complexes.

Table des matières

Introduction général	1
Chapitre 1 : Les robots	2
1.1 Introduction	2
1.2 Définitions	2
1.3 Historique de la robotique	3
1.4 Les lois de la robotique	6
1.5 Structure d'un robot	7
1.6 Type des robots	7
1.7 Fonctionnement des robots évolués	8
1.7.1 Perception	8
1.7.2 Décision	9
1.7.2 Action	9
1.8 Les robots mobiles	10
1.8.1 définition des robot mobile	10
1.8.2 Les classe des robot mobile a roues	12
1.8.3 applications des robots mobile	14
1.9 Conclusion	15
Chapitre 2 : Moteur à courant continu	16
2.1 Introduction	16
2.2 Description de la machine à courant continu	16

2.3 Descriptif du fonctionnement	20
2.3.1 Fonctionnement en moteur	20
2.3.2 Fonctionnement en génératrice	20
2.4 Les réducteurs	21
2.4.1 Introduction	21
2.4.2 Définition	21
2.4.3 le rapport de réduction	22
2.5 Conclusion	23
Chapitre 3 : Les applications Android.....	24
3.1 Introduction	24
3.2 Présentation Android	25
3.3 Présentation d'App inventor	26
3.4 Créé un projet	27
3.4.1 Application permettant la commande du robot.....	30
3.4.2 Connexion de la smartphone avec le robot par Bluetooth	31
3.4.3 mise en fonctionnement du robot	32
3.4.4 Création de l'application «svr application robotique »	34
3.5 Conclusion	35
Chapitre 4 : Arduino conception et programmation.....	36
4.1 Introduction	36
4.2 Définition du module Arduino	36
4.3 Les gammes de la carte Arduino	37

4.4 La constitution de la carte Arduino uno	40
4.4.1 Partie matérielle	40
4.4.2 Partie programme	44
4.5 Les Accessoires de la carte Arduino	52
4.5.1 Communication	53
<i>a Le module shield Arduino Wifi</i>	53
<i>b Le Module XBee</i>	54
<i>c Le module Arduino Bluetooth</i>	54
4.5.2 Les capteurs	58
<i>a Introduction</i>	58
<i>b Le capteur ultrason</i>	59
4.5.3 Les drivers	60
<i>a Introduction</i>	60
<i>b Pont-h L298</i>	61
<i>c Les moteurs électriques</i>	65
4.5.4 Autre accessoire	66
<i>a Breadboard</i>	66
<i>b Fils pour arduino</i>	67
<i>c Bouton poussoir</i>	67
4.6 Programmation du robot (robot télécommandé et détecte obstacle)	68
4.6.1 List du besoin	68
4.6.2 Montage du robot	70

4.6.3 Code source de projet	71
4.7 Conclusion	74
Conclusion générale	75
Annexe	76
Bibliographie	78

Liste des figures

<i>Figure 1.1.</i> Robot wabot-1, robot unimate et robot shakey.....	5
<i>Figure 1.2.</i> Robot aibo, robot spirit et rob.....	6
<i>Figure 1.3.</i> Classique des interactions d'un robot avec son environnement.....	11
<i>Figure 1.4.</i> Exemples de robots utilisés dans différentes applications.....	14
<i>Figure 2.1.</i> Principe de fonctionnement moteur courant continu.....	16
<i>Figure 2.2.</i> Schéma de continu vu en coupe latérale. La partie ronde en gris au centre est le rotor, l'autre partie en gris constitue le stator.....	17
<i>Figure 2.3.</i> Schéma électrique idéalisé.....	19
<i>Figure 2.4.</i> Les réducteurs.....	22
<i>Figure 2.5.</i> Schéma d'un réducteur.....	26
<i>Figure 3.1.</i> Site mit app inventor.....	27
<i>Figure 3.2.</i> Première interface de la création App Inventor.....	29
<i>Figure 3.3.</i> L'interface de l'application.....	30
<i>Figure 3.4.</i> Code source d'application 1.....	31
<i>Figure 3.5.</i> Code source d'application 2.....	32
<i>Figure 3.6.</i> Code source d'application 3.....	33
<i>Figure 3.7.</i> Code source d'application 4.....	33
<i>Figure 3.8.</i> Photo de notre application de contrôle.....	34
<i>Figure 4.1.</i> La carte Arduino UNO.....	38
<i>Figure 4.2.</i> Microcontrôleur ATmega328.....	40
<i>Figure 4.3.</i> Constitution de la carte Arduino UNO.....	44

<i>Figure 4.4.</i> Interface IDE Arduino.....	49
<i>Figure 4.5.</i> Paramètre de la carte.....	50
<i>Figure 4.6.</i> Description du programme sur arduino.....	51
<i>Figure 4.7.</i> Les étapes de téléchargement du code.....	52
<i>Figure 4.8.</i> Module shield wifi.....	53
<i>Figure 4.9.</i> Module XBee.....	54
<i>Figure 4.10.</i> Type de modules Bluetooth.....	55
<i>Figure 4.11.</i> Module Bluetooth hc-06.....	56
<i>Figure 4.12.</i> Capteur ultrason hc-sr04.....	59
<i>Figure 4.13.</i> L298n double pont-h.....	62
<i>Figure 4.14.</i> Moteur avec motoréducteurs et roue.....	65
<i>Figure 4.15.</i> Mini breadboard.....	66
<i>Figure 4.16.</i> Fils pour arduino.....	67
<i>Figure 4.17.</i> Bouton poussoir.....	67
<i>Figure 4.18.</i> Schéma de montage du robot.....	70
<i>Figure 4.19.</i> Le robot.....	70
<i>Figure 4.20.</i> Code source 1.....	71
<i>Figure 4.21.</i> Code source 2.....	71
<i>Figure 4.22.</i> Code source 3.....	72
<i>Figure 4.23.</i> Code source 4.....	72
<i>Figure 4.24.</i> Finale du projet.....	73

Liste des tableaux

<i>Tableau 1.1.</i> Classes des robots mobiles à roue.....	13
<i>Tableau 4.1.</i> Avantages et inconvénients du capteur distance.....	59
<i>Tableau 4.2.</i> Caractéristique et spécification pont-h I298n.....	63
<i>Tableau 4.3.</i> Détails technique du circuit I298n.....	64
<i>Tableau 4.4.</i> List du besoin dans notre projet.....	69

Listes des acronymes et abréviations

MIT : Institut de technologie du Massachusetts

IDE : Environnement de développement intégré

PCB : Circuit imprimé

TX : transmission de données

RX : réception de données

PWM : modulation de largeur d'impulsions

LED : diode électroluminescente

RAM : appelée RAM de l'anglais Random Access Memory traduit en français par mémoire à accès direct

EPPROM : La mémoire EEPROM est un type de mémoire morte

SRAM : La mémoire vive statique

VIN : la tension d'entrée qui alimente la carte Arduino

GND : Niveau de tension de référence circuit

SPI : Interface Série Périphérique

USB : universel serial bus

TTL : Transistor-Transistor Logique

INO : Fichier de Package arduino

APK : Fichier de Package d'Application Android

MHZ : mégahertz

KBS : kilo byte par seconde

INPUT : Entrée.

BREADBOARD : plaque d'expérimentation

OUTPUT : Sortie.

PIN : Broche

LOOP : Boucle.

LOW : Bas.

Bibliographie

- [1] ROBOTIQUE : ISTIA, Université Angers .Jean-Louis Boimond (PDF).
- [2] Dictionnaire encyclopédique en un volume, publié en 1905 sous la direction de Claude Augé (Petit Larousse illustré).
- [3] : A. ALLOUI, A. HAJ Brahim. «Proposition d'une solution multi-agent pour la commande et la coopération multi -robot mobile». Mémoire d'ingénieur d'état en automatique, Université Biskra, Juin 2007.
- [4] Le site web :
<http://www.thomasta.com/soutien%20scolaire/informatique%20et%20telecommunication/info%20et%20tel/robotique.html>
- [5] Robotique Mobile - david.filliat@ensta-paristech.fr (PDF).
- [6] B. BAYLE. «Robotique Mobile», Ecole Nationale supérieur de Strasbourg, France, 2008-2009.
- [7] le site web : https://fr.wikipedia.org/wiki/Moteur_%C3%A0_courant_continu
- [8] Boîte de vitesse (moto-réducteur) eDrawing mercredi 7 juin 2006, par LECOURTIER Sébastien
- [9] le site web : <https://fr.wikipedia.org/wiki/Android>
- [10] Lycée Eugène IONESCO TP COMMANDE DU ROBOT LEGO®MINDSTORM™NTX PAR APPLICATION ANDROID UTILISATION D'APP INVENTOR
- [11] <http://www.generationrobots.com/fr/152-arduino>. Consulter le : mars 2016.
- [12] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino ».
- [13] X.HINAULT. www.mon-club-elec.fr.

Bibliographie

[14] <http://www.acm.uiuc.edu/sigbot/tutorials/2009-11-17-arduino-basics>. consulter le: avril 2016.

[15] Jean- Noël, « livret Arduino en français », centre de ressources art sensitif .

[16] le site web : <http://eskimon.fr/2498-arduino-annexes-g-utiliser-module-bluetooth-hc-05>

[17] <http://www.generationrobots.com/fr/content/65-les-capteurs-a-ultrasons-pour-les-robots>

[18] le site web : http://wiki.mchobby.be/index.php?title=Pont-H_L298N

Chapitre 4 Arduino conception et programmation

4.1 Introduction

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi de système embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation. On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

4.2 Définition du module Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme).

Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino. [11]

Chapitre 4 Arduino conception et programmation

4.3 Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique :

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega32U4

Chapitre 4 Arduino conception et programmation

- L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération. [12]

Parmi ces types, nous avons choisi une carte Arduino UNO (carte Basique). L'intérêt principal de cette carte est de faciliter la mise en œuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal à 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation. La carte est illustrée dans la figure si dessous.



Figure 4.1. La carte Arduino UNO.

Chapitre 4 Arduino conception et programmation

- **Pourquoi Arduino uno :**

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit :

- Le prix (réduits) : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré-assemblées coûtent moins de 2500 Dinars).
- Multi plateforme : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- Un environnement de programmation clair et simple : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- Logiciel Open Source et extensible : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- Matériel Open source et extensible : les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. [13]

Chapitre 4 Arduino conception et programmation

4.4 La constitution de la carte Arduino uno :

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

4.4.1 Partie matérielle :

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

- **Le microcontrôleur atmega328 :**

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors ; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure 2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.



Le composant CMS



Le composant classique

Figure 4.2. Microcontrôleur ATmega328.

Chapitre 4 Arduino conception et programmation

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- La mémoire Flash : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).
- RAM : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- EEPROM : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [14]

- **Les sources de l'alimentation de la carte :**

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme

Suit :

- VIN : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- 5V : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino).

Le 5V régulé fourni par cette broche peut donc provenir soit de la tension

Chapitre 4 Arduino conception et programmation

d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de toute autre source d'alimentation régulée.

- 3.3V : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA.
- GND : Broche de masse (ou 0V).
- **Les entrées et les sorties :**

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode ()`, `digitalWrite ()` et `digitalRead ()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digital Write (broche, HIGH)`.

En plus, certaines broches ont des fonctions spécialisées :

- Interruptions Externes : Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. -Impulsion PWM (largeur d'impulsion modulée) : Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analog Write ()`.
- SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- I2C : Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou

Chapitre 4 Arduino conception et programmation

interface TWI (Two-Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils").

- LED : Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead ()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analog Reference ()` du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

- **Les ports de communications:**

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 Programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Chapitre 4 Arduino conception et programmation

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil :

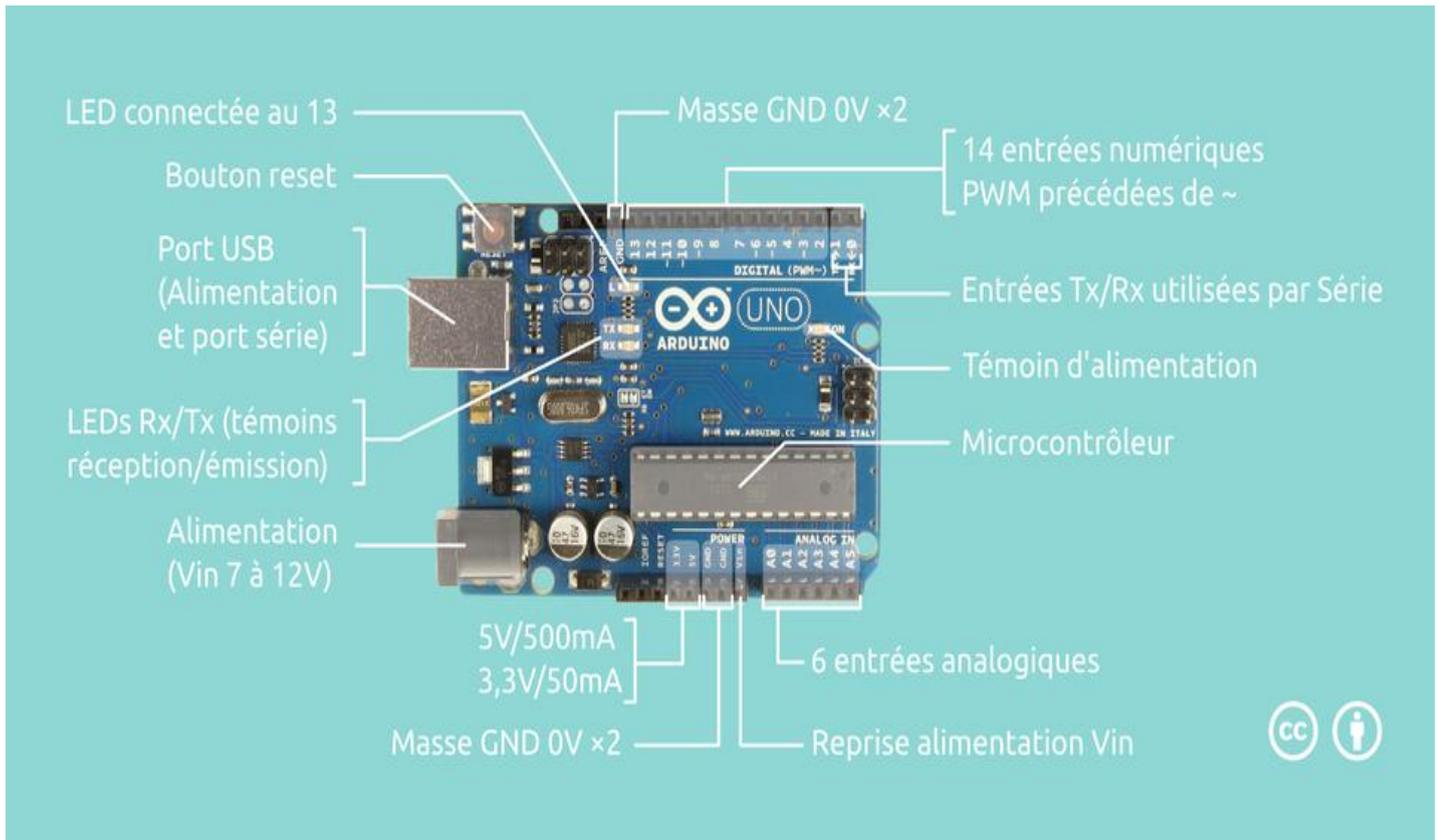


Figure 4.3. Constitution de la carte Arduino UNO.

4.4.2 Partie programme :

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

Chapitre 4 Arduino conception et programmation

- **L'environnement de la programmation :**

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino. [15]

- **Structure générale du programme (IDE Arduino) :**

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation.

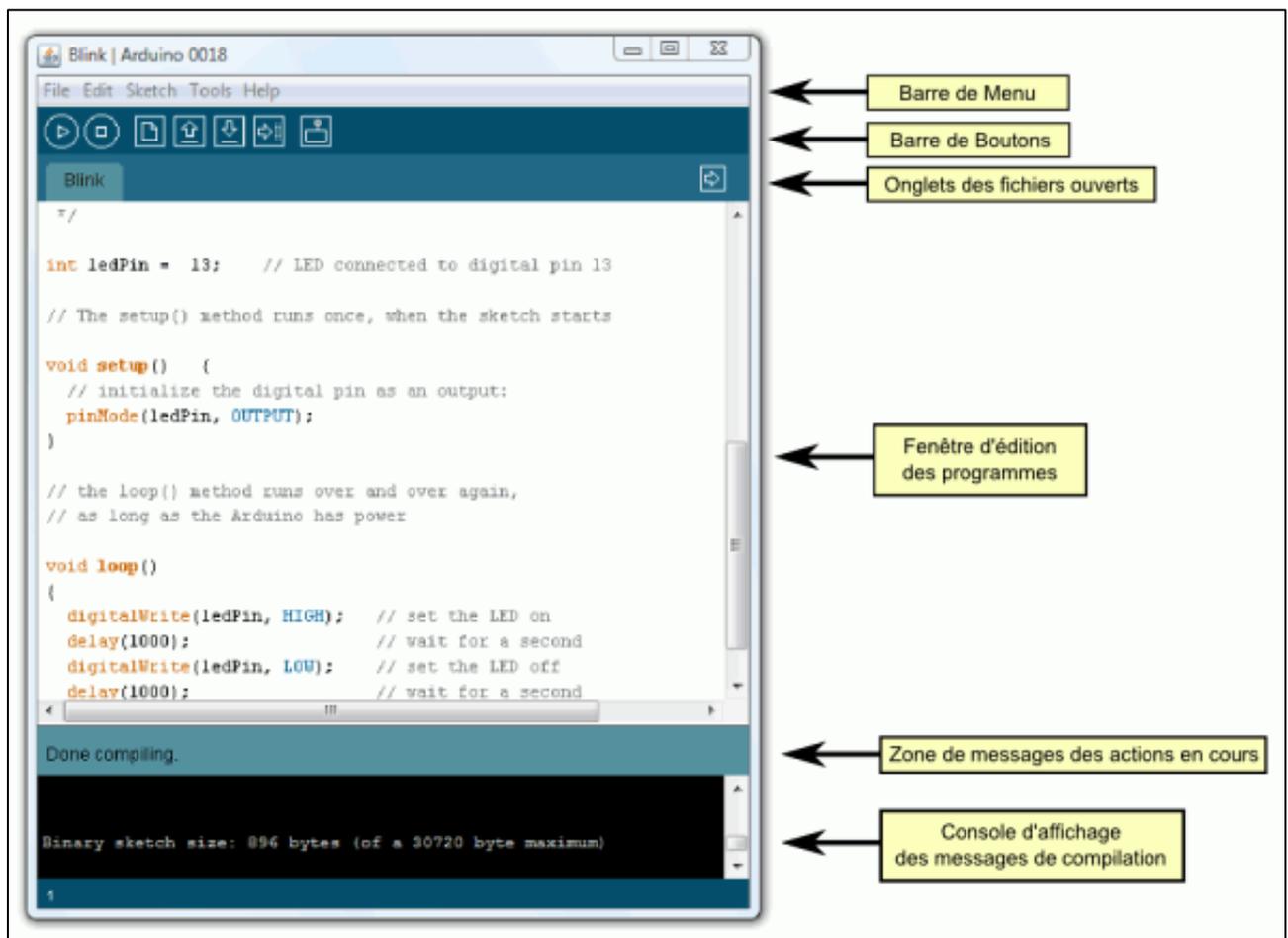


Figure 4.4. Interface IDE Arduino.

Chapitre 4 Arduino conception et programmation

- **Injection du programme :**

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 12) comme à titre d'exemple cette figure suivante.

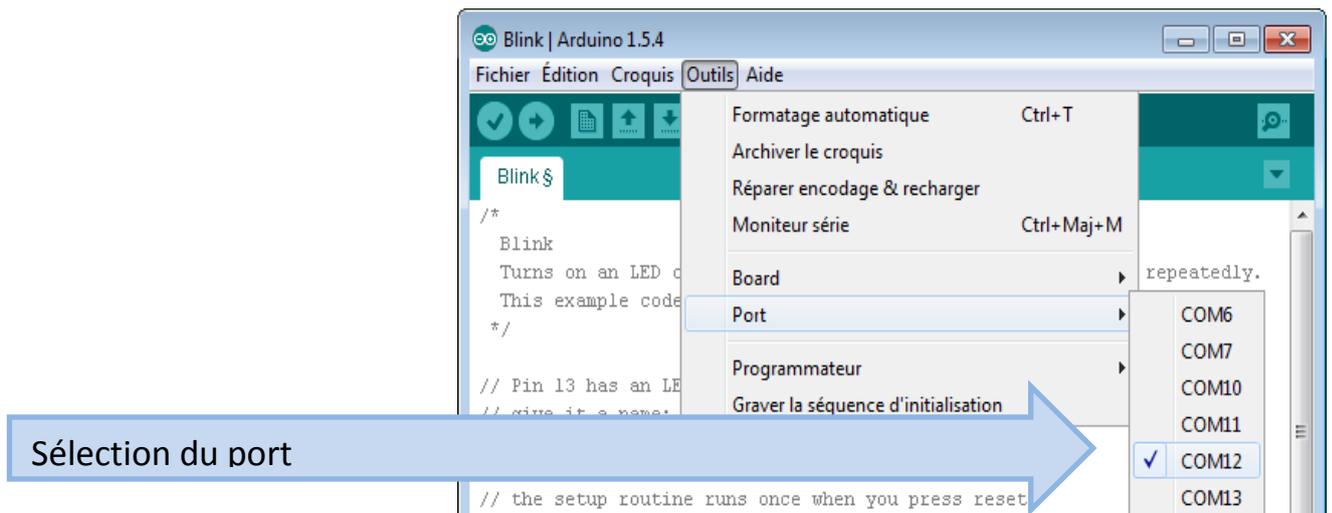


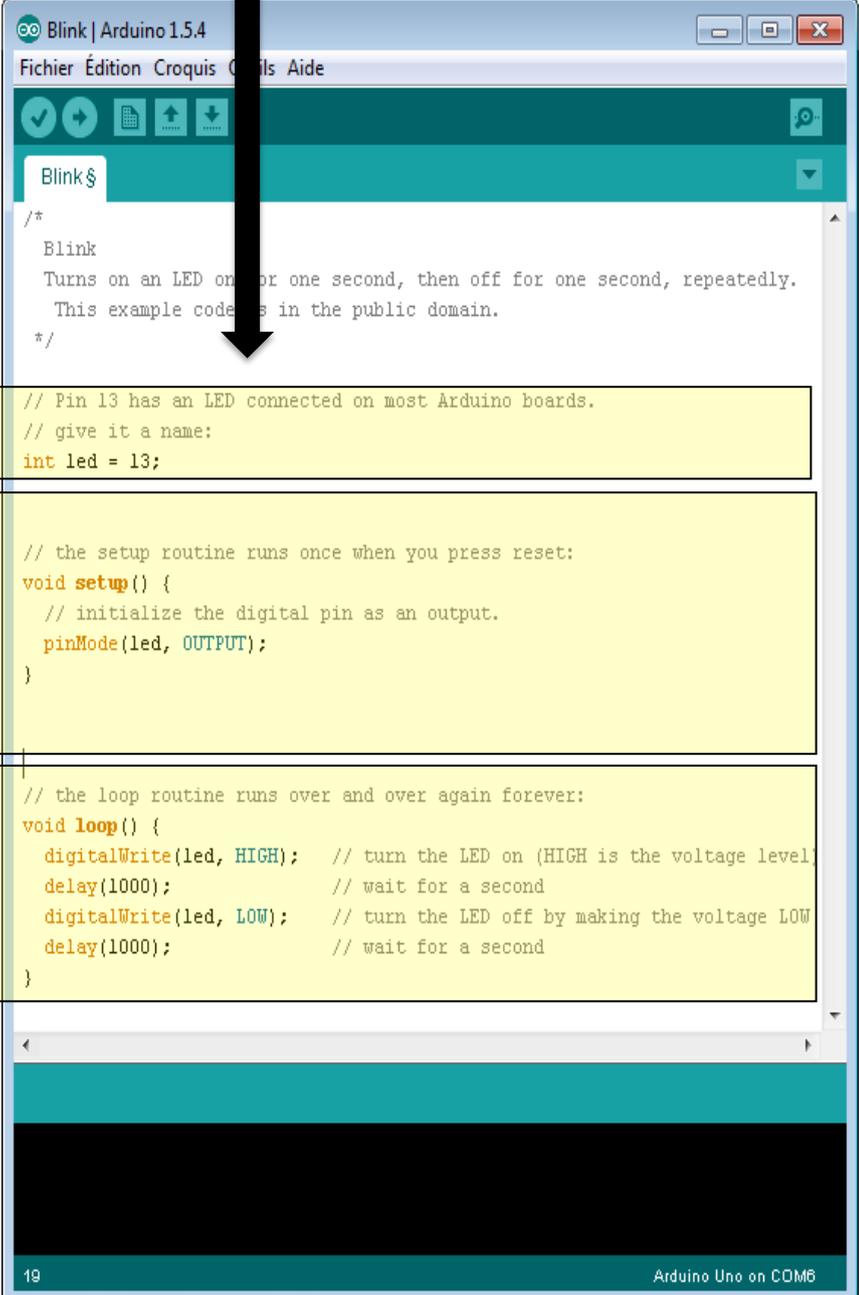
Figure 4.5. Paramètre de la carte.

- **Description du programme :**

Un programme arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

Chapitre 4 Arduino conception et programmation

Les commentaires



	<pre>/* Blink Turns on an LED on for one second, then off for one second, repeatedly. This example code is in the public domain. */</pre>
Définition des constantes et variables globales	<pre>// Pin 13 has an LED connected on most Arduino boards. // give it a name: int led = 13;</pre>
Fonction principale : VOID SETUP () Initialisation des ressources de la carte Configuration des entrées/sorties Définition de la vitesse de fonctionnement Du port série, etc. Cette partie ne sera exécutée qu'une seule fois	<pre>// the setup routine runs once when you press reset: void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT); }</pre>
Fonction boucle : VOID LOOP () Description du fonctionnement général Du programme Gestion des interactions entre les entrées/sorties Cette partie sera exécutée en boucle	<pre>// the loop routine runs over and over again forever: void loop() { digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level) delay(1000); // wait for a second digitalWrite(led, LOW); // turn the LED off by making the voltage LOW delay(1000); // wait for a second }</pre>

Figure 4.6. Description du programme sur arduino.

- **Les étapes de téléchargement du programme :** Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers carte Arduino via le port USB.

Chapitre 4 Arduino conception et programmation

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

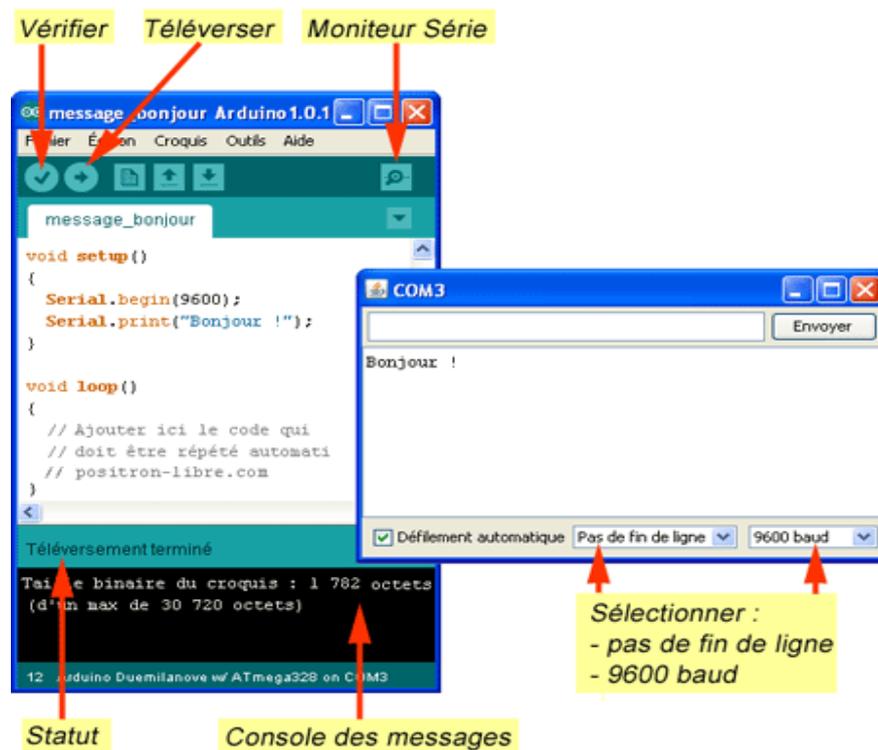


Figure 4.7. Les étapes de téléchargement du code.

4.5 Les Accessoires de la carte Arduino :

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

Chapitre 4 Arduino conception et programmation

4.5.1 Communication :

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications ; on peut éclaircir actuellement quelques types.

a Le module shield Arduino Wifi :

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.

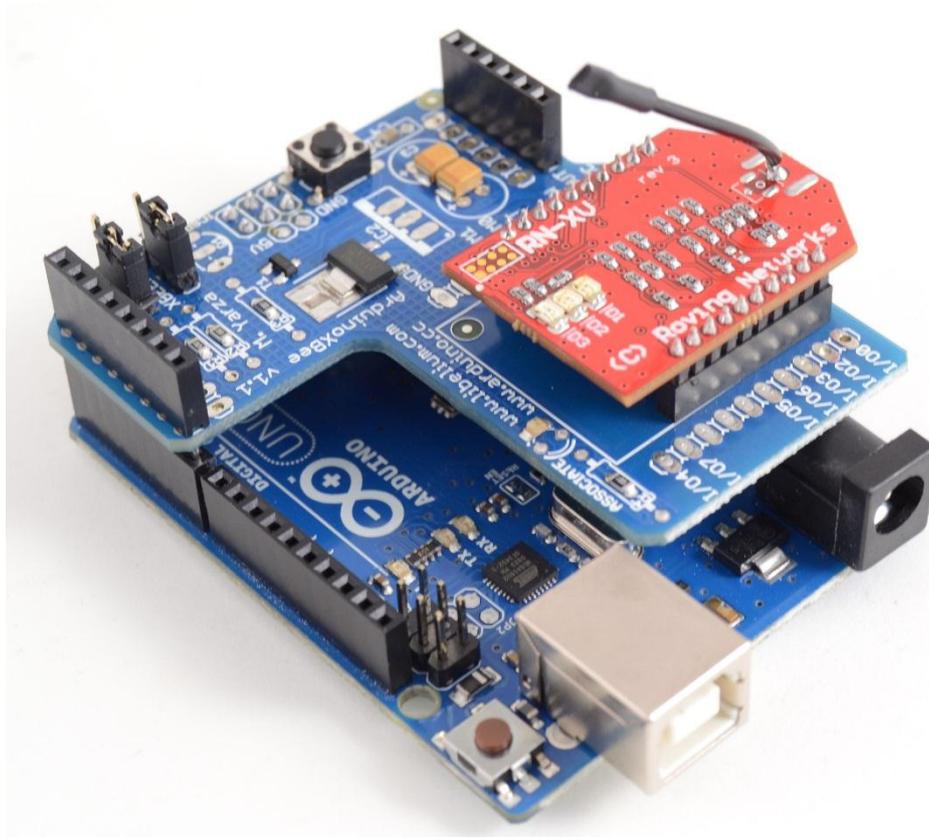
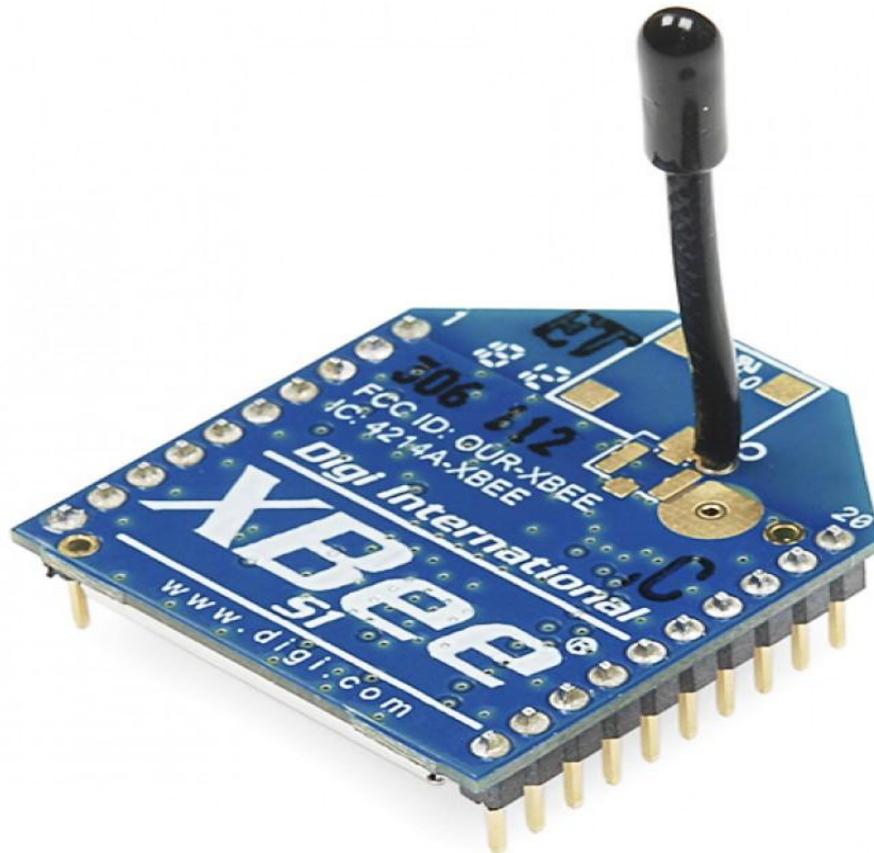


Figure 4.8. Module shield wifi.

b Le Module XBee :

Chapitre 4 Arduino conception et programmation

Ce module permet de faire de la transmission sans fil, faible distance /consommation/débit/ prix.



*Figure 4.9.*Module XBee.

c Le module Arduino Bluetooth :

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.

Chapitre 4 Arduino conception et programmation



*Figure 4.10.*Type de modules Bluetooth.

✓ *Présentation du module Bluetooth HC-06 :*

Introduction :

Comme expliqué plus tôt, le module utilisé se nomme HC-06 et est trouvable assez facilement pour 1800 dinars algérien. Il est aussi gros que le pouce et est en fait un montage d'un module Bluetooth sur un petit PCB. Cela permet de s'affranchir de certaines contraintes comme la soudure du module (qui est très délicate), la conversion 5V -> 3.3V, la régulation de l'alimentation (3.3V de nouveau) ou encore l'ajout de LEDs de signal. Tout d'abord, un ensemble de broches. VCC et GND pour l'alimentation (5V), Rx/Tx pour la communication. [16]

Chapitre 4 Arduino conception et programmation

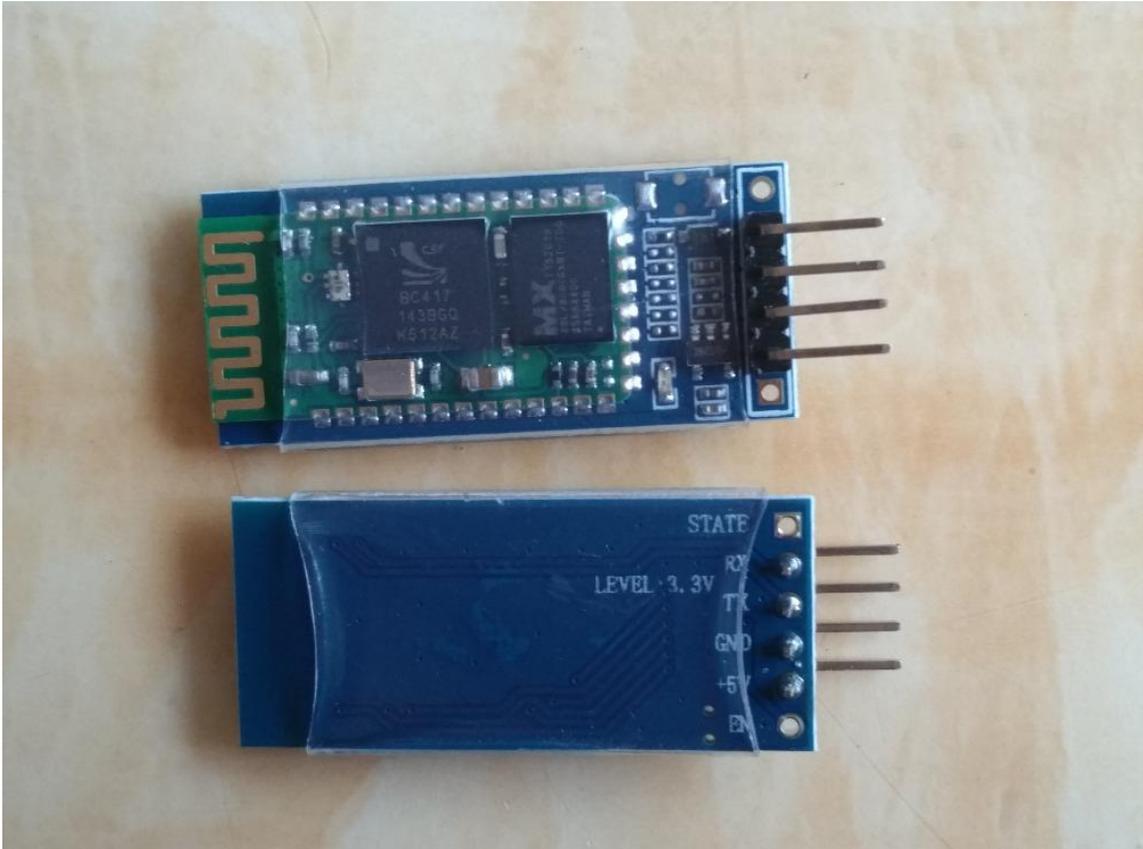


Figure 4.11. Module Bluetooth hc-06.

Avantages du Bluetooth :

- Consomme moins que le Wifi
- Peu encombrant (cet avantage n'est pas pris en compte avec un shield Arduino)
- Meilleur débit
- Module moins chère que le Wifi
- Le Bluetooth permet le mode "parked", qui signifie qu'une trame peut être envoyée à tout le monde en même temps.

✓ Communication entre Bluetooth hc-06 et arduino :

Chapitre 4 Arduino conception et programmation

Maintenant que le module est connecté, il est temps de le mettre en œuvre avec notre Arduino ! Pour garder les choses simples, nous allons simplement faire une connexion qui permet de renvoyer tout sur la voie série de l'ordinateur. [16]

- ✓ Utiliser la bibliothèque :

Pour mettre en œuvre cette bibliothèque c'est assez simple. Il va tout d'abord falloir l'inclure dans notre projet. Au choix, soit en cliquant sur « Library/Import/SoftwareSerial » dans l'IDE Arduino (inclus de base) ou alors en ajoutant la ligne suivante en haut de votre programme :

```
1.#include <SoftwareSerial.h>
```

Une fois que la bibliothèque est là, il nous faudra l'utiliser. Cette bibliothèque nous donne l'accès à un objet nommé... SoftwareSerial.

On peut donc créer une instance de cet objet en écrivant :

```
SoftwareSerial mavoieserie (Rx, Tx) ;
```

Une fois cela fait, tout est prêt ! En effet, cette bibliothèque offre les mêmes fonctions que la voie série habituelle, la seule différence sera qu'au lieu d'utiliser `Serial` on utilisera `mavoieserie`, comme on peut le voir dans le listing ci-dessous :

```
1. mavoieserie.begin (speed) ; // démarre la voie série à la vitesse speed
```

```
2. mavoieserie.available () ; // retourne le nombre de caractère à lire
```

```
3. mavoieserie.read () ; // retourne le prochain caractère reçu
```

```
4. mavoieserie.write (val) ; // envoie le char "val" sur la voie série
```

Notre module fonctionnant par défaut en 9600 bauds, nous allons pouvoir le préparer en écrivant :

Chapitre 4 Arduino conception et programmation

```
1. #include <SoftwareSerial.h>
2.
3. SoftwareSerial mavoieserie (10, 11);
4.
5. void setup () {
6. mavoieserie.begin(9600);
7. }
```

Vous devriez maintenant être en mesure d'utiliser votre module Bluetooth HC-06. Il ne reste plus qu'à coder une petite application sur votre PC ou un téléphone et vous pourrez faire du pilotage à distance complètement autonome, sans fil relié à un ordinateur pour recevoir les ordres.

4.5.2 Les capteurs :

a Introduction :

Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis. Il fournit cette information grâce à une électronique à laquelle il est associé.

b Le capteur ultrason :

Les capteurs ultrasons fonctionnent en mesurant le temps de retour d'une onde sonore inaudible par l'homme émise par le capteur. La vitesse du son étant à peu près stable, on en déduit la distance à l'obstacle. [17]

Les capteurs ultrasons fournis ont souvent la forme d'une paire d'yeux car il y a deux parties essentielles :

Chapitre 4 Arduino conception et programmation

- L'émetteur
- Le récepteur



Figure 4.12. Capteur ultrason hc-sr04.

- Voici un tableau présentant les principaux avantages et inconvénients des différentes

Capteur de distance :

	Ultrason	Infrarouge	Laser
Portée	De 1 à 250 cm	De 5 à 80 cm	Plusieurs mètres à plusieurs dizaines de mètres selon les modèles.

Chapitre 4 Arduino conception et programmation

Directivité	Cône d'environ 30°	Cône d'environ 5°	Les plus directifs (de l'ordre du degré, voire du demi-degré)
Précision	Relativement précis mais la précision diminue avec la distance, l'angle de mesure et les conditions de température et de pression.	Relativement précis mais la précision diminue avec la distance.	Sont précis avec un bruit de quelques centimètres sur des mesures de plusieurs mètres.
Coût	Peu chers	Peu chers	Relativement chers
Sensibilité aux interférences	Sensible à la température et à la pression. Egalement sensible aux autres robots utilisant la même fréquence ce qui peut poser problème dans une compétition.	Sont sensibles aux fortes sources de lumière qui contiennent un fort rayonnement infrarouge. Sont également sensibles à la couleur et à la nature des obstacles.	Ne peut pas détecter les objets réfléchissant la lumière laser (vitres, objets chromés,...)

Tableau 4.1. Avantages et inconvénients du capteur distance.

4.5.3 Les drivers :

a Introduction

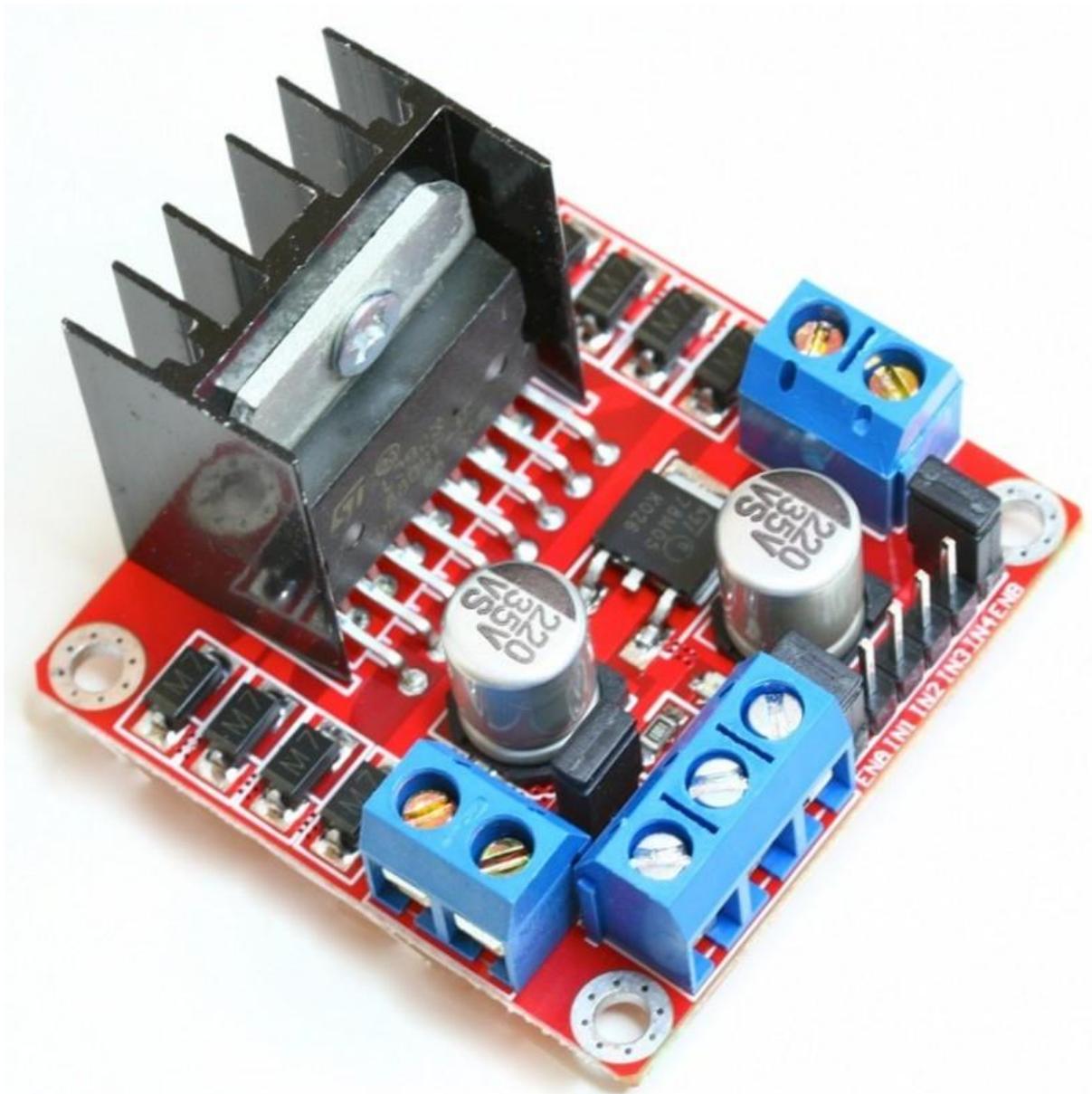
Il existe plusieurs drivers comme des cartes auxiliaires qui peuvent être attachées avec L'arduino afin de faciliter la commande ; on peut citer quelques types.

Chapitre 4 Arduino conception et programmation

b Pont-h L298 :

Ce break out board est un Double Pont-H destiné au contrôle de moteur continu (H-Bridge Motors Driver). Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation. C'est un module extrêmement utile pour le contrôle de robots et ensembles mécanisés. Il peut contrôler deux moteurs courant continu ou un moteur pas-à-pas 4 fils 2 phases. Il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

Il peut piloter des charges inductives comme des relais, solénoïdes, moteurs continus et moteurs pas-à-pas. Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retour. Il s'agit d'un module prêt à l'emploi. [18]



*Figure 4.13.*L298n double pont-h.

Ce tableau contient la caractéristique et la spécification du pont-h :

Chapitre 4 Arduino conception et programmation

Caractéristiques :	Spécifications :
<ul style="list-style-type: none"> • Léger, petit • Des capacités hors-pair pour contrôle moteur • Diodes de protections • Un dissipateur (pour dissiper la chaleur en cas de forte charge) • Un sélecteur pour sélectionner la source d'alimentation • 4 Sélecteurs pour les résistances pull up • Sortie pour 2 moteurs continu/ 1 moteur pas-à-pas (4 bobines, deux phases) • Indicateur LED pour sens de rotation moteur • Indicateur LED pour alimentation 5V • 4 trous de fixation standard 	<ul style="list-style-type: none"> • Composant de contrôle en puissance : L298N • Alimentation de la charge : de +6V à +35V • Courant Max (en pointe) : 2A • Tension de commande logique Vs : de +5 à +7V (alimentation interne de +5V) • Courant de commande logique : de 0 à 36mA • Tensions pour contrôle du sens : Low - 0.3V~1.5V, high : 2.3V~Vss • Tensions pour contrôle "Enable": Low - 0.3V~1.5V, high: 2.3V~Vss • Puissance Max : 25W (Température 75 °C) • Température de fonctionnement : de -25°C à +130°C • Dimensions : 60mm x 54mm • Poids : ~48g

Tableau 4.2. Caractéristique et spécification pont-h l298n.

Ce tableau contient la description technique du pont-h :

Chapitre 4 Arduino conception et programmation

Nom du connecteur	Direction	Description	Utilisation
VMS GND	/	Connecté à l'alimentation externe (6V~35V)	<p>ENA(ENB) raccordés à un niveau haut (HIGH) activera MOTORA(MOTORB).</p> <p>IN1(IN3) raccordés à 5V et IN2(IN4) à GND MOTORA(MOTORB) tournera dans le sens horlogique.</p> <p>IN1(IN3) raccordés à GND et IN2(IN4) à 5V MOTORA(MOTORB) tournera dans le sens Anti-horlogique.</p> <p>Si vous voulez contrôler la vitesse, vous pouvez connecter ENA(ENB) sur une sortie PWM.</p>
ENA	Entrée	Compatible TTL, peut donc être utilisé directement avec Arduino. Entrée "Enable" : un niveau bas LOW désactive le Pont A	
IN1	Entrée	Compatible TTL. Entrée de commande de sens du pont A.	
IN2	Entrée	Compatible TTL. Entrée de commande de sens du pont A.	
ENB	Entrée	Compatible TTL. Entrée "Enable» : un niveau bas LOW désactive le Pont B	
IN3	Entrée	Compatible TTL. Entrée de commande de sens du pont B.	
IN4	Entrée	Compatible TTL. Entrée de commande de sens du pont B.	
MOTORA	Sortie	Sortie du pont A. Commande en puissance du Moteur A	
MOTORB	SORTIE	Sortie du pont A. Commande en puissance du Moteur B	
5V +5V	/	Sortie 5V	
Jumper activant le régulateur 5V	/	5V Chip Enable Jumper. Le régulateur 5 volts est activé sur le jumper est connecté.	

Chapitre 4 Arduino conception et programmation

Tableau 4.3. Détails technique du circuit l298n.

c Les moteurs électriques :

Mini motoréducteur 6V (dc moteur) :

Ce moteur + engrenages, par définition, motoréducteur possède une tension de commande de 6V et une vitesse de rotation de 180 tours/min.



Figure 4.14. Moteur avec motoréducteurs et roue.

Spécifications :

- Rapport de réduction : 1:120
- Vitesse hors-charge (3V) : 100 tours/min
- Vitesse hors-charge (6V) : 200 tours/min
- Courant hors-charge (3V) : 60mA

Chapitre 4 Arduino conception et programmation

- Courant hors-charge (6V) : 71mA
- Courant d'arrêt (3V) : 260mA
- Courant d'arrêt (6V) : 470mA
- Dimension : 55mm x 48.3mm x 23mm
- Poids : 45g

4.5.4 Autre accessoire :

a Bread board :

La Breadboard n'est pas une planche à pain comme son nom l'indique, mais une "plaque à essai" ou plaquette de prototypage rapide pour l'électronique. Il n'y a pas besoin de souder les composants sur la Breadboard, car c'est prévu pour tester rapidement des circuits électroniques.



Figure 4.15. Mini breadboard

Cette plaque à essai est vraiment simple à utiliser. Dans une Breadboard standard, les deux lignes du haut et du bas sont reliées électriquement. De même, chaque colonne au centre est composée de points reliés électriquement. La création d'un circuit électronique est ainsi grandement facilitée. Il suffit d'enficher quelques composants dans les trous et d'utiliser quelques fils électriques.

Chapitre 4 Arduino conception et programmation

b Fils pour arduino :

Généralement utilisé pour les projets de l'électronique (pour Arduino) ils sont flexibles, d'usage, réutilisable, facile à tracer, facile à connecter et déconnecter sans la soudure.

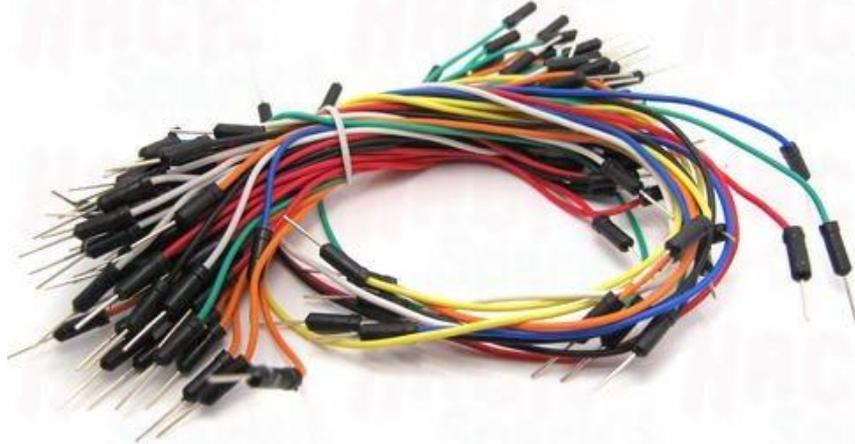


Figure 4.16. Fils pour arduino.

c Bouton poussoir :

Ces boutons poussoirs nous servent de capteur de collision. Les BP sont installés de façon à ce que le module appuie dessus lorsqu'il rencontre un obstacle. De ce fait, une interruption sera déclenchée au niveau de l'arduino.



Figure 4.17. Bouton poussoir.

Chapitre 4 Arduino conception et programmation

Spécifications :

- Couleur : Noir
- Courant de contact : 100mA
- Courant de contact c.c. max.: 100mA Diamètre: 12mm
- Durée de vie, mécanique : 250000
- Largeur (externe) : 11.4mm Longueur, course max.: 0.8mm
- Longueur/hauteur : 11.5mm

4.6 Programmation du robot (robot télécommandé et détecte obstacle) :

4.6.1 List du besoin :

- ❖ Arduino uno.
- ❖ Module Bluetooth hc-06.
- ❖ Fils.
- ❖ Bouton poussoir.
- ❖ Batterie 12volts..
- ❖ Pont-h l298n.
- ❖ Capteur ultrasons.
- ❖ Smartphone avec système androïde.
- ❖ Mini breadboard.
- ❖ Châssis avec 2 minis moteurs réducteurs et roue libre.

Chapitre 4 Arduino conception et programmation

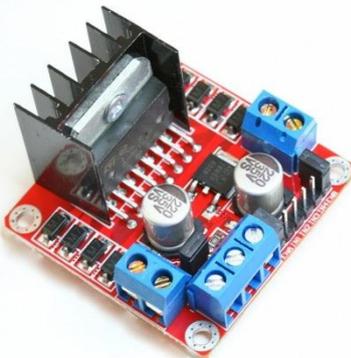
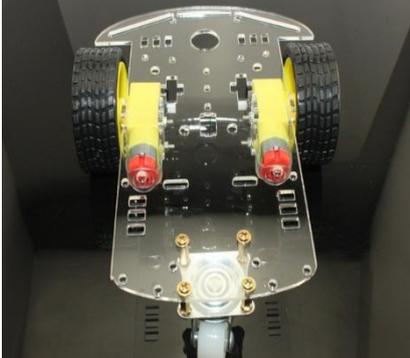
<p>Arduino uno</p> 	<p>fils</p> 	<p>Pont h I298n</p> 	<p>Chassis 2 moteur</p> 
<p>Mini bread board</p> 	<p>Bluetooht hc-06</p> 	<p>Smartphone android</p> 	<p>Capteur ultrason</p> 
<p>Bouton pousseire</p> 	<p>Batterie 12v</p> 		

Tableau 4.4. List du besoin dans notre projet.

Chapitre 4 Arduino conception et programmation

4.6.2 Montage du robot :

- Schéma du montage Avec les site www.fritzing.org :

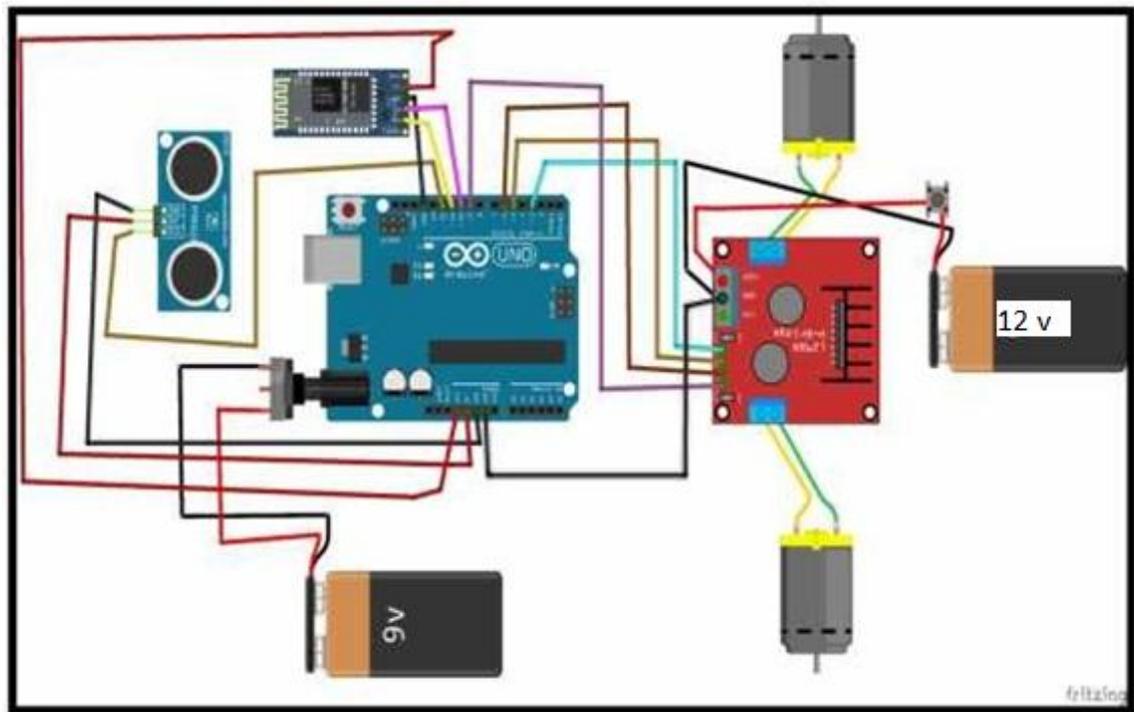


Figure 4.18. Schéma de montage du robot.

- Apres le montage voici le résultat :



Figure 4.19. Le robot.

Chapitre 4 Arduino conception et programmation

4.6.3 Code source de projet :

- la première partie :

```
#include <SoftwareSerial.h>
#include "Ultrasonic.h"
Ultrasonic ultrasonic (12,13); //define pin of ultrasonic
boolean cond = false;
int distance = 0;
boolean Rotate=true;
SoftwareSerial BT(10, 11); //TX, RX respetively
String readdata;
const int Motor1Pin1 = 3;
const int Motor1Pin2 = 5;
const int Motor2Pin2 =6;
const int Motor2Pin1 = 9;
int discm;
int fDistance;
int lDistance;
int rDistance;
```

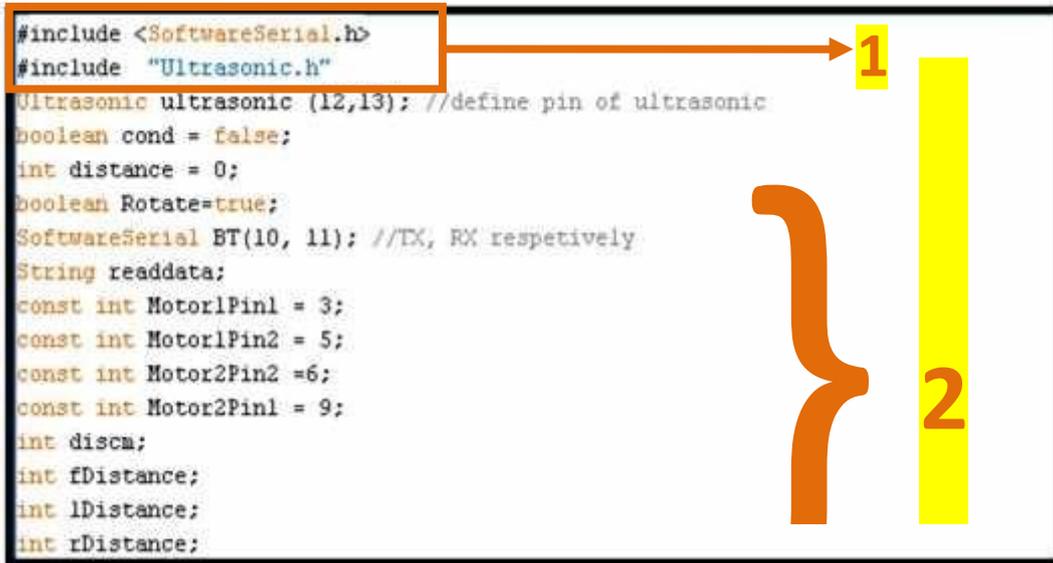


Figure 4.20. Code source 1.

- 1- Déclaration de la bibliothèque du Bluetooth et capteur ultrason.
- 2- Déclaration des pins du Bluetooth ultrason et les moteurs.

- La deuxième partie du programme .INO :

```
int scan(){
    distance=ultrasonic.Ranging(CM);
    delay (20);
    discm = distance * 1.27; // /29/2
    return discm ;
    delay(10);
}

void moveBackward(){
    digitalWrite (Motor1Pin1, HIGH);
    digitalWrite (Motor1Pin2, LOW);
    digitalWrite (Motor2Pin1, LOW);
    digitalWrite (Motor2Pin2, HIGH);
}
```

Figure 4.21. Code source 2.

Chapitre 4 Arduino conception et programmation

3- Fonction scan : pour ultrasons allant Module HC - SR04.

4- Fonction marche-avant : pour le robot et commande de la vitesse du moteurs avec l'instruction " `digitalwrite (pin, valeurs)` " valeurs est de 0 à 255.

- La troisièmes parties :

```
//initialization all components and all pins
void setup() {
  BT.begin(9600);
  Serial.begin(9600);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(9, OUTPUT);
}
```

Figure 4.22. Code source 3.

5- La responsabilité de " `Void setup` " est la broche d'initialisation pour le mettre les sortie ou les entrées. Ainsi que réglée la vitesse du bit de communication par seconde par l'instruction « `Serial.begin (9600) ;` » du module Bluetooth hc-06.

- La quatrième partie du programme .INO :

```
void loop() {
  blu();
  if(cond == true){
    //check if arduino receive data
    if (readdata.length() > 0) {
      Serial.println(readdata); //print data after Reception
      //if arduino Receive word 'forward' from app android to do fonction forward.
      if(readdata == "forward")
      (
        moveForward();
        delay(100);
      )
    }
  }
}
```

Figure 4.23. Code source 4.

Chapitre 4 Arduino conception et programmation

6- la boucle `Void loop ()` : est le programme principal et cette fonction pour permettre de changer le programme et de répondre, et cette boucle est fonction (répétition).

7- l'instruction « `if (readdata.lenght ()>0) {Serial.println (readdata) ; } »`

Permis de lire les valeurs qui arrive par Bluetooth et exécuté la fonction qui concerne sa fonction soit marche avant, marche arrière, gauche, droit, stop et changer la vitesse du robot.

- Finalement voici les résultats finals du robot mobile et ça application androïde la commande à distance

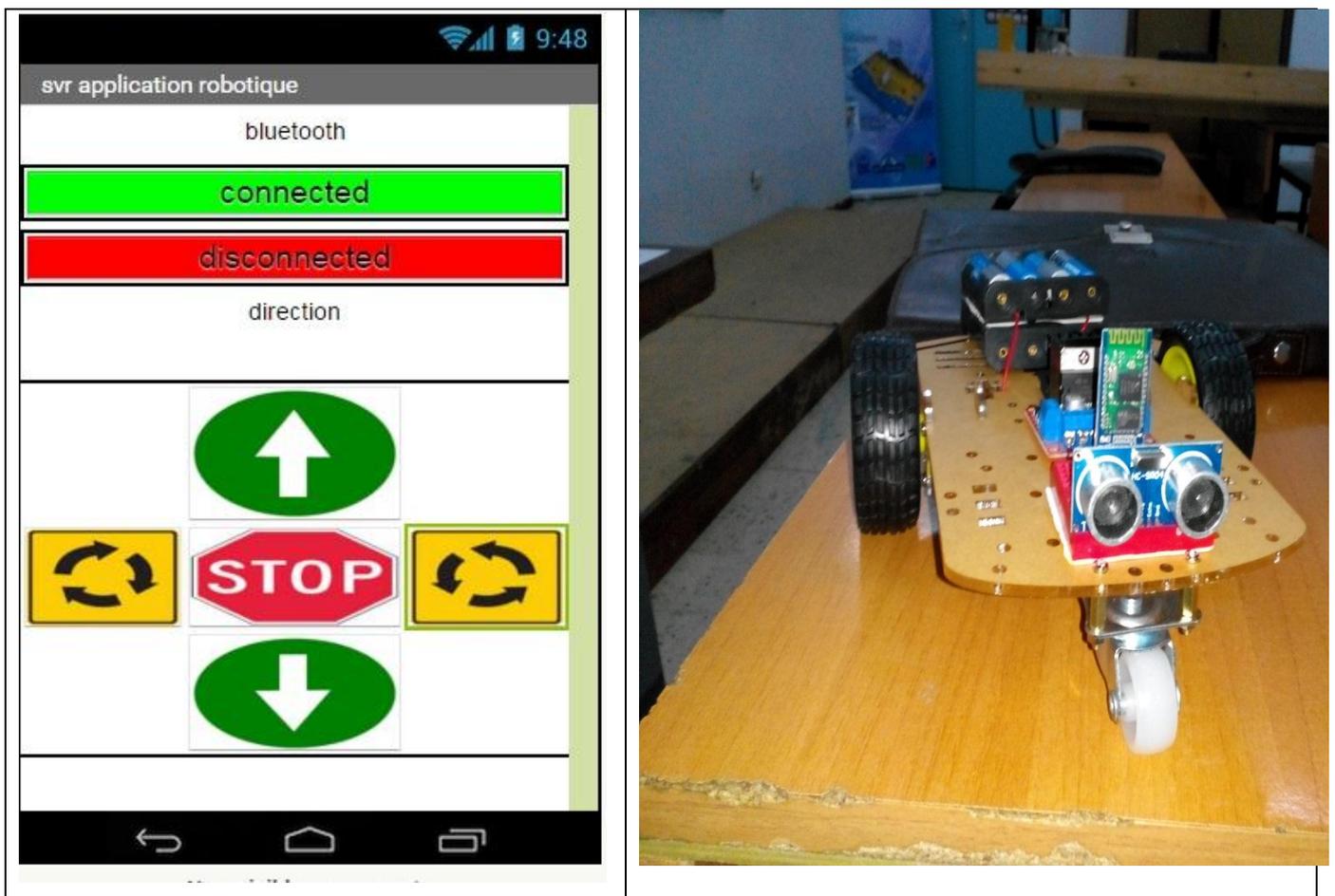


Figure 4.24 . Finale du projet.

Chapitre 4 Arduino conception et programmation

4.7 Conclusion :

Dans ce chapitre, nous avons projeté la lumière sur une carte d'acquisition qui est l'Arduino donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité des différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino ; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

Annexes

La fiche technique de pont-h L 298n :

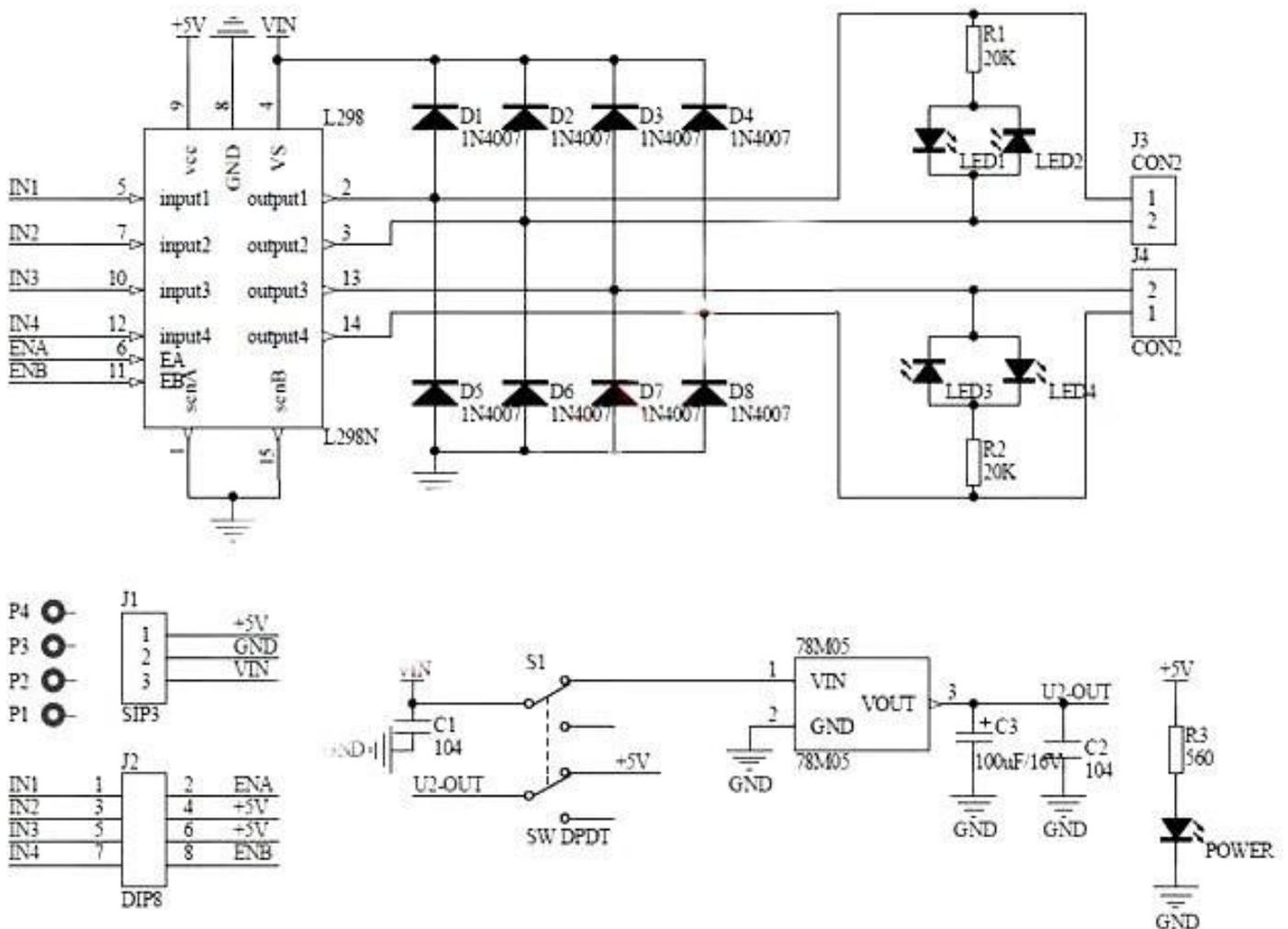


Schéma simplifié de la carte Arduino UNO :

