

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE TECHNOLOGIE  
DEPARTEMENT: ELECTRONIQUE  
N° : 2021/ESEM



FILIERE : SCIENCES ET TECHNOLOGIE  
OPTION : ELECTRONIQUE DES SYSTEMES  
EMBARQUES

**Mémoire présenté pour l'obtention  
Du diplôme de Master Académique**

**Par: KADDOUR SABER et ZANAT SOHEYB**

**Intitulé**

**Etude et réalisation d'un robot mobile commandé via  
androïde**

**Soutenu devant le jury composé de:**

.....	Université MSILA	Président
TABBAKH MOSTEFA	Université ... ..	Rapporteur
.....	Université ... ..	Examineur

**Année universitaire : 2020 /2021**

**Résumé :** Aujourd'hui l'électronique 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 sommes arrivés dans notre projet à la création d'un "robot" qui peut éviter tout obstacle et est commandé à travers une application du système android.

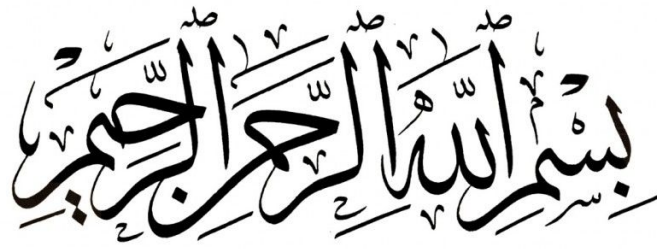
**Mots clés :** électronique ; arduino ; obstacle ; robot ; microcontrôleurs ; système android.

**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.

**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 مسبقه بعلم الإلكترونيات و دون دراسة تعقيدات الدوائر الإلكترونية , وتتكون هذه اللوحة من دائرة إلكترونية تحتوى على متحكم دقيق قابل للبرمجة عن طريق برنامج بيئة التطوير المتكاملة للأردوينو و الذي يمكن تحميله مجانا لجميع أنظمه 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.



## *Dédicaces*

*Je dédie ce travail :*

- ✚ A mon cher père et ma chère mère Que Dieu préserve bonne santé et longue vie*
- ✚ A mes chers frères et sœurs.*
- ✚ A toute ma famille.*
- ✚ A mes chers amis*
- ✚ Ainsi que la promotion Master ESEM sans oublier tous mes enseignants durant tout mon cursus*
- ✚ A tous ceux qui ont sacrifié leur temps pour la science et à tous ceux qui utilisent la science pour le bien et la prospérité de l'humanité.*
- ✚ Merci à tous.*

***SABER KADDOUR***

## *Dédicaces*

*Je dédie ce travail :*

- ✚ A mon cher père et ma chère mère Que Dieu préserve bonne santé et longue vie*
- ✚ A mes chers frères et sœurs.*
- ✚ A toute ma famille.*
- ✚ A mes chers amis*
- ✚ Ainsi que la promotion Master ESEM sans oublier tous mes enseignants durant tout mon cursus*
- ✚ A tous ceux qui ont sacrifié leur temps pour la science et à tous ceux qui utilisent la science pour le bien et la prospérité de l'humanité.*
- ✚ Merci à tous.*

***SOHEYB ZANAT***

## *Remerciements*

Nous remerciant **ALLAH** qui nous aide et nous donne la patience et le courage durant ces longues années d'étude.

Nous tiens à remercier grandement nos Encadreur Dr: **TABBAKH MOUSTEFA** pour sa grande disponibilité et ses précieux conseils.

Nous remerciant également tous les enseignants du département d'électronique d'université de M'sila

plus spécialement les membres de jury de notre travail.

Nous remercions également vivement l'étudiant, frère et ami **MOUSSA QUENANI**

Enfin, nous adressons nos plus sincères remerciements à tous nos amis et collègues qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Merci à tous.

# SOMMAIRE

## Chapitre i : Généralité sur la robotique

Introduction .....	3
La Robotique .....	3
Historique de la robotique .....	3
Définition 4	
Les types des robots .....	4
Les robots manipulateurs .....	4
Les types des robots manipulateurs .....	5
a) Robots cylindriques .....	5
b) Robots rectilignes .....	6
c) Robots sphériques .....	6
d) Robots articulés .....	7
e) Robots SCARA .....	7
Les robots mobiles .....	7
Architecture des robots mobiles .....	7
Classification des Robots Mobiles .....	8
Les robots mobiles à roues .....	8
a) Robot unicycle .....	8
b) Robot tricycle .....	9
c) Robot voiture .....	10
e) Robot omnidirectionnel .....	10

# SOMMAIRE

1. Les robots mobiles à chenilles .....	11
2. Les robots mobiles marcheurs .....	12
3. Les robots mobiles rampants .....	12
<b>Domaines d'utilisation des robots .....</b>	<b>13</b>
<b>Avantages et inconvénients des robots .....</b>	<b>14</b>
<b>I.3 Conclusion .....</b>	<b>14</b>

## chapire II Circuit de commande Arduino

<b>Circuit de commande : Arduino</b> .....	<b>15</b>
Introduction .....	15
Présentation générale de module Arduino .....	16
Description technique .....	16
Alimentation .....	16
Horloge .....	17
Reset .....	17
Entrées/sorties .....	18
Mémoire .....	19
Les gammes de la carte Arduino .....	20
La carte Arduino UNO .....	22
Présentation .....	22
les Caractéristiques techniques .....	24
Avantages .....	25
<b>Chapitre 3 : Les applications Android</b> .....	<b>26</b>
Introduction .....	26
Présentation Android .....	26
Présentation d'App inventor .....	27
Créé un projet .....	28
Application permettant la commande du robot .....	30
Connexion de la smartphone avec le robot par Bluetooth .....	30
mise en fonctionnement du robot .....	31
Création de l'application «svr application robotique» .....	33
Conclusion .....	34
<b>Chapitre 4 : Arduino conception et programation</b> .....	<b>35</b>
Introduction .....	35
Partie matérielle .....	35

Partie programme.....	35
Les Accessoires de la carte Arduino .....	38
Communication .....	38
a Le module shield Arduino Wifi .....	38
b Le Module XBee .....	39
c Le module Arduino Bluetooth .....	39
Les capteurs.....	42
a Introduction.....	42
<i>b Le capteur ultrason</i> .....	42
Les drivers .....	42
a Introduction.....	42
b Pont-h L298 .....	43
c Les moteurs électriques .....	45
Autre accessoire.....	46
<i>a Breadboard</i> .....	46
<i>b Fils pour arduino</i> .....	47
<i>c Bouton poussoir</i> .....	47
Programmation du robot (robot télécommandé et détecte obstacle).....	48
List du besoin .....	48
Montage du robot .....	48
Code source de projet .....	49
Conclusion .....	52
<b>Conclusion générale .....</b>	<b>54</b>
<b>Annexe.....</b>	<b>55</b>
<b>Bibliographie .....</b>	<b>61</b>

## Liste des figures

### Chapitre I

<i>Figure I.1: Robot cylindrique</i> .....	5
<i>Figure I.2 : Robot rectiligne</i> .....	6
<i>Figure I.3 : Robot sphérique</i> .....	6
<i>Figure I.4 : Robot articulé</i> .....	6
<i>Figure I.5 : Robot SCARA</i> .....	7
<i>Figure I.6 : Architecture d'un robot mobile</i> .....	8
<i>Figure I.7 : Robot mobile unicycle</i> .....	9
<i>Figure I.8 : Robot mobile tricycle</i> .....	9
<i>Figure I.9: Robot mobile voiture</i> .....	10
<i>Figure I.10: Robot mobile omnidirectionnel</i> .....	10
<i>Figure I.11 : Robots mobiles à chenilles</i> .....	11
<i>Figure I.12 : Robots mobiles marcheurs</i> .....	12
<i>Figure I.13: Robot mobile rampant</i> .....	12

### Chapitre II

<b>Figure</b>	<b>Nom de figure</b>	<b>Numéro de page</b>
<b>Fig(II.1)</b>	Schéma explicatif des différentes broches du microcontrôleur ATmega328	<b>20</b>
<b>Fig(II.2)</b>	Arduino UNO	<b>21</b>
<b>Fig(II.3)</b>	Arduino Mega	<b>21</b>
<b>Fig(II.4)</b>	Arduino Nano	<b>21</b>
<b>Fig(II.5)</b>	Arduino Leonardo	<b>21</b>
<b>Fig(II.6)</b>	Arduino LilyPad	<b>21</b>
<b>Fig(II.7)</b>	Arduino DUE	<b>21</b>
<b>Fig(II.8)</b>	Intel Galileo	<b>21</b>
<b>Fig(II.9)</b>	Description détaillée du brochage de la carte Arduino Uno	<b>22</b>

### Chapitre III

<i>Figure III.1. Site mit app inventor</i> .....	28
<i>Figure III.2. Première interface de la création App Inventor</i> .....	29

## Liste des figures

---

<i>Figure III.3.</i> L'interface de l'application.....	30
<i>Figure III.4.</i> Codesourcé d'application 1.....	31
<i>Figure III.5.</i> Codesourcé d'application 2.....	32
<i>Figure III.6.</i> Codesourcé d'application 3.....	32
<i>Figure III.7.</i> Codesourcé d'application 4.....	33
<i>Figure III.8.</i> Photo de notre application de contrôle.....	38

## Chapitre VI

<i>Figure VI.1.</i> Interface IDE Arduino .....	36
<i>Figure VI.2.</i> Paramètre de la carte.....	36
<i>Figure VI.3.</i> Description du programme sur arduino.....	37
<i>Figure VI.4.</i> Les étapes de téléchargement du code.....	38
<i>Figure VI.5.</i> Module shield wifi.....	39
<i>Figure VI.6.</i> Module XBee.....	39
<i>Figure VI.7.</i> Type de modules .....	39
<i>Figure VI.8.</i> Module Bluetooth hc-06.....	40
<i>Figure VI.9.</i> Capteur ultrason hc-sr04.....	42
<i>Figure VI.10.</i> L298n double pont-h.....	43
<i>Figure VI.11.</i> Moteur avec motoréducteurs et roue.....	45
<i>Figure VI.12.</i> Mini breadboard.....	46
<i>Figure VI.13.</i> Fils pour arduino.....	47
<i>Figure VI.14.</i> Bouton poussoir.....	47
<i>Figure VI.15.</i> Schéma de montage du robot.....	49
<i>Figure VI.16.</i> Le robot.....	49
<i>Figure VI.17.</i> Code source 1.....	50
<i>Figure VI.18.</i> Code source 2.....	50
<i>Figure VI.19.</i> Code source 3.....	51

## Liste des figures

---

<i>Figure VI.20.</i> Code source 4.....	51
<i>Figure VI.21.</i> Finale du projet.....	52

# Liste des tableaux

---

## Liste des tableaux

### Chapitre I

*Tab I.1: différents types de robots à roues*.....11

### Chapitre II

**Tab II.1** :Les caractéristiques de l'Arduino Uno .....25

### Chapitre VI

*Tableau 4.1.* Caractéristique et spécification pont-h l298n.....44

*Tableau 4.2.* Détails technique du circuit l298n.....45

# Introduction Générale

---

## Introduction Générale.

### 1 Généralités

La forte augmentation des ventes de smart phone et de tablettes électronique se fait en même temps qu'une adoption rapide par le grand public des technologies de la domotique ainsi que l'autopilotage. Au fond, le smart phone, avec sa connectivité Bluetooth intégrée, devient une télécommande universelle pour toute la maison et les équipements électriques. 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 carte qui permet de contrôler n'importe quel appareil, véhicule, machine à travers un smart phone ou à une tablette. Cette carte électronique réalisée rend la commande et le contrôle facile et souple lors du pilotage à distance.

### 2 Position du problème

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

Le smart phone occupe la première place d'objets qui ne nous quittent pas donc notre travail se concentre sur l'utilisation de ce dernier avec bien sûr sa liaison avec un système ou une carte de commande (carte d'interface) telle que l'Arduino.

### 3 Objectif du projet

Dans ce projet trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage (Arduino) : son langage de programmation, sa construction, son principe de fonctionnement.
- Le deuxième consiste à réaliser une carte électrique capable d'exécuter une action entre un smart phone et une carte d'interfaçage (Arduino) en expliquant les différents blocs de sa construction.
- Le troisième est de réaliser une application sous smart phone et de la programmer sous l'environnement JAVA afin de simplifier la commande et de montrer l'intérêt de la programmation orientée objet.

### 4 Présentation du mémoire

Le premier chapitre consiste à introduire une description sur la robotique.

Le deuxième chapitre sera consacré à une étude approfondie sur les cartes d'interface telles que l'Arduino, puis, on mettra la lumière sur un modèle de base qui est (Arduino UNO) sa construction son environnement de programmation et son principe de fonctionnement afin de

## Introduction Générale

---

simplifier son utilisation.

Dans le troisième chapitre et le quatrième, on présentera un algorithme; on expliquera le fonctionnement de notre programme, son déclenchement à l'action à distance et cela sur deux modes de commande (domotique et l'autoguidage) après le chargement du code simulé dans la carte Arduino et de réaliser une application capable de gérer une telle commande sous smarte phone.

Enfin, on terminera avec une conclusion générale qui résumera l'intérêt de notre étude : les différents résultats obtenus expérimentalement seront donnés sur annexe (A) et (B) sous forme de photos, donnant ainsi un aperçu sur les performances des carte d'interfaçage, et une idée sur les problèmes à résoudre ultérieurement.

**Introduction :**

Ces dernières années, les robots manipulateurs ont un impact considérable sur de nombreux aspects de la vie moderne, de la fabrication industrielle aux soins de santé, le transport et l'exploration de l'espace et les profondeurs de la mer. Ceci s'explique par le fait que de plus en plus, les processus industriels sont automatisés, afin de répondre à des problématiques de maîtrise de la qualité produite, de la productivité et des coûts. Une des méthodes d'automatisation repose sur la robotisation. Dans les prochaines années, des robots seront aussi omniprésents dans des processus industriels comme les ordinateurs personnels. En outre, ces robots utilisent des capteurs pour bien mener les différentes tâches et accomplir leurs missions.

Dans le but d'expliquer l'utilité des robots, ce chapitre sera consacré à la présentation des généralités sur la robotique.

**La Robotique :****Historique de la robotique :**

La robotique est passée par plusieurs générations comme suit [1] :

- 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.
- 1963 : Utilisation de la vision pour commander un robot.
- 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, il est équipé 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 est 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.
- Les robots mobiles actuels : A présent la plupart des travaux de recherche portent sur les problèmes de perception. Ainsi, la planification de trajectoires, l'analyse et la modélisation de l'environnement de robot, sont appliquées sur des robots mobiles commerciaux. Également la recherche actuelle s'oriente vers la conception mécanique des robots mobiles pour des applications hautement spécialisées, comme l'exploration sous-marine, les robots volants et le micro robot.

### **Définition :**

Le terme (Robot) prend son origine du mot slave (Paboma) (se prononce robota) qui veut dire en russe travail ou en tchèque corvée ou travail forcé. Il désigne aussi une machine à l'aspect humain, capable de se mouvoir et d'agir grâce à un mécanisme automatique pouvant effectuer certaines opérations, et capable par fois de modifier de lui-même son cycle de fonctionnement et d'exercer un certain choix.

### **Les types des robots :**

Il existe deux grandes familles de robots :

- Les robots manipulateurs (fixe).
- Les robots mobiles.

### **Les robots manipulateurs :**

Un robot manipulateur se présente sous forme d'un bras doté d'un certain nombre de segments articulés. Il est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. En parallèle, les robots sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement en toute sécurité. En effets, les

robots manipulateurs sont utilisés dans des applications industrielles pour effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage.

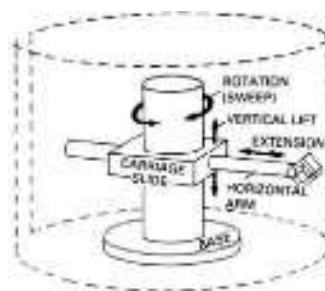
### Les types des robots manipulateurs :

Les robots manipulateurs se présentent sous plusieurs formes, qui sont réparties en cinq grandes catégories :

- Robots cylindriques
- Robots rectilignes
- Robots sphériques
- Robots articulés
- Robots SCARA

#### a) Robots cylindriques :

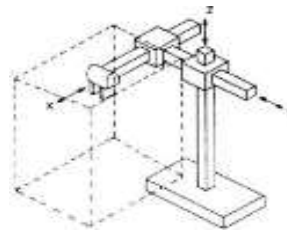
Le robot cylindrique, comme l'illustre la figure I.1, est composé essentiellement de deux axes de mouvement, dont le premier pour le mouvement en haut et en bas et le second est représenté par la rotation à travers la jonction de base. De plus, le bras horizontal peut se déplacer à l'intérieur et à l'extérieur, ce qui donne un troisième axe de mouvement [2].



*Figure I.1 : Robot cylindrique.*

#### b) Robots rectilignes :

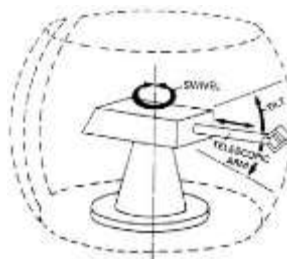
Le robot rectiligne est représenté à la figure I.2 possède trois axes de mouvement (x, y, z). Pour cette raison, il est parfois appelé Robot cartésien. Les robots de type de configuration exploitent le vérin pneumatique [2].



*Figure I.2 : Robot rectiligne.*

**c) Robots sphériques :**

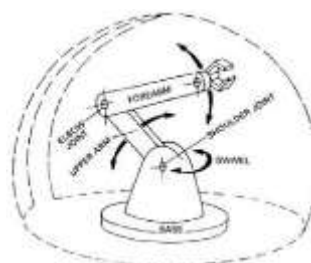
Le robot sphérique donné à la figure I.3 est de grande taille avec un bras télescopique qui assure un mouvement à l'intérieur ou à l'extérieur. Les mouvements de base du robot sphérique sont des rotations qui s'effectuent à la base et la rotation angulaire du bras horizontal, soit vers le haut ou vers le bas en haut ou en bas [2].



*Figure I.3: Robot sphérique.*

**d) Robots articulés :**

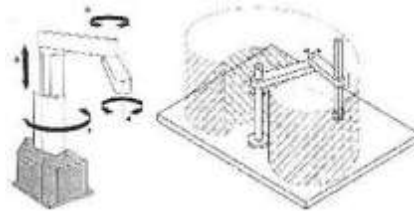
Le bras articulé du robot comme l'illustre la figure I.4 ressemble à un bras humain. Il se compose de deux éléments, nommés l'avant-bras et le bras supérieur [3]



*Figure I.4: Robot articulé.*

**e) Robots SCARA :**

Le terme « SCARA » signifie « Sélective Compliance Arm for Robot Assembly »  
Un robot SCARA est défini dans la norme ISO, en tant qu'un robot comportant deux liaisons pivotales parallèles pour fournir conformément à un plan sélectionné comme la montre la figure I.5. Il est peut-être considéré comme un cas particulier d'un robot cylindrique [4].



*Figure I.5: Robot SCARA.*

**Les robots mobiles :**

Un robot mobile est celui qui peut se déplacer dans son environnement de façon autonome. Pour ce faire, le robot doit pouvoir naviguer. Par conséquent, la portée et la précision de navigation requise varie en fonction de la taille du robot et du type de la tâche à réaliser.

**Architecture des robots mobiles :**

L'architecture des robots mobiles, est présentée à la figure I.6. Elle composée de quatre éléments à savoir :

- La structure mécanique et la motricité.
- Les organes de sécurité.
- Le système de traitement des informations et gestion des tâches.
- Le système de localisation.

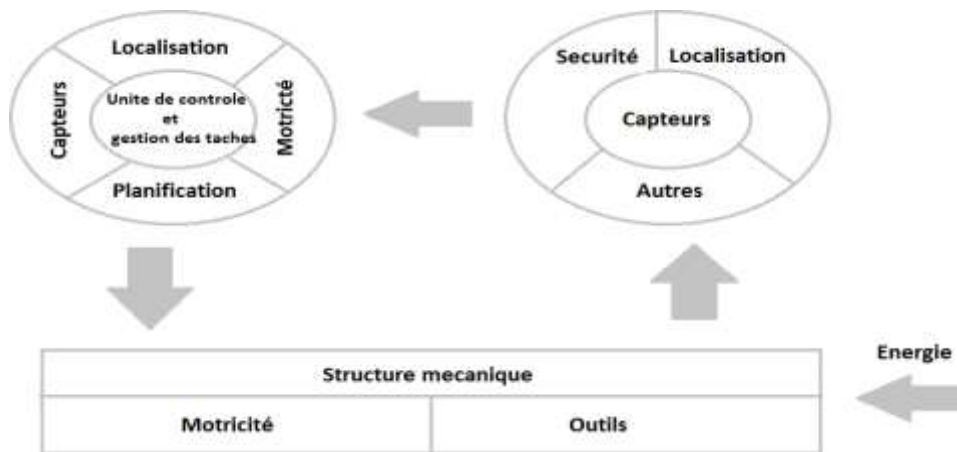


Figure I.6: Architecture d'un robot mobile.

### Classification des Robots Mobiles :

Selon le système de locomotion, on peut distinguer quatre types de robots mobiles :

#### 1. Les robots mobiles à roues :

La mobilité par roues est la structure mécanique la plus utilisée. Ce type de robot assure un déplacement avec une accélération et une vitesse rapide mais nécessite un sol relativement plat. On distingue plusieurs classes de robots à roues, déterminées principalement par la position et le nombre de roues utilisées.

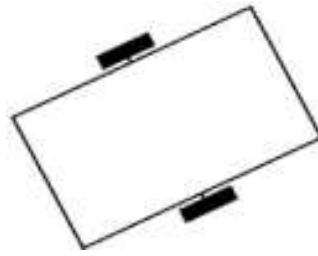
Nous citerons ici les quatre classes principales de robots mobiles à roues [5] :

##### a) Robot unicycle :

Un robot de type unicycle est actionné par deux roues indépendantes comme la montre la figure I.7, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

En effet, le fait que c'est un robot non-holonyme, il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

Sa commande peut être très simple, car il est assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites [6].

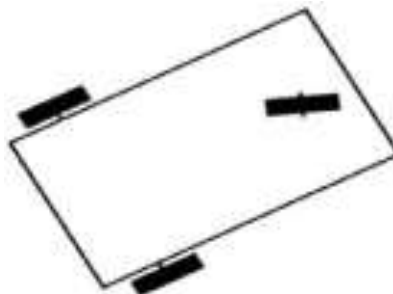


*Figure I.7: Robot mobile unicycle.*

### **b) Robot tricycle**

Un robot de type tricycle représenté par la figure I.8 est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable.

Né au moins, le fait que c'est un robot non-holonome, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable [6].



*Figure I.8: Robot mobile tricycle.*

### **c) Robot voiture :**

Un robot de type voiture donné à la figure I.9 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 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, en raison que le robot voiture peut être ramené au robot tricycle 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.

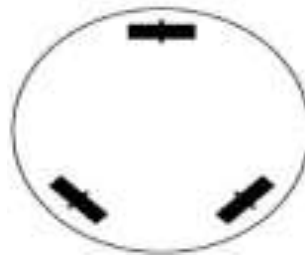


*Figure I.9: Robot mobile voiture*

#### e) Robot omnidirectionnel :

Un robot omnidirectionnel représenté à la figure I.10 est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées, orientables et 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 au détriment d'une complexité mécanique bien plus grande [6].



*Figure I.10: Robot mobile omnidirectionnel.*

#### ➤ Comparaison des différents types de robots mobiles à roues :

Nous pouvons observer dans le tableau ci-dessous un récapitulatif des avantages et des inconvénients des différents types de robots à roues :

Type du robot	Avantage	Inconvénient
<b>Unicycle</b>	<ul style="list-style-type: none"> <li>* Stable</li> <li>* Rotation sur soi-même</li> <li>* Complexité mécanique faible</li> </ul>	<ul style="list-style-type: none"> <li>* Non-holonome</li> </ul>
<b>Tricycle</b>	<ul style="list-style-type: none"> <li>* Complexité mécanique Modérée</li> </ul>	<ul style="list-style-type: none"> <li>* Non-holonome</li> <li>* Peu stable</li> <li>* Pas de rotation sur soi-même</li> </ul>
<b>Voiture</b>	<ul style="list-style-type: none"> <li>* Stable</li> <li>* Complexité mécanique modérée</li> </ul>	<ul style="list-style-type: none"> <li>* Non-holonome</li> <li>* Pas de rotation sur soi-même</li> </ul>
<b>Omnidirectionnel</b>	<ul style="list-style-type: none"> <li>* Holonome</li> <li>* Stable</li> <li>* Rotation sur soi-même</li> </ul>	<ul style="list-style-type: none"> <li>* Complexité mécanique Importante</li> </ul>

*Tab I.1* : différents types de robots à roues

**2. Les robots mobiles à chenilles :**

Les robots mobiles à chenilles illustrés par la figure I.11, présentent l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation de ces robots est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence (présence de boue, herbe...).



*Figure I.11*: Robots mobiles à chenilles.

### 3. Les robots mobiles marcheurs :

Les robots mobiles marcheurs donnés à la figure I.12 sont destinés à réaliser des tâches variées dont l'accès à un site est difficile et dangereux pour l'être humain. Leur structure est dotée de plusieurs degrés de liberté ce qui permet un rapprochement avec les robots manipulateurs. On distingue les robots marcheurs (figure à deux jambes (humanoïdes), à quatre pattes (type cheval), et à six pattes (type araignée).



*Figure I.12: Robots mobiles marcheurs.*

### 4. Les robots mobiles rampants :

La reptation est une solution de locomotion pour un environnement de type « tunnel » qui conduit à réaliser des structures filiformes. Le système est composé d'un ensemble de modules ayant chacun plusieurs mobilités. Ici aussi les techniques utilisées découlent des méthodes de locomotion des animaux et des insectes comme la montre la figure I.13.

Les applications de ce type de robots sont très spécialisées et les architectures des robots sont en général spécifiques à l'application visée [7].



*Figure I.13: Robot mobile rampant.*

**Domaines d'utilisation des robots :**

- **Les robots industriels :** sont des robots utilisés dans un environnement de fabrication industrielle. Ils sont utilisés dans la fabrication des automobiles, des composants et des pièces électroniques, des médicaments et de nombreux produits.
- **Robots domestiques ou ménagers :** Robots utilisés à la maison. Ce type de robots comprend de nombreux appareils très différents, tel que les aspirateurs robotiques, robots nettoyeurs de piscines, balayuses, nettoyeurs gouttières et autres robots qui peuvent faire différentes tâches.
- **Robots en médecine et chirurgie :** Les robots semblent avoir de l'avenir à l'hôpital. Par exemple le Robodoc aide à réaliser certaines opérations de chirurgie. Le robot infirmier est encore en projet. Le cyber squelette HAL aide les personnes à se déplacer. Ainsi, le robot patient permet aux futurs chirurgiens-dentistes d'apprendre à soigner sans faire des dégâts.

**Avantages et inconvénients des robots :**

Un système robotique consiste non seulement en des robots mais aussi d'autres dispositifs et systèmes qui sont utilisés avec le robot pour effectuer la tâche nécessaire.

**❖ Les avantages des robots sont [8] :**

- La robotique et l'automatisation peut dans de nombreuses situations accroître la productivité, la sécurité, l'efficacité, la qualité et la cohérence des produits.
- Les robots peuvent travailler dans un environnement dangereux, sans besoin de soutien de vie, ou de préoccupations concernant la sécurité.
- Les Robots n'ont pas besoin d'éclairage, de climatisation, de ventilation ou de protection contre le bruit.
- Les robots travaillent continuellement, sans ressentir une fatigue ou l'ennui, et ne nécessitent pas une assurance médicale ou de vacances.
- Les robots sont de précision répétable à tous les moments.
- Les robots peuvent être beaucoup plus précis que les êtres humains. Précision linéaire d'un robot typiquement est de 10 à 20 microns.
- Ils se perfectionnent sans cesse.

### ❖ **L'inconvénient des robots :**

- Ils manquent de capacité de réagir en cas d'imprévu, à moins que les situations soient comprises et les réponses soient incluses dans le système. Les mesures de sécurité sont nécessaires pour s'assurer qu'ils ne blessent pas les opérateurs et n'endommagent pas les machines. On peut également citer d'autres inconvénients comme [8] :
- Réponse inadéquate.
- Le manque de prise de décision,
- Consommation de l'énergie.
- Ils peuvent causer des dommages à d'autres appareils, et la blessure à l'homme.
- Les robots sont coûteux en raison du coût initial de l'équipement, d'installation, le besoin de périphériques, le besoin de formation et la nécessité de la programmation (les robots ont besoin des humains).
- Peur de voir les robots voler nos emplois
- Les capacités des robots sont limitées

## **I.3 Conclusion**

A travers ce chapitre nous avons présenté une introduction aux deux catégories principales des robots, ainsi que d'autres spécificités de la robotique. Nous avons ensuite abordé le domaine d'utilisation des robots, ainsi que leurs avantages et inconvénients. Nous avons pu constater l'évolution actuelle des robots proposés par divers constructeurs permettant de présenter une alternative aux machines-outils en utilisant une commande adéquate.

Le chapitre suivant s'intéresse ainsi à la présentation des cartes Arduino et le système Android.

## II.1 Circuit de commande: Arduino

### II .1.1 Introduction

Les cartes Arduino sont conçues pour réaliser des prototypes et des maquettes de cartes électroniques pour l'informatique embarquée. Ces cartes permettent un accès simple et peu coûteux à l'informatique embarquée. De plus, elles sont entièrement libres de droit, autant sur l'aspect du code source (Open Source) que sur l'aspect matériel (Open Hardware). Ainsi, il est possible de refaire sa propre carte Arduino dans le but de l'améliorer ou d'enlever des fonctionnalités inutiles au projet.

Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique embarquée de par sa simplicité. En effet, beaucoup de librairies et de fonctionnalités de base occulte certains aspects de la programmation de logiciel embarquée afin de gagner en simplicité. Cela en fait un langage parfait pour réaliser des prototypes ou des petites applications dans le cadre de hobby. [8]

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 a 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation.[17]

Le système Arduino permet de :

- contrôler les appareils domestiques
- fabriquer votre propre robot
- faire un jeu de lumières
- communiquer avec l'ordinateur
- télécommander un appareil mobile (modélisme)
- etc... [9]

## II.1.2 Présentation générale de module Arduino

Arduino est une gamme de circuits électroniques open source basés pour la plupart sur un microcontrôleur du fabricant Atmel. Ces circuits intègrent les composants nécessaires pour permettre une utilisation rapide et simple du microcontrôleur. Cette simplification vise à rendre accessibles à tous : la création et la programmation d'objets ou dispositifs interactifs. Ces objets peuvent contenir toutes sortes de capteurs, d'indicateurs lumineux ou d'interrupteurs que l'on souhaite faire intervenir.

Entre autres, les cartes Arduino sont équipées de connecteurs standardisés pour brancher des modules compatibles appelés 'shields'. Ces derniers sont des circuits d'une taille plus ou moins semblable à celle de l'Arduino et qui viennent s'empiler sur ces connecteurs. Ils proposent des extensions matérielles qui permettent d'ajouter des fonctionnalités originales à son projet. En plus de ces connecteurs, les cartes possèdent toutes une connectique USB permettant de programmer facilement le microcontrôleur qu'elles embarquent.

Arduino propose en plus gratuitement un environnement de développement intuitif avec lequel la création de programmes est simple, même pour les débutants. De plus, cet environnement est compatible avec les principaux systèmes d'exploitation : Windows, Mac OS et Linux.

Il existe en plus une grande variété de cartes Arduino qui possède chacune leurs avantages et inconvénients. Il faudra alors adapter son choix en fonction du projet que l'on souhaite réaliser. [12]

## II.1.3 Description technique

Comme énoncé précédemment, il existe beaucoup de cartes Arduino différentes, mais elles possèdent toutes des éléments en commun.

### II.1.3.1 Alimentation

Le microcontrôleur présent en général sur les cartes Arduino est alimenté par une tension de 5V. En fonction du modèle de la carte, cette tension peut être fournie soit par une des prises d'alimentation présentes sur la carte, soit par la prise USB utilisée pour la connecter à un ordinateur. La valeur de la tension à fournir sur une des prises d'alimentation doit être comprise entre 7 et 12V mais cette tension n'a pas besoin d'être stabilisée en raison de la présence d'un régulateur de tension sur la carte.

Il peut exister deux prises d'alimentation sur une carte Arduino. La première prise est souvent la plus simple à mettre en place. Il s'agit d'une simple prise jack standard, présente sur les cartes Arduino Uno et Arduino Mega, sur laquelle il suffit de raccorder un bloc d'alimentation fournissant la tension requise pour démarrer sa carte. Cette prise n'est pas présente sur toutes les cartes Arduino, notamment pour les cartes les plus petites pour lesquelles ces prises n'étaient pas vitales.

La seconde prise en revanche est présente sur tous les modèles. Il s'agit d'une broche nommée Vin présente sur un des cotés de la carte. Il suffit alors de raccorder la branche positive de son alimentation à cette broche, comme on le ferait pour une pile, et la branche négative sur une broche GND.

La carte possède également deux broches de tensions stabilisées de 3,3V et de 5V. Ces deux tensions sont généralement utilisées pour alimenter les shields qui viendront s'ajouter sur l'Arduino mais elles peuvent être utilisées pour d'autres circuits.

### **II.1.3.2 Horloge**

L'horloge détermine la fréquence ou la rapidité à laquelle les instructions seront exécutées par le microcontrôleur. Cette vitesse peut varier d'un microcontrôleur à l'autre et est exprimée en hertz(Hz). Si les PC et Mac peuvent atteindre des fréquences de plusieurs gigahertz, la fréquence est en revanche beaucoup plus faible pour les microcontrôleurs, avec des fréquences de quelques mégahertz seulement. Cette fréquence est déterminée par un oscillateur à quartz ou résonateur céramique. Ces composants électroniques sont les seuls capables de délivrer un signal électrique carré stable (signal alternant rapidement entre deux valeurs hautes et basses) nécessaire au bon fonctionnement du microcontrôleur. Sur les cartes Arduino, l'horloge est généralement cadencée à 16 MHz.

### **II.1.3.3 Reset**

La reset est une fonction physique permettant au microcontrôleur, comme son nom l'indique, de réinitialiser son état. Un microcontrôleur exécute en effet les instructions contenues dans sa mémoire de manière cyclique et infinie. Il n'est ainsi pas rare, notamment lors de la conception de son circuit et infinie. Il n'est ainsi pas rare, notamment lors de la conception de

son circuit, que le programme soit soumis à une erreur bloquant le bon déroulement du programme.

La fonction reset permet alors d'interrompre le programme et de le remettre à zéro. Il est possible d'effectuer cette même opération en coupant et en remettant le courant sur la carte Arduino. Lors de sa mise sous tension, elle commencera toujours par reprendre le programme du début, ce qui équivaut donc à une réinitialisation. Il est toutefois plus aisé d'utiliser cette fonction reset matérialisée sur la carte par un bouton poussoir.

### II.1.3.4 Entrées/sorties

Les entrées et sortie sont les moyens que possède le microcontrôleur pour communiquer avec le monde extérieur. Typiquement, on appelle entrées / sorties les pattes métalliques qui donnent cette forme familière aux puces électroniques. Ces pattes reçoivent ou émettent des signaux logiques qui peuvent alors être interprétés par le microcontrôleur ou d'autres circuits. Les cartes Arduino facilitent l'accès à ces entrées / sorties en les reportant sur le pourtour du circuit. En utilisant des connecteurs femelles standards, d'un pas de 2,54mm, il devient très simple d'ajouter ou de retirer des composants sans devoir forcément passer par une étape de soudure. C'est sur ces connecteurs que se branchent les shields et les différents circuits additionnels employés.

Le nombre de broches peut varier d'un microcontrôleur à l'autre, et même d'une carte Arduino à l'autre et il peut être intéressant d'étudier les différentes cartes avant de choisir celle adaptée à son projet. Généralement, ces broches fonctionnent sous une tension maximale de 5V et ne peuvent recevoir et fournir que 40 mA de courant.

On peut ranger ces broches par catégorie suivant leur fonctionnalité :

- Les broches digitales : ces broches fournissent des données digitales sous forme de signaux logiques. Elles ne peuvent donc contenir que deux valeurs, un 0 logique correspondant à une absence de tension et une 1 logique correspondant à une tension de 5V. certaines de ces broches digitales possèdent une seconde fonctionnalité. C'est le cas notamment des broches PWM (pulsed Width Modulation ou modulation de large impulsion) ou de certaines broches de communication. Sur les cartes Arduino, les broches digitales sont généralement nommées par un simple numéro, sauf dans le cas de l'Arduino Nano sur laquelle ce numéro est précédé de 1 lettre D.

- Les broches analogiques : ces broches ne fonctionnent qu'en entrée, c'est-à-dire qu'elles ne servent qu'à la lecture de données. Elles acceptent des tensions comprises entre 0 et 5V. ces tensions sont ensuite utilisées par un convertisseur analogique/numérique qui s'occupe de traduire ces valeurs physiques en données numériques sur 10 bits, soit des données comprises entre 0 et 1023. Les broches analogiques sont nommées sur les cartes Arduino par un chiffre précédé de la lettre A.
- Les broches d'alimentation : ces broches servent à alimenter des shields ou circuits externes. Elles permettent également d'alimenter la carte Arduino sans passer par la prise jack présente sur certains modèles. Ces broches sont réunies sous l'étiquette POWER.
- Les broches de communication : il existe trois ports de communication sur les carte Arduino : le port série, le port I<sup>2</sup>C et le port SPI. Ces ports ne disposent pas de broches dédiées à leur fonctionnement. Ils partagent en revanche leur fonctionnalité avec d'autres broches. Les broches partagées sont différentes d'une carte à l'autre et c'est pourquoi il est recommandé de consulter la documentation de chaque carte afin de connaître les numéros des broches correspondant à ces ports.

### II.1.3.5 Mémoire

Les microcontrôleurs ATmega dont sont équipées la plupart des cartes Arduino disposent de trois types de mémoires : la mémoire flash, la SRAM (static Random Access Memory ou mémoire vive statique) et le EEPROM (Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et reprogrammable).

La mémoire flash et l'EEPROM sont deux mémoires non volatiles dites mortes, c'est-à-dire qu'elles conservent leurs données même en l'absence de courant électrique. C'est d'ailleurs dans la mémoire flash qu'est contenu le programme qui sera exécuté par le microcontrôleur lors de chaque mise sous tension. Cette mémoire flash contient également un bootloader, ou chargeur d'amorçage, sur quelques octets.

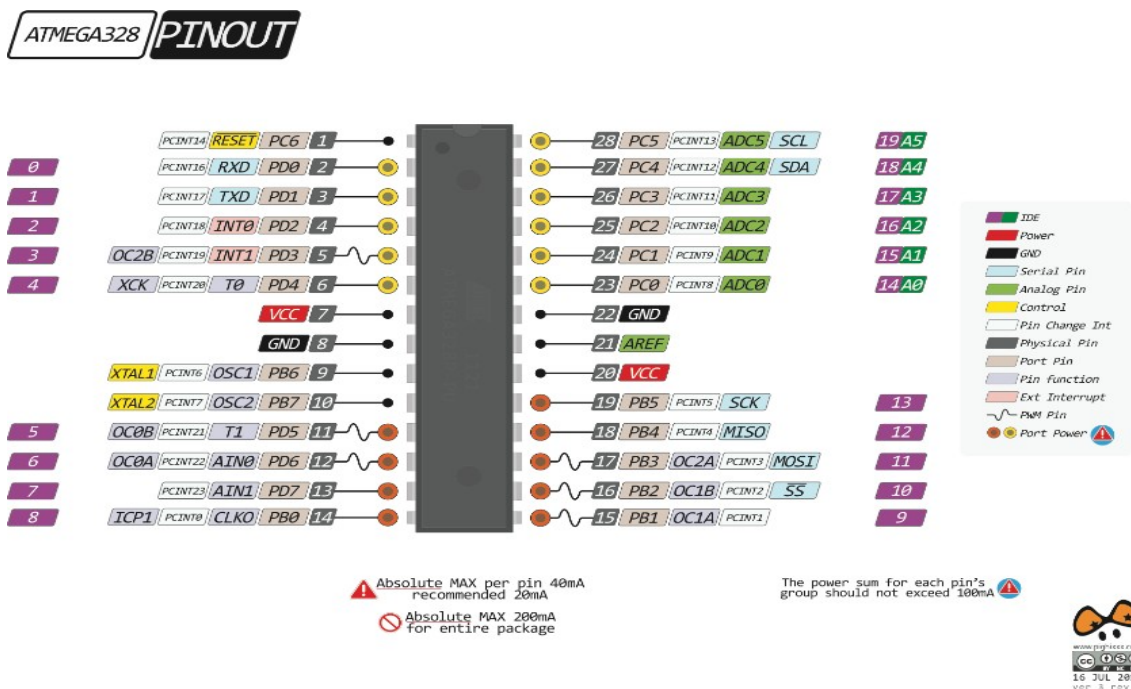
Le bootloader est un programme s'exécutant au démarrage de chaque machine, habituellement pour choisir quel système d'exploitation utiliser.

Dans le cas de l'Arduino, le bootloader présent initialement sur chaque carte permet de simplifier le transfert de programmes sur le microcontrôleur. De manière simple, lors de la

mise sous tension de la carte, le bootloader vérifie s'il y a une tentative d'écriture d'un nouveau programme. Si c'est le cas, le programme présent dans la mémoire flash sera remplacé, sinon, le programme présent sur la mémoire flash est démarré. Il est possible de se passer de ce bootloader pour gagner un peu d'espace dans la mémoire mais il sera nécessaire dans ce cas de passer par un programmeur spécifique.

L'EEPROM sert quant à elle à stocker des variables que l'on souhaite conserver d'une mise hors tension à l'autre. Elle peut être utile si l'on souhaite stocker des données globales qui sont susceptibles d'être réutilisées tout au long de la durée de vie de son projet.

Ces deux types de mémoires mortes présentent l'avantage de la conservation de données, mais la technologie employée pour ce genre de mémoire réduit significativement leur durée de vie. On estime ainsi que ces deux mémoires ne peuvent supporter que 100000 écritures. Les zones de la mémoire qui auront dépassé leur espérance de vie deviendront simplement inutilisables. Malheureusement, dans les rares cas de pannes, aucune erreur ne sera relevée et le comportement du programme en cours d'exécution risque d'être aléatoire. La lecture de ces mémoires n'affecte pas leur durée de vie et est donc illimitée. [12]



FigII.1: Schéma explicatif des différentes broches du microcontrôleur ATmega328

### II.1.4 Les gammes de la carte Arduino

- Arduino Uno
- Arduino Mega
- Arduino Nano

- LilyPad Arduino
- Arduino Leonardo
- Arduino DEU
- Arduino Intel Galileo. [10]

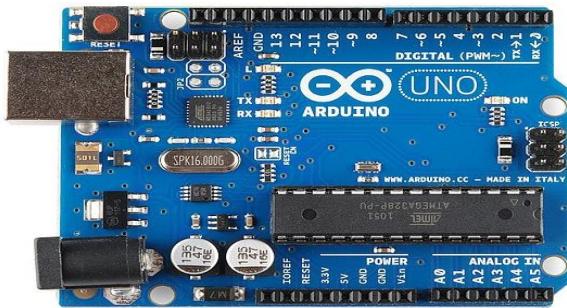


Fig II.2: Arduino UNO



FigII.3: Arduino MEGA



Fig II.4: Arduino NANO



Fig II.5: Arduino LEONARDO

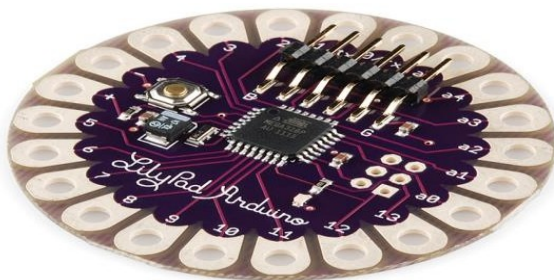
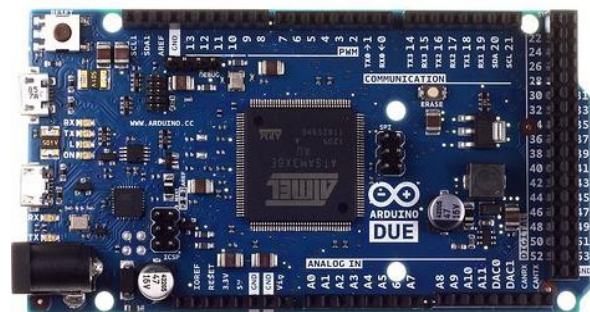


Fig II.6: Arduino LilyPad



FigII.7: Arduino DUE



Fig II.8: Intel Galileo

## II.1.5 La carte Arduino Uno

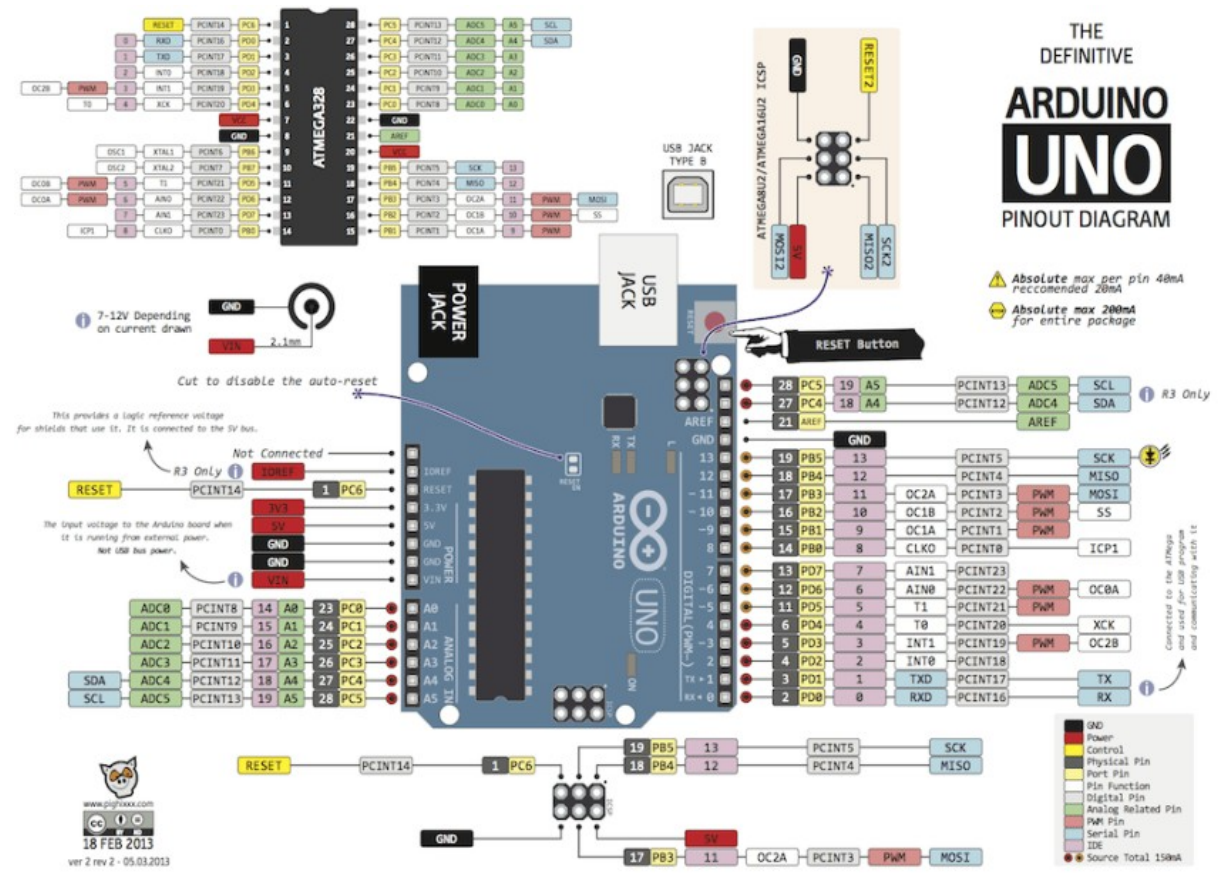
### II.1.5.1 Présentation

La carte Arduino Uno est le produit populaire parmi les cartes Arduino. Parfaite pour débiter la programmation Arduino, elle est constituée de tous les éléments de base pour construire des objets d'une complexité relativement faible.

La carte Arduino Uno, comme son nom l'indique, a été la première à utiliser la version de programmation Arduino 1.0, et elle est devenue le symbole de l'univers Arduino.

La carte Arduino Uno est constituée de 14 broches d'entrées/sorties digitales, dont six sont utilisables en PWM, de 6 broches d'entrées analogiques, d'une connectique USB, d'une connectique d'alimentation, d'un port ICSP et d'un bouton RESET.[11]

La description de toutes les connectiques de la carte Arduino UNO est présentée dans l'image ci-dessous :



FigII.9 : Description détaillée du brochage de la carte Arduino Uno

L'alimentation de la carte Arduino Uno se fait normalement entre 7V et 12V de courant continu. Cependant, il est possible de faire fonctionner la carte Arduino Uno au maximum

entre 6V et 20V. En deca de 6V, la carte n'est plus stable. Au-delà de 20V, le régulateur de tension surchauffe, et peut endommager la carte.

L'alimentation de la carte peut se faire à travers le port USB lorsqu'il est branché sur l'ordinateur, ou via la connectique d'alimentation avec le port jack ou l'entrée d'alimentation. L'alimentation est sélectionnée de manière automatique par la carte Arduino. La source d'alimentation fournissant le meilleur voltage sera sélectionnée comme source d'alimentation par la carte.

En utilisant la connectique d'alimentation, une batterie ou un chargeur spécifique délivrant un courant continu de 9V convient parfaitement pour alimenter d'autres éléments.

L'entrée d'alimentation (VIN) permet d'utiliser une broche afin d'alimenter votre carte Arduino en électricité. Il est conseillé d'utiliser une alimentation entre 7V et 12V de courant continu pour ne pas endommager la carte.

La broche 5V est une connexion de sortie permettant de récupérer un courant généré par le régulateur de la carte. La broche 3.3V permet aussi de récupérer un courant mais de 3.3V et de 50 mA au maximum.

Les prises de terre ou GND permettent de fermer le circuit.

Les broches d'entrées et sorties pour la carte Arduino Uno peuvent être décrites en deux parties.

Les 14 broches d'entrées et sorties digitales sont utilisables comme leur nom l'indique en entrée ou en sortie en utilisant comme leur nom l'indique en entrée ou en sortie en utilisant les fonctions `pinMode`, `Digital Read/` ou `Digital Write`. Chaque broche opère à 5V et peut fournir ou recevoir au maximum 40mA. De plus, chaque broche dispose d'une résistance interne de 20 à 50 k $\Omega$ , non connectée par défaut.

Parmi ces broches, plusieurs possèdent des rôles plus étendus :

- Broches couplées de liaison série : broches 0 (RX) et 1(TX). Ces broches sont particulières puisque leur rôle peut être étendu à une entrée série (RX) et une sorties séries (TX).
- Broches interrupteurs : broches 2 et 3. Ces broches peuvent jouer le rôle d'interrupteurs sur différents composants en utilisant la fonction `attach Interrupt()`.
- Broches PWM : broches 3, 5, 6, 9, 10 et 11. Ces broches sont utilisées comme sorties PWM en utilisant la fonction `analog Write ()`.
- Broche SPI : 10(SS), 11(MOSI), 12(MISO) et 13(SCK). Ces broches peuvent être utilisées pour gérer une communication SPI.

- Broche Led-13 : sur la carte Arduino UNO, il existe une LED qui est contrôlable directement à travers la broche 13 en utilisant les valeurs HIGH pour allumer et LOW pour éteindre la LED.

La carte Arduino UNO possède également six broches d'entrées analogiques étiquetées de A0 à A5. Elles mesurent l'entrée de courant sur 5V sur une résolution de 10 bits, soit sur une échelle de 0 à 1023.

Sur les broches A4, appelée aussi broche SDA, et A5, appelée broche SCL, il est possible de gérer la communication I<sup>2</sup>C.

D'un point de vue plus technique, cette carte se base sur le processeur ATmega328, un puissant microcontrôleur disposant d'une mémoire flash qui donne des performances très élevées tout en ayant une basse consommation.

Il dispose d'une mémoire de 32 Ko, et de 2 Ko de SRAM. Il possède également une mémoire de 2 Ko d'EEPROM, programmable en utilisant la librairie EEPROM. [12]

### II.1.5.2 les Caractéristiques techniques

Microcontrôleur	ATmega328
Tension de fonctionnement	5v
Tension d'Input (recommandée)	7-12v
Tension d'Input (limites)	6-20v
Pins I/O digitales	14 (dont 6 fournissent une sortie PWM)
Pinces d'E / S numériques PWM	6
Pins Input Analogiques	6
Courant DC par pin I/O	20 mA
Courant DC pour la broche 3.3V	50 mA
Mémoire flash	32 Ko (ATmega328P) Dont 0,5 Ko utilisé par bootloader
SRAM	2KB(ATmega328)
EEPROM	1 KB (ATmega328P)
Vitesse de l'horloge	16 MHz
LED_BUILTIN	13
Longueur	68.6 mm
Largeur	53.4 mm

Poids	25 g
-------	------

**TabII.1: Les caractéristiques de l'Arduino Uno. [13]**

### **II.1.5.3 Avantages**

- Pas cher
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso etc...)
- Existence de « shield » (boucliers en français) : ce sont des cartes supplémentaires qui

Se connectent sur le module Arduino pour augmenter les possibilités comme par exemple : afficheur graphique couleur, interface et hernet, GPS, etc...

Par sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle.

On constate sur la toile, que par le biais d'Arduino, de nombreux curieux redécouvrent l'électronique.[11]

### 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 stockage 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.

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.

### 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 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]

### Présentation d'Appinventor



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 événements 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.

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

### Créé un projet:

Depuis un navigateur internet ouvrir le compte google.

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

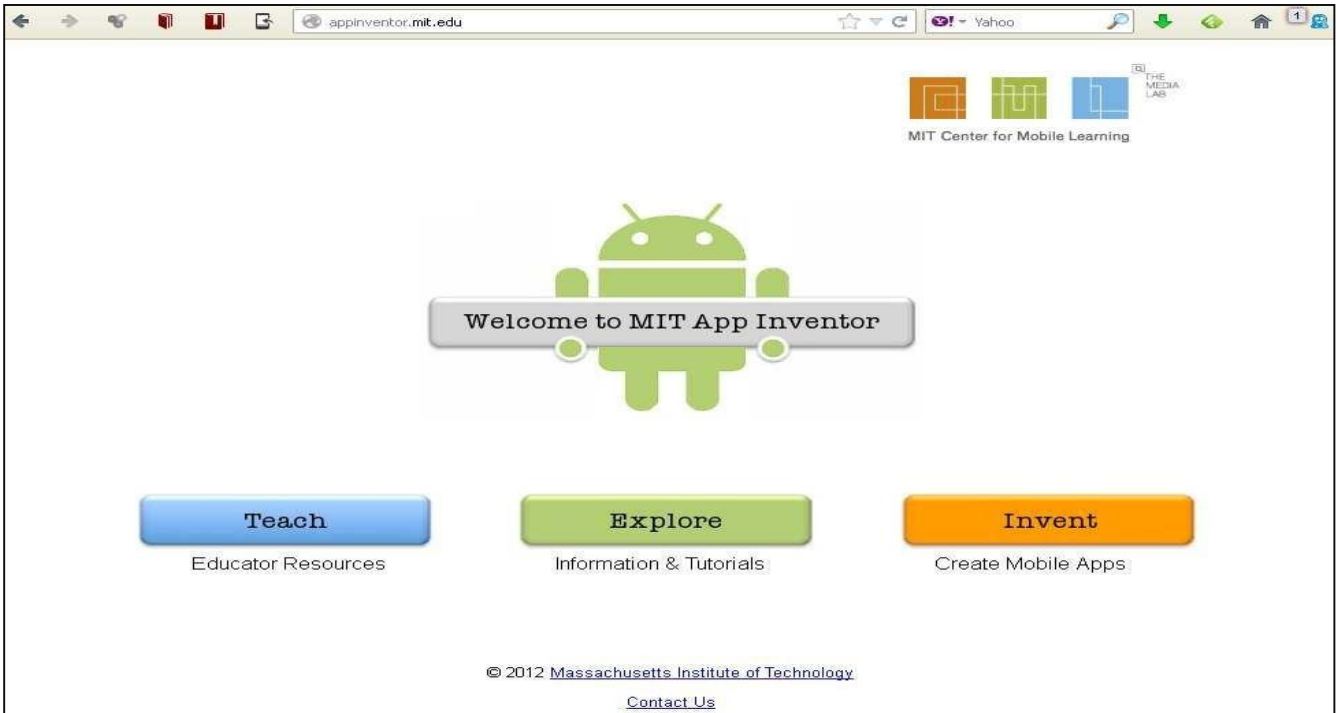
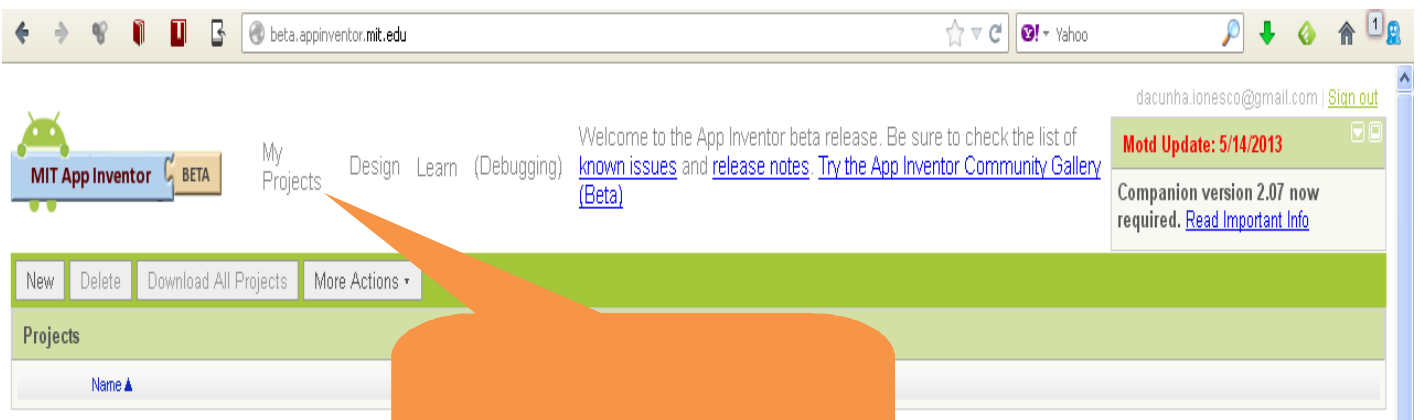


Figure 3.1. Site mit app inventor.

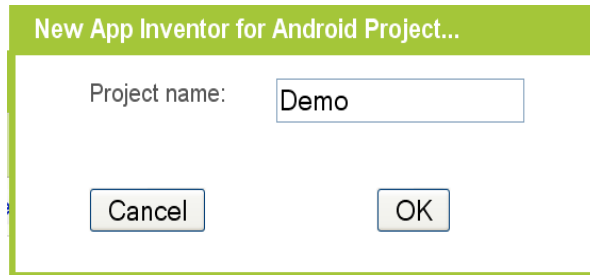
- Cliquer sur



Cliquer sur

New

. Donner un nom au nouveau projet.



Et voilà Interface graphique d'utilisateur :

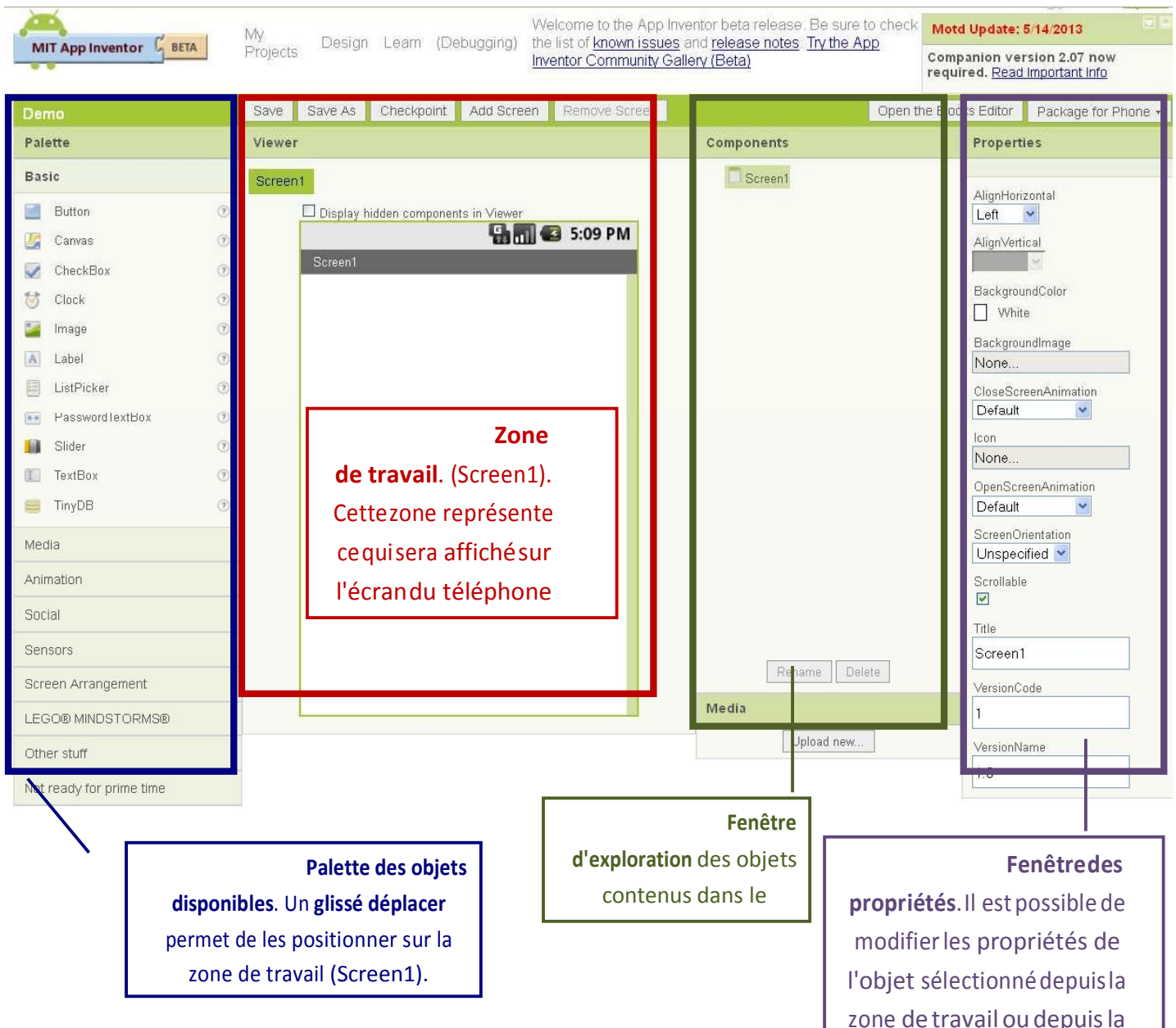


Figure 3.2. Première interface de la création App Inventor

**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 :

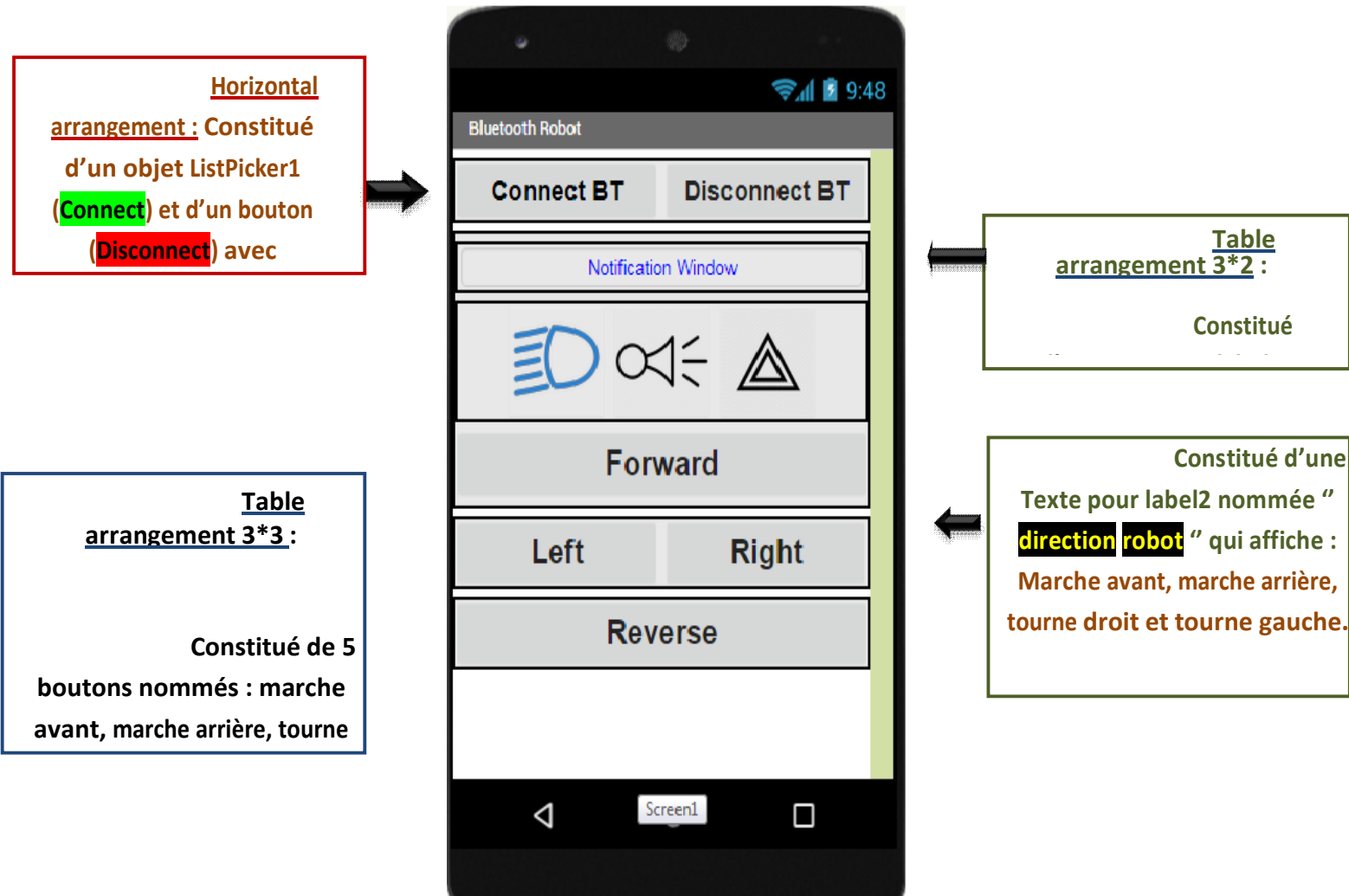
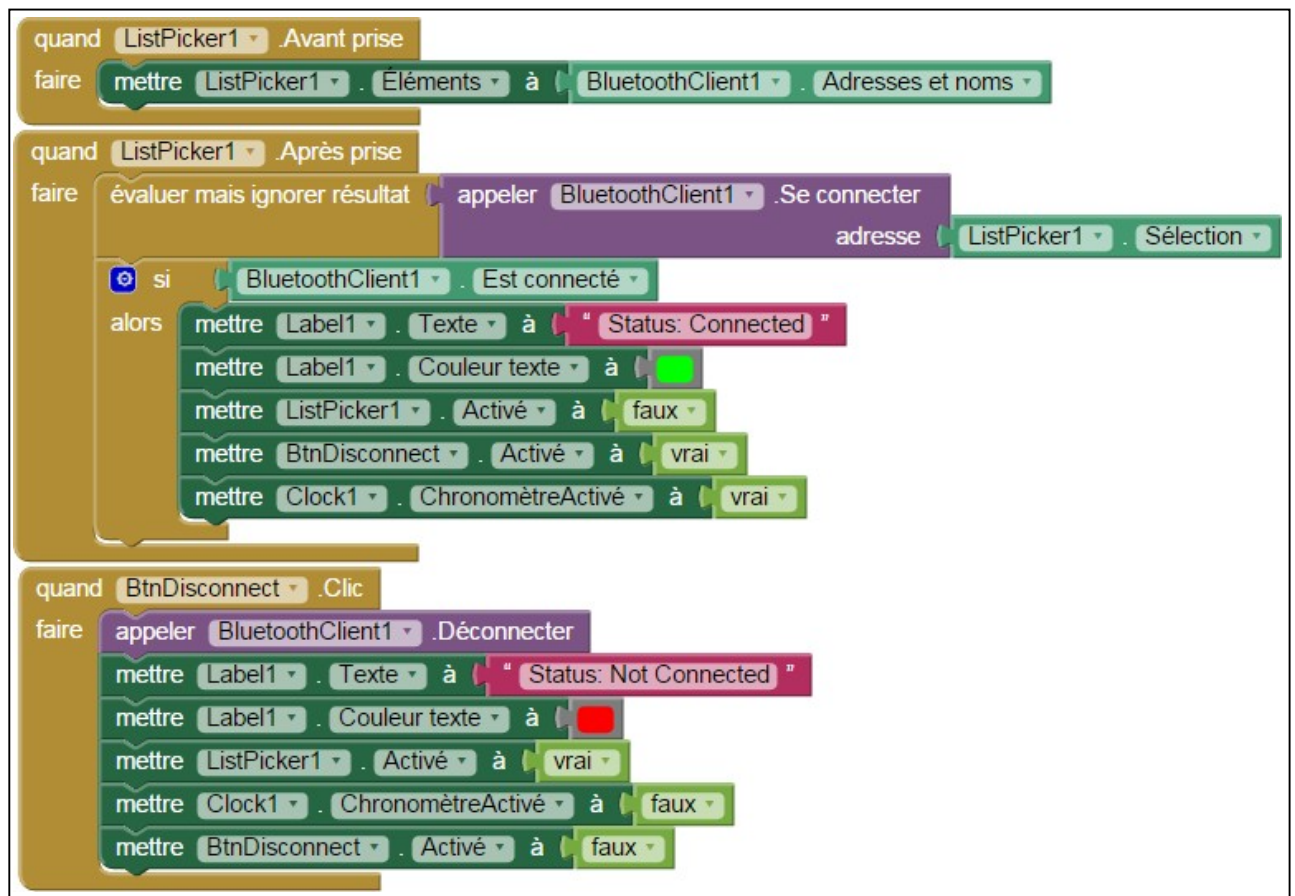


Figure 3.3.L'interface de l'application.

**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.

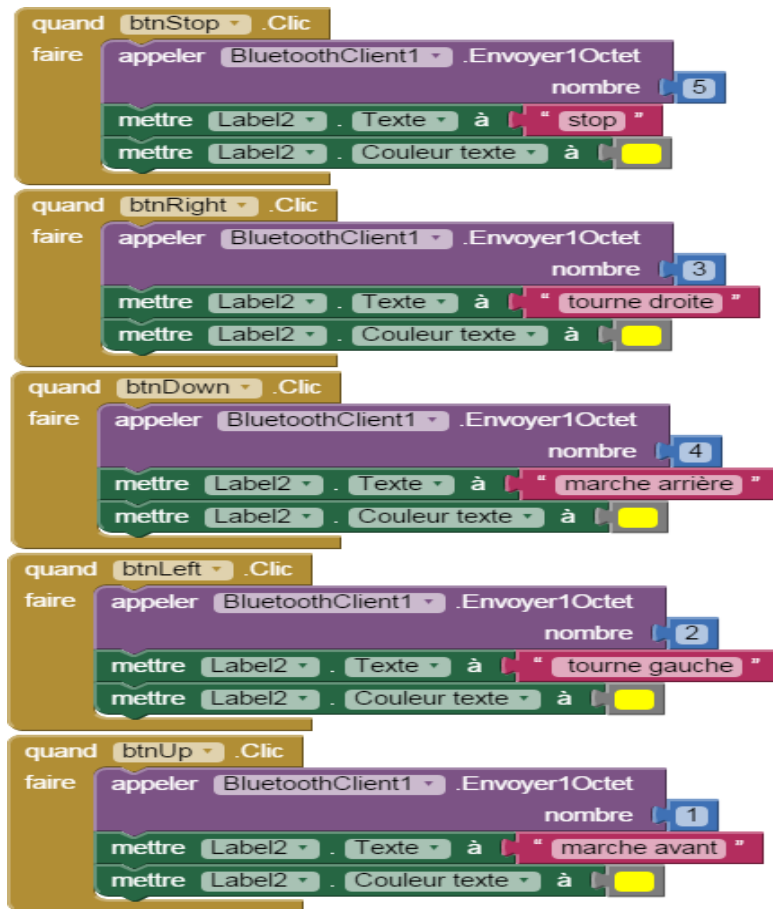
- L'objet **ListPicker1** permettra de choisir dans une liste, un robot parmi ceux appariés en Bluetooth avec le smartphone
- L'objet **texte pour label1** permettra l'affichage du fonctionnement de la connexion avec Bluetooth du robot si elle est connecté ou pas.
- le bouton **btndisconnect** est pour déconnecter BluetoothClient 1

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 connexionBluetooths'établie. Si le smartphoneest 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.

#### **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 :



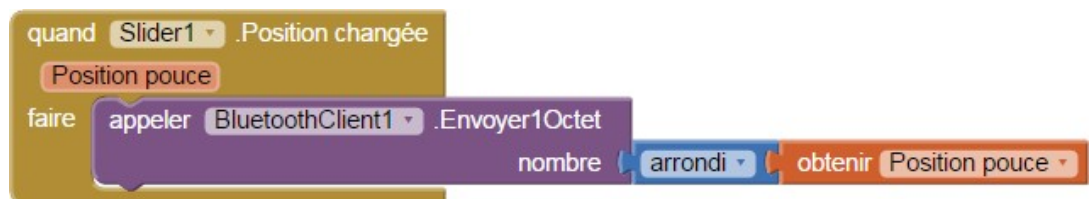
**Figure 3.5.** Code source d'application 2.

-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 à choisir la valeur minimale égale à 20 et la valeur maximale égale à 255. Lorsque la position du curseur se change elle a envoie 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 :

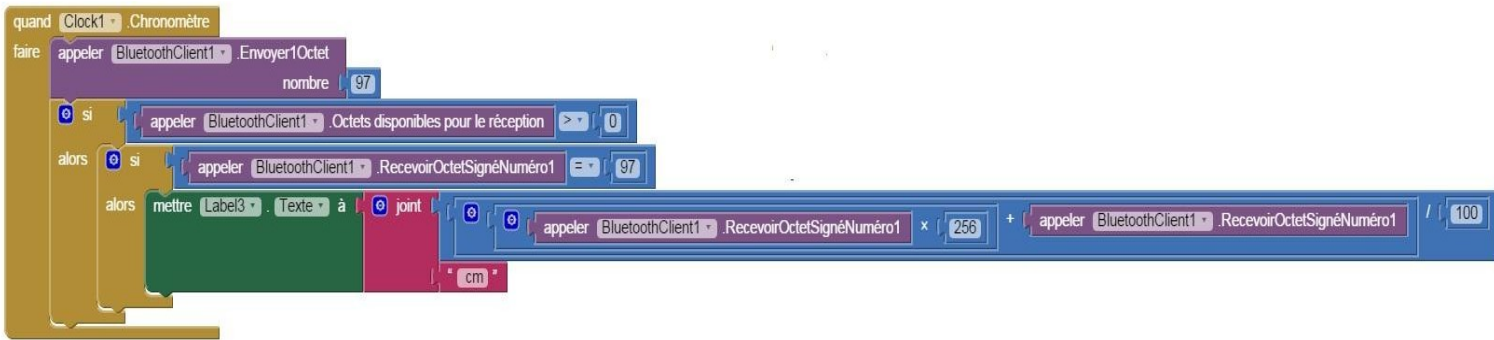


Figure 3.7. Code source d'application 4.

On a utilisé pour ça le capteur ultrason.

-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 androïde reçoit cette information de distance et d'écrit sur l'écran.

**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'abords'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.

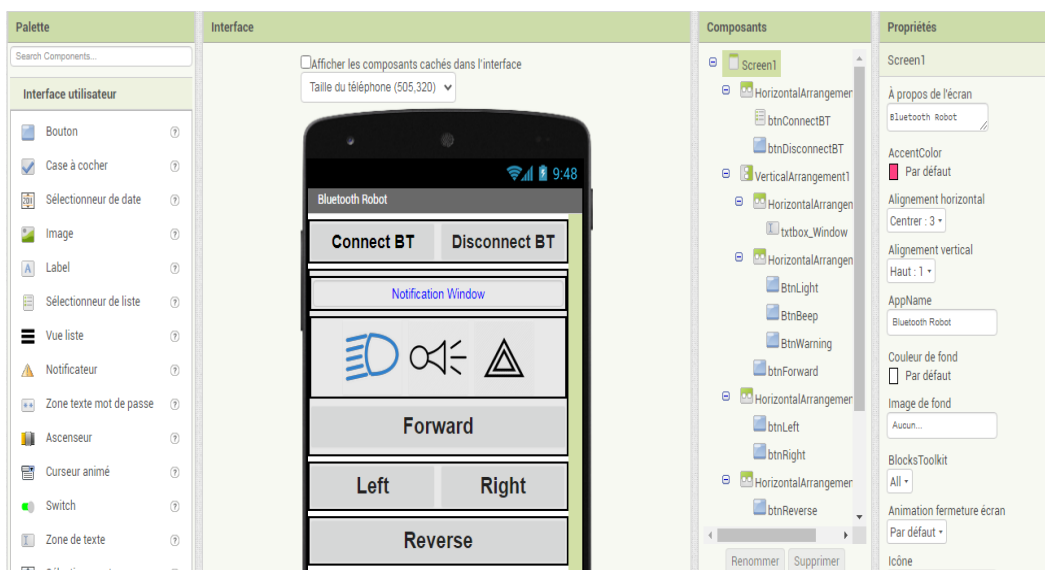


Figure 3.8. Photo de notre application de contrôle.

### **COCLUSION :**

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

## 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.

### **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).

#### □ **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.

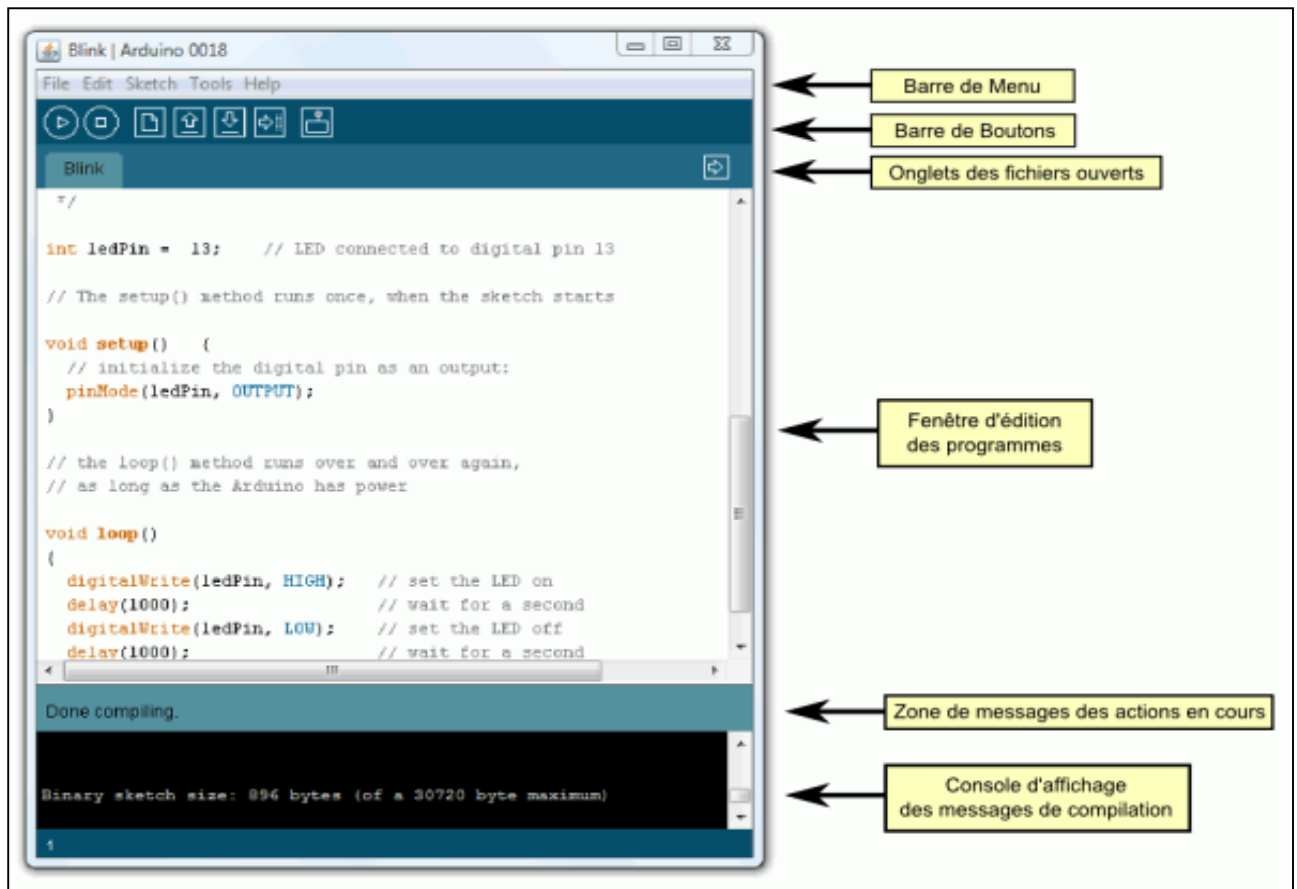


Figure 4.1. Interface IDE Arduino.

□ **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 (COM12) comme à titre d'exemple cette figure suivante.

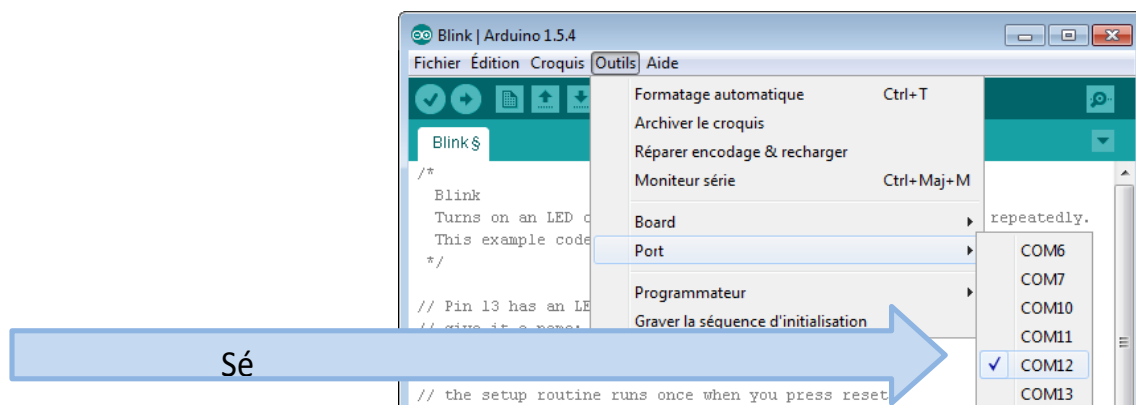


Figure 4.2. Paramètre de la carte.

□ **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.

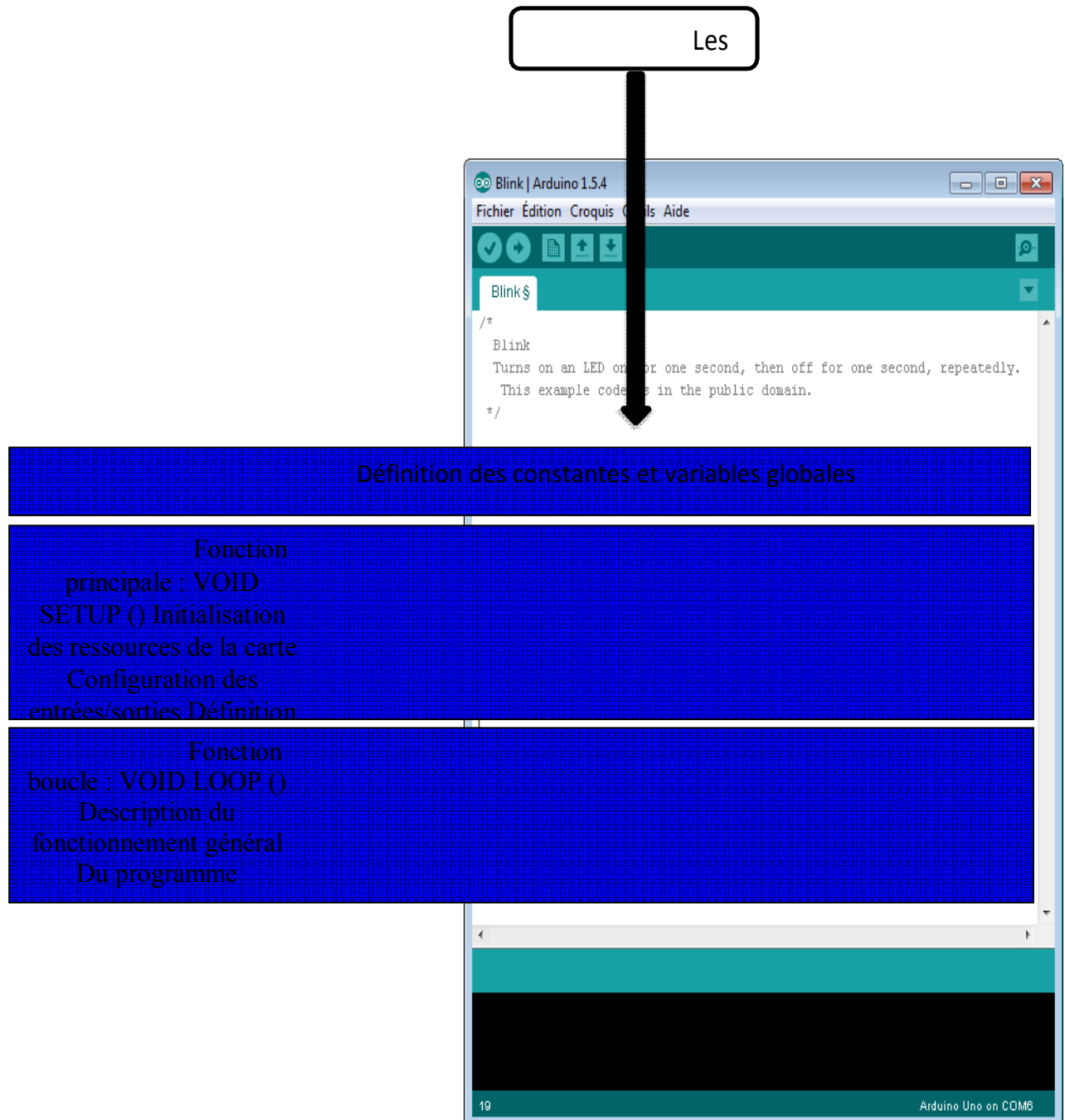
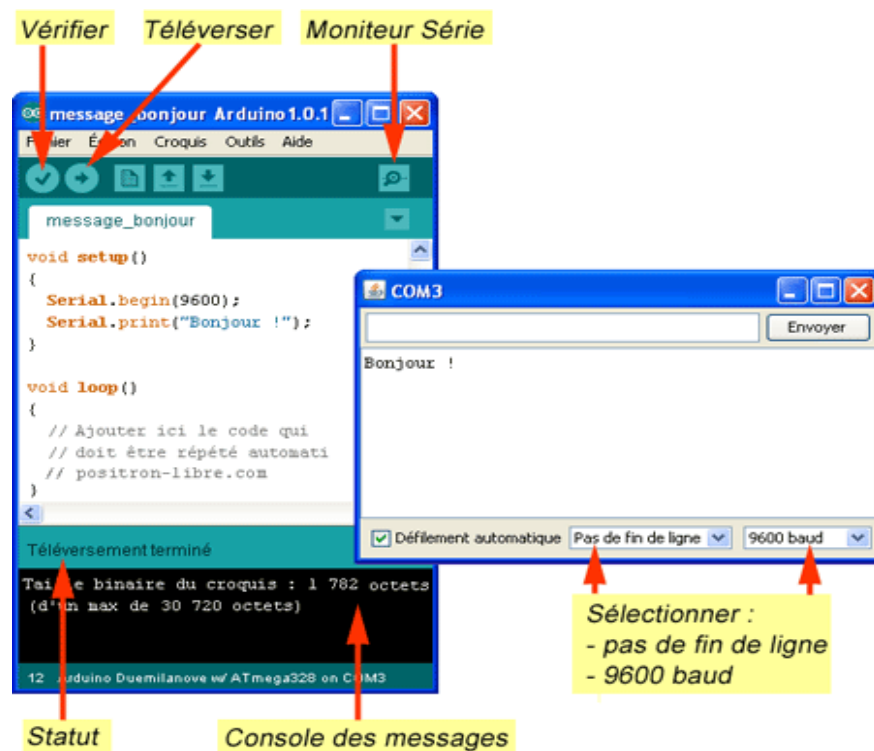


Figure 4.3. Description du programme sur arduino.

□ **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.

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 4.4.** Les étapes de téléchargement du code.

#### Les Accessoires de la carte Arduino :

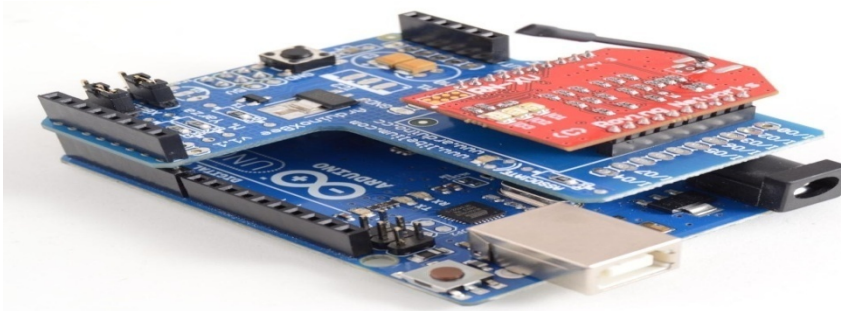
La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

#### 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 :

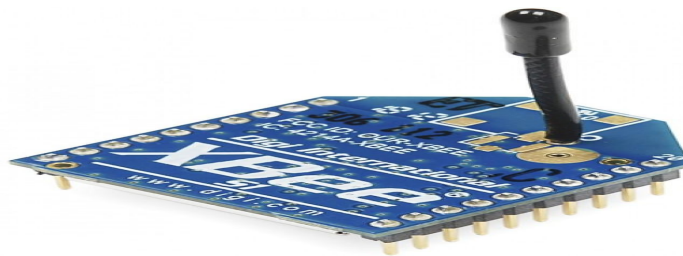
Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.



*Figure 4.5.* Module shield wifi.

**b** *Le Module XBee :*

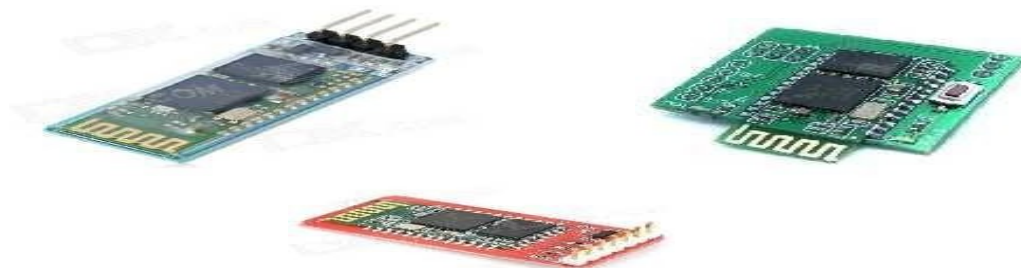
Ce module permet de faire de la transmission sans fil, faible distance /consommation/débit/ prix.



*Figure 4.6.* 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.



*Figure 4.7.* Type de modules Bluetooth.

✓ **Présentation du module Bluetooth HC-05:**

**Introduction :**

Comme expliqué plus tôt, le module utilisé se nomme HC-05 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]



**Figure 4.8.** Module Bluetooth hc-05.

**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-05 et arduino :**

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 :

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

Vous devriez maintenant être en mesure d'utiliser votre module Bluetooth HC-05. 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.

### Les capteurs :

#### a Introduction :

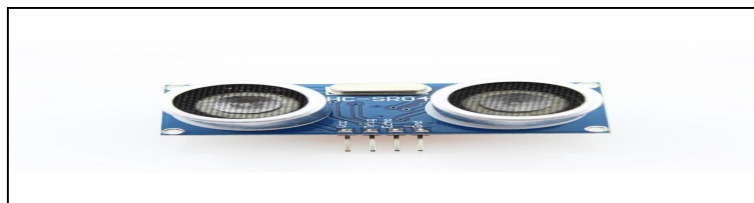
Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis. Il fournit cette information grâce à une électronique à laquelle il est associé.

#### 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 :

- L'émetteur
- Le récepteur



*Figure 4.9.* Capteur ultrason hc-sr04

### 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.

#### 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 contrôler 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 4.10.* L298n double pont-h.

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

Caractéristiques :	Spécifications :
--------------------	------------------

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

**Tableau 4.1.** Caractéristique et spécification pont-h l298n.

Ce tableau contient la description technique du pont-h :

Nom du connecteur	Direction	Description	Utilisation
VMS GND	/	Connecté à l'alimentation externe (6V~35V)	ENA(ENB) raccordés à un niveau haut (HIGH) activera MOTORA(MOTORB).
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.	IN1(IN3) raccordés à 5V et IN2(IN4) à GND MOTORA(MOTORB) tournera dans le sens horlogique. IN1(IN3) raccordés à GND et IN2(IN4) à 5V MOTORA(MOTORB) tournera dans le sens Anti-horlogique.

IN2	Entrée	Compatible TTL. Entrée de commande de sens du pont A.	Si vous voulez contrôler la vitesse, vous pouvez connecter ENA(ENB) sur une sortie PWM.
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.	

**Tableau 4.2.** 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 4.11.** 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
- Courant hors-charge (6V) : 71mA
- Courant d'arrêt (3V) : 260mA
- Courant d'arrêt (6V) : 470mA
- Dimension : 55mm x 48.3mm x 23mm
- Poids : 45g

**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 4.12.** 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.

***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 4.13.***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 4.14.***Bouton poussoir.

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

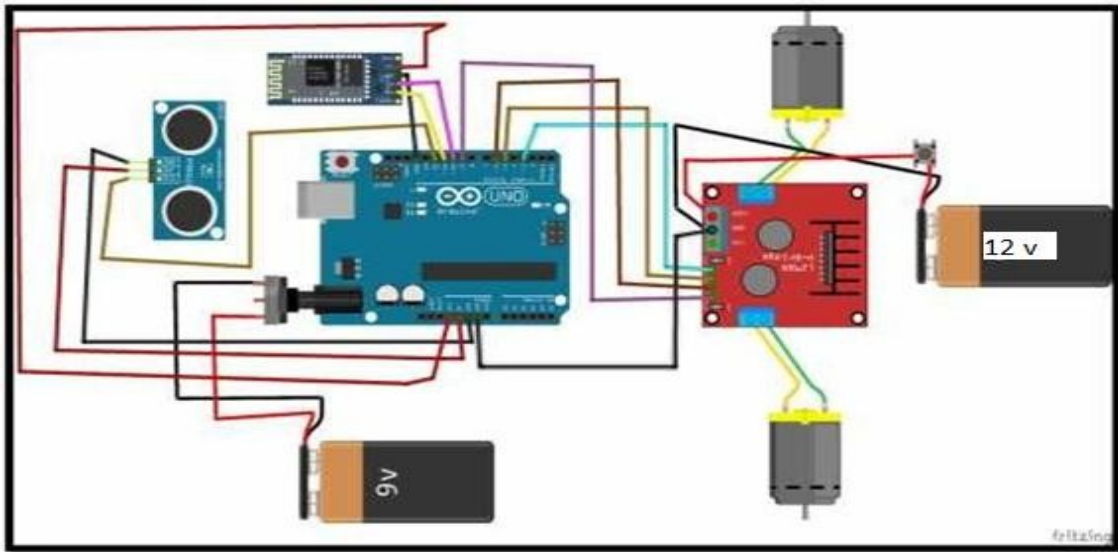
**Programmation du robot (robot télécommandé et détecte obstacle):**

**List du besoin :**

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

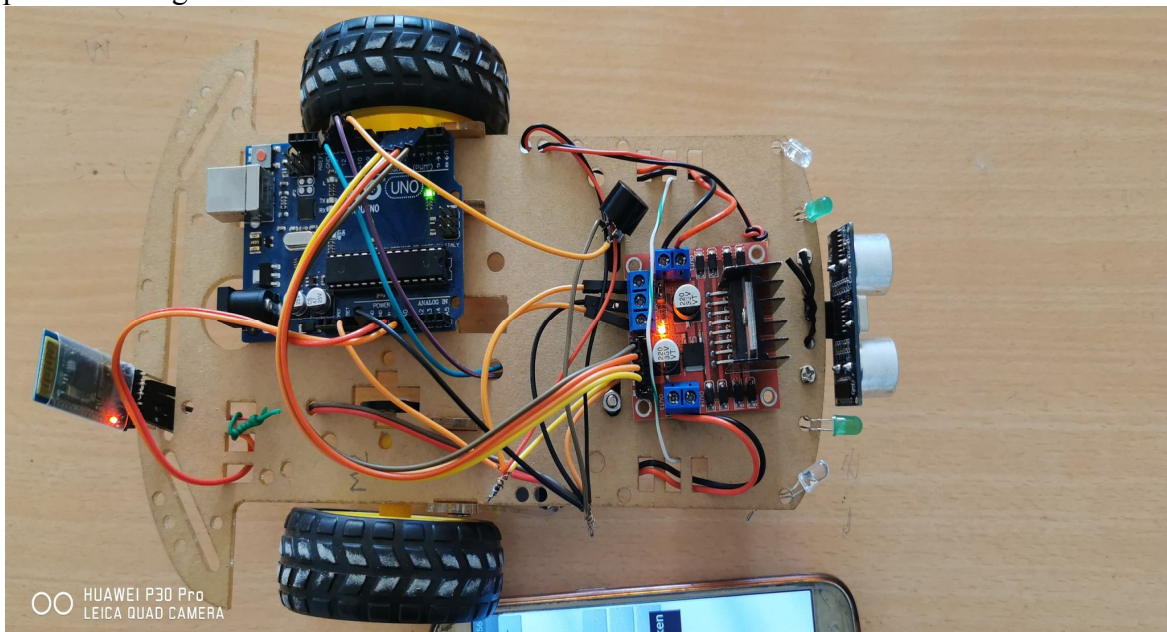
**Montage du robot:**

- Schéma du montage Avec les site [www.fritzing.org](http://www.fritzing.org) :



*Figure 4.15.* Schéma de montage du robot.

- Après le montage voici le résultat :



*Figure 4.16.* Le robot.

**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 4.17.* Code source 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 4.18.* Code source 2.

- Fonction scan : pour ultrasons allant Module HC - SR04.
- 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 4.19.* Code source 3.

- 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);
      }
    }
  }
}
```

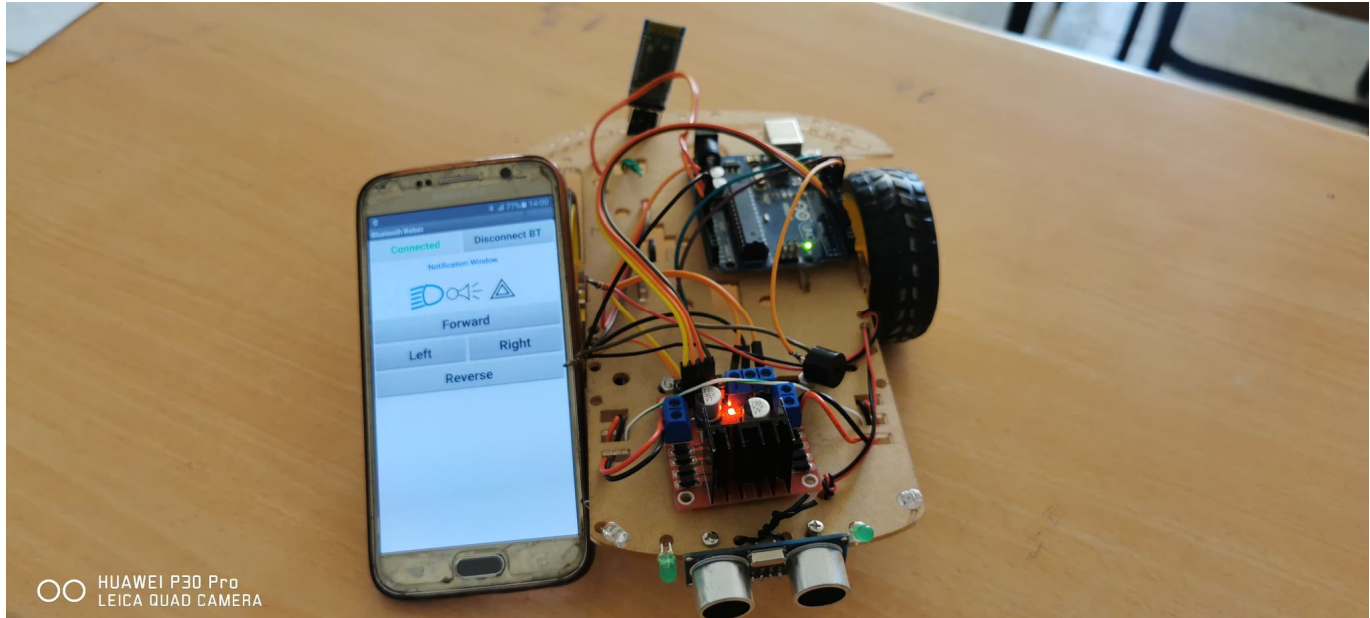
*Figure 4.20.* Code source 4.

- 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).
- 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 4.21.* Finaleduprojet.

### **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

## BIBLIOGRAPHIE

- [1] : A. ALLOUI, A. HADJ 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.
- [2]: Mechatronics; Auteur: V.S. Bagad, ISBN9788184314908, Edition 2008.
- [3]: Computer Aided Manufacturing, C. Elanchezhian, G.ShanmugaSundar, First Edition 2005, Second Edition 2007.
- [4]: Robotics for Electronics Manufacturing: Principles and Applications in cleanroom automation; Auteur: Karl Mathia, ISBN978-0-521-87652-0Hardback.
- [5]: Robotics and control; Auteur: Mittal &Nagrath, ISBN 0-07-048293-4.
- [6] : BelkhadriaKhemisti, "commande d'un robot mobile par réseaux de neurones artificiels" Mémoire en vue de l'obtention du diplôme de magister en électronique, Option : Robotique. Université El HadjLakhdarBatna.
- [7] : A. PRUSKI. « Robotiquegénérale » Edition Ellipase 1988.
- [8] : Robotics (par AppuuKuttan) ; Auteur : AppuuKuttan, ISBN978-81-89866-38-9
- [9] <http://www.centreantipoisons.be/monoxyde-de-carbone/le-monoxyde-de-carbone-co-en-d-tail/quelles-sont-les-concentrations-toxiques-de>
- [10] [https://fablab.univ-tlse3.fr/wiki/images/9/.../Cours\\_arduino\\_v0.2.pdf](https://fablab.univ-tlse3.fr/wiki/images/9/.../Cours_arduino_v0.2.pdf)
- [11] [projet.eu.org/pedago/sin/tutos/arduino.odt](http://projet.eu.org/pedago/sin/tutos/arduino.odt)
- [12] <http://www.ShieldList.org/>
- [13] <http://www.louisreynier.com> « c'est quoi Arduino ».
- [14] Nicolas GOILAV Geoffrey LOI «Arduino» (page12, 13, 14, 15, 16, 17, 18,267, 268, 302, 303)

```

quand btnConnectBT . Avant prise
faire mettre btnConnectBT . Eléments à BluetoothClient1 . Adresses et noms

quand btnConnectBT . Après prise
faire
mettre btnConnectBT . Sélection à appeler BluetoothClient1 . Se connecter
adresse btnConnectBT . Sélection
si BluetoothClient1 . Est connecté
alors
mettre btnConnectBT . Couleur texte à
mettre btnConnectBT . Texte à " Connected "
appeler TextToSpeech1 . Parler
message " blue tooth connected "
mettre Clock1 . ChronomètreActivé à vrai
mettre btnConnectBT . Activé à faux
sinon
mettre btnConnectBT . Couleur texte à
mettre btnConnectBT . Texte à " Connect "
mettre btnConnectBT . Activé à vrai
appeler NoBluetooth . Afficher Alerte
notice " Bluetooth not Connected "

quand btnDisconnectBT . Clic
si BluetoothClient1 . Est connecté
alors
appeler BluetoothClient1 . Déconnecter
appeler TextToSpeech1 . Parler
message " blue tooth disconnected "
faire
si BluetoothClient1 . Est connecté
alors
appeler BluetoothClient1 . Déconnecter
appeler TextToSpeech1 . Parler
message " blue tooth disconnected "
mettre Clock1 . ChronomètreActivé à faux
mettre btnConnectBT . Activé à vrai
mettre btnConnectBT . Texte à " Connect "
appeler NoBluetooth . Afficher Alerte
notice " Bluetooth not Connected "

initialise global command à " S "

quand Clock1 . Chronomètre
faire
si BluetoothClient1 . Est connecté
alors
appeler BluetoothClient1 . Envoyer texte
texte obtenir global command
sinon
mettre btnConnectBT . Couleur texte à

quand btnForward . Retiré
faire mettre global command à " S "

quand btnForward . Enfoncé
faire mettre global command à " F "

quand btnReverse . Retiré
faire mettre global command à " S "

quand btnReverse . Enfoncé
faire mettre global command à " B "

quand btnLeft . Retiré
faire mettre global command à " S "

quand btnLeft . Enfoncé
faire mettre global command à " L "

quand btnRight . Retiré
faire mettre global command à " S "

quand btnRight . Enfoncé
faire mettre global command à " R "

```

# Annexe

---

```
initialise global light à 0
initialise global beep à 0
initialise global warning à 0

quand BtnLight - Clic
faire
  si obtenir global light = 0
  alors
    appeler BluetoothClient1 - Envoyer texte
    texte "W"
    mettre global light à 1
  sinon
    appeler BluetoothClient1 - Envoyer texte
    texte "W"
    mettre global light à 0

quand BtnBeep - Clic
faire
  si obtenir global beep = 0
  alors
    appeler BluetoothClient1 - Envoyer texte
    texte "P"
    mettre global beep à 1
  sinon
    appeler BluetoothClient1 - Envoyer texte
    texte "P"
    mettre global beep à 0

quand BtnWarning - Clic
faire
  si obtenir global warning = 0
  alors
    appeler BluetoothClient1 - Envoyer texte
    texte "C"
    mettre global warning à 1
  sinon
    appeler BluetoothClient1 - Envoyer texte
    texte "C"
    mettre global warning à 0
```

## CODE PORGRAMME AERDUINO :

```
arduino_code1
1 #include <SoftwareSerial.h> // import the serial library
2 #include <HCSR04.h>
3
4 UltrasonicDistanceSensor distanceSensor(13, 12); // Initialize sensor that uses digital pins 13 and 12.
5 SoftwareSerial BT(14, 15); // RX, TX
6
7 #define IN1 2
8 #define IN2 3
9 #define IN3 4
10 #define IN4 5
11
12 #define light_pin 9
13 #define beep_pin 10
14 #define warning_pin 11
15
16 char BluetoothData; // the data given from Computer
17 int t = 10;
18 bool obs = false;
19
20 void setup() {
21     Serial.begin(9600);
22     BT.begin(9600);
23     pinMode(IN1, OUTPUT);
24     pinMode(IN2, OUTPUT);
25     pinMode(IN3, OUTPUT);
26     pinMode(IN4, OUTPUT);
27 }
28
```

```
arduino_code1
28
29 void loop() {
30   if (distanceSensor.measureDistanceCm() > 10){obs = false;}else{obs = true;}
31   if (Serial.available()) {
32     BluetoothData = Serial.read();
33     delay(t);
34   }
35   if (BT.available()) {
36     BluetoothData = BT.read();
37     Serial.println(BluetoothData);
38     delay(t);
39     switch (BluetoothData) {
40       case 'F':
41         if (obs == false){
42           digitalWrite(IN1,HIGH); // left_motor
43           digitalWrite(IN2,LOW); //
44           digitalWrite(IN3,HIGH); // right_motor
45           digitalWrite(IN4,LOW); //
46         }else{
47           digitalWrite(IN1,LOW); // left_motor
48           digitalWrite(IN2,LOW); //
49           digitalWrite(IN3,LOW); // right_motor
50           digitalWrite(IN4,LOW);
51         }
52         delay(t);
53         break;
54       case 'L':// right
55         digitalWrite(IN1,HIGH); // left_motor
```

```
55     digitalWrite(IN1,HIGH); // left_motor
56     digitalWrite(IN2,LOW); //
57     digitalWrite(IN3,LOW); // right_motor
58     digitalWrite(IN4,LOW); //
59     delay(t);
60     break;
61 case'R'://left
62     digitalWrite(IN1,LOW); // left_motor
63     digitalWrite(IN2,LOW); //
64     digitalWrite(IN3,HIGH); // right_motor
65     digitalWrite(IN4,LOW); //
66     delay(t);
67     break;
68 case'B':
69     digitalWrite(IN1,LOW); // left_motor
70     digitalWrite(IN2,HIGH); //
71     digitalWrite(IN3,LOW); // right_motor
72     digitalWrite(IN4,HIGH); //
73     delay(t);
74     break;
75 case'S':
76     digitalWrite(IN1,LOW); // left_motor
77     digitalWrite(IN2,LOW); //
78     digitalWrite(IN3,LOW); // right_motor
79     digitalWrite(IN4,LOW); //
80     delay(t);
81     break;
82 case'W':
```

```
arcuino_code1
80     delay(t);
81     break;
82 case 'W' :
83     digitalWrite(light_pin,HIGH); // left_motor
84     delay(t);
85     break;
86 case 'w' :
87     digitalWrite(light_pin,LOW); // left_motor
88     delay(t);
89     break;
90 case 'P' :
91     digitalWrite(beep_pin,HIGH); // left_motor
92     delay(t);
93     break;
94 case 'p' :
95     digitalWrite(beep_pin,LOW); // left_motor
96     delay(t);
97     break;
98 case 'C' :
99     digitalWrite(warning_pin,HIGH); // left_motor
100    delay(t);
101    break;
102 case 'c' :
103    digitalWrite(warning_pin,LOW); // left_motor
104    delay(t);
105    break;
106 }}
```