



Université Mohamed Khider de Biskra  
Faculté des Sciences et de la Technologie  
Département de génie électrique

# MÉMOIRE DE MASTER

Sciences et Technologies  
Automatique  
Automatique et informatique industrielle

Réf. : .....

---

Présenté et soutenu par :  
**BELLILI Sana'a**

Le : dimanche 7 juillet 2019

## Conception et Réalisation d'un Robot d'Inspection des Pipelines à base de Carte Arduino Uno

---

### Jury :

Dr.	TOUBA Mouamed Moustafa	MCA	Université de Biskra	Président
Dr.	BOUMEHREZ Mouhamed	Pr	Université de Biskra	Rapporteur
Dr.	MIHI Asia	MCB	Université de Biskra	Examineur

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



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Automatique  
Option : Automatique et informatique industrielle

Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:

**MASTER**

*Thème*

**Conception et Réalisation d'un Robot  
d'Inspection des Pipelines à base de Carte  
Arduino Uno**

**Présenté par :**

*BELLILI Sana'a*

**Avis favorable de l'encadreur :**

*BOUMEHREZ Mouhamed*

**Avis favorable du Président du Jury**

*TOUBA Mouhamed Moustafa*

**Cachet et signature**

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



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Automatique  
Option : Automatique et informatique industrielle

## *Thème :*

# Conception et Réalisation d'un Robot d'Inspection des Pipelines à base de Carte Arduino Uno

**Proposé par** : BOUMEHREZ Mouhamed  
**Dirigé par** : BOUMEHREZ Mouhamed

**RESUMES (Français et Arabe)**

## Résumé :

Conception et réalisation d'un robot mobile capable d'inspecter les pipelines pour la détection des fissures et des mauvaises soudures. Le système de détection des fissures est composé par un détecteur de gaz. Le robot mobile dont la structure est adaptée avec la géométrie du pipeline se déplace le long de celui-ci et détecte l'existence de fuites de gaz dues à des fissures.

## ملخص :

تصميم وتنفيذ روبوت متنقل قادر على فحص خطوط الأنابيب للكشف عن الشقوق واللحامات السيئة. يتكون نظام الكشف عن الشقوق من كاشف الغاز. يكتشف الروبوت المتحرك الذي يتكيف هيكله مع هندسة خط الأنابيب على طول وجود تسربات الغاز بسبب التشققات.

## *Dédicaces*

*Du profond de mon cœur, je dédie ce modeste travail à tous ceux qui  
me sont chers.*

*A mon cher père que rien au monde ne vaut tes efforts fournis, jour  
et nuit, pour mon éducation et mon bien être.*

*A ma très chère aimée mère qui m'a donnée la vie tu es l'exemple  
de dévouement qui n'a pas cessé de m'encourager et de prier  
pour moi, Que Dieu, le tout-puissant, te préserve et t'accorde  
santé, longue vie et bonheur.*

*J'espère qu'ils trouveront dans ce travail toute ma  
reconnaissance et tout mon amour.*

*A mes très chères sœurs, **ROUFIEDA, OUMAMA AMINA**, et  
surtout ma chérie **NOUR EL-IMANE** qui était toujours à mes  
côtés, et mon cœur **MOHAMED AMINE**.*

*A tous mes amies: **KANZA, ROUMAÏSSA, NIHAD** qu'elles avec  
lesquels j'ai partagé des moments de joie et de bonheur, à tous  
ceux qui ont été à mes côtés jusqu'à aujourd'hui.*

*A tous les honorables enseignants qui ont contribué à ma  
formation.*

*A Toute la promotion Automatique et informatique industrielle  
2018 / 2019.*

*Tous ceux qui m'aiment.*

*Tous ceux que j'aime.*

**BELLILI SANA'A.**

## *Remerciements*

*Avant tout, je remercie **ALLAH** de nous avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la volonté, la patience et le courage durant ces longues années d'étude.*

*Au terme de ce modeste travail, je tiens à exprimer ma profonde gratitude et mes vifs remerciements au professeur **BOUMEHREZ.M** mon encadreur pour le soutien, l'aide et les conseils qu'il m'a dispensé. Qu'elle trouve dans ce travail un hommage vivant à sa haute personnalité.*

*Mes vifs remerciements vont également aux membres du jury **TOUBA.M** et **MIHI.A**, pour l'intérêt qu'ils ont porté à ma recherche en acceptant d'examiner mon travail et de l'enrichir par leurs propositions, je les remercie tout particulièrement pour l'attention et le temps qu'ils y ont consacrés pour lire ce manuscrit.*

*Je remercie mes parents pour leurs contributions, leurs soutiens, leurs patiences, leurs prières et leurs encouragements.*

*Je remercie, aussi, à toute l'équipe du laboratoire de génie électrique pour le soutien moral et pour leur accueil bienveillant et leurs conseils avisés.*

*Je souhaite adresser mes plus sincères remerciements à tous mes enseignants, mes proches et mes amis, pour leur encouragement au cours de la réalisation de ce travail et leur présence chaleureuse.*

*Enfin nous remercions tous ceux qui ont contribué de près ou de loin à la concrétisation de Ce travail.*

## Liste des figures :

### *Chapitre 01 :*

<b>Figure I.1</b> -Principales fonctions d'une application en robotique mobile .....	6
<b>Figure I.2</b> -Robots soudeurs sur une chaîne d'assemblage automobile .....	8
<b>Figure I.3</b> -Robot éclairateur tout-terrain.....	9
<b>Figure I.4</b> -Robot médical .....	10
<b>Figure I.5</b> -Détail du RAT (Rock Abrasion Tool), l'outil de prélèvement géologique. ....	11
<b>Figure I.6</b> -Robot aspirateur .....	11
<b>Figure I.7</b> -Robot tondeuse .....	12
<b>Figure I.8</b> -Robot intelligent.....	13
<b>Figure I.9</b> -Système multi robot.....	14

### *Chapitre 02 :*

<b>Figure II.1</b> -Carte Arduino Uno .....	22
<b>Figure II.2</b> -Le logo présent sur une carte Arduino authentique .....	22
<b>Figure II.3</b> -Fenêtre graphique de l'IDE .....	25
<b>Figure II.4</b> -Brochage de la carte Arduino Uno .....	30
<b>Figure II.5</b> -Carte Arduino .....	30
<b>Figure II.6</b> -Interface IDE Arduino.....	31
<b>Figure II.7</b> -Paramétrage de la carte étape 1 .....	32
<b>Figure II.8</b> -Paramétrage de la carte étape 2 .....	32
<b>Figure II.9</b> -Exemple de modules bluetooth.....	36
<b>Figure II.10</b> -Module shield wifi .....	37
<b>Figure II.11</b> -DriverL298N.....	38

## *Chapitre 03:*

<b>Figure III.1</b> -Schéma bloc .....	40
<b>Figure III.2</b> -Le GSM shield .....	41
<b>Figure III.3</b> -Brochage du GSM shield .....	42
<b>Figure III.4</b> -Module capteur de gaz MQ2.....	43
<b>Figure III.5</b> -Branchement de MQ2 avec l'arduino.....	44
<b>Figure III.6</b> -Module GPS .....	45
<b>Figure III.7</b> -Branchement de GPS à l'arduino .....	47
<b>Figure III.8</b> -Réalisation du système .....	57
<b>Figure III.9</b> -Test de rotation de deux petits moteurs à DC .....	58
<b>Figure III.10</b> -Tester un capteur de gaz MQ2 .....	59
<b>Figure III.11</b> -GPS test.....	60
<b>Figure III.12</b> -Exemple du message reçu .....	61



## Le sommaire :

Introduction générale .....	1
-----------------------------	---

## *Chapitre 01 :*

I.1 -Introduction .....	3
I.2 Historique .....	3
I.2.1. Automates (1ère génération) .....	3
I.2.2 -Robots (2ème génération) .....	4
I.2.3 -Robots dotés d'IA (3ème génération) .....	4
I.3 -Notion de base .....	4
I.3.1 Automate .....	4
I.3.2 -Robot .....	4
I.3.3 -Robot mobile .....	6
I.4 -Composantes d'une cellule robotisée .....	6
I.4.1 -Le mécanisme .....	6
I.4.2 -La perception .....	7
I.4.3 -La commande .....	7
I.4.4 -L'interface homme-machine .....	7
I.4.5 -Le poste de travaille les dispositifs péri-robotiques .....	7
I.5 -Classification des robots mobiles .....	7
I.6 -Domaines d'application de la robotique mobile... .....	8
I.6.1 -L'industrie .....	8
I.6.2 -Le domaine militaire. ....	9
I.6.3 -La santé. ....	9
I.6.4 -Utilisation civile .....	10
I.6.5 -L'usage domestique.....	11
I.7 -Système multi robot.....	13

I.7.1 -Définition .....	13
I.7.2 -Pourquoi un système multi robots .....	14
I.7.3 -Transport d'objets par un système multi-robots.....	15
I.7.3.1 -Système multi-robots homogènes .....	15
I.7.3.2 Systèmes multi-robots hétérogènes .....	15
I.7.4 -Les problèmes de la coordination dans les systèmes multi-robots .....	15
I.8 -Composants décisionnels d'un robot mobile .....	16
I.8.1 -La vision.....	16
I.8.2 -Localisation .....	17
I.8.3 -Représentation de l'environnement.....	17
I.8.4 -Planification .....	18
I.9 -Conclusion .....	19

## *Chapitre 02 :*

II .1 -Introduction .....	20
II.2 -Historique .....	20
II.3 -Qu'est-ce que c'est ? .....	21
II.4 -Pourquoi utiliser une carte Arduino ?.....	22
II.5 -Applications possibles.....	23
II.6 -Spécification technique de la carte Aduino .....	23
II.6.1 -Caractéristiques techniques.....	23
II.6.2 -logiciel .....	24
II.6.3 -Alimentation .....	25
II.6.4 -Les mémoires .....	26
II.6.5 -Entrées/Sorties numériques.....	27
II.6.6 -Broches analogiques .....	28
II.6.7 -Autres broches .....	28

II.6.8 -Communication.....	29
II.6.9 -Brochage de la carte Arduino Uno .....	29
II.7 -Partie programme .....	31
II.7.1 -L’environnement de la programmation .....	31
II.7.2 -Structure générale du programme (IDE Arduino) .....	31
II.7.3 -Injection du programme .....	32
II.7.4 -Description du programme .....	33
II.7.5 -Commentaires .....	33
II.7.6 -Définition des variables .....	33
II.7.7 -Programmation des interactions void loop .....	34
II.7.8 -Les étapes de téléchargement du programme .....	34
II.8 -Les accessoires de la carte Arduino .....	35
II.8.1 -Communication.....	35
II.8.1.1 -Les modules Bluetooth HC-05 et HC-06.....	35
II.8.1.2 -Le module shield Arduino wifi .....	36
II.8.2 - La commande des moteurs .....	37
II.8.3 -Les relais .....	38
II.9 -Conclusion .....	38

## *Chapitre 03:*

III.1 -Introduction .....	39
III.2 -Schéma bloc du système réalisé.....	39
III.3 -Les différentes étapes de la réalisation .....	40
III.4 -Lesdifférents modules du système .....	41
III.4.1 -GSM shield.....	41
III.4.1.a -Caractéristique électrique de GSM Shield .....	42
III.4.1.b -Brochage de la carte GSM shield .....	42

III.5 -Le capteur de gaz MQ2.....	43
III.5.1 -Caractériqtiq électrique de capteur MQ2.....	43
III.5.2 -Branchement du capteur MQ2 .....	44
III.6 -Le GPS « Global Positioning System » .....	44
III.6.1 –Fonctioonnement du GPS .....	45
III.6.2 –Précision du GPS .....	45
III.6.3 –Récepteur GPS.....	46
III.6.4 -Caractéréristiq électrique .....	46
III.6.5 -Branchement du GPS.....	46
III.7 -La carte de commande des moteurs .....	47
III.8 -Partie du programme IDE .....	47
III.8.1 -Programme "Send SMS" .....	47
III.8.2 -Programme "Receive SMS".....	49
III.8.3 –Programme concernant le fonctionnement du GPS.....	52
III.8.4 –Programme concernant de test de détecteur de gaz MQ2.....	56
III.9 -Tests du système .....	56
III.9.1-Test de la partie moteurs .....	57
III.9.2 -Test du module GSM.....	58
III.9.3 -Test du module GPS .....	59
III.9.4 -Test global .....	60
III.10 –Conclusion .....	61
Bibliographie	



*Introduction générale.*

## *Introduction générale*

L'homme a toujours souhaité se libérer du travail dans ce qu'il a d'astreignant, de fatiguant et d'inintéressant. Les progrès qu'ont connus les technologies depuis un demi-siècle ont permis de répondre en bonne parti à son désir par l'intermédiaire de deux outils génériques : l'ordinateur pour les travaux mentaux et le robot pour les travaux physiques. Le premier a pris une certaine avance sur le second grâce à la découverte des semi-conducteurs, ces matériaux solides qui peuvent trier des informations sans mobilité mécanique. Le second est toujours tributaire de la vieille mécanique aux pièces fragiles et mobiles et au coût significatif.

Malgré son aspect récent, la robotique tire ses origines des civilisations les plus antiques. Il est peut être même vrai que toutes les autres sciences ont existé juste pour permettre à la robotique de se développer afin d'aider l'homme à créer cet « esclave » qui appliquera ses ordres au doigt et à l'œil et qui le libérera à jamais du travail.

De l'homme préhistorique qui s'empressa d'inventer des extensions technologiques de ses membres afin de faciliter son travail, comme la massue par exemple, jusqu'au robot mobile autonome le plus performant, la robotique mobile n'a cessé de connaître des progrès spectaculaires. En effet, les générations de robots mobiles se sont vite succédé offrant à chaque fois plus de performances en matière de perception, d'autonomie et de pouvoir décisionnel. Ces progrès étaient un résultat inévitable des développements qu'ont connu la mécanique, la microélectronique et l'informatique. C'est pour cela qu'en très peu de temps, les générations de robots se succèdent pour voir naître une branche de la robotique visant de plus amples horizons : c'est l'ère de la robotique mobile.

L'objectif principal de notre travail est de réaliser un robot mobile, d'inspection de pipeline, à base d'une carte électronique Arduino. La détection du gaz dégagé est utilisée pour la recherche des fissures et des mauvaises soudures. L'existence du gaz signifie la présence d'une fissure ou d'une mauvaise soudure.

Ce mémoire est organisé en trois chapitres de la manière suivante :

- Le premier chapitre présente certaines généralités concernant la robotique mobile.
- Le deuxième chapitre est une description générale de la carte Arduino Uno.
- Dans le troisième chapitre, on présente le robot réalisé et le robot réalisé, ses composants et son fonctionnement avec ses trois niveaux : mécanique, électronique et informatique.
- Enfin, la conclusion présente le bilan de ce travail et les perspectives envisagées.



*Chapitre 01 : Généralité sur les robots  
mobile.*

## I.1 -Introduction

La révolution robotique porte de grands espoirs et des problèmes non moins importants. Nous risquons d'oublier que ces machines sont dirigées par des algorithmes, et connectées en permanence à leurs fabricants. Un autre risque est que certains d'entre nous finissent par croire qu'elles ont « pour de vrai » les émotions qu'elles simuleront à la perfection. Enfin, de la même manière que nos téléphones mobiles nous ont rendus moins tolérants à l'attente, il y a un risque que ces machines nous rendent moins tolérants au caractère imprévisible de l'humain. Mais face à chacun de ces risques, des remèdes existent. Ils sont à la fois législatifs, technologiques et éducatifs.

## I.2 -Historique

On peut schématiquement distinguer trois principales ères en robotique : les automates, les robots n'étant pas dotés d'intelligence artificielle (IA), et ceux disposant d'une IA [1].

### I.2.1 -Automates (1ère génération)

Un automate contrairement à un robot (même s'il ne dispose pas d'IA), obéit uniquement à un programme préétabli, que ce soit de manière mécanique ou électronique. De ce fait, il n'y a aucune adaptation possible entre l'automate et son environnement. On attribue la paternité du tout premier automate de l'humanité à Architos de Tarente (4ème siècle avant J.C). Il s'agissait d'une représentation d'un pigeon capable de voler et étant propulsé par de la vapeur. Malheureusement, aucun vestige ni aucun schéma ou représentation fidèle n'ont été retrouvés.

Les toutes premières traces d'automates remontent à l'antiquité par Héron d'Alexandrie au 1er siècle après J.C., ses réalisations ornèrent les temples et les théâtres de la ville égyptienne. En 1495 Léonard de Vinci présenta un chevalier humanoïde capable de s'asseoir, de relever sa visière et de bouger ses bras. L'automate le plus célèbre est le canard mécanique de Jacques De Vaucanson, capable d'ingurgiter de la nourriture et de la digérer tout en se déplaçant. De Vaucanson aurait également présenté en 1738 un second automate représentant un homme jouant d'un instrument à vent.

### **I.2.2 -Robots (2ème génération)**

Les robots de 2ème génération disposent d'organe(s) sensoriel(s), autrement dit des capteurs, pouvant influencer sur leurs comportements. Ils sont donc relativement adaptables à leur environnement. Le chien électrique de Hammond et Miessner (1915) est le premier robot de ce genre. Il se déplace selon la luminosité de l'endroit grâce à son capteur optique. Le chien Phillipdog de Henri Piraux en 1928, et le renard de Ducrocq (1953) fonctionnent selon le même principe. Walter Grey équipe en 1950 sa tortue cybernétique de capteurs tactiles et lumineux.

### **I.2.3 -Robots dotés d'IA (3ème génération)**

Pour être intégré dans cette famille, un robot doit parvenir à effectuer une tâche par lui-même, sans aucune aide extérieure. En 1973, l'université de Waseda présente le tout premier humanoïde "intelligent" dénommé Wabot-1. Il est doué de la vision, peut manipuler des objets, effectue un semblant de marche et est même capable de débiter une conversation en japonais. Hi-T-Hand d'Hitachi en 1974 manipule des aiguilles à travers des trous grâce à la détection de force. Vers la fin des années 70 Hans Morava présente les premiers robots capables d'évoluer à l'extérieur, de façon autonome.

## **I.3 -Notions de base**

Dans cette partie nous allons introduire quelques notions concernant notre sujet, on peut citer les suivantes :

### **I.3.1 -Automate**

Un **automate** est un dispositif reproduisant en autonomie une séquence d'actions prédéterminées sans l'intervention humaine. Le système fait toujours la même chose, ou s'adapte à des conditions environnementales perçues par ses capteurs. L'automate est un objet programmé [2].

### **I.3.2 -Robot**

D'un point de vue technique, un **robot** est une machine dotée de capteurs qui lui permettent de percevoir son environnement, de moteurs l'autorisant à bouger et à agir sur

cet environnement, et d'un système électronique ou informatique qui contrôle, au moins en partie, ce qu'effectue le robot en fonction de ce qu'il perçoit. En pratique, cette définition recouvre une très vaste diversité de machines. Par exemple, les bras articulés et programmables dans les usines automobiles, les aspirateurs qui font le ménage tout seul, les robots ludiques et électroniques des magasins de jouets, ou les robots bio-mimétiques à la forme de singes ou de poissons que l'on rencontre parfois dans les laboratoires universitaires de recherche [3].

Cette diversité n'est pas seulement une diversité de formes, c'est aussi une diversité de fond. Les mécanismes de fonctionnement interne diffèrent beaucoup d'un robot à l'autre et on peut les distinguer selon quatre axes :

**L'autonomie** : il existe des robots qui agissent sans le besoin qu'un humain ne les guide et d'autres dont le comportement est, soit influencé par un humain soit, presque totalement contrôlé par un humain. Par exemple, dans une usine, les robots qui travaillent à la chaîne et répètent toujours le même geste le font souvent de manière autonome.

**L'adaptabilité** : certains robots adoptent un comportement différent selon ce qu'ils perçoivent, c'est-à-dire qu'ils s'adaptent aux changements dans leur environnement, alors que d'autres se comportent toujours de la même manière.

**L'apprentissage** : il y a des robots dont le comportement, même s'il est adaptatif, est figé au départ et une fois pour toute par le programmeur, alors que d'autres robots sont capables d'acquérir de nouveaux comportements et de nouvelles connaissances par leurs expériences.

**La créativité** : certains robots, enfin, sont capables d'inventer des solutions et des comportements non prédits par leur concepteurs, et même de s'inventer des objectifs qui ne sont pas pré-programmés (mais cela ne veut pas dire qu'ils soient capables de tout inventer ou de tout faire, loin de là). C'est le cas par exemple des robots équipés de mécanismes dits de « curiosité » qui les poussent à s'intéresser aux choses nouvelles de leur point de vue, comme ceux construits par les chercheurs en robotique développementale [3].

### I.3.3 -Robot mobile

Un robot mobile est un véhicule doté de moyens de locomotion qui lui permettent de se déplacer. Suivant son degré d'autonomie il peut être doté de moyens de perception et de raisonnement [4].

Un robot mobile autonome est un système doté de moyens de perception de son environnement partiellement ou totalement inconnu. Il est capable de prendre des décisions pour accomplir une tâche donnée sans intervention humaine. Nous pouvons mettre en évidence les principales fonctions qui interviennent dans une application bâtie autour d'un robot mobile comme montré sur le schéma de la figure I.1 [5].

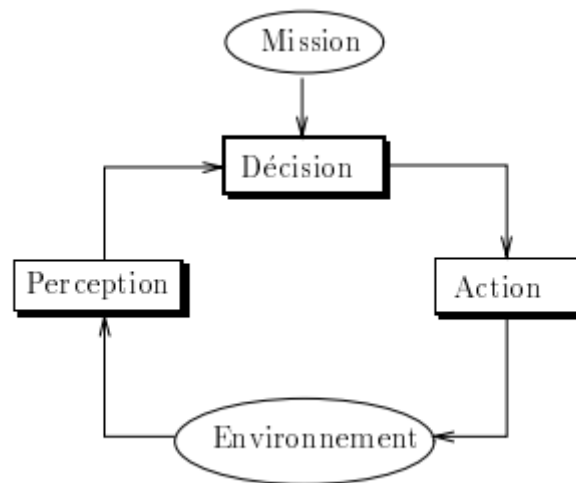


Figure I.1 -Principales fonctions d'une application en robotique mobile.

## I.4 -Composantes d'une cellule robotisée

### I.4.1 -Le mécanisme

Dans notre étude, nous avons pris le système d'un robot ayant une structure ressemble beaucoup à celle du bras humain, il permet de remplacer ou de prolonger son action (le terme manipulateur exclue implicitement les robots mobiles). Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

#### **I.4.2 -La perception**

Elle permet de gérer la relation entre le robot et son environnement. Les organes de perception sont des capteurs dits "proprioceptifs" lorsqu'ils mesurent l'état interne du robot, et "extéroceptifs" lorsqu'ils recueillent des informations sur l'environnement.

#### **I.4.3 -La commande**

Elle synthétise les consignes des asservissements pilotant les actionneurs. A partir de la fonction de perception et des ordres de l'utilisateur, elle permet d'engendrer les actions du robot.

#### **I.4.4 -L'interface homme-machine**

A travers laquelle l'utilisateur programme les tâches que le robot doit exécuter.

#### **I.4.5 -Le poste de travail les dispositifs péri-robotiques**

Ils constituent l'environnement dans lequel évolue le robot. [20,21].

### **I.5 -Classification des robots mobiles**

La classification des robots mobiles se fait suivant plusieurs critères (degré d'autonomie, système de locomotion, énergie utilisée ...). La classification la plus intéressante, et la plus utilisée est selon leur degré d'autonomie. Un robot mobile autonome est un système automoteur doté de capacités décisionnelles et de moyens d'acquisition et de traitement de l'information qui lui permettent d'accomplir sous contrôle humain réduit, un certain nombre de tâches, dans un environnement non complètement connu. On peut citer quelques types :

- Véhicule télécommandé par un opérateur qui lui impose chaque tâche élémentaire à réaliser,
- Véhicule télécommandé au sens de la tâche à réaliser,
- Le véhicule contrôle automatiquement ses actions,
- Véhicule semi- autonome réalisant sans l'aide de l'opérateur des tâches prédéfinies,
- Véhicule autonome qui réalise des tâches semi- définies.

Les principaux problèmes particuliers liés à la conception de tels robots sont :

1. La conception mécanique liée à la mobilité,
2. La détermination de la position et de l'orientation,
3. La détermination du chemin optimal [4].

## I.6 -Domaines d'application de la robotique mobile

La robotique est un domaine en plein essor depuis quelques années. Les évolutions technologiques, dépassant sans cesse nos espérances, permettent maintenant de réaliser des solutions technologiques s'adaptant au moindre problème.

Par conséquent, la robotique est utilisée dans des domaines extrêmement rigoureux et exigeants. Nous allons explorer ces différents domaines [6].

### I.6.1 -L'industrie

Le but premier des robots est de remplacer l'homme dans des activités fastidieuses ou onéreuses pour l'employeur. Les robots ont donc commencé à être utilisés dans les chaînes d'assemblage industrielles. Dans ces chaînes d'assemblage, on retrouve des robots soudeurs, manipulateurs, peintres ... etc



**Figure I.2** -Robots soudeurs sur une chaîne d'assemblage automobile.

### I.6.2 -Le domaine militaire

Les robots sont de plus en plus utilisés dans le domaine militaire. En effet, la miniaturisation permet aujourd'hui de créer des robots discrets mais dotés de nombreux capteurs, ce qui est idéal pour des missions d'espionnage ou d'éclairage.



Figure I.3 -Robot éclaireur tout-terrain .

### I.6.3 -La santé

Les robots commencent à être de plus en plus dans le domaine médical, qu'il s'agisse de « simples » échographies ou d'opérations chirurgicales plus délicates. En fait ces robots ne sont pas complètement autonomes mais ils assistent les médecins ou chirurgiens, jusqu'à permettre des opérations médicales à distance (télémédecine). On parle de surgétique (mot né de l'anglais « surgery » : chirurgie) c'est-à-dire tout ce qui consiste à introduire les derniers outils des technologies informatiques et robotiques dans la pratique médico-chirurgicale. Cette pratique de « chirurgie assistée » est émergente donc bien que peu répandue, elle est en phase de devenir la chirurgie du futur.





Figure I.4 -Robot médical.

#### I.6.4 -Utilisation civile

De part leur polyvalence, de plus en plus de tâches sont confiées aux robots. Voici par exemple Yukitaro, un robot japonais conçu pour déneiger :

Il est équipé de deux caméras et d'un GPS. Il circule dans la rue de manière autonome en engouffrant la neige avec sa « bouche » pelleuse, puis il la rejette sous forme de blocs de glace.

Les robots sont également utilisés à des fins scientifiques. En témoignent les robots de la NASA Spirit et Opportunity, destinés à explorer la planète Mars et à récolter toutes sortes d'informations sur cette planète.

Ces deux robots sont équipés de caméras (dont une microscopique), de spectromètres (pour étudier la composition minéralogique des roches martiennes), d'un équipement informatique (assimilable à un micro-ordinateur), et d'un outil de perforation des roches pour effectuer des prélèvements. Ces robots peuvent être reprogrammés à distance, ce qui les rend opérationnels à long terme et d'autant plus polyvalents.



**Figure I.5** -Détail du RAT (Rock Abrasion Tool), l’outil de prélèvement géologique.

### **I.6.5 -L’usage domestique**

La démocratisation de la robotique a conduit, ces dernières années, à voir de nombreux robots s’installer chez les particuliers pour effectuer des tâches à la place de leur possesseur. En effet, ceux-ci sont capables de faire le ménage, tondre la pelouse, nettoyer la piscine... Ce qui conduit certains clients (aisés) à se procurer ces domestiques contemporains.

Robot aspirateur, fonctionnant de manière autonome sur tous types de sol. Grâce à ses capteurs, il évite les escaliers, et retourne à sa base pour se recharger sans assistance. Il est possible de délimiter une surface précise à nettoyer avec les accessoires fournis.



**Figure I.6** -Robot aspirateur.

Robot tondeuse, certes moins efficace que ses homologues manuels, cette tondeuse permet quand même de rafraîchir sa pelouse pendant que l'on désherbe les rosiers ou mieux, pendant que l'on profite du beau temps ou de la piscine :

On trouve ce genre de tondeuse à un prix débutant aux environs de 2000€, jusqu'à 4000€ ou plus suivant les caractéristiques : temps d'autonomie, modèle hybride ou non, efficacité de tonte, capacité à s'adapter au relief...



**Figure I.7** -Robot tondeuse.

Enfin, la robotique, autrefois réservée à des applications précises ou coûteuses, est aujourd'hui de plus en plus utilisée à titre ludique. En effet, les robots compagnons par exemple sont des objets de plus en plus convoités : les applications « basiques » de jouet pour enfant, jusqu'à l'humanoïde destiné à remplacer une présence humaine.

Robosapien (trois versions existantes), un petit humanoïde capable d'interagir avec son environnement. Il peut également évoluer les autres robots de sa gamme, du constructeur WowWee (Roboquad, Robopanda, Robope [6].



**Figure I.8** -Robot intelligent.

### **I.7 -Système multi robot**

La notion des systèmes multi-agent a débuté dans les années 1990, notamment dans des travaux regroupant des robots mobiles rassembleurs d'objets [7], des colonies des robots marcheurs.

L'objectif dans ce type de tâches est de transporter un objet par un ensemble de robots qui peuvent être homogènes ou hétérogènes [8].

#### **I.7.1 -Définition**

Une flotte de robots autonomes est constituée de plusieurs robots capables de partager des données et d'effectuer ensemble une ou plusieurs tâches. ces robots sont capables de communiquer entre eux et coopérer pour améliorer l'efficacité d'exécution des tâches ou encore pour permettre d'exécuter des tâches impossibles à exécuter de façon individuelle [8,9].

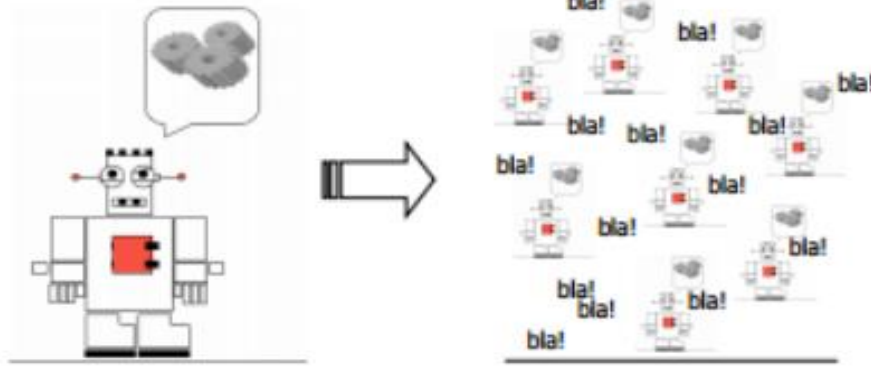


Figure I.9 -Système multi robot .

### I.7.2 -Pourquoi un système multi robots

Un **système multi robots** peut effectuer des tâches difficiles ou même impossibles à accomplir par un seul robot. Une équipe de robots fournit une certaine redondance, elle contribue à l'accomplissement d'une tâche de manière collaborative. Ce système offre plusieurs avantages par rapport aux systèmes mono-robot ; les motivations les plus communes pour le développement de solutions de systèmes multi robots sont les suivants [10] :

- La complexité de la tâche à réaliser est trop élevée pour un seul robot donc elle nécessite une coopération d'un ensemble de robots.
- La construction de plusieurs robots simples avec des ressources bornées est beaucoup plus facile que la construction d'un seul robot complexe et puissant.
- L'introduction de plusieurs robots augmente la robustesse grâce à la redondance.
- La rapidité : pour obtenir un niveau de performance élevé les tâches sont effectuées en parallèle. A titre d'exemple : l'exploration parallèle d'un environnement inconnu par un ensemble de robots mobiles, dans le but de faire soit une cartographie de l'environnement, soit la réalisation d'une tâche de fourragement. Ces deux tâches peuvent être réalisées par un seul robot, mais l'ajout d'autres robots va faire en sorte d'accélérer l'exécution des tâches en question.

- La robustesse : les performances du contrôle dans un SMR ne sont pas trop affectées en cas de défaillance d'un robot.
- La flexibilité : il est possible d'exécuter les tâches désirées, de diverses manières. Ceci est induit principalement par la redondance des entités robotiques.

### I.7.3 -Transport d'objets par un système multi-robots

#### I.7.3.1 -Système multi-robots homogènes

Un **système multi-robots homogènes** est un ensemble de robots qui ont les mêmes caractéristiques, ils doivent coopérer entre eux; ils s'auto-organisent pour soulever, porter pousser et déposer l'objet qu'un seul robot ne pourrait réaliser tout seul. Des travaux dans ce domaine ont été réalisés, on peut citer par exemple les travaux de Stiwel :plusieurs robots porteurs qui se positionnent en dessous d'une palette chargée pour la déplacer. Les robots mobiles disposent d'un capteur de force qui leur sert d'information pour contrôler le déplacement de la palette d'une manière stable et distribuée [10].

#### I.7.3.2 -Systèmes multi-robots hétérogènes

Un **SMR hétérogènes** est un ensemble de robots mixtes, c'est-à-dire les robots ne sont pas identiques, il peut y avoir des robots fixes et des robots mobiles; dans ce cas chaque robot s'occupe de sa propre sous tâche, les robots manipulateurs fixes s'occupent du dépôt et le soulèvement d'objet et les robots mobiles s'occupent de son transport [8,10].

### I.7.4 -Les problèmes de la coordination dans les systèmes multi-robots

Un système multi-robots est considéré comme un système distribué, puisque les tâches sont distribuées sur l'ensemble des robots du système. L'un des problèmes des systèmes multi-robots est la coordination, ce problème est séparé en quatre parties, la communication, la computation, la configuration, et la coordination [11].

- La communication : définir comment les agents communiquent entre eux.

- La computation : elle définit la mise en œuvre du comportement des agents individuels. Elle détermine ainsi ce qui est communiqué.
- La configuration : elle définit la structure d'interaction et elle indique quel sont les agents existants dans le système et quel sont les agents qui peuvent communiquer entre eux.
- La coordination : elle définit les modes d'interaction, c'est-à-dire déterminé à quel moment certaines communications ont lieu.

## **I.8 - Composants décisionnels d'un robot mobile**

Afin de faire fonctionner un robot mobile, plusieurs modules logiciels sont mis à contribution. Ces modules peuvent servir à interpréter les données perçues par les capteurs afin d'y extraire des informations, ou à traiter des commandes de haut niveau pour générer d'autres commandes à un niveau inférieur. Les modules les plus fréquemment utilisés sont les modules de vision, localisation, de navigation, de planification et de séquençement d'activités du robot. Ces aspects de ces modules logiciels seront donnés avec plus de détails dans les sections suivantes [12].

### **I.8.1 -La vision**

La **vision** fournit une grande quantité de données en provenance de l'environnement et permet d'entreprendre une interaction intelligente avec les environnements dynamiques (éviter d'obstacles mobiles, rendez-vous avec autres agents mobiles. . .). De ce fait, il n'est pas surprenant de trouver une grande quantité de recherche sur le développement de capteurs qui essaient d'imiter le système visuel humain. De plus, les capteurs visuels utilisés par les robots intelligents doivent avoir les mêmes sensibilités et réponses à la lumière que notre système de vision. En robotique, au cours des deux dernières décennies, les innovations technologiques concernant la fabrication de caméras et l'évolution des ordinateurs ont permis d'intégrer des systèmes complexes de vision dans les systèmes embarqués, que ce soit sur des robots mobiles pour la navigation autonome ou sur des

véhicules pour l'aide à la conduite. La vision artificielle revêt une importance toute particulière car elle permet de fournir à la machine les capacités nécessaires pour réagir avec son environnement; elle fournit les représentations à partir desquelles le robot prend des décisions[12,13].

### **I.8.2 -Localisation**

Un robot mobile doit toujours connaître sa situation pour se déplacer d'un point à un autre, soit en identifiant des repères, on parle de localisation absolue, soit tout simplement en mesurant les distances parcourues et les directions empruntés depuis sa position initiale. Les méthodes de localisation se regroupes en deux catégories, soit :

-La localisation à l'estime ou relative qui est obtenue par des informations issues de capteurs proprioceptifs .

-La localisation absolue qui est obtenue par des informations issues de capteurs extéroceptifs.

-Localisation à l'estime ou relative: obtenue par des informations issues des capteurs proprioceptifs, et consiste à déterminer la variation des coordonnées de position lors d'un déplacement en mesurant tout simplement les distances parcourues et les directions, empruntées depuis sa position initiale.

-Localisation absolue: obtenue par des informations issues de capteurs extéroceptifs; le robot doit toujours connaître sa situation pour se déplacer d'un point à un autre en identifiant des repères artificiels, la méthode des balises est la plus employée[14,28].

### **I.8.3 -Représentation de l'environnement**

La localisation a besoin, sous une forme ou une autre, d'une représentation informatique de l'environnement. Il existe deux types principaux de cartes : les grilles d'occupation et les graphes. Dans le premier cas, l'environnement est quadrillé en cases.



Durant l'exploration, à l'aide des sonars ou d'un télémètre laser, chacune de ces zones se voit attribuer une probabilité d'être occupée par un obstacle. Le résultat est une image probabiliste des murs et des espaces libres. Cette représentation donne lieu à une localisation sous forme de coordonnées qui désignent une case du quadrillage.

L'un des inconvénients des grilles d'occupation vient de la grande quantité de données qu'il faut manipuler. L'exploitation de ces représentations est coûteuse en temps de calcul [29].

Les cartes topologiques ne s'intéressent qu'aux points clefs du milieu. Elles ont la forme d'un graphe où les sommets figurent des endroits spécifiques (intersections de couloirs, changement de classe de perception...) et les arcs l'existence d'un chemin entre deux sommets. C'est une vue plus abstraite de l'environnement que celle proposée par les grilles d'occupation. Elle vise à ne conserver que l'information utile.

Si le quadrillage est précis, il manque de discernement. Un long couloir uniforme ne nécessite pas à priori une accumulation de données sous formes de milliers de probabilités. Notons que ces deux types de cartes ne sont pas exclusifs [30].

#### **I.8.4 -Planification**

Une fois que le robot dispose d'une représentation de l'environnement et d'une estimation de sa position, il convient de s'en servir pour générer les tâches qui lui permettront d'atteindre son objectif. Pour cela plusieurs méthodes sont possibles. Le choix d'une méthode de planification est guidé par deux questions :

1. Quel type d'espace utilisé : l'espace de travail ou l'espace des configurations?
2. Quel type de méthodes : des méthodes exactes ou des méthodes approchées?

Les méthodes exactes sont basées sur une exploitation complète de la description de l'environnement. Par opposition, les méthodes approchées réalisent tout d'abord une discrétisation de l'environnement sous forme de grilles régulières ou irrégulières. L'espace libre ainsi représenté est un sous-ensemble de l'espace libre réel.

Alors que les méthodes exactes sont susceptibles d'être complètes, les méthodes approchées ne le sont jamais [31].

### **I.9 -Conclusion :**

Dans ce chapitre, nous avons donné une idée générale sur la robotique, l'historique des robots, leurs structures, leurs utilisations et les différents types de robots ainsi que leurs classifications et domaines d'applications et cela va nous servir à construire notre projet.

## *Chapitre 02 : La carte Arduino.*

## **II.1 -Introduction**

Arduino est une plate-forme basée sur un matériel et un logiciel "Open Source". Elle se compose d'une carte programmable à base d'un microcontrôleur et d'un logiciel prêt à l'emploi appelé Arduino IDE (Integrated Development Environment), qui est utilisé pour écrire et transférer le programme sur la carte.

Arduino fournit un facteur de forme standard qui brise les fonctions du microcontrôleur dans un package plus accessible.

## **II.2 -Historique**

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea. Ils rencontraient un problème majeur à cette période (avant 2003 - 2004) : les outils nécessaires à la création de projets d'interactivité étaient complexes et onéreux . Ces coûts souvent trop élevés rendaient difficiles le développement par les étudiants de nombreux projets et ceci ralentissait la mise en œuvre concrète de leur apprentissage.

Jusqu'alors, les outils de prototypage étaient principalement dédiés à l'ingénierie, la robotique et aux domaines techniques. Ils sont puissants mais leurs processus de développement sont longs et ils sont difficiles à apprendre et à utiliser pour les artistes, les designers d'interactions et, plus généralement, pour les débutants.

Leur préoccupation se concentra alors sur la réalisation d'un matériel moins cher et plus facile à utiliser. Ils souhaitaient créer un environnement proche de Processing, ce langage de programmation développé dès 2001 par Casey Reas et Ben Fry, deux anciens étudiants de John Maeda au M.I.T., lui-même initiateur du projet DBN.

En 2003, Hernando Barragan, pour sa thèse de fin d'études, avait entrepris le développement d'une carte électronique dénommée Wiring, accompagnée d'un environnement de programmation libre et ouvert. Pour ce travail, Hernando Barragan réutilisait les sources du projet Processing basé sur un langage de programmation facile d'accès et adapté aux développements de projets de designers, la carte Wiring a donc inspiré le projet Arduino (2005).

Comme pour Wiring, l'objectif était d'arriver à un dispositif simple à utiliser, dont les coûts seraient peu élevés, les codes et les plans « libres » (c'est-à-dire dont les sources sont ouvertes et peuvent être modifiées, améliorées, distribuées par les utilisateurs eux-mêmes) et, enfin, « multi-plates-formes » (indépendant du système d'exploitation utilisé).

Conçu par une équipe de professeurs et d'étudiants (David Mellis, Tom Igoe, Gianluca Martino, David Cuartielles, Massimo Banzi ainsi que Nicholas Zambetti), l'environnement Arduino est particulièrement adapté à la production artistique ainsi qu'au développement de conceptions qui peuvent trouver leurs réalisations dans la production industrielle.

Le nom Arduino trouve son origine dans le nom du bar dans lequel l'équipe avait l'habitude de se retrouver. Arduino est aussi le nom d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort » [18].

### **II.3 -Qu'est-ce que c'est ?**

**Une carte Arduino** est une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout un chacun dans le but de créer facilement des systèmes électroniques [17].



**Figure II.1** -Carte Arduino Uno.



**Figure II.2** -Le logo présent sur une carte Arduino authentique.

#### **II.4 -Pourquoi utiliser une carte Arduino ?**

Car elle est :

- Accessible et pas très cher et le logiciel est fourni gratuitement.
- Compatible avec toutes les plateformes (Windows, Mac OS, Linux).

- Une liberté quasi absolue :
  - Le logiciel est gratuit et open source, développé en Java,
  - Le matériel (la carte électronique) dont le schéma est en libre,
  - Circulation sur internet à condition que le nom Arduino ne soit employé que pour les cartes officielles [16].

## II.5 -Applications possibles

L'arduino peut être utilisé dans diverses applications telles que :

- Le contrôle des appareils domestiques,
- Donner une intelligence à un robot,
- Réaliser des jeux de lumière,
- Télécommander un appareil mobile,
- etc. ...

## II.6 -Spécification technique de la carte Arduino

### II.6.1 -Caractéristiques techniques

- Microcontrôleur ATmega328P,
- Tension de fonctionnement 5V,
- Tension d'entrée (recommandée) 7-12V,
- Tension d'entrée (limite) 6-20V,
- E / S numériques Pins 14 (dont 6 fournissent la sortie PWM\*),
- PWM numérique E / S Pins 6,
- Pins d'entrée analogique 6,
- DC Courant par I O Pin / 20 mA,
- Courant DC pour 3.3V Pin 50 mA.

- Mémoire flash 32 KB (ATmega328P) dont 0,5 KB utilisé par le bootloader,
- SRAM 2 KB (ATmega328P),
- EEPROM, 1 KB (ATmega328P),
- Vitesse de l'horloge 16 MHz,
- Longueur 68,6 mm,
- Largeur 53,4 mm.
- Poids 25 g [19],

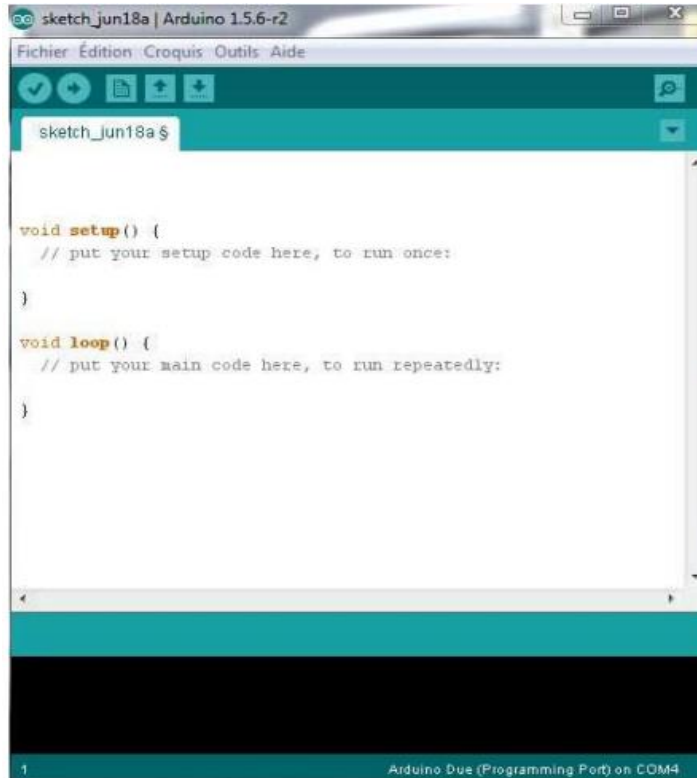
### **II.6.2 -Logiciel de programmation**

L'environnement de programmation Arduino ,est un logiciel de programmation des modules Arduino , une application écrite en Java inspirée du langage Processing, libre et multiplateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série ( Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler les programmes en ligne de commande.

Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne maitrisant le C ou le C++.

Un programme utilisateur 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 code, comme lors d'une programmation classique. Cette structure se décompose en deux parties, Fig. (II.3).





**Figure II.3** -Fenêtre graphique de l'IDE .

- Fonction configuration : configuration des entrées/sorties et éléments à configurer (cette partie ne sera exécutée qu'une seule fois) dans le partie void setup().
- Fonction principale : description du fonctionnement général du programme (gestion des interactions entre les entrées/sorties) dans la partie void loop ( ) [22].

### **II.6.3 -Alimentation**

Diverses possibilités s'offrent pour alimenter cette carte. Ces trois façons sont expliquées

Plus précisément, la carte nécessite une tension de 5V pour pouvoir fonctionner. C'est la tension que délivre le port USB.

Dès lors que nous alimentons l'Arduino via le jack ou les pins VIN et GND, un régulateur de tension prend le relais entre l'alimentation et la carte. La tension entrante doit être comprise entre 6 et 20V. Ce sont les limites maximum.

Toutefois, il est préférable de restreindre cet intervalle pour garantir un bon fonctionnement de la carte :

- En dessous de 7V, la carte peut être instable avec une tension en sortie de régulateur pouvant être inférieure aux 5V nécessaires.
- Au dessus de 12V, le régulateur peut chauffer de manière trop importante et se mettre en sécurité, voire endommager la carte.
- La tension idéale est de 9V.

Le jack d'alimentation a le + au centre et le – à l'extérieur. Une diode présentée sur la carte protège d'un mauvais branchement.

Quelle que soit la source d'alimentation, la carte la sélectionne automatiquement. On peut donc avoir simultanément l'Arduino connecté en USB et alimenté par une autre source comme un adaptateur secteur 9V [23].

### II.6.4 -Les mémoires

#### **Mémoireflash**

Elle est de 32 ko dont 0,5 ko sont utilisés pour le *bootloader*, son programme de démarrage. Cette mémoire est l'équivalent du disque dur pour l'ordinateur. C'est la place que nous avons pour stocker le programme.

#### **SRAM**

Équivalent à la mémoire RAM, elle sert à stocker le résultat des variables. Sa taille est de 2 ko. Comme la RAM, cette mémoire est volatile, à l'extinction de la carte, les valeurs disparaissent.

#### **EEPROM**

Mémoire en dur, elle permet de sauvegarder des valeurs de variables et ceci même à l'extinction de la carte. Sa taille est de 1 kB. Comme une carte SD, le nombre de réécritures est limité.

Ces mémoires sont limitées en taille. Il s'agit donc d'optimiser au maximum afin qu'une carte Arduino puisse recevoir le programme et l'exécuter correctement [39].

### II.6.5 -Entrées/Sorties numériques

Elles sont numérotées de 0 à 13. Chacune de ces broches peuvent être utilisée comme entrée ou sortie digitale avec les fonctions :

- `pinMode()`,
- `digitalWrite()`,
- `digitalRead()`.

Le voltage est à 5V, en sortie numérique, HIGH veut dire que la broche est mis à 5V et LOW à 0V.

En sortie, ces broches peuvent fournir chacune un courant maximum de 40 mA. Mais il est recommandé de ne pas dépasser 20mA. A noter aussi que la consommation totale toutes broches confondues ne doit pas dépasser 200mA.

Si vous avez besoin de courant sur plusieurs broches à la fois, la somme des courants est encore limitée.

A noter qu'une broche de sortie peut soit « fournir » du courant (SOURCE en anglais), soit « consommer » du courant (SINK en anglais). Les valeurs de courant maximum ne sont pas exactement les mêmes dans ces deux cas.

En entrée, elles ont une résistance de tirage (pull-up) interne de 20-50 kOhms qui est désactivée par défaut. Cette résistance qui est à l'intérieur du circuit, permet de « tirer » le niveau de cette entrée au 5V (HIGH) lorsque rien n'y est connecté ou ne la relie au 0V. Cela permet d'éviter d'en ajouter une à l'extérieur.

Pour activer cette résistance de tirage, il suffit de déclarer l'entrée de cette façon :

➤ `pinMode (Pin_xx, INPUT_PULLUP);`

Certaines broches ont aussi des fonctions spécialisées en plus de celles de base :

- la fonction la plus connue est PWM, qui est décrite plus précisément ici : La PWM , Les broches concernées sont 3, 5, 6, 9, 10 et 11.

- Voie série sur les broches 0 (Rx) et 1 (Tx) : R pour "Received "et T pour "Transmit". Ces broches sont connectées à l'Atmega 16U2 qui s'occupe de la voie série-USB. Il est donc fortement déconseillé d'effectuer un branchement de cette voie série lors d'un téléchargement de programme.
- Interruptions externes sur les broches 2 et 3.
- la broche 13 est un peu particulière car une LED incorporée sur la carte lui est reliée. Cette broche reste malgré tout utilisable si l'on accepte de voir la Led clignoter!
- les broches 10 (SS), 11 (MOSI), 12 (MISO) et 13 (SCK) sont utilisées pour communiquer avec le protocole SPI(Serial Peripheral Interface) permettant des transferts à haute vitesse avec de nombreux périphériques comme les écrans graphiques par exemple [23].

### II.6.6 -Broches analogiques

La carte possède 6 broches analogiques numérotées de A0 à A5. Par défaut, elles mesurent le voltage de 0 à 5 V. Toutefois, il est possible de changer cette limite supérieure en utilisant la broche AREF et la fonction analogReference().

Ces broches sont associées à des convertisseurs analogiques/numériques intégrés de 10 bits qui donnent une valeur comprise entre 0 et 1023.

De même que les broches numériques, certaines ont des fonctions spécialisées :

- les broches A4 (SDA) et A5 (SCL) sont utilisées pour communiquer en utilisant la Bibliothèque Wire. Elles constituent le bus I2C (Integrated Integrated Circuit) [15].

### II.6.7 -Autres broches

#### **GND**

Ce sont les masses. Elles servent à fermer le circuit.

#### **3V3**

Cette broche délivre du 3V3 pour un ampérage maximum de 50 mA. Elle peut être utilisée pour alimenter un composant à ce voltage.

### **AREF**

Voltage de référence pour les broches analogiques, à utiliser avec la fonction `analogReference()`.

### **Reset**

Mettre cette broche à LOW permet de redémarrer la carte. Elle est beaucoup utilisée avec les shields pour permettre un redémarrage lorsque le bouton physique de la carte est inaccessible.

### **IOREF**

Cette broche donne le voltage de référence à lequel la carte fonctionne. Dans le cas du Uno, elle sera à 5V. Sert surtout pour les shields [23].

## **II.6.8 -Communication**

Pour communiquer, l'Arduino Uno peut employer différents protocoles comme nous l'avons vu précédemment :

- la voie série avec le port USB ainsi que les broches numériques 0 et 1. Précisons qu'à l'initialisation de la voie Série (par exemple quand on allume le terminal de l'IDE), la carte Arduino se réinitialise.
- SPI avec les broches numériques 10, 11 12 et 13.
- I2C / Wire avec les broches analogiques A4 et A5
- de plus, la librairie Software Serial permet d'utiliser n'importe quelle broche numérique pour communiquer en voie série [15,25].

## **II.6.9 -Brochage de la carte Arduino Uno**

La figure II.4 montre le brochage de la carte Arduino.

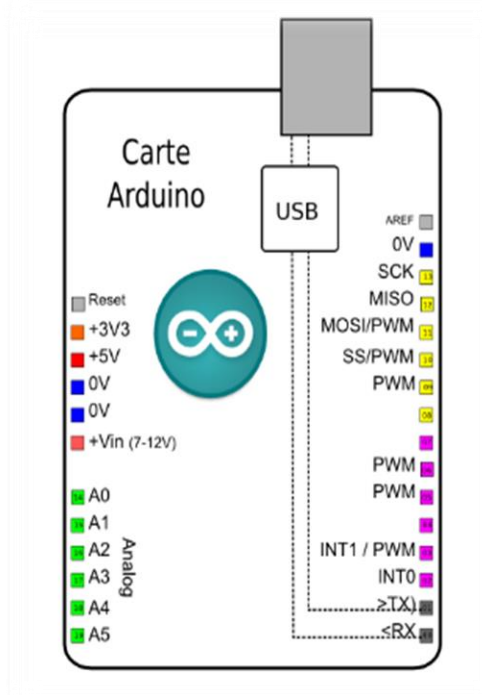


Figure II.4 -Brochage de la carte Arduino Uno [24].

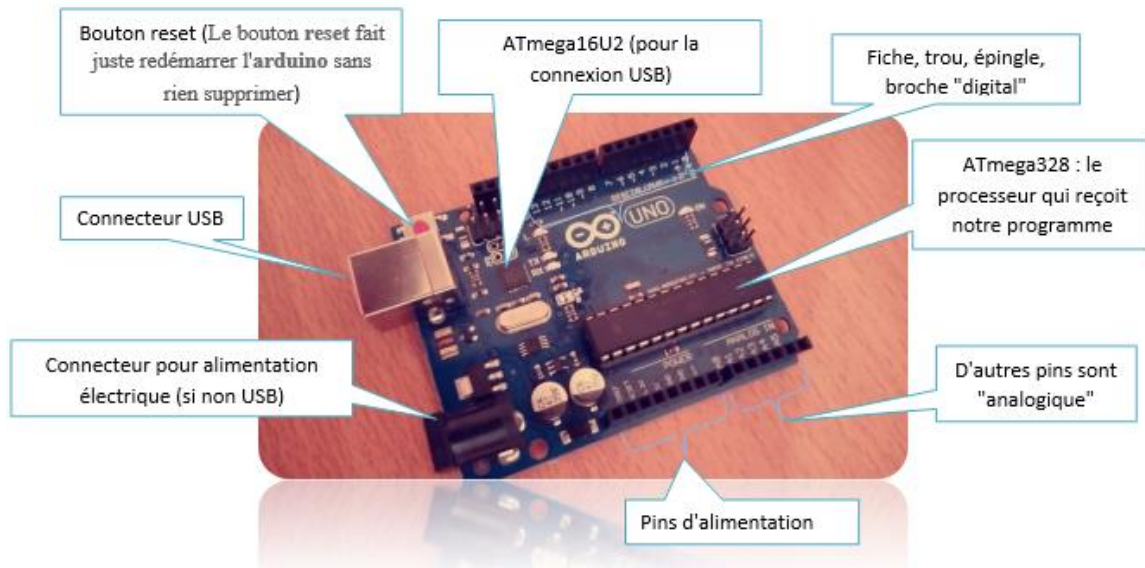


Figure II.5 -Carte Arduino.

## II.7 -Partie programme

La carte Arduino peut être programmée par Arduino IDE, l'environnement de programmation open-source pour Arduino téléchargeable gratuitement (pour Mac OS X, Windows, et Linux ) [36].

### II.7.1 -L'environnement de 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 logiciel est appelé IDE Arduino [26].

### II.7.2 -Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, c'est une interface souple et simple et exécutable sur n'importe quel système d'exploitation. Arduino basé sur la programmation en C [26].

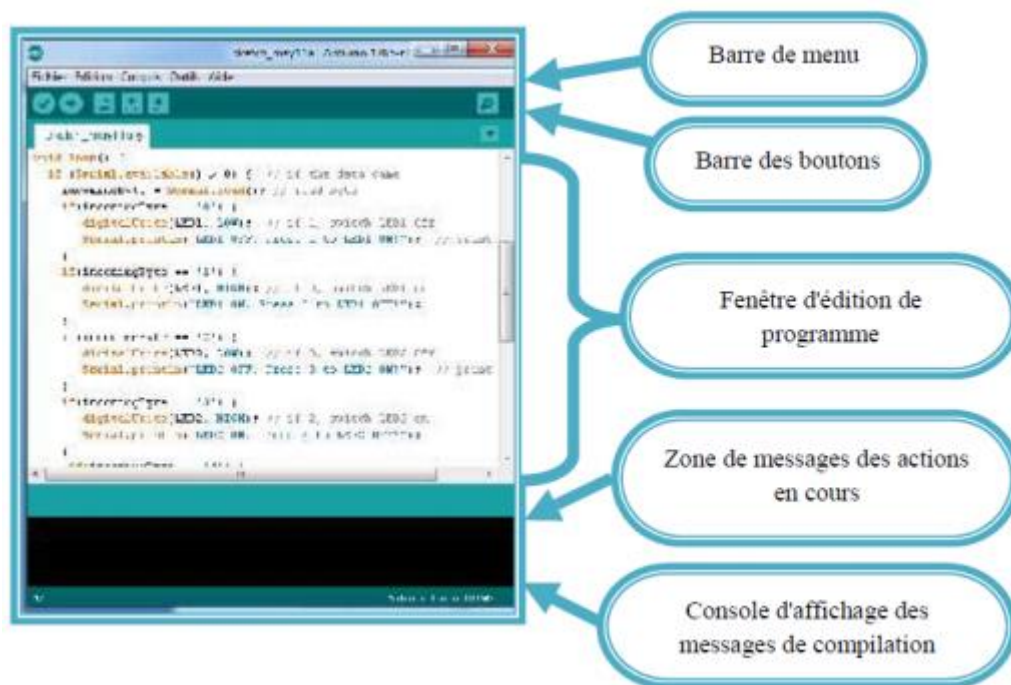


Figure II.6 -Interface IDE Arduino.

### II.7.3 –Transfert du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (dans notre Arduino UNO) et le numéro de port USB (dans notre cas COM 3) comme représenté à titre d'exemple sur la figure suivante (II.8).

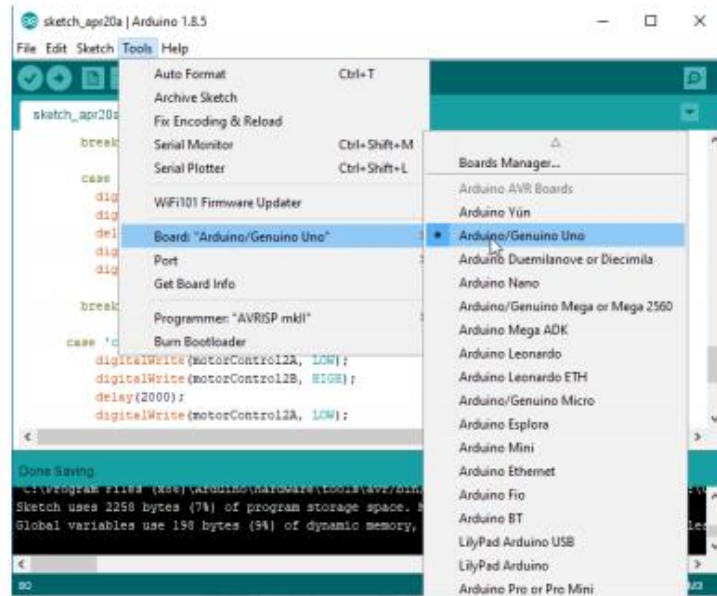


Figure II.7 -Paramétrage de la carte étape 1.

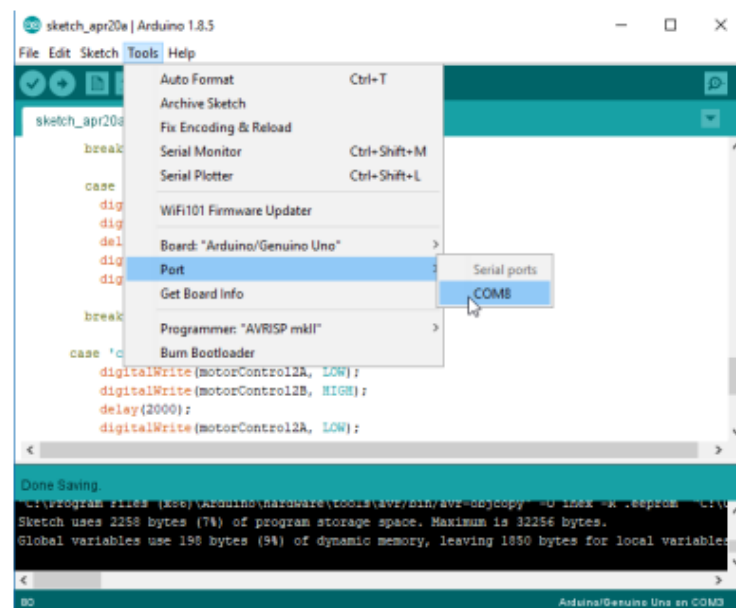


Figure II.8 -Paramétrage de la carte étape 2.



### II.7.4 -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.

### II.7.5 -Commentaires

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme. Par ex :

- 1 /\* programme de commande DC moteur avec Smartphone via "GSM"-----  
---
- 2 \*/-----  
----

### II.7.6 -Définition des variables

Pour notre montage, on va utiliser une sortie numérique de la carte qui est, par exemple la 7<sup>ème</sup> sortie numérique, cette variable doit être définie et nommée, ici moteur pin 7, la syntaxe est pour désigner un nombre entier est int [26]. Par ex :

- 3 int moteur = 7; // mettre le moteur au pin 7

La Configuration des entrées et des sorties se fait dans la fonction void setup (). Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques. Ici on va configurer moteur pin en sortie, pin mode (nom, état) est une des quatre fonctions relatives aux entrées / sorties numériques:

- 4 void setup() {-----
- 5 // mettre le moteur 1 comme sortie:-----

- 6 pinMode(motor 1, OUTPUT); // lorsque le pin 7 est actif le moteur

tourne-

- 7\* }-----

### II.7.7 -Programmation des interactions "void loop ()"

Dans cette boucle, on définit les opérations à effectuer dans l'ordre:

digital write (nom, état) est une autre des quatre fonctions relatives aux entrées /sorties numériques.

-delay (temps en mili-seconde) est la commande d'attente entre deux instructions.

-chaque ligne d'instruction est terminée par un point virgule.

-ne pas oublier les accolades qu'encadrent la boucle.

- 8\* void loop() {-----
- 9\* digitalWrite ( moteur 1,HIGH); -----
- 10\* delay (3000);-----
- 11\* digitalWrite(moteur 1, LOW);-----
- 12 \*delay (1000);-----
- 13\* }-----

### II.7.8 -Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la 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 [37].

### II.8 -Les accessoires de la carte Arduino

La carte arduino généralement est associée aux accessoires qui simplifient les réalisations [38]. Il existe une multitude de shields et de modules conçus spécialement pour la carte arduino.

#### II.8.1 -Communication

##### II.8.1.1 -Les modules Bluetooth

Il existe plusieurs sortes de modules Bluetooth compatibles Arduino et utilisables sur un plaque d'essai. Les plus connus sont les modules HC-05 et HC-06. On les distingue par le nombre de pattes d'entrées / sorties :

- **HC-05** : 6 sorties. Ce module peut être « maître » (il peut proposer à un autre élément Bluetooth de s'appairer avec lui) ou « esclave » (il ne peut que recevoir des demandes d'appairage)..

- **HC-06** : 4 sorties. Ce module ne peut être qu'esclave.

Les deux modules peuvent être utilisés en mode COMMANDE, pour les programmer avec des « commandes AT », ou en mode DATA, pour échanger des données.

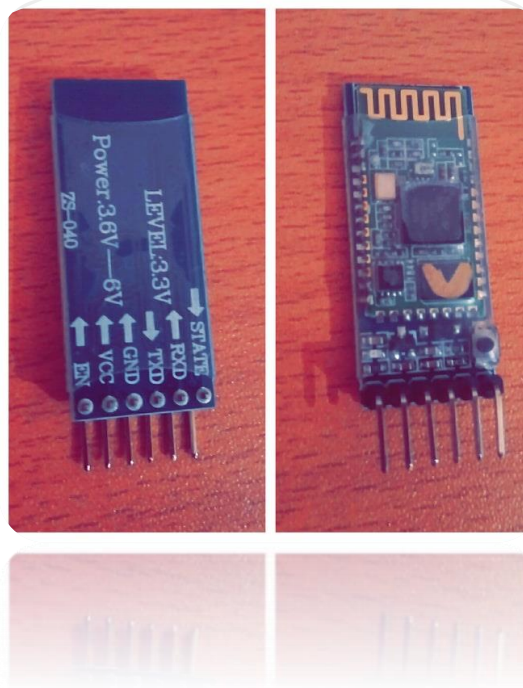


Figure II.9 -Exemple de modules bluetooth.

### II.8.1.2 -Le module shield Arduino wifi

Le bouclier Arduino Wifi permet à une carte Arduino de se connecter à Internet en utilisant la bibliothèque Wifi et de lire et d'écrire une carte SD en utilisant la bibliothèque SD « *Secure Digital* » [38].



**Figure II.10** -Module shield wifi [38].

### **II.8.2 –La commande des moteurs:**

La commande des moteurs à courant continu de faible puissance se fait à l'aide de moteur shield à base de circuits intégrés pont en H.

En général, un moteur shield est basé sur le composant L298N ou un composant similaire, 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 à courant continu ou un moteur pas-à-pas 4 fils 2 phases [40]. Il est conçu pour supporter des tensions plus élevées, des courants maximum de 2 A, tout en proposant une commande logique TTL (basse tension, courant faible, 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.

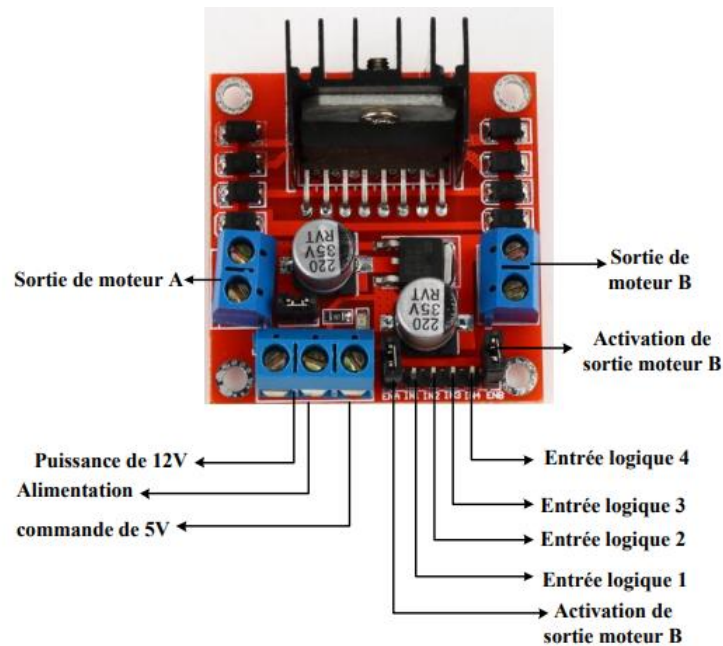


Figure II.11 -DriverL298N.

### II.8.3 -Les relais

C'est un composant qui possède une bobine (électro-aimant) qui lorsqu'elle est parcourue par un courant électrique agit sur un ou plusieurs contacts. Le relais est une solution à la commande en puissance. Il assure en outre une isolation galvanique en mettant en œuvre un mouvement mécanique [27].

### II.9 -Conclusion

Dans ce chapitre, nous avons donné un aperçu sur la carte Arduino, le cœur de notre système et son environnement de programmation.

Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino, (la partie matérielle et la partie programmation) avec plus de détails.

*Chapitre 03 : Partie conception et  
réalisation.*

### III.1 Introduction

Ce chapitre est consacré à la description du système réalisé, sa conception, sa réalisation et son fonctionnement.

Le robot mobile est un moyen excellent pour inspecter et surveiller les pipelines pour la détection des fissures et les mauvaises soudures par la détection des fuites du gaz. Pour bien atteindre les objectifs, nous avons utilisé un robot mobile muni d'un détecteur de gaz qui va se déplacer le long des pipelines et détecte la présence du gaz qui se dégage à travers des fissures.

### III.2 -Schéma bloc du système réalisé

Le schéma ci-dessous représente le schéma bloc du robot réalisé. Le robot est centré sur une carte Arduino avec un capteur de gaz, un GSM shield, un module GPS, un moteur shield, une batterie, des panneaux solaires et un circuit de charge de la batterie.



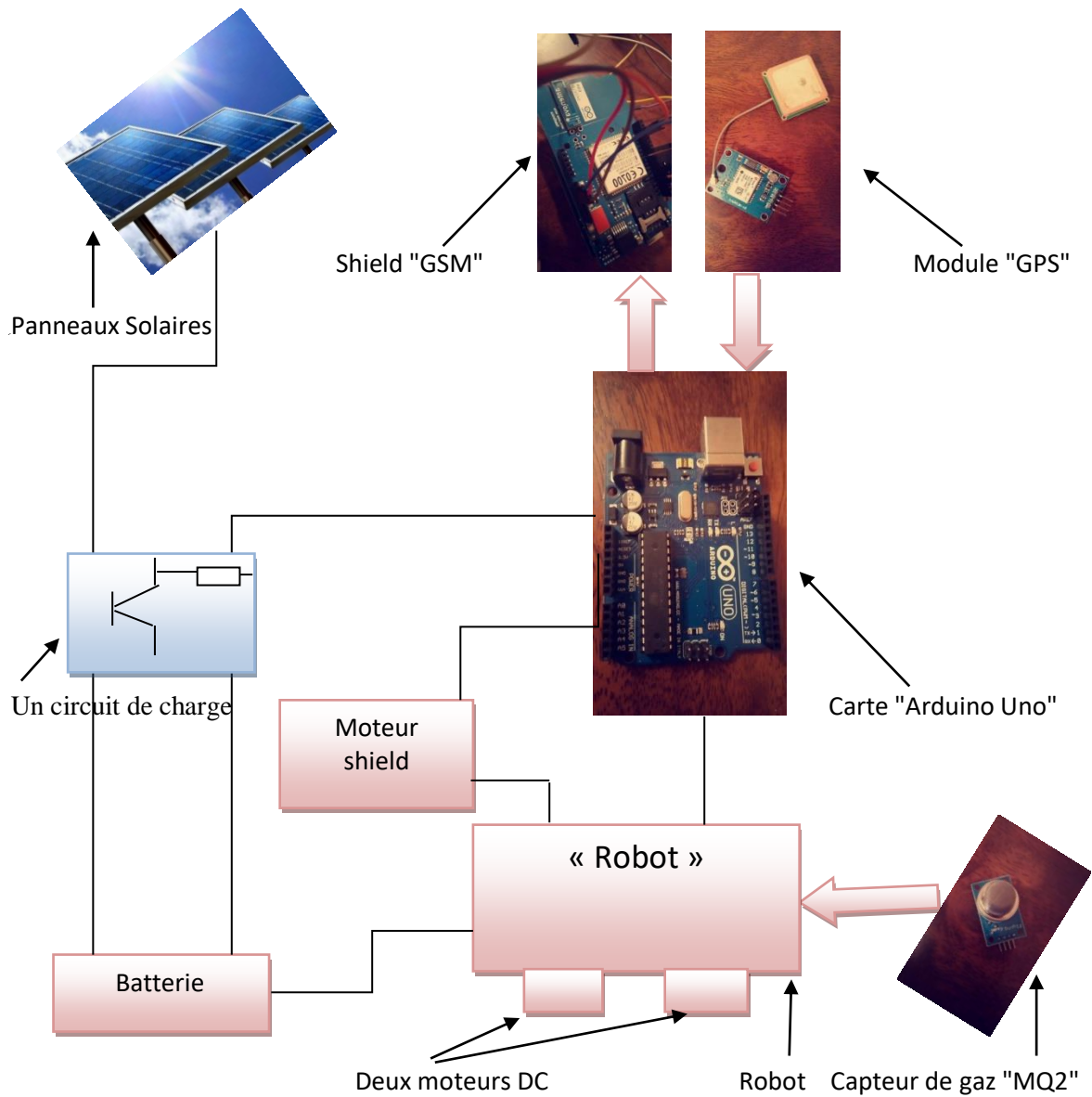


Figure III.1 -Schéma bloc.

### III.3- Les différentes étapes de la réalisation

Notre réalisation pratique a été faite en trois parties :

- La réalisation et test de chaque partie séparément.
- Réalisation du système complet.
- Test du système.

### **III.4- Les différents modules du système**

#### **III.4.1 GSM Shield**

Cette platine GSM/GPRS permet d'échanger des données en utilisant les réseaux mobiles tout en gardant une consommation électrique faible. Elle est basée sur le SIM900 Quadri-bande GSM / GPRS. Elle est contrôlée via des commandes AT (GSM 07.07, 07.05 et SIMCOM améliorée), et entièrement compatible avec Arduino Uno et Mega2560.



**Figure III.2** -Le GSM shield.

Le GSM Shield, développé en partenariat avec (Telefónica Digital), permet d'utiliser une connexion téléphonique mobile pour réaliser de l'échange de données.

Avec le GSM Shield, il est possible de relier la carte Arduino à Internet en utilisant une connexion GPRS, et envoyer et recevoir des appels vocaux ou encore des SMS [32].

III.4.1.a -Caractéristiques électriques de GSM Shield

Paramètre :	Min	Typique	Max	Unité
Tension d'alimentation	4.5	5	5.5	V
Consommation de courant (impulsion)	-	-	2000	mA
Consommation de courant (Continue)			500	mA
Vitesse de transmission	-	9600	-	bqs

Tab III.1 -Caractéristiques électriques de GSM Shield.

III.4.1.b -Brochage de la carte GSM shiel

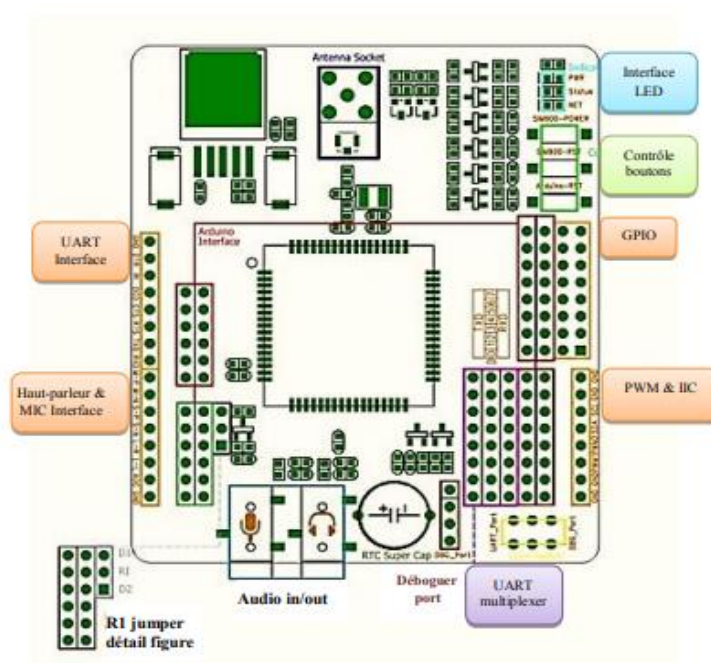


Figure III.3 -Brochage du GSM shield.

### **III.5 - Le capteur de gaz MQ2**

Module basé sur le capteur de gaz MQ2 permettant de détecter le GPL, le butane, le propane, le méthane, l'alcool, l'hydrogène et la fumée. Il est caractérisé par sa haute sensibilité et son temps de réponse rapide. La sensibilité est réglable par potentiomètre.

Ce module se raccorde sur une entrée analogique d'une carte compatible Arduino, comme on peut brancher sa sortie digitale directement à une entrée/sortie digitale de la carte Arduino [33].



**Figure III.4** –Module capteur de gaz MQ2.

#### **III.5.1 -Caractéristiques électriques de capteur MQ2**

- Entrée 5V.
- Sortie analogique
- Sortie digitale TTL
- Puissance : 800mw en chauffe.

- Détecte : la fumée, le propane, le butane, le méthane, l'alcool, l'hydrogène [33].

### III.5.2 -Branchement du capteur MQ2

Le branchement du module avec la carte Arduino est représenté sur la figure suivante:

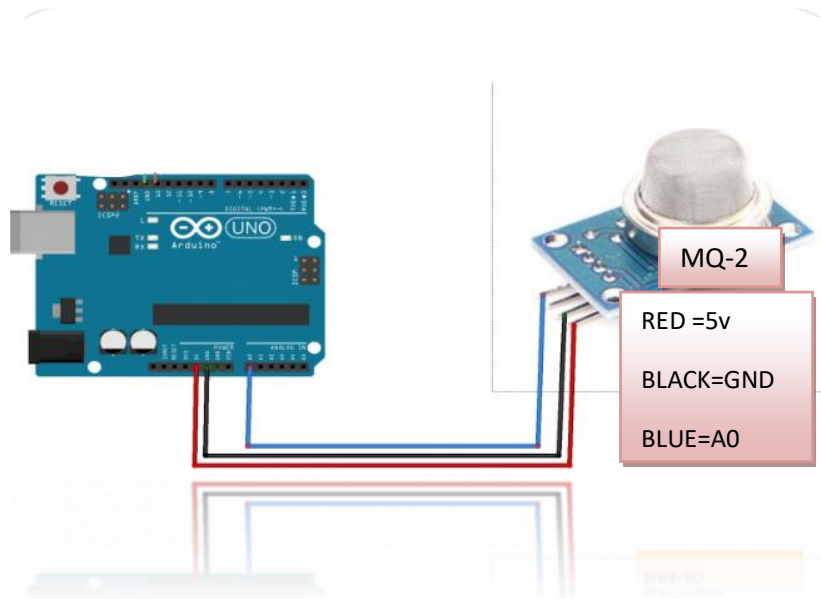
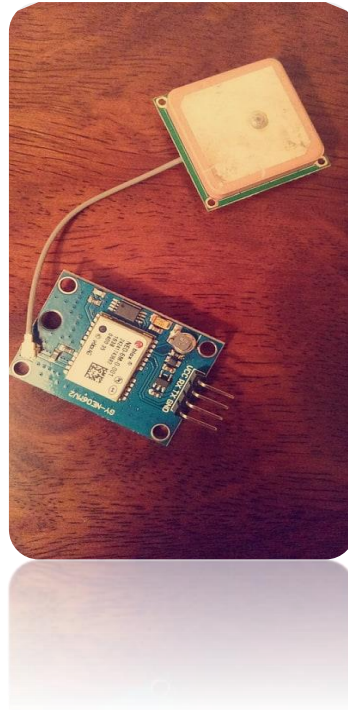


Figure III.5 -Branchement de MQ2 avec l'Arduino.

### III.6 -Le GPS "*Global Positioning System*"

GPS est l'abréviation du *Global Positioning System*, le système de géolocalisation par satellite. Le réseau GPS se compose de 24 satellites (plus 4 satellites en réserve) actuellement en fonctionnement, développé par l'armée américaine, est mis à disposition des civils. Il permet de déterminer les coordonnées géographiques de n'importe quel point situé à la surface du globe. Sa précision peut atteindre 1 mètre. Le GPS s'utilise en association avec une carte pour se repérer et se positionner: randonnées, voile, trek,... [34].



**Figure III.6** -Module GPS.

### **III.6.1-Fonctionnement du GPS**

Le principe de fonctionnement du GPS repose sur la mesure de la distance d'un récepteur par rapport à plusieurs satellites (les satellites sont répartis de telle manière que 4 à 8 d'entre eux soient toujours visibles). Chaque satellite émet un signal, capté sur terre par le récepteur, permettant ainsi de mesurer très précisément la distance séparant l'émetteur du récepteur grâce au temps de parcours.

Avec la réception des signaux de quatre satellites (trois pour obtenir le point d'intersection des trois sphères, un quatrième pour la synchronisation du temps), le récepteur mobile est capable de calculer sa position géographique par triangulation.

### **III.6.2 -Précision du GPS**

Dégradée volontairement par l'armée américaine jusqu'en 2000, la précision du GPS est aujourd'hui de l'ordre du mètre, mais dépend du matériel utilisé et du nombre de satellites en visibilité.

### **III.6.3 - Récepteur GPS**

L'association d'un récepteur GPS et d'un logiciel de cartographie permet d'obtenir un système de guidage routier efficace (affichage d'une carte avec les directions et guidage audio par synthèse vocale), développé sous différentes formes système embarqué en voiture, boîtier autonome avec récepteur intégré, assistant personnel ou smart phone associé à un récepteur GPS [34].

### **III.6.4 - Caractéristiques électriques**

- De type GY-NEO6MV2
- Alimentation 3V-5V
- Avec Flight Control EEPROM MWC
- Antenne large céramique, fort signal, 25 x 25 mm
- Paramètres de configuration sauvegardés en EEPROM
- Avec batterie de backup de données
- LED témoin de signal
- Module miniaturisé 25 x 35 mm
- Trou de fixation de 3mm
- Baud rate par défaut 9600
- Compatible avec de nombreux modules de contrôle de vol [35].

### **III.6.5 - Branchement du GPS**

Le branchement est représenté sur la figure suivante :

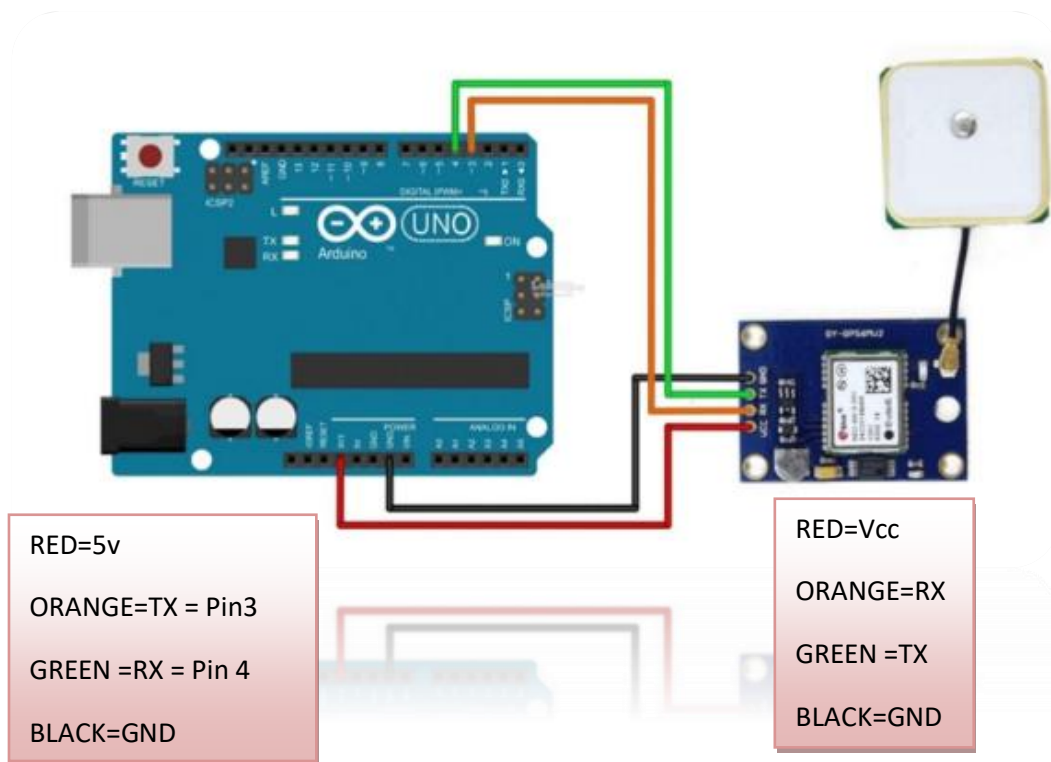


Figure III.7 - Branchement de GPS à l'Arduino.

### III.7 -La carte de commande des moteurs

La carte Arduino UNO ne peut pas gérer directement des moteurs à courant continu. Il faut ajouter un moteur shield.

On va utiliser un moteur shield L298N (chapitre II, page 37).

### III.8 -Partie du programme IDE

Dans cette partie, nous allons présenter des programmes pour chaque composant.

#### III.8.1 -Programme "Send SMS"

```
#include <GSM.h>  
  
#define PINNUMBER ""  
  
// initialize the library instance
```



```
GSM gsmAccess;

GSM_SMS sms;

void setup() {

  // initialize serial communications and wait for port to open:

  Serial.begin(9600);

  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

  Serial.println("SMS Messages Sender");

  // connection state

  boolean notConnected = true;

  // Start GSM shield

  // If your SIM has PIN, pass it as a parameter of begin() in quotes

  while (notConnected) {

    if (gsmAccess.begin(PINNUMBER) == GSM_READY) {

      notConnected = false;

    } else {

      Serial.println("Not connected");

      delay(1000);

    }

  }

  Serial.println("GSM initialized");

}

void loop() {

  Serial.print("Enter a mobile number: ");

  char remoteNum[20]; // telephone number to send sms
```

```
readSerial(remoteNum);  
  
Serial.println(remoteNum);  
  
// sms text  
  
Serial.print("Now, enter SMS content: ");  
  
char txtMsg[200];  
  
readSerial(txtMsg);  
  
Serial.println("SENDING");  
  
Serial.println();  
  
Serial.println("Message:");  
  
Serial.println(txtMsg);  
  
// send the message  
  
sms.beginSMS(remoteNum);  
  
sms.print(txtMsg);  
  
sms.endSMS();  
  
Serial.println("\nCOMPLETE!\n");  
  
}
```

### **III.8.2 -Programme " Receive SMS"**

```
#include <GSM.h>  
  
// PIN Number for the SIM  
  
#define PINNUMBER ""  
  
// initialize the library instances  
  
GSM gsmAccess;  
  
GSM_SMS sms;
```

```
// Array to hold the number a SMS is retrieved from

char senderNumber[20];

void setup() {

  // initialize serial communications and wait for port to open:

  Serial.begin(9600);

  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

  Serial.println("SMS Messages Receiver");

  // connection state

  boolean notConnected = true;

  // Start GSM connection

  while (notConnected) {

    if (gsmAccess.begin(PINNUMBER) == GSM_READY) {

      notConnected = false;

    } else {

      Serial.println("Not connected");

      delay(1000);

    }

  }

}
```

```
}

Serial.println("GSM initialized");

Serial.println("Waiting for messages");

}

void loop() {

  char c;

  // If there are any SMSs available()

  if (sms.available()) {

    Serial.println("Message received from:");

    // Get remote number

    sms.remoteNumber(senderNumber, 20);

    Serial.println(senderNumber);

    // An example of message disposal

    // Any messages starting with # should be discarded

    if (sms.peek() == '#') {

      Serial.println("Discarded SMS");

      sms.flush();

    }

    // Read message bytes and print them
```

```
while (c = sms.read()) {  
  
    Serial.print(c);  
  
}  
  
Serial.println("\nEND OF MESSAGE");  
  
// Delete message from modem memory  
  
sms.flush();  
  
Serial.println("MESSAGE DELETED");  
  
}  
  
delay(1000);  
  
}
```

### **III.8.3 -Programme concernant le fonctionnement du "GPS"**

```
#include <GSM.h>  
  
//#define PINNUMBER ""  
  
// initialize the library instance  
GSM gsmAccess;  
  
GSM_SMS sms;  
  
#include "TinyGPS++.h"  
  
#include <SoftwareSerial.h>  
  
//SoftwareSerial serial_connection(4,5); //RX=pin 5 ; TX=pin 4  
  
TinyGPSPlus gps;  
  
//char data;
```

```
double latitude;
double longitude;
double alt; //altitude
double vitesse;
unsigned long nbre_sat;
char senderNumber[20];
char remoteNumber[20]= "0*****";
//char message;
String message;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    //serial_connection.begin(9600);
    Serial.println("GPS start");
}
void loop() {
    // put your main code here, to run repeatedly:
    while(serial_connection.available())
    {
        gps.encode(serial_connection.read());
    }
    if(gps.location.isUpdated())
    {
        latitude = gps.location.lat();
        longitude = gps.location.lng();
        alt = gps.altitude.meters();
    }
}
```

```
vitesse = gps.speed.kmph();
nbre_sat = gps.satellites.value();
String message ;
message="gaz detected at" ;
message.concat(latitude);
message= message + "alt :";
message.concat(alt);
message = message +"vitess" ;
message.concat(vitesse);
message = message +"nbre sat";
message.concat(nbre_sat);

Serial.print("Satellite count:");
// Serial.println(gps.satellites.value());
Serial.println(nbre_sat);
Serial.print("Latitude:");
// Serial.println(gps.location.lat(), 6);
Serial.println(latitude);
Serial.print("longitude:");
// Serial.println(gps.location.lng(), 6);
Serial.println( longitude);
Serial.print("Speed MPH:");
// Serial.println(gps.speed.mph());
Serial.println( vitesse);
Serial.print("Altitude Feet:");
```

```
// Serial.println(gps.altitude.feet());
    Serial.println( alt);
Serial.print("Date :");
Serial.println(gps.date.value()); // Raw date in DDMMYY format (u32)
Serial.print("Time :");
Serial.println(gps.time.value()); // Raw time in HHMMSSCC format (u32)
Serial.println(gps.time.centisecond()); // 100ths of a second (0-99) (u8)
Serial.println("");
delay(2000);
}
sendSMS();
}
void sendSMS(){
Serial.print("Message to mobile number: ");
Serial.println(remoteNumber);

// sms text
Serial.println("SENDING");
Serial.println();
Serial.println("Message:");
Serial.println(message);
// send the message
sms.beginSMS(remoteNumber);
sms.print(message);
sms.endSMS();
Serial.println("\nCOMPLETE!\n");
```



}

**III.8.4 -Programme de test de détecteur de gaz "MQ2"**

```
int analog_pin=A0;

int temp;

void setup() {
  Serial.begin(9600)
}

void loop() {
  // put your main code here, to run repeatedly:

temp=analogRead(analog_pin);

if(temp>250)
{
  Serial.println(temp);
  Serial.println(" dangerous!! ");
  delay(1000);
}
}
```

**III.9 –Tests du système**

Dans les figures suivantes nous allons présenter les tests du chaque partie du système.

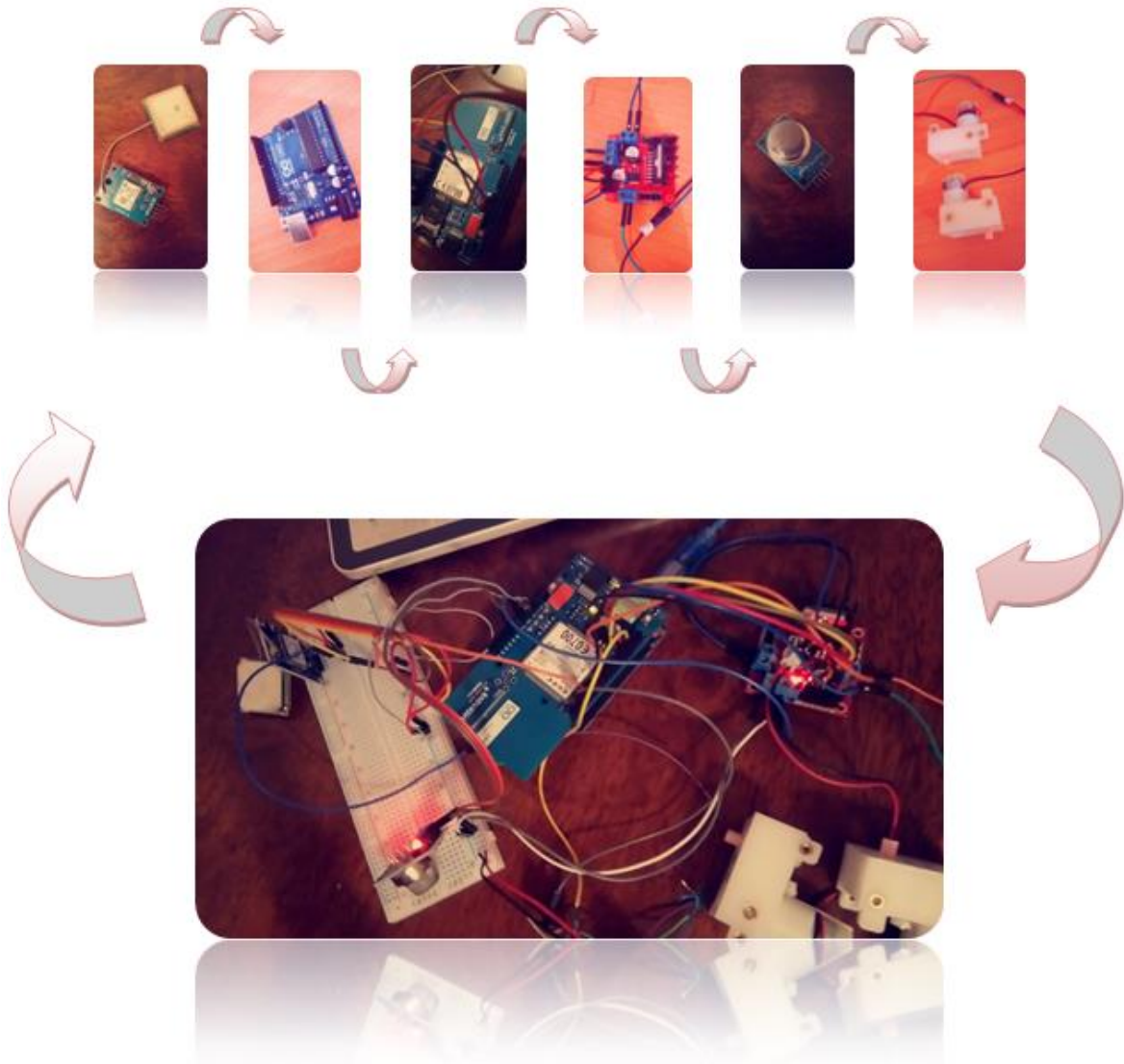


Figure III.8 -Réalisation du Système.

### III.9.1 Test de la partie moteurs

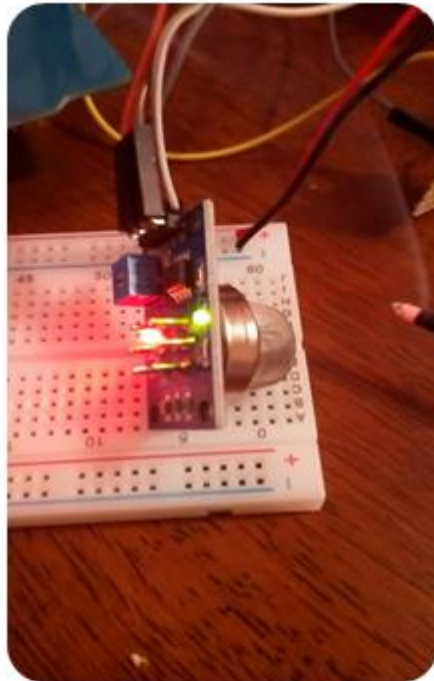
- Nous avons réalisé cet essai en premier lieu en utilisant un shield moteur et deux moteurs à DC, Figure (III.9).



Figure III.9 -Test de rotation de deux petits moteurs à DC.

### III.9.2 Test du module GSM

- Dans la figure ci-dessous, nous avons testé le capteur de gaz avec le "GSM".



Si le rebot détecte du gaz, le "GSM" envoie un "SMS" au smartphone "gaz detected".

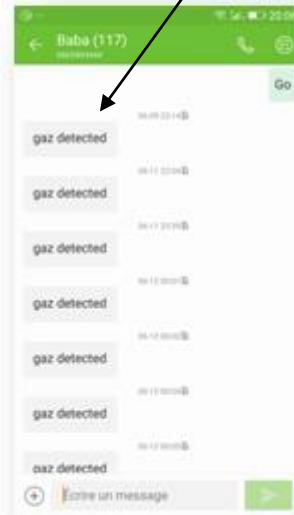
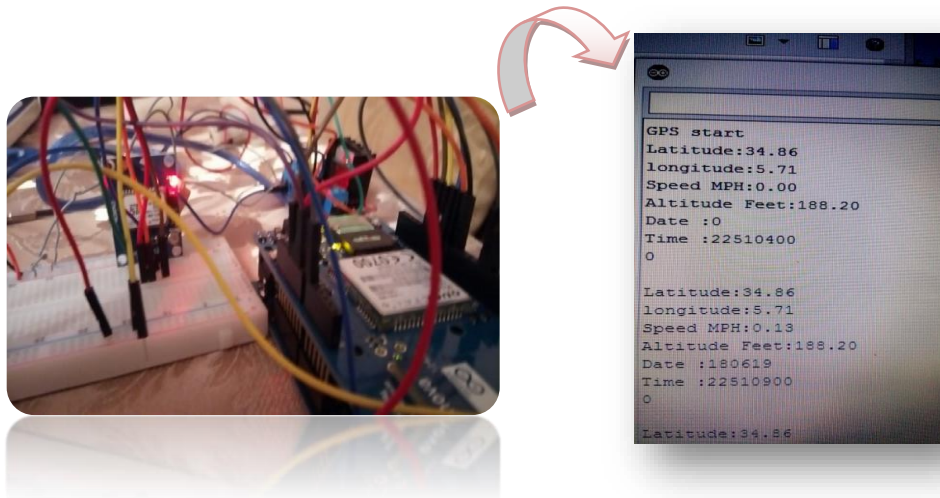


Figure III.10 -Tester un capteur de gaz MQ2.

### III.9.3 Test du module GPS

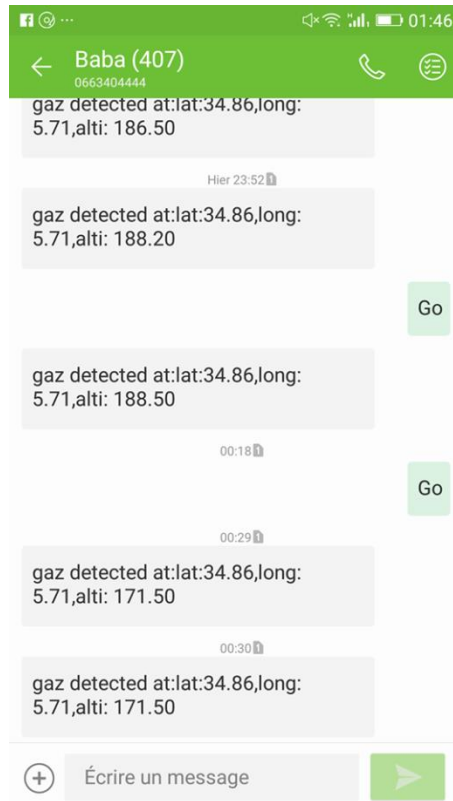
- Dans la figure ci-dessous, nous avons testé le capteur de gaz avec le GPS.



**Figure III.11** -GPS test.

#### **III.9.4 Test global.**

Nous envoyons un message texte du smartphone au GSM pour faire fonctionner les deux moteurs et guider ce robot le long de la ligne du pipeline. Le capteur de gaz détecte la présence du gaz. En cas de présence du gaz, la carte Arduino lie les données du module GPS et envoie un message texte au smartphone avec un avertissement d'existence du gaz en indiquant les coordonnées du lieu. Après signalisation de la présence du gaz, le robot continue l'inspection le long de pipeline. Si le niveau de la batterie descend au dessous d'un niveau critique, la carte Arduino donne un ordre au robot de s'arrêter et une recharge de la batterie démarre automatiquement par les panneaux solaires. Une fois la batterie chargée, l'Arduino redonne l'ordre au robot de démarrer et l'opération se répète.



**Figure III.12** –Exemple du message reçu

### III.10 -Conclusion

Le but de notre projet a été atteint. On a réalisé un système robotique excellent, ce système peut nous sauver des risques survenus du gaz des fuites et des mauvaises soudures.

On a étudié chaque partie toute seule, après on a fait l'essai de chaque partie comme indiquent les figures.

Enfin on a construit un système global intelligent qui peut détecter les fuites du gaz.

| *Conclusion générale.*

## *Conclusion générale*

La Conception et la réalisation d'un robot mobile capable d'inspecter les pipelines pour la recherche des fissures et des mauvaises soudures par la détection des fuites du gaz étaient l'objectif principal de notre projet.

Nous avons choisi de réaliser un système autonome commandé par GSM et alimenté par cellules solaires. Ce projet nous a permis d'acquérir diverses connaissances dans de multiples domaines notamment en électronique (choix de composants, réalisation de la carte,..) en télécommunication (système de GSM, protocole de connexion ...) et en informatique (langage C, ordinogramme ,.....).

Ce robot peut être utilisé pour l'exploration des pipelines pour la recherche des fissures ou de mauvaises soudures car il permet la détection du gaz dégagé par les fissures ou les mauvaises soudures.

Le robot se déplace le long du pipeline en inspectant avec son capteur la présence du gaz. Si le gaz est détecté, la carte Arduino, entre en communication avec le module GPS pour relever les coordonnées géographiques (longitude, latitude et altitude) de l'endroit et envoie par le module GSM un message SMS au Smartphone contenant un avertissement de détection de gaz avec les coordonnées géographiques du lieu suspecté.

Lorsque le niveau de la batterie atteint un niveau critique, le robot s'arrête pour recharger la batterie et un SMS est envoyé pour renseigner le superviseur des coordonnées géographiques du lieu. Une fois la batterie chargée, le robot continue son parcours en envoyant un SMS au superviseur.

Ce robot peut être utilisé pour d'autres applications par le remplacement du capteur par un capteur approprié et l'adaptation du mode de fonctionnement selon l'application.

Il peut être utilisé, par exemple, pour la détection des métaux en remplaçant le détecteur du gaz par un détecteur de métaux

Comme perspective, ce robot peut être amélioré par l'ajout d'autres capteurs et une caméra et des actionneurs et il deviendra un robot multifonction qui peut être programmé selon la mission envisagée.



## Bibliographie

- [1] : DJILALI, B. O. U. R. et MAKHLOUF, CHERIEF. Commande d'un robot mobile Khepera IV. 2017.
- [2] : FLAUS, Jean-Marie. *Cybersécurité des systèmes industriels*. ISTE Editions, 2019.
- [3] : OUDEYER, Pierre-Yves. OÙ VONT LES ROBOTS?.
- [4] : JAULIN, Luc. *La robotique mobile: cours et exercices*. ISTE editions, 2015.
- [5] : KHADRAOUI, Djamel. *La commande référencée vision pour le guidage automatique de véhicules*. 1996. Thèse de doctorat. Clermont-Ferrand 2.
- [6] : <http://jf.duchet.pagesperso-orange.fr/robotique/Pages/applications.htm>
- [7] : Jean-Louis Deneubourg Ralph Beckers, Owen E. Holland. From local actions to global tasks : Stigmergy and collective robotics. In Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, pages 181–189, Cambridge
- [8] : HAZAN, Mr A. et BOUYOUCHEF, Imene. Coordination de robots pour le transport d'objets
- [9] : Intergiciels pour systèmes multi-robots : état de l'art Stefan-Gabriel Chitic, Julien Ponge, Olivier Simonin, Université de Lyon
- [10] : L Adouane. These : Architecture des Contrôle comportementales et Réactives pour la Coopération d'un Groupe de Robots Mobiles
- [11] : Matthis Radestock , S Eisenbach. Cordination in Evolving Systems. Departments of computing Imperial College of Science, Technology and Medicine .
- [12] : Siegwart, R., Nourbakhsh, R., "Introduction to autonomous mobile robots", The MIT Press 155, 2004

[13] : Avina J, G., "Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structuré". Thèse de doctorat, Institut National Polytechnique de Toulouse, 2005.

[14] : Bayle, B., "Robotique mobile : note de cours", école Nationale Supérieure de Physique de Strasbourg, 2009

[15] : AMARA, Mohammed et DERBOUZ, Mahdjouba Asma. REALISATION D'UNE CARTE DE COMMANDE D'UNE STATION DE POMPAGE A BASE D'ARDUINO. 2018

[16] : ZIOUECHE, DJAWED et MEBREK, SALAH EDDINE. *Réalisation d'un capteur de présence piloté par Arduino*. Thèse de doctorat

[17] : <https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/742-decouverte-de-larduino/3414-presentation-darduino/#1-10774-quest-ce-que-cest>

[18] : Kossigan Roland ASSILEVI (Togo), Mamadou COULIBALY (Mali), Maurin DONNEAUD (France), Cédric DOUT RIAUX (France), Simon LAROCHE (Canada-Québec), Florian PIT T ET (Suisse), Marie-Paul UWASE (Rwanda), << LIVRE ARDUINO >> ,2011

[19] : [www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php)

[20] : KHOUKHI, Amar. *Planification multi-objectif de trajectoires pour manipulateurs robotiques par lagrangien augmenté et techniques neuro-floues*. École polytechnique, 2007.

[21] : Wisama. K, Dombre. E, 'Modélisation Identification et Commande des robots', 2ème édition, Hermes Science Publications, Paris, 1999.

[22] : BOUMERDAS, ABDELATIF et LOKMAN, NEMOUR AHMED. L'ETUDE DE LA QUALITE D'ENERGIE AL'AIDE D'UNE CARTE ARDUINO.'22'

[23] : <https://www.locoduino.org/spip.php?article69>

[24] : [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.MaterielUno](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielUno)

[25] : Jean-Noel Montagé, <<Atelier Arduino (Initiation à la mise en oeuvre matérielle et logicielle de l'Arduino )>> , Centre de Ressources Art Sensitif , novembre 2006

[26] : AHMED, BOUCHERIT et MOULOUD, MERAIMI. Etude et réalisation d'une carte de contrôle par Arduino via le système Android. 2018.

[27] : A. Grimault, J. Querard « Articl Procédé et dispositif de commutation d'un relais électromagnétique ». EP2312598 A1.

[28] : Durieu C., "Algorithmes de localisation d'un robot mobiles dans un milieu balisé par mesure de distance ou d'angle de gisement", Thèse de Doctorat, Université de Paris sud, 1989.

[29] : Nagatani, K., Choset, H., Thrun, S., "Towards exact localization without explicit localization with the generalized Voronoi graph", in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 342-348, 1998.

[30] : Thrun, S., Bucken, A., "Integrating Grid-based and topological maps for mobile robot navigation", In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 944-950, 1996.

[31] : Reignier, P., "Pilotage Réactif d'un Robot Mobile, Etude de Lien entre la Perception et l'Action", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1994.

[32] : <http://wiki.iteadstudio.com/IComSat?setlang=fr>

[33] : <https://www.gotronic.fr/art-capteur-de-gaz-mq2-sen0127-22964.htm>

[34] : PARKINSON, Bradford W., ENGE, Per, AXELRAD, Penina, *et al.*(ed.). *Global positioning system: Theory and applications, Volume II*. American Institute of Aeronautics and Astronautics, 1996.

[35] : <http://tiptopboards.com/338-module-gps-gy-neo6mv2-.html>

[36] : بو شةالة , عبد اللطيف, نصير, *et al.* Conception et Réalisation d'un Suiveur du Soleil Commandé par une Carte Arduino. 2018.

[37] : [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.DebuterInstallationWindows](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.DebuterInstallationWindows)

[38] : <https://arduino103.blogspot.com/2013/01/shield-wifi-pour-arduino.html>

[39] : Nicolas GOILAV Geoffrey LOI «Arduino » page (12-18.)

[40] : <http://pecquery.wixsite.com/arduino-passion/copie-de-le-detecteur-a-ultrasons-h>

:

