



وزارة البحث العلمي والتعليم العالي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة عبد الحميد بن باديس مستغانم
Université Abdelhamid Ibn Badis de Mostaganem
كلية العلوم والتكنولوجيا



Faculté des Sciences et de la Technologie
Département de génie des électrons

N° d'ordre : M...../GE/2019

MEMOIRE

Présenté pour obtenir le diplôme de

MASTER EN GENIE ELECTRONIQUE

Option : **Electronique des Systèmes Embarqués**

Par :

- EBATA-ATIPO Hugor
- BENTOUNSI Mohamed

Thème

**Réalisation et commande d'un Robot
détecteur d'obstacle**

Soutenu le 08 / 07 / 2019 devant le jury composé de :

Président : Mr BENOUALI.A	MAA	UMAB- Mostaganem
Examineur 1: Mr BENAOUALI.M	MAA	UMAB- Mostaganem
Examineur 2 : Mme BENCHELLAL.A	MCB	UMAB- Mostaganem
Encadreur : Mme BECHIRI.F	MCB	UMAB- Mostaganem

Année Universitaire : 2018 / 2019

Dédicaces

Je dédie ce Travail à mes parents un signe de reconnaissance pour tout ce qu'ils ont consenti, des efforts rien que pour me voir réussir, et voilà l'occasion est venue.

A ceux qui m'ont donné la vie, symbole de bonté, de fierté, de sagesse et de patience.

A ceux qui sont la source de mon inspiration et de mon courage, à qui je dois de l'amour et de la reconnaissance,

A mes parents.

A mes frères et sœurs je vous réserve une place dans mon cœur et dans mes pensées.

A toutes ma famille, mes ami(es) et mes ami(es) de la promotion et tous ceux qui ont contribué à ma réussite.

Hugor EBATA- ATIPO et Mohamed BENTOUNSI

Remerciements

Nous tenons à remercier au premier lieu le Dieu tout puissant de nous avoir accordé la force et le courage de mener ce travail à terme.

Nous tenons à adresser nos sincères remerciements à notre encadreur pour le thème, Mme F. BECHIRI de l'Université ABDELHAMID IBN BADIS pour sa disponibilité, pour sa lecture, suggestion et remarques et surtout pour sa confiance sans limite mise en nous tout au long de ce projet.

Nous vous prions de bien vouloir agréer le témoignage de notre plus vive reconnaissance et notre profond respect.

Nous présentons également nos sincères remerciements à toute l'équipe pédagogique du département génie électrique de l'université de Mostaganem, une équipe efficace qui nous a appris énormément de choses ces cinq(05) dernières années

Nous remercions particulièrement nos familles pour leurs soutiens moraux Pour ce travail, merci de nous avoir encouragés, et cru en nous.

Nous remercions également nos ami(es), nos collègues de classe, qui nous ont encouragés tout au long de ce projet et nous ont beaucoup aidés.

Enfin, nos remerciements aux membres de jury qui ont fait l'honneur d'accepter de juger ce modeste travail.

Table des matières

Dédicace	i
Remerciements	ii
Table des matières	1
Table des figures	3
Introduction générale	6
Chapitre I : Rappel sur les systèmes embarqués	7
I.1 : Introduction	8
I.2 : Définition d'un système embarqué	8
I.3 : Caractéristiques principales d'un système embarqué	8
I.4 : Les étapes de réalisation d'un système embarqué	9
I.5 : Conclusion	9
Chapitre II : Etude théorique du projet	10
II.1 : Introduction	11
II.2 : Cahier des charges du projet	11
II.3 : Schéma synoptique du projet	12
II.4 : Le matériel utilisé dans le projet	13
II.4.1 : Une carte Arduino Uno :	13
II.4.2 : un servomoteur	14
II.4.3 : un capteur ultrason	15
II.4.4 : moteur à courant continu DC.....	18
II.4.5 : le circuit intégré L298N.....	20
II.4.6: un module Bluetooth.....	21
II.4.7: Panneaux solaire.....	23

II.4.8: Régulateur de charge.....	23
II.4.9: Batterie 12v.....	24
II.5 : Conclusion	25
Chapitre III : Description du Robot	26
III.1: Définition du Robot.....	27
III.2: Schéma synoptique du Robot.....	27
III.3: Types du Robot.....	28
III.4: Application du Robot mobile.....	29
III.5: Application Industrielle.....	29
III.6: Types de marche du Robot.....	31
Chapitre IV : Réalisation pratique du Robot.....	33
IV.1 : Introduction	34
IV.2 : Schéma électronique du projet	34
IV.3 : logiciel.....	35
IV.3.1 : La conception du logiciel sous l'environnement Arduino	36
IV.3.2 : Fonctionnalités du logiciel	37
IV.3.3 : Organigramme de haut niveau du logiciel	39
IV.4 : Montage complet du robot.....	43
IV.5 : Conclusion	45
Conclusion générale et perspectives	46

Annexes	47
Annexe1 :Microcontrôleurs.....	47
Annexe 1.1 : Définition d'un microcontrôleur	47
Annexe 1.2 : Généralités	47
Annexe 1.3 : Architecture interne de base d'un système à processeur	49
Annexe 1.4 : Langages de programmation des microcontrôleurs	49
Annexe 1.5 : Quelques critères du choix d'un microcontrôleur	51
Annexe 2 : L'Arduino	52
Annexe 2.1 : Qu'est-ce que l'Arduino ?	52
Annexe 2.2 : La carte Arduino	52
Annexe 2.3 : Le logiciel Arduino	54
Bibliographie et sites internet.....	57

Table des figures

Figure I.1 : Etapes de réalisation d'un système embarqué.....	9
Figure II.1 : Schéma synoptique du projet	12
Figure II.2 : estimation du coût du projet.....	13
Figure II.3 : Un servomoteur	14
Figure II.4.1 : Emission et Réception des obstacles.....	15
Figure II.4.2 : Transformations des ondes.....	16
Figure II.4.3 : Diagramme des obstacles	17
Figure II.4.4: Capteur ultrason.....	18
Figure II.5.1 : Relations entre les Vitesses.....	18
Figure II.5.2: Moteur à courant continu DC	19
Figure II.5.3: Moteur DC3.7v et les roues	20

Figure II.6 : Le circuit intégré L298N.....	21
Figure II.7 : Un module Bluetooth.....	22
Figure II.8: Panneaux solaire.....	23
Figure II.9 : Régulateur de charge.....	24
Figure II.10 : Batterie 12v.....	24
Figure III.1 : Schéma synoptique d'un robot.....	27
Figure III. 2 Robot d'exploration.....	29
Figure III.3 :Robot militaire.....	29
Figur III.4 : véhicule Autoguide.....	30
Figure III.5 :Robot aspirateur.....	30
FigureIV.1 : simulation avec le logiciel proteus.....	34
Figure IV.2: Schéma électronique du projet	35
Figure IV.3: Partie déclaration dans le programme	36
Figure IV.4: Fonction « setup () » du programme	37
Figure IV.5: Fonction « loop () » du programme	38
Figure IV.6: Schéma bloc.....	39
Figure IV.7 : Organigramme du haut niveau du logiciel.....	42
Figure IV.8 : Montage final du robot sans les panneaux solaires	44
Figure IV.9 : Montage final du robot avec les panneaux solaires	45
Figure A.4 : La carte Arduino Uno.....	49
Figure A.5 : L'interface utilisateur du logiciel Arduino	51

Abstract

Robotiser is a term that has become more and more interesting due to its importance in all fields, precisely in the field of electronics and power electronics; on the other hand, it is an imperative need for the realization of

several of spots that are difficult for a human being, or the same, that requires routine when they are done. It is of such an advantage in terms of quality, accuracy and speed that the use of this technique ensures a complete integration to the field of work, either to carry out instructions at the requested time, or to exchange information, and all this surely passes in an inevitably flexible way.

The flagship of technological progress, robots are today present in all sectors (industry, administration, our own apartment). The various treatments they allow to achieve will lead to the improvement of our daily needs.

The major interest of this project is attributed to the search for a solution simplifying the mobility of the robot avoiding any obstacle; with a programmable circuit at a reduced price, to try to avoid any waste of electricity.

Introduction générale

La robotique permet d'aider l'homme dans les tâches difficiles, répétitives ou pénibles. De plus, elle constitue le rêve de substituer la machine à l'homme dans ces tâches.

Les facultés de perception et de raisonnement des robots progressent chaque jour actuellement et plus encore dans l'avenir, ils sont appelés à jouer un rôle de plus en plus

important dans notre vie.

La robotique comporte deux grands pôles d'intérêt: la robotique de manipulation (robotique industrielle) et la robotique mobile. Un des problèmes majeurs de la robotique mobile est la planification de mouvement. Autour de ce problème de planification de mouvement de nombreuses études ont été réalisées dans le but de développer des méthodes générales pour guider les robots. Le robot aura un capteur de distance devant lui pour éviter les obstacles, les quatre moteurs DC et les différents capteurs sont pilotés à l'aide d'une carte Arduino Uno. Le mode manuel définit un robot commandé par un Bluetooth via une application Android (App_robot) et la programmation se fera en langage Arduino qui ressemble au C++. En effet, notre projet consiste à la réalisation et commande d'un robot détecteur d'obstacle. Ce robot change de direction en présence d'un obstacle détecté par le capteur ultrason. Nous verrons tous les détails de ce projet dans les différents chapitres qui suivent. Ce robot n'étant ni plus ni moins qu'un système embarqué. Après l'introduction générale, nous ferons un rappel sur les systèmes embarqués en premier chapitre. Le second chapitre sera consacré à une étude théorique du projet et dans le troisième chapitre, nous ferons la réalisation pratique du système pour terminer avec les tests finaux et la validation et nous terminerons par une conclusion générale.

CHAPITRE I

Rappel sur les systèmes embarqués

I.1 : Introduction

Nous donnons dans ce chapitre un petit rappel sur les systèmes embarqués. Ces systèmes électroniques utilisent les composants électroniques classiques et les composants à processeur autrement dit les composants analogiques et les composants numériques. Les systèmes embarqués sont présents dans notre vie quotidienne et y jouent un rôle important. Grâce à ses systèmes, nous communiquons à distance, nous nous déplaçons plus rapidement, nous avons plus de sécurité et de confort dans nos maisons, etc....

I.2 : Définitions d'un système embarqué

Un système embarqué est un système électronique et informatique qui est spécialisé dans une tâche bien précise. C'est un système électronique et informatique particulier, conçu pour effectuer des tâches spécifiques avec des ressources limitées et des contraintes variées (consommation d'énergie, espace mémoire restreint, etc.).

Un système embarqué est aussi défini comme étant un composant électronique issu de la combinaison de matériel et de logiciel et qui interagit avec son environnement. Ces systèmes sont dit embarqués, parce qu'ils sont toujours intégrés dans des systèmes beaucoup plus larges pour y jouer un rôle bien spécifique. Notons qu'avant l'arrivée des microprocesseurs, ces systèmes étaient tout simplement appelés systèmes électroniques et ils étaient constitués de différents composants classiques reliés entre eux par des liaisons point à point. La composante logicielle n'existait pas.

I.3 : Caractéristiques principales d'un système embarqué

Les principales caractéristiques d'un système embarqué sont :

- C'est un système principalement numérique.
- Il dispose des ressources limitées surtout les mémoires.
- Il peut être autonome ou non.
- Il est constitué de logiciel et de matériel.
- Il ne possède généralement pas d'entrées/sorties standards tels qu'un port série RS232, USB, etc. ni de périphériques classiques comme un clavier d'ordinateur mais plutôt des

périphériques limitées telles que des boutons poussoirs, des claviers matricielles, des LED, des écrans LCD, etc.

- Il peut être ou non ouvert au monde extérieur pour des mises à jour ou pour communiquer.

Exemple :

- un distributeur de billet
- un robot
- une carte à puce
- les systèmes de contrôle d'un avion

I.4 : Les étapes de réalisation d'un système embarqué

La réalisation d'un système embarqué, quel que soit sa génération, obéit à un certain nombre d'étapes présentées dans l'organigramme ci-dessous :

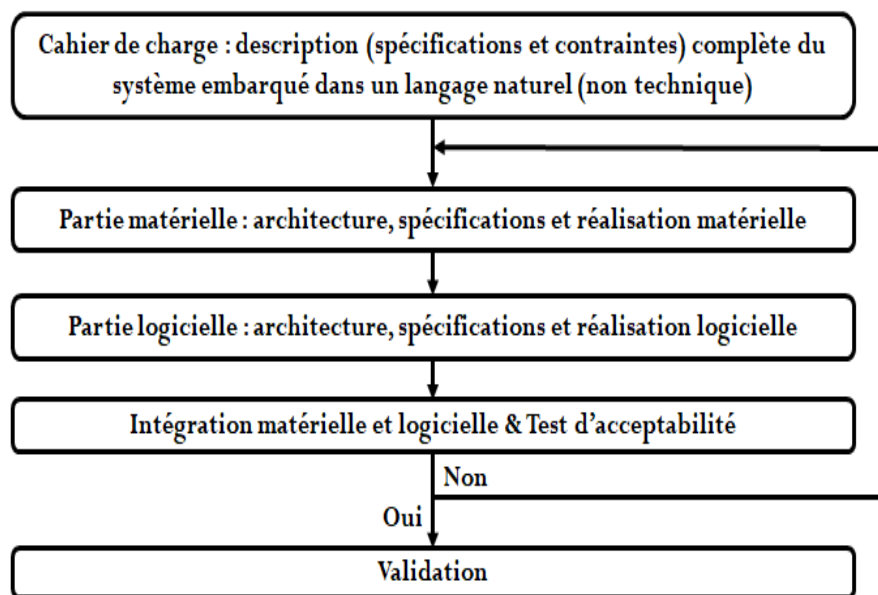


Figure I.1 : Etapes de réalisation d'un système embarqué

I.5 : Conclusion

Dans ce chapitre, nous avons décrit de façon brève les systèmes embarqués. Nous avons vu ce qu'est un système embarqué, ses caractéristiques principales et les étapes de conception d'un système embarqué.

CHAPITRE II

Etude théorique

Description des composants utilisés

II.1 : Introduction

Dans ce chapitre nous allons faire une étude théorique de notre projet la réalisation et commande d'un robot détecteur d'obstacle .Ce Robot n'est qu'un système embarqué, nous allons donc suivre naturellement les étapes de conception d'un système embarqué que nous avons déjà évoqué dans le premier chapitre. Nous allons débiter par la rédaction du cahier des charges du projet, ensuite nous verrons les spécifications matérielles et logicielles, et dans le chapitre suivant nous entamerons la réalisation pratique.

II.2 Cahier de charges du projet

Le cahier de charges d'un projet est une description complète du projet dans le langage naturel (non technique) compréhensible par toute personne ingénieur ou non ingénieur. Dans ce cahier, le client fait ressortir l'ensemble des contraintes que doit respecter son produit : l'encombrement, le temps réel, la consommation, le coût, etc....

Le cahier de charge de notre Robot détecteur d'obstacle est le suivant :Le panneau solaire capte l'énergie envoyée par le soleil vers la terre et la converti en courant électrique continu. Cette énergie sera transmise au régulateur de charge pour charger la batterie et de limiter sa charge. La batterie 12V va transmettre en suite cette énergie reçue au driver moteur pour alimenter les moteurs. Le robot étant en mouvement quant il détecte l'obstacle, il prend une autre direction automatiquement. Le servomoteur qui est un moteur à courant continue va piloter un driver qui va régler constamment sa position, on parle dans ce cas du système asservi, En utilisant le réducteur qui va augmenter le couple des moteurs. Le capteur ultrason permet de mesurer la distance, l'émetteur envoie une série d'impulsion sonore à 40khz une fréquence inaudible pour l'oreille humaine et en rencontrant l'obstacle le signal revient sous forme d'un écho au récepteur.

II.3 : Schéma synoptique du projet

Le schéma synoptique du projet est présenté comme suit :

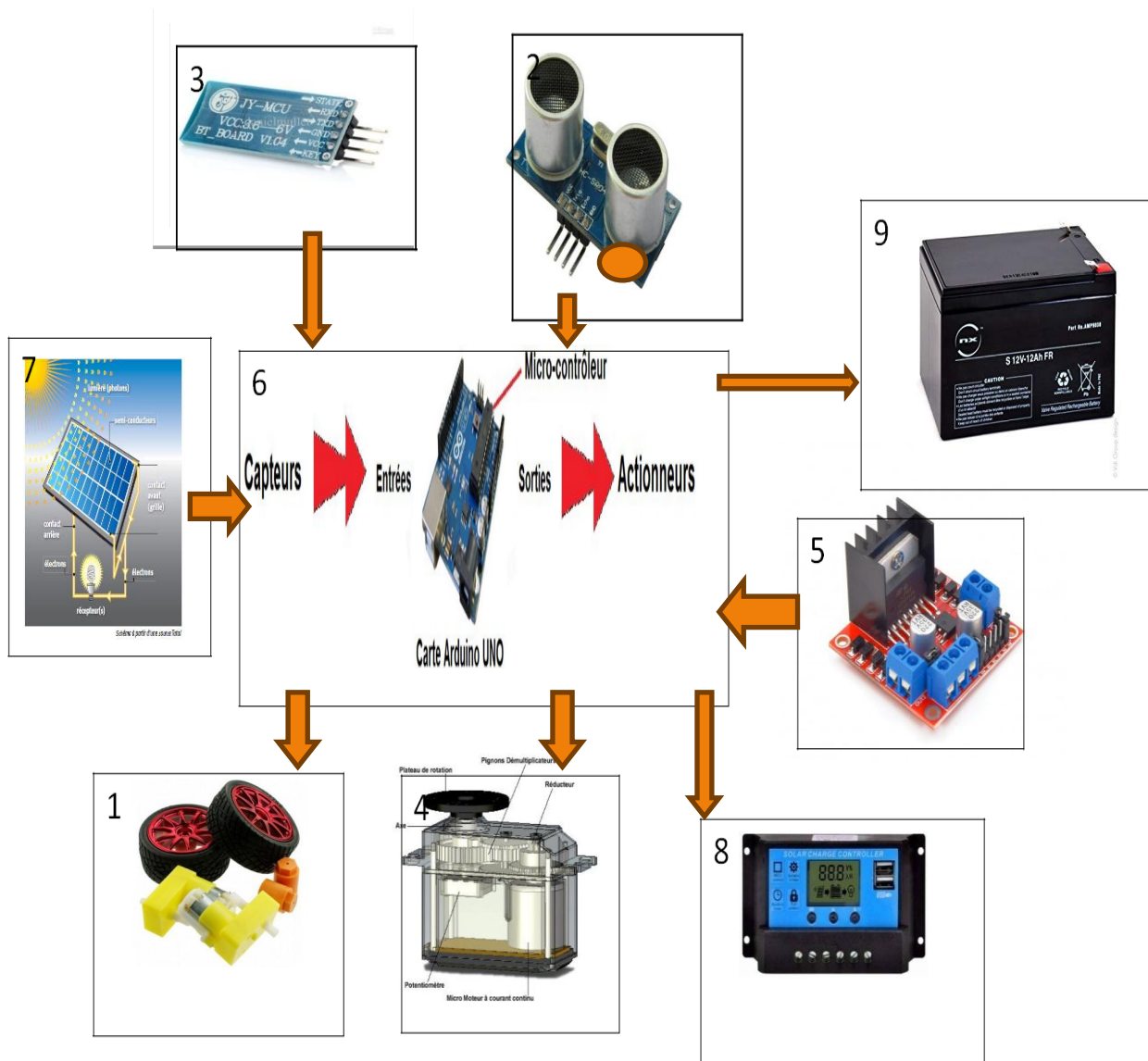


Figure II.1 : Schéma synoptique du projet

A partir de ce schéma synoptique, on peut réaliser notre projet et estimer également son coût comme le montre le tableau

9 _ Batterie 12V	1800DA
8 _ Régulateur de charge	3000DA
7 _ 2 panneaux solaires	2400DA
2 _ Capteur ultrason	500DA
1 _ Les moteurs et les roues	5200DA
Capteurs suiveur de ligne	1500DA
3 _ Bluetooth	1200DA
Plaque d'essai	450DA
6 _ Carte arduino Uno	6500DA
4 _ Servomoteur	700DA
5 _ Circuit I298N	1000DA
Support des batteries	450DA
Pile	300DA
TOTAL	29500DA

Figure II.2 : Estimation du coût du projet

II.4 : Matériel utilisé dans le projet

II.4.1 : Une carte Arduino Uno

Dans tout système embarqué, il existe au moins un composant intégrant une unité de traitement d'informations (le processeur), c'est le (ou ce sont les) cerveau(x) du système

embarqué. Il existe plusieurs types de composants utilisés pour cette fonction parmi lesquels nous avons les microcontrôleurs, les microprocesseurs, les cartes à microprocesseurs ou à microcontrôleurs dont les plus populaires sont celles à base d'Arduino. Mais, il en existe d'autres comme Raspberry Pi, Pcdino, le launchpad MSP430 etc...

Dans ce projet, nous avons utilisé l'une de ces cartes à microcontrôleur à savoir l'Arduino, plus précisément la carte Arduino Uno (plus de détails sur la carte dans « Annexe 2.2 »). Ce choix est motivé par plusieurs raisons surtout le nombre d'entrées sorties de la carte, l'espace mémoire, mais aussi par la facilité d'utilisation d'Arduino que ce soit dans le câblage ou la programmation, et les différents guides disponibles sur le site officiel de l'Arduino (www.arduino.cc) (Arduino est open source).

II.4.2 : Servomoteur

Un servomoteur, son nom vient du latin « servus » qui signifie esclave. Mais avant de s'atteler de son exploration interne, nous allons avant tout voir à quoi il sert. Les fils, qui sont au nombre de trois (nous y reviendrons)

L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal

Le boîtier qui le protège ... mais aussi de plusieurs éléments que l'on ne voit pas

Un moteur à courant continu des engrenages pour former un réducteur (en plastique ou en métal) un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent). Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu[7]. Voilà une image 3D de vue de l'extérieur et de l'intérieur d'un servomoteur. Le servomoteur a besoin de trois fils de connexion pour fonctionner. Deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande : rouge: pour l'alimentation positive (4.5V à 6V en général) noire marron: pour la masse (0V) orange, jaune, blanc, ... : entrée du signal de commande

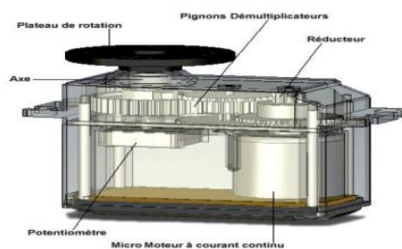


Figure II.3 : Servomoteur

II.4.3 : Capteur ultrason

Les capteurs ultrasonores utilisent des ondes de fréquence non perceptible par l'oreille humaine, généralement dans la fourchette 20Khz-200khz. De la même manière que les télémètres laser, ils sont basés sur le principe de la mesure du temps aller-retour lors de la réflexion sur un obstacle. C'est la méthode employée par certains animaux pour percevoir leur environnement, comme les chauves-sous ou les dauphins l'écholocalisation.

Un avantage de ces capteurs est que contrairement aux télémètres, l'onde qu'ils émettent n'étant pas focalisée, ils perçoivent beaucoup plus facilement des éléments filiformes comme des pieds de chaises ou des grillages. Par contre leur portée est faible, et ils sont moins adaptés aux milieux de propagations non [8] isotropes comme l'air.

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme des vêtements puissent être difficiles à détecter.

1-Principes de fonctionnement des ultrasons

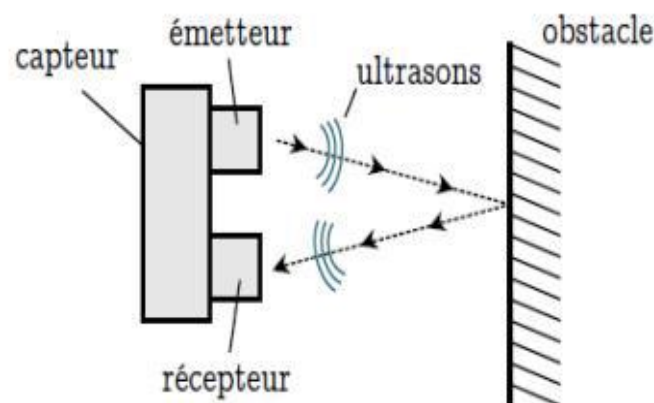


Figure II.4.1 : Emission et réception des obstacles

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. Les ultrasons sont des ondes infra-acoustique, qui oscillent à des fréquences supérieures au seuil acoustique. Ceci est important à plusieurs niveaux :

- Le premier est que cette fréquence n'interfère pas avec le bruit audible produit par la majorité des corps physiques de notre environnement.

- La deuxième est que ces fréquences sont très précises, rapides et peuvent être personnalisées. Les capteurs fournis ont souvent la forme d'une paire car il y a deux parties essentielles :

• L'émetteur

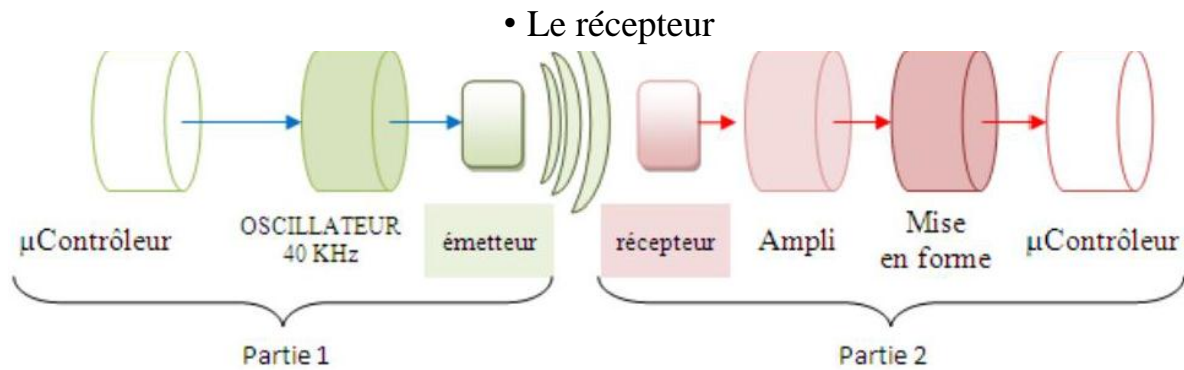


figure II.4.2 : Transformation d'onde

L'émetteur émet un son à une fréquence définie et le récepteur collecte le son répercuté par les obstacles. La distance aux objets est calculée par le temps mis par le son pour revenir au récepteur.

2-L'émission:

La fréquence du signal de l'émetteur couramment utilisé est égale à 40 KHz, sa production nécessite l'un des TIMER du microcontrôleur. Cette fréquence n'est efficace que si l'émetteur ultrason est alimenté par une tension au moins égale à 4.5 V

3-La réception:

- Le capteur transforme les variations de pression acoustique en signal électrique.
- Ce signal électrique analogique de quelques millivolts est amplifié.
- Le signal amplifié est mis en forme pour pouvoir être traité par microcontrôleur. Le signal fourni par le capteur a une amplitude de l'ordre de 10 mV, celle-ci diminue avec la distance.

Son exploitation au niveau du microcontrôleur nécessite une amplitude de quelques volts qui varie de 0 à 5 V donc nécessite son amplification. Nous avons choisi d'utiliser le module le plus disponible sur le marché : **HC-SR04**

4- Caractéristiques de HC-SR04:

- Dimensions : 45 mm x 20 mm x 15 mm
- Plage de mesure : 2 cm à 400 cm
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : 15 °
- Largeur d'impulsion sur l'entrée de déclenchement : 10 µs (Trigger Input Pulse width)

5- Facteurs d'influence :

Le capteur ultrasonique comporte certains inconvénients :

- Il est très vite inopérant lorsqu'il y a présence **d'échos parasites**, en présence de poussière ou encore lorsque le niveau mesuré se trouve dans un endroit trop étroit.
- Aucun fonctionnement possible dans le vide.
- Il y a une zone que l'on appelle **zone morte** et qui correspond à la distance minimum que doit avoir le niveau à détecter par rapport au capteur pour que celui-ci fonctionne correctement

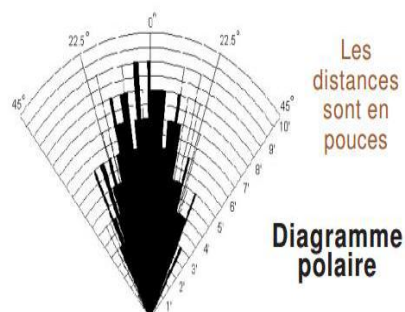


Figure II.4.3 : diagramme des obstacles

6- Spécifications et limites:

Paramètre	Min	Type	Max	Unité
Tension d'alimentation	4.5	5.0	5.5	V
Courant de repos	1.5	2.0	2.5	mA
Courant de fonctionnement	10	15	20	mA
Fréquence des ultrasons	-	40	-	kHz



*Figure II.4.4 : capteur ultrason **HC-SR04***

II.4.4 : Moteurs DC

Les moteurs DC avec des aimants permanents sont très courant dans les différentes applications, avec de petits dimensions, lorsqu'on recherche beaucoup de puissance a bas prix. A cause de leur vitesse assez importante, ils sont utilisés dans plusieurs sortes de transmissions (pour obtenir beaucoup de couple avec peu de vitesse).

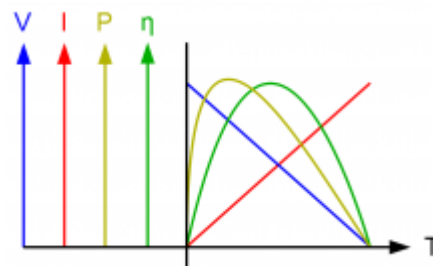


Figure II.5.1 : Relation entre les moteurs

Le graphe parfait de la relation entre la vitesse (V), le courant (I), la puissance (P), le rendement (η) et le couple (T) d'un moteur DC

Les moteurs DC à aimant permanent sont d'une fabrication assez simple et les commander est assez élémentaire. Bien que les contrôler soit assez simple, leur vitesse n'est pas déterminée précisément par le signal de contrôle parce qu'il dépend de nombreux autres facteurs, dont le couple appliqué sur l'arbre et le courant d'alimentation. La relation entre le couple et la vitesse d'un moteur DC parfait est linéaire, ce qui signifie que : plus grand est l'effort sur l'arbre, plus petite est la vitesse de l'arbre et plus grand est le courant dans la bobine.

Les moteurs DC utilisent la tension DC et ne nécessitent pas de contrôle électronique supplémentaire puisque toutes les communications nécessaires sont faites dans le moteur. Lorsque le moteur fonctionne, deux broches statiques glissent dans le commutateur de rotation et garde la tension dans la bobine. La direction de la rotation du moteur est déterminée par la polarité du courant appliqué. Si le moteur ne doit tourner que dans une seule

direction, alors le courant peut passer à travers un relai où tout autre connexion simple. Si le moteur doit tourner dans plusieurs directions, on utilisera un circuit électronique appelé H-bridge (ou “demi-pont”).

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

1-Les caractéristiques techniques des moteurs

- 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

2-Varier la vitesse(PWM)

Le réglage de la vitesse de rotation du moteur à courant continu possède une relation directe entre sa tension d'alimentation et sa vitesse de rotation. En effet, plus la tension à ses bornes est élevée et plus son axe tournera rapidement (dans la limite de ses caractéristiques évidemment). Cependant le microcontrôleur d'Arduino n'est capable de produire que des tensions de 0 ou 5V. En revanche, il peut “simuler” des tensions variables comprises entre 0 et 5V. Nous obtenons une tension moyenne (comprise entre 0 et 5V) en sortie de la carte Arduino.



Figure II.5.2 : moteur DC 3.7V et les roues

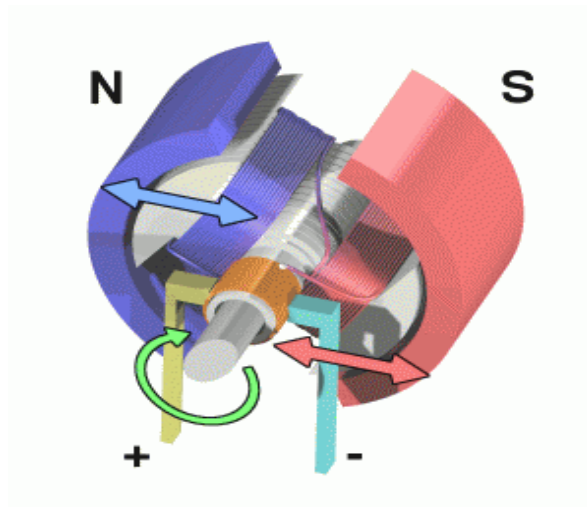


Figure II.5.3 : moteur à courant continu et sa structure interne

II.4.5 : Composant L298N

Ce break out board est un Double pont-H destiné au contrôle de moteur continu(H-Bridge Motors Drive).il est base sur le composant l298N qui est un double pont –H conçu spécifiquement pour ce cas d'utilisation.

C'est un module extrêmement utile pour contrôler de robot et ensembles mécanisés. Il peut contrôler deux moteurs courant continu ou un moteur pas –à-pas quatre (4) fils et deux (2) phases.

Il est conçu pour supporter les tensions plus élevées, des courants importants en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleurs). 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 sorties en puissance sont déjà protégées par des diodes anti-retour.il s'agit d'un module prêt a l'emploi. Caractéristiques Leger, petit des capacités hors-pair pour contrôler moteurs diodes de protection un dissipateur (pour dissiper la chaleur en cas de forte charge) un sélecteur pour sélectionner la source d'alimentation quatre sélecteurs pour les résistances pull up sortie deux moteurs continu et un moteur pas-à-pas (4 bobines, deux phases) indicateurs LED pour sens de rotation moteur indicateur LED pour alimentation 5V 4 trous de fixation standard.[11]

1- Spécification du composant

Le L298N alimente une charge : de +6V à 35V courant max (en pointe) : 2A tension de commande logique V_{ss} : de +5 à 7V (alimentation interne de +5V) courant de commande logique de 0 à 36mA tensions pour contrôler du sens :

Low-0.3~1.5V high :2.3V~V_{ss} tensions pour contrôle "Enable" :Low-0.3~1.5V, high :2.3V~V_{ss} puissance max :25W(température 75) température de fonctionnement : de -25 à +130 dimensions :60mmx54mm poids :48 idées d'applications pilotages d moteurs continu(eg :voiture téléguidée ,montage divers à base de moteurs) pilotage de moteur pas -à-pas4 fils deux phases précautions s'assurer que le pole positif soit raccordé à WMS et le pole négatif à GND la tension d'entrée (étage de puissance) ne doit pas excéder 35V détails techniques.

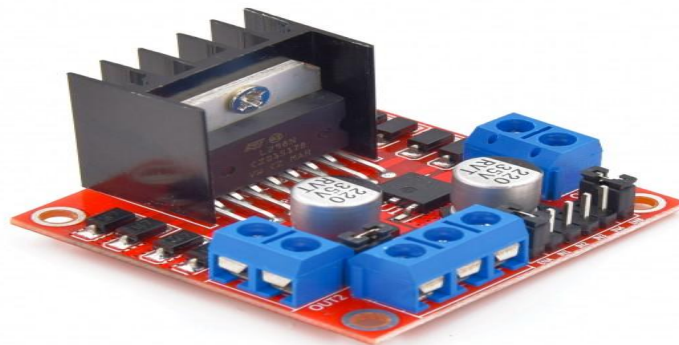


Figure II.6 : Composant L298N

II.4.6 : Module Bluetooth HG-06

Bluetooth est une spécification de l'industrie des télécommunications. Elle utilise une technique radio courte distance destinée pour simplifier les connexions entre les appareils électroniques. Nous allons utiliser le module Bluetooth afin de communiquer l' Arduino avec Smartphone (Android).

- GND (carte Bluetooth) ⇒ Arduino GND

- VCC (carte Bluetooth) ⇒ Arduino 5V
- TX-O (carte Bluetooth) ⇒ Arduino RX (0)
- RX-I (carte Bluetooth) ⇒ Arduino TX (1)

Nous allons commencer par nous connecter à l'Arduino par Bluetooth. Le mot [9]de passe est 1234. Une fois connecté, nous allons installer une application pour lui envoyer des chiffres.

1- Caractéristiques du module Bluetooth HG-06

Le module Bluetooth est doté :

- Module de type JY-MCU / HC-06 Master
- Paramétrage par défaut du port série: 9600.
- Alimentation de 3.3 à 5V DC
- LED indicatrice : statut de connexion.
- Fonctionnement bluetooth sur la bande 2.4 -2.8 GHz.
- Modulation: GFSK (Gaussian Frequency Shift Keying)
- Puissance d'émission: <= 4dBm, classe 2
- Vitesse: asynchrones: 2.1 Mbps (max)/160 kbps, synchrones: 1 Mbps/1 Mbps
- Puissance Classe: (+ 6dBm)
- Courant: éplucher-35mA, connecté-8mA
- Température:-40 ~ + 105 degrés Celsius
- Défini par l'utilisateur vitesse de transmission: 4800, 9600, 19200, 38400, 57600, 115200, 230400,460800, 921600, 1382400.
- Dimension: 35.7mm * 15.2mm
- Le module est appairer avec un mot de passe (1234) modifiable.



Figure II.7: Module Bluetooth

II.4.7 : Panneaux solaire

Le module solaire photovoltaïque est un système capable de capter l'énergie envoyée par le soleil vers la Terre, et de la convertir en courant électrique continu. Ce procédé est rendu possible par les cellules photovoltaïques qui composent le module. Chaque cellule est produite à l'aide d'un matériau semi-conducteur appelé le silicium. ... En effet, la lumière du soleil se compose de photons qui vont venir frapper la surface du **panneau solaire** photovoltaïque.[13]

Le fonctionnement d'un panneau solaire photovoltaïque repose sur plusieurs éléments :

- Un module composé de cellules photovoltaïques transformant l'énergie en tension électrique continue;
- Un système de montage, également appelé système d'intégration à la toiture qui ancre le panneau photovoltaïque au bâti ;
- Un abrévement qui établit une jonction étanche entre le panneau et le reste de la toiture.

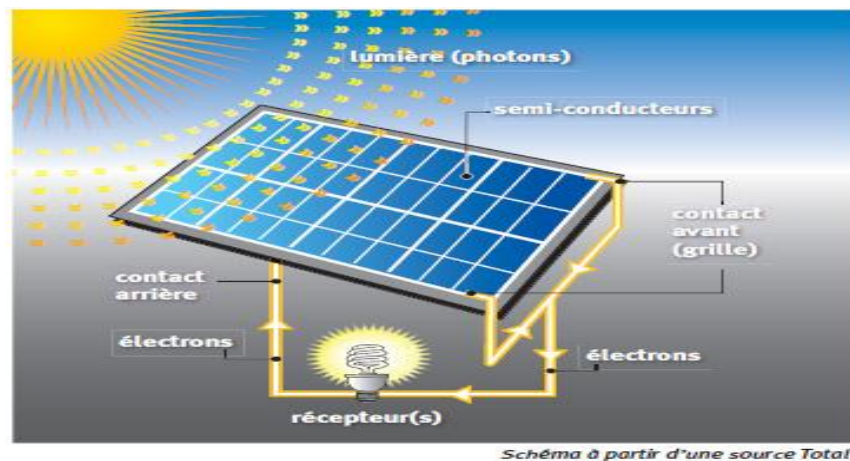


Figure II.8: panneaux solaire

II.4.8 : Régulateur de charge

Le régulateur de charge est utilisé dans les systèmes autonomes de l'énergie solaire photovoltaïque. Il peut également être appelé contrôleur de charge. Le régulateur de charge/décharge est associé à un générateur photovoltaïque, il a [14] pour rôle, entre autres, de contrôler la charge de la batterie et de limiter sa décharge. Sa fonction est primordiale car elle a un impact direct sur la durée de vie de la batterie.



Figure II.9 : Régulateur de charge

II.4.9 : Batterie 12v

La batterie 12v est une batterie utilisé pour stocker l'énergie venant du panneaux solaire. L'électricité produite par les panneaux solaires en journée y sera stockée, avant d'être utilisée au moment où on a vraiment besoin. la batterie de stockage solaire nous permet de stocker puis consommer notre énergie solaire quand nous le souhaitons. La durée de vie d'une batterie dépend de plusieurs variables : le nombre de cycles, la puissance de charge prélevée, la profondeur de décharge des cycles, l'état de charge ou encore la température[15] ambiante de la pièce.



Figure II.10 : batterie 12v

II.5 : Conclusion

Dans ce chapitre, nous avons fait une étude théorique de notre projet réalisation et commande d'un détecteur d'obstacle. Nous avons rédigé un cahier des charges du projet dans lequel nous décrivons les spécifications du projet. Ensuite nous avons élaboré un schéma synoptique du projet à partir duquel nous avons pu réaliser ce projet. Nous terminons ce chapitre par des détails sur chaque composant ainsi que son rôle dans le Robot.

CHAPITRE III

Description du Robot

III.1-Définition d'un robot

Un **robot** est un dispositif mécatronique, il allie de la mécanique, de l'électronique et de l'informatique. Il accomplit automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitive ou impossible pour les humains, soit des tâches plus simples mais en les réalisant mieux que ce que ferait un être humain.

III.2-Schéma synoptique d'un robot

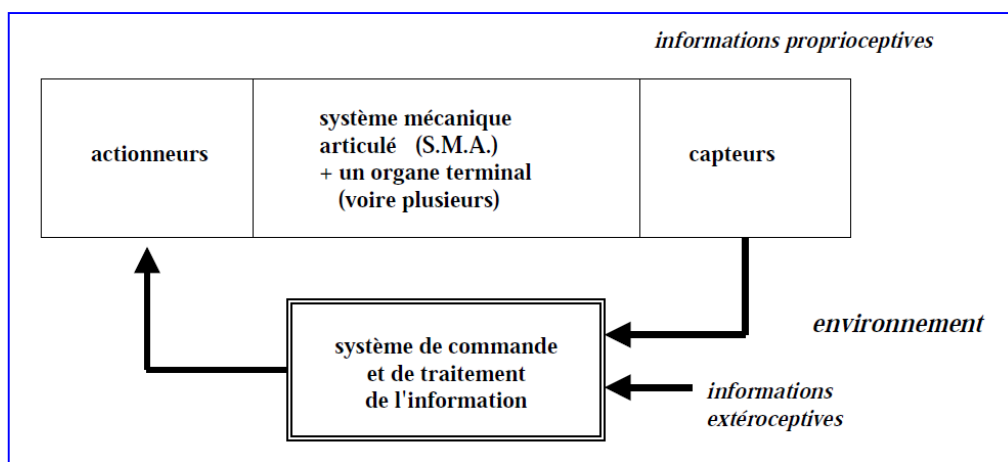


Figure III.1 : Schéma synoptique d'un robot

- ✓ **Le système mécanique articulé (S.M.A.)** est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action. Son rôle est d'amener l'organe terminal dans une situation (position et orientation) donnée. Son architecture est une chaîne de corps, généralement rigides, assemblés par des liaisons appelées articulations.
- ✓ **La motorisation** est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations.
- ✓ **Les capteurs** sont des organes de perception : proprioceptifs lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).

- ✓ **L'organe terminal** : dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, ...), ou à les transformer (outils, torche de soudage, pistolet de peinture, ...). Il s'agit d'une interface permettant au robot d'interagir avec son environnement.

- ✓ **La partie commande** synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur.

III.3 Types de robots

Il existe deux types de robots :

- Robots manipulateurs
- Robots mobiles

III.3.1- Robot manipulateur

Ce sont des robots à base fixe constitués d'un bras. Le bras est un ensemble de corps reliés par des liaisons (rotoïdes ou prismatiques). En général le bras manipulateur se termine par un outil ou un organe de préhension (pince, main, etc...). La plupart des robots sont fixes et leurs environnements parfaitement défini à l'avance. Grâce à la chute des prix et la fiabilité des composants électroniques.

III.3.2-Robot mobile

Les robots mobiles sa liberté de mouvement lui donne son autonomie pour interagir dans le monde des humains. Le robot mobile est d'abord une base mobile ; à ce titre il peut être utilisé pour le transport. Mais, dès lors qu'il est équipé d'un bras manipulateur ou d'une caméra, ses applications sont plus nombreuses (surveillance, exploration, ...).

Les déplacements représentent l'aspect le plus important d'un robot mobile. Les outils dont il peut être équipé doivent être considérés comme des robots de type fixe et indépendants dans un premier temps. Dans notre travaille on s'intéresse à ce type de robot, nous allons réaliser

un véhicule autoguide a base d'une Arduino Uno comme il est présenté dans tous ce qui suit. Il existe plusieurs applications dans le robot mobile

III.4-Application des robots mobiles :

III.4.1. Robot d'exploration des planètes

A l'image de Séjourner une mission qui marquait le grand retour des Américains sur la planète rouge, plus de vingt ans après les missions Viking. C'était aussi la première fois qu'un robot mobile déambulait sur une autre planète que la Terre. Elle a constitué un succès total pour la NASA, notamment car elle a validé des prouesses technologiques qui n'avaient jamais été utilisées (jusqu'alors, tels les coussins gonflables, dits « airbags », pour se poser sur le sol. Mars Pathfinder est aussi remarquable au vu de son coût, très bas par rapport à d'autres missions spatiales robotisées. L'atterrissage est enfin une prouesse, car deux tiers des vaisseaux envoyés vers la planète Mars n'ont jamais atteint leur objectif final.



Figure III.2 : Robots d'exploration



Figure III.3 : Robots militaires

III.4.2. Les robots militaires

Un robot militaire, aussi appelé arme autonome, est un robot, autonome ou contrôlé à distance, conçu pour des applications militaires. Les drones sont une sous classe des robots militaires. Des systèmes sont déjà actuellement en service dans un certain nombre de forces armées, avec des succès remarquables, tel que le drone "Predator", qui est capable de prendre des photographies de surveillance, et même à lancer des missiles air-sol AGM-11

«Hellfire" II ou des GBU-12 "Paveway" II dans le cas du MQ-1 et du MQ-9. Les études se poursuivent car ce type d'engins offre des possibilités prometteuses. Ces robots posent des

problèmes éthiques et légaux. Cela a conduit des associations ou ONG à mener des actions de sensibilisation à ces problèmes pour encadrer l'utilisation de ces robots militaires.

III.5 Application industrielle :

Un véhicule à guidage automatique, ou véhicule autoguidé (en anglais : Automatique Guided Véhicules ou (AGV) ou Véhicule à Guidage Automatique (VGA) est un robot qui se déplace de façon autonome sans l'intervention humaine. Les technologies d'automatisation traditionnelles sont le filoguidée, le laser guidage et l'autoguidage. Cependant, d'autres technologies aussi performantes existent telles que le géo guidage et le guidage par ultrason. Les AGV sont le plus souvent utilisés dans des applications industrielles pour déplacer de manière autonome des marchandises dans une usine, un entrepôt ou un atelier. Ainsi, l'expression chariot automatique pour désigner un AGV trouve son origine dans les premiers AGV présents sur le marché : des chariots de manutention pour le transport de palettes. En effet, le premier AGV a vu le jour dans les années 50, initialement développé pour l'industrie automobile. De nos jours, leurs applications ont été élargies et ils ne sont plus limités au secteur industriel. Ils sont également utilisés dans des hôpitaux, musés, aéroports...



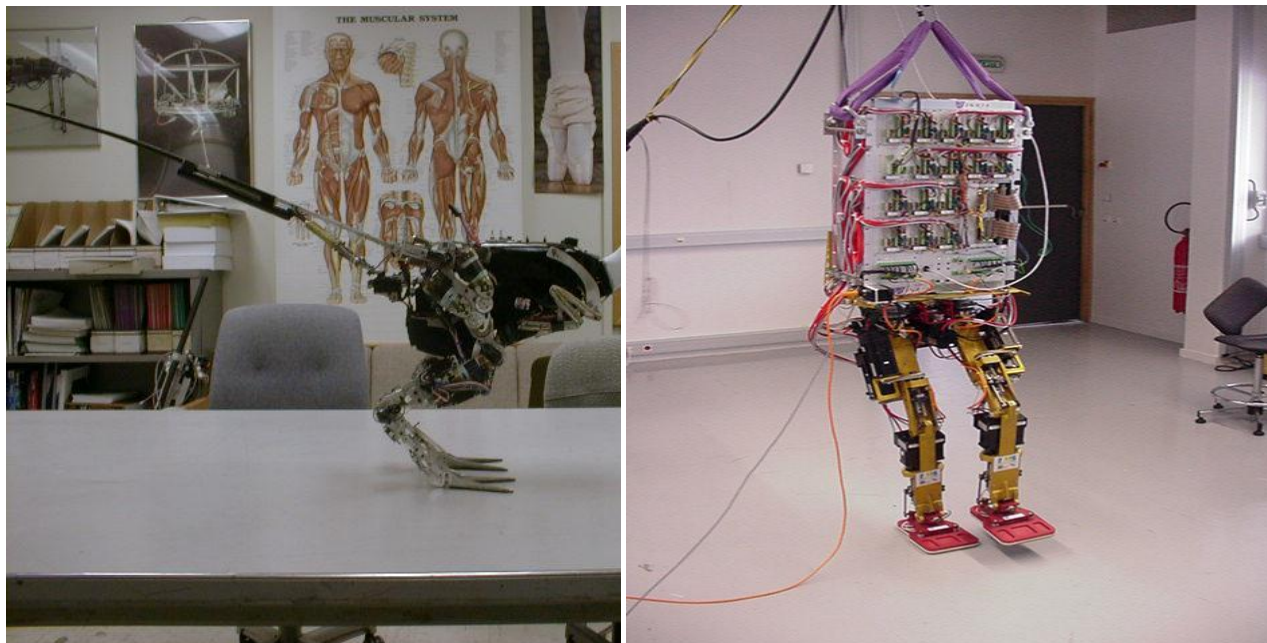
Figure III.4. Véhicule autoguide

III.5.1 Robot de liaison en usine : Application domestiques :Un robot aspirateur (ou aspirateur robot) est un aspirateur robotisé, c'est-à-dire capable de réaliser le travail d'un aspirateur de manière autonome, sans intervention d'un être humain. C'est en 2009 que s'ouvre le principal marché de robots domestiques dans le monde. Le Roobas, fabriqué par la société Irobot, a été vendu à plus de cinq millions d'exemplaires en 2009.



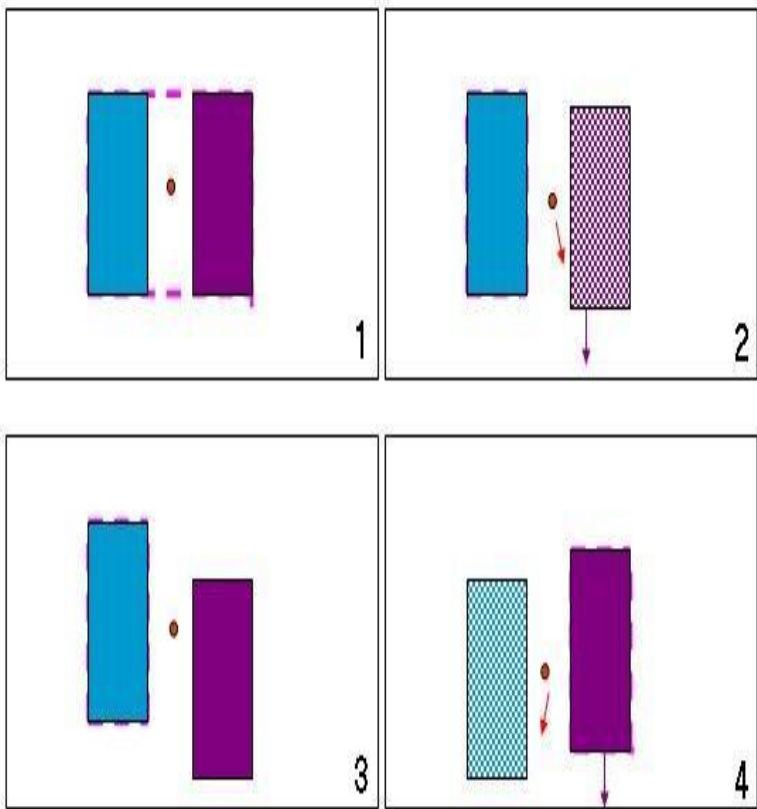
Figure III.5 : Robot aspirateur

III.5.2 Les Robots bipèdes



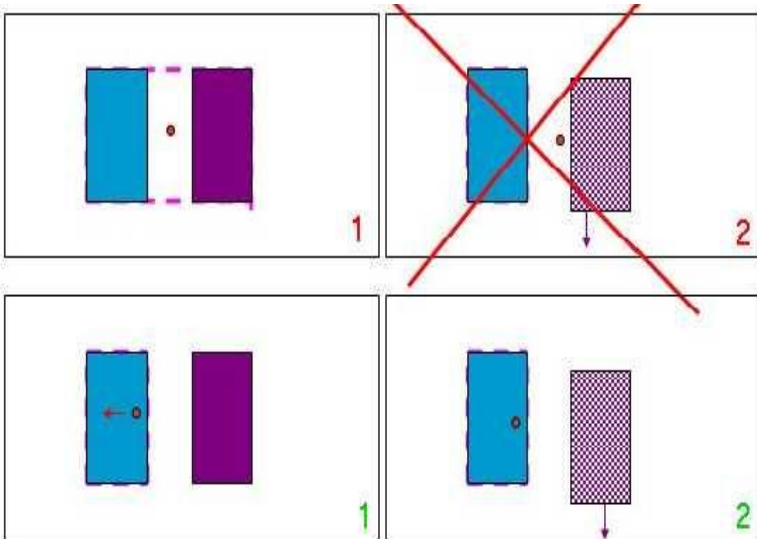
III.6. Dans la robotique on trouve deux types de marches :

- **Marche dynamique**
 - Perte d'équilibre entre chaque pas.
 - Difficulté : s'arrêter en pleine marche.



➤ **Marche quasi-statique :**

- Le robot ne sera jamais en déséquilibre.



CHAPITRE IV

Réalisation pratique du robot

IV.1 : Introduction

Dans ce chapitre, nous allons passer à la réalisation pratique du robot. Nous allons d'abord commencer par détailler le fonctionnement du Robot, ensuite nous réaliserons le logiciel afin de pouvoir simuler le projet sur un logiciel dédié. Après la simulation, nous testerons le projet si tout marche comme prévu, nous procéderons avec les tests finaux et la validation.

IV.2 : Schéma électronique du projet

Pour mieux prédire le comportement de notre robot la simulation est exigée, on a réalisé ce Schéma suivant pour la version basic (détecteur d'obstacle) : Cette méthode basée sur un entête de détection dynamique afin de garantir la précision ainsi une décision un peu intelligente par rapport au méthode classique. Donc en pratique, on obtiendra le montage suivant :

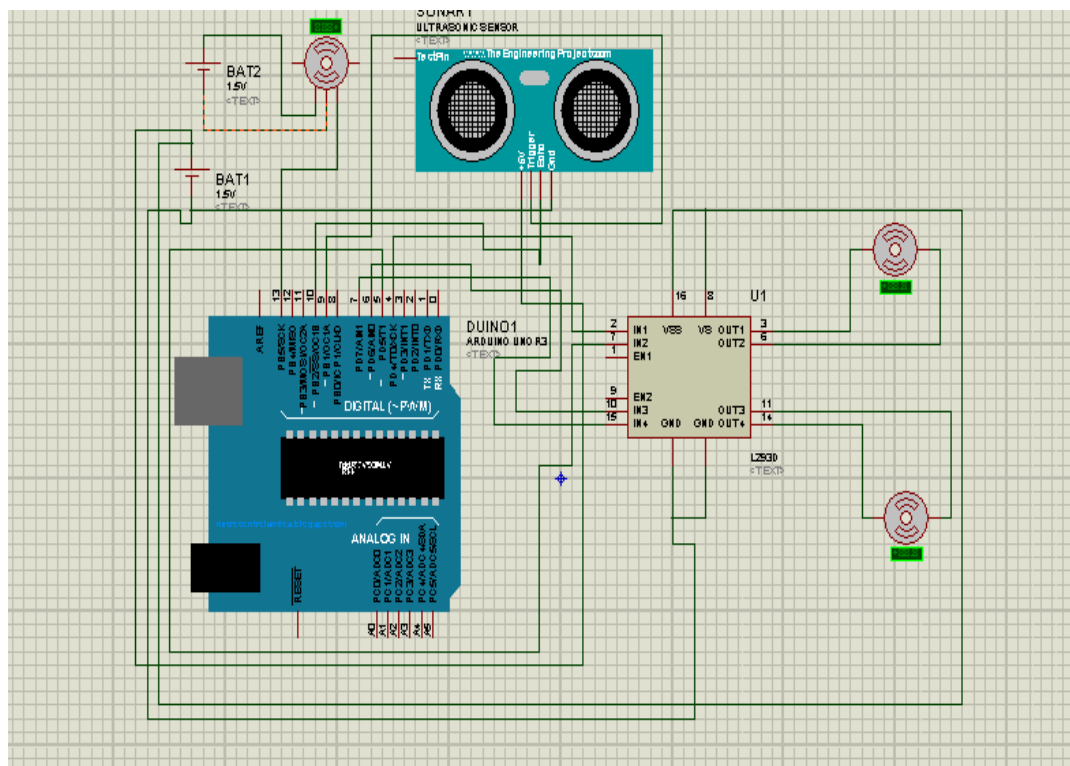


Figure IV.1 : Simulation avec le logiciel proteus

Ce schéma a été réalisé avec le logiciel Proteus 8 Professional, qui est en même temps le logiciel que nous avons choisi pour simuler notre projet. Les labels de même nom sont connectés. Dans ce schéma certains composants que nous devons utiliser la réalisation du montage final n'existe pas. Mais on a simulé au moins avec les composants que l'on trouve dans la librairie.

IV.2.1. Logiciel fritzing

Fritzing est un projet de logiciel libre, destiné aux à l'électronique. Il a notamment pour vocation de favoriser l'échange de circuits électroniques libres et d'accompagner l'apprentissage de la conception de circuits. Le logiciel conçu par la faculté de sciences appliquée de l'Université de Potsdam et dont le développement est assuré par la fondation, également nommée Fritzing, est un logiciel d'édition de circuit imprimé. Il est disponible dans seize langues dont le français. Il est adapté aux débutants où confirmés en électronique pour faire rapidement des circuits simples, et est également un bon outil didactique pour apprendre à bidouiller en électronique par la pratique

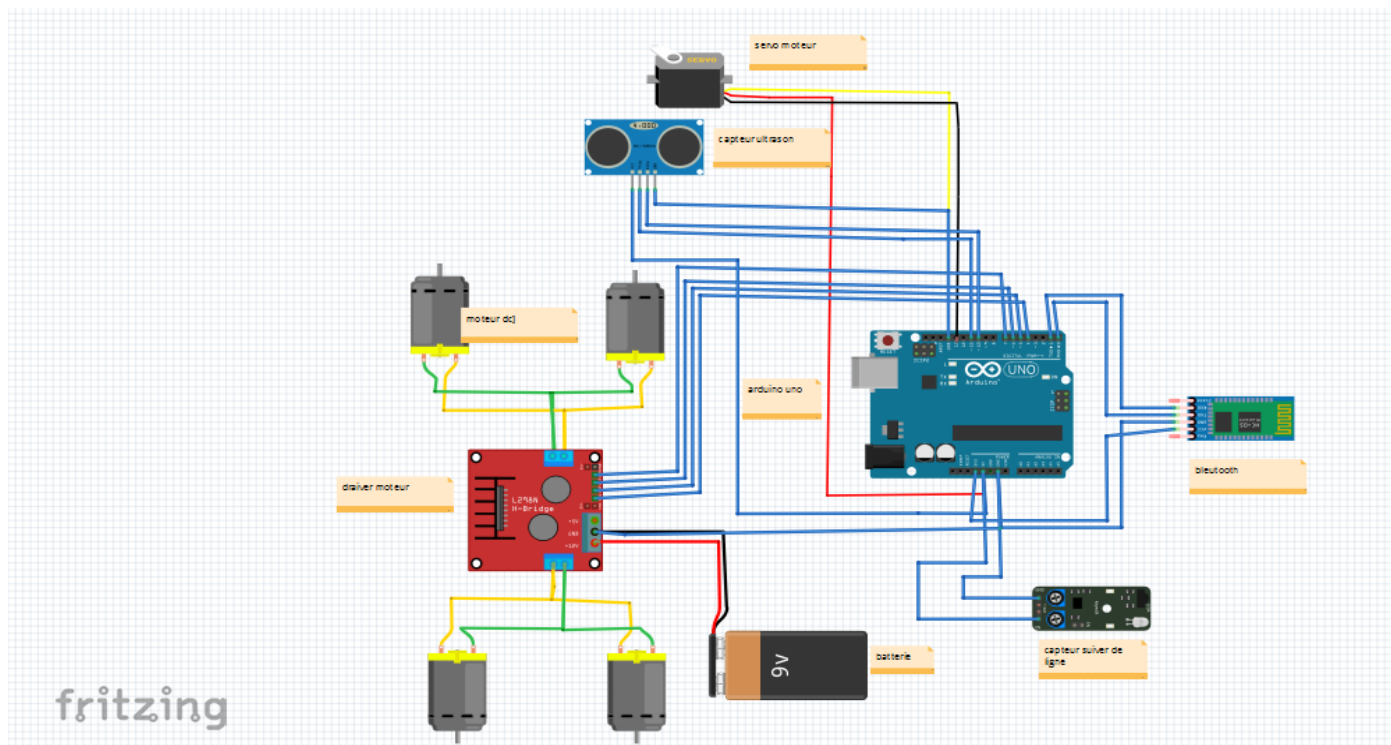


Figure IV.2 : schéma électronique du robot

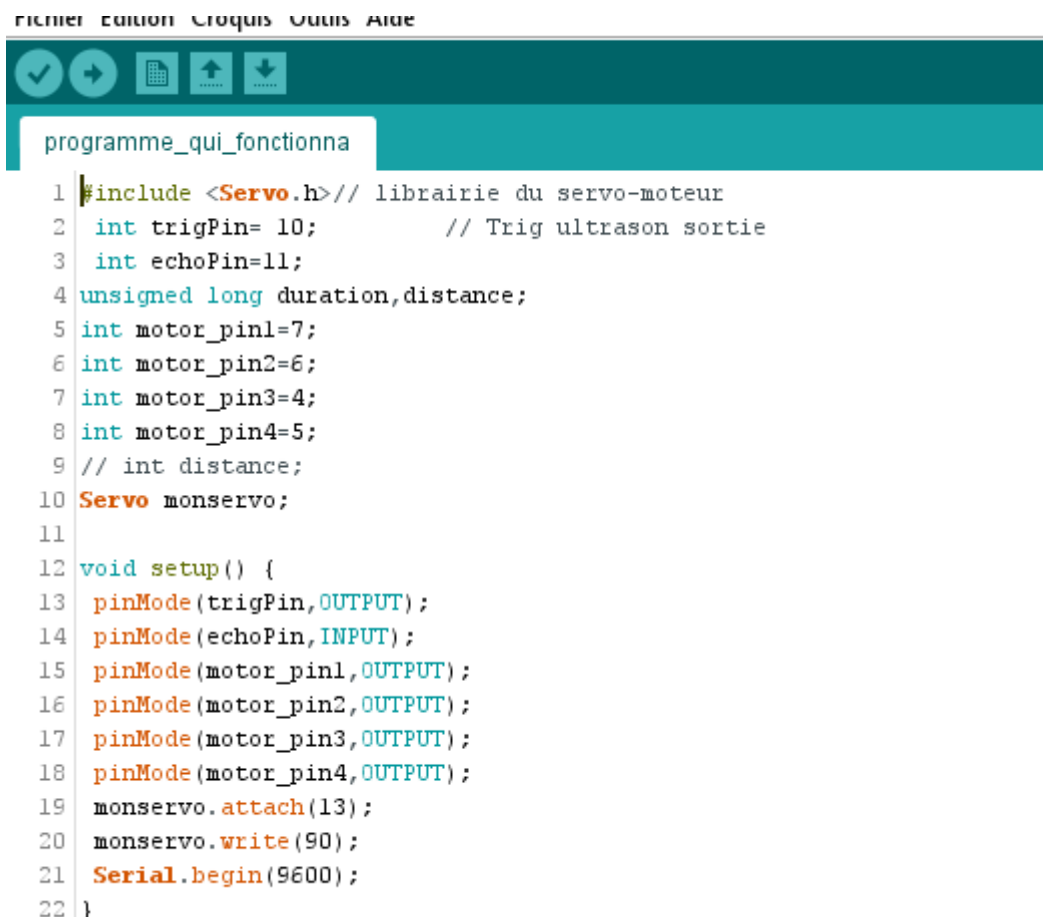
IV.3 : Logiciel

IV.3.1 : La conception du logiciel sous l'environnement Arduino

La partie logicielle est la partie la plus complexe de ce projet. Le programme est évidemment écrit en langage Arduino.

Des détails sur l'environnement de développement et la structure d'un programme Arduino sont donnés dans « Annexe 2.3 ». Comme tout programme Arduino, notre programme se décompose en trois (03) parties :

- Déclaration des variables, des constantes, l'inclusion des bibliothèques...



```
1 #include <Servo.h> // librairie du servo-moteur
2 int trigPin= 10;      // Trig ultrason sortie
3 int echoPin=11;
4 unsigned long duration,distance;
5 int motor_pin1=7;
6 int motor_pin2=6;
7 int motor_pin3=4;
8 int motor_pin4=5;
9 // int distance;
10 Servo monservo;
11
12 void setup() {
13   pinMode(trigPin,OUTPUT);
14   pinMode(echoPin,INPUT);
15   pinMode(motor_pin1,OUTPUT);
16   pinMode(motor_pin2,OUTPUT);
17   pinMode(motor_pin3,OUTPUT);
18   pinMode(motor_pin4,OUTPUT);
19   monservo.attach(13);
20   monservo.write(90);
21   Serial.begin(9600);
22 }
```

Figure IV.3 : Partie déclaration dans le programme

- La fonction d'initialisation « *setup* () » :

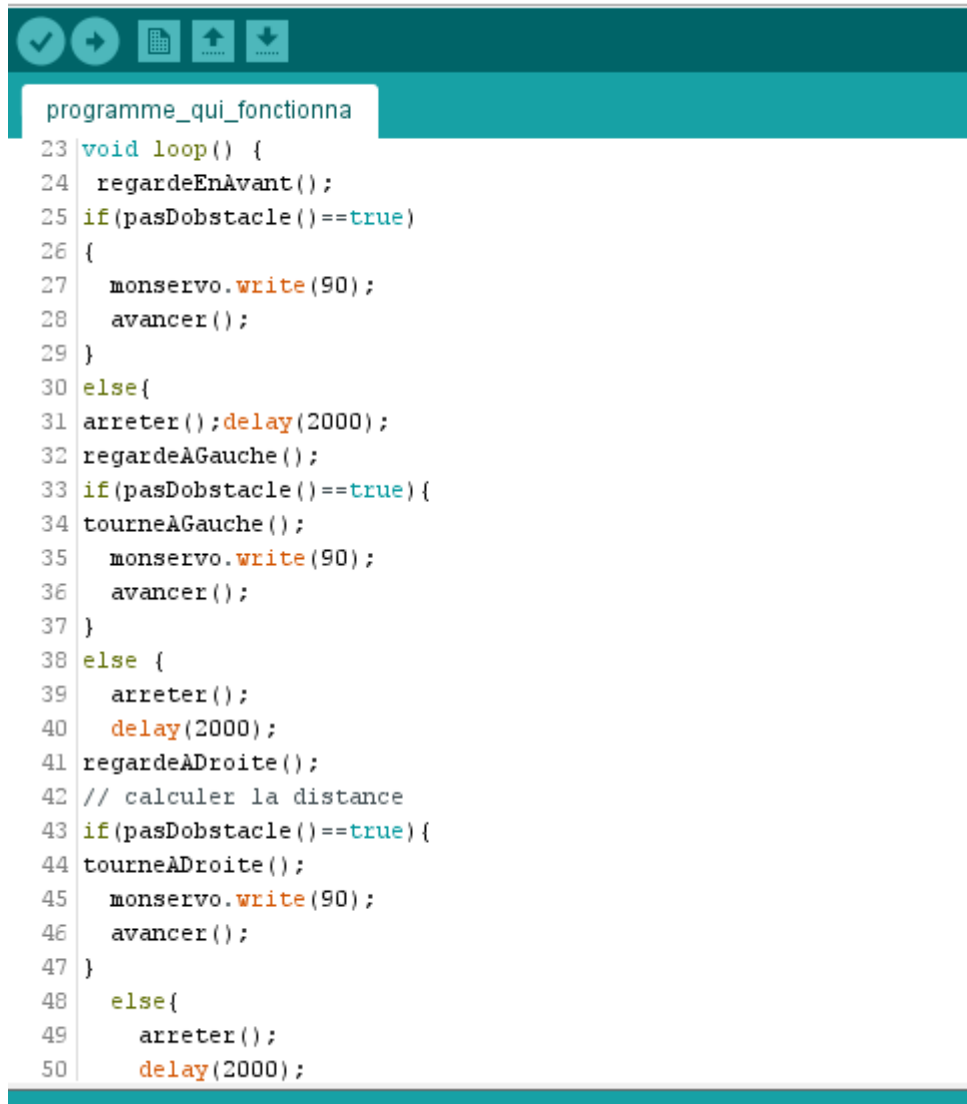
```

programme_qui_fonctionna
--
12 void setup() {
13   pinMode(trigPin,OUTPUT);
14   pinMode(echoPin,INPUT);
15   pinMode(motor_pin1,OUTPUT);
16   pinMode(motor_pin2,OUTPUT);
17   pinMode(motor_pin3,OUTPUT);
18   pinMode(motor_pin4,OUTPUT);
19   monservo.attach(13);
20   monservo.write(90);
21   Serial.begin(9600);
22 }
23 void loop() {
24   regardeEnAvant();
25   if(pasDobstacle()==true)
26   {
27     monservo.write(90);
28     avancer();
29   }
30   else{
31     arreter();delay(2000);
32     regardeAGauche();
33     if(pasDobstacle()==true){
34     tourneAGauche();
35     monservo.write(90);
36     avancer();
37   }
38   else {
--

```

Figure IV.4 : Fonction « setup () » du programme

- La fonction d'initialisation « *loop* () » :



```
programme_qui_fonctionna
23 void loop() {
24   regardeEnAvant();
25   if(pasDobstacle()==true)
26   {
27     monservo.write(90);
28     avancer();
29   }
30   else{
31     arreter();delay(2000);
32     regardeAGauche();
33     if(pasDobstacle()==true){
34       tourneAGauche();
35       monservo.write(90);
36       avancer();
37     }
38     else {
39       arreter();
40       delay(2000);
41     regardeADroite();
42     // calculer la distance
43     if(pasDobstacle()==true){
44       tourneADroite();
45       monservo.write(90);
46       avancer();
47     }
48     else{
49       arreter();
50       delay(2000);
```

Figure IV.5 : Fonction « loop () » du programme

IV.3.3 : Organigramme

Le schéma bloc explicatif du notre robot est présenté sur cette figure

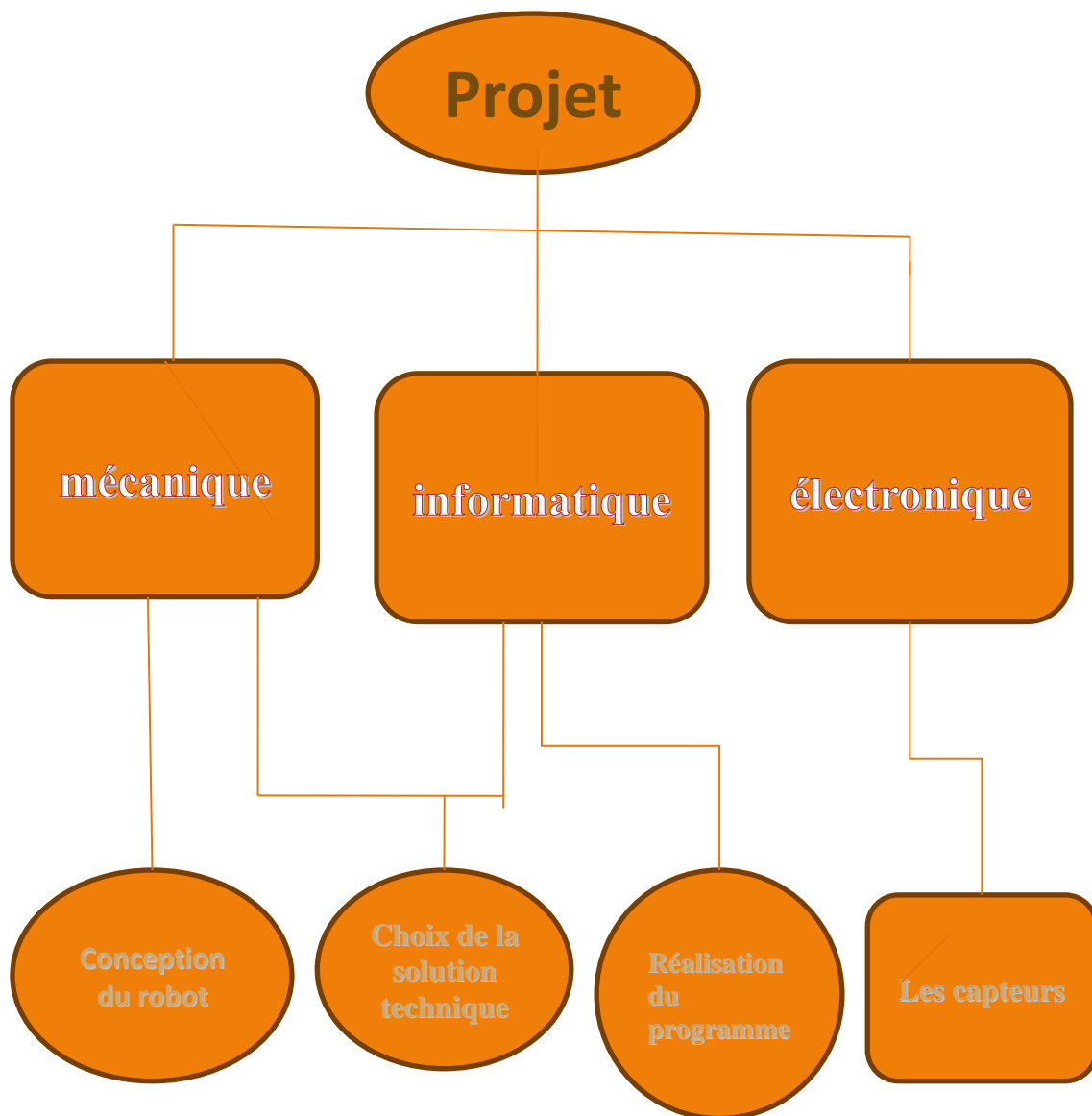
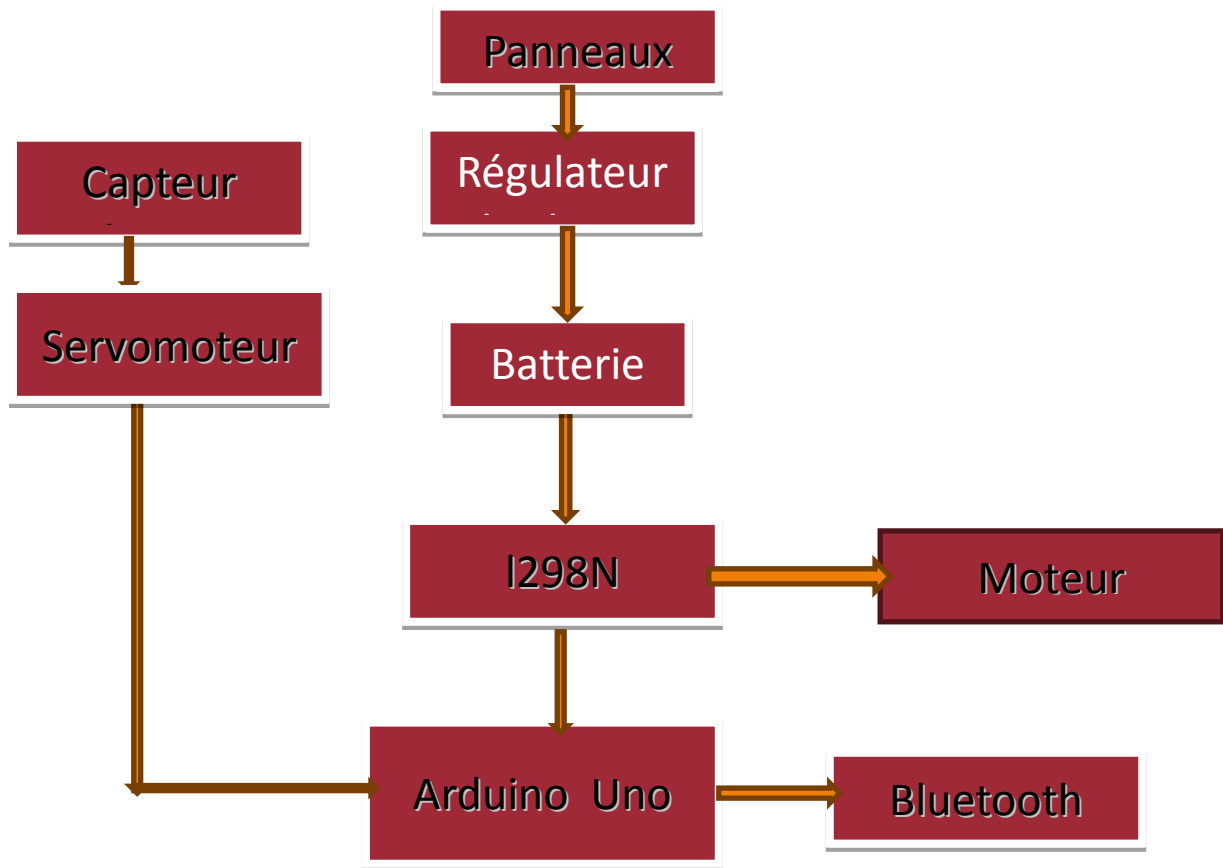
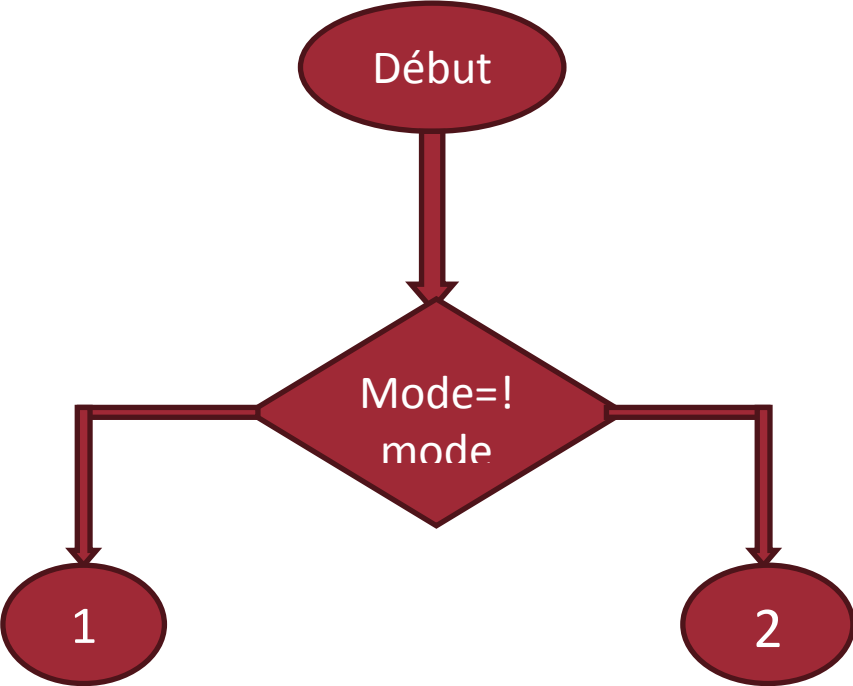


Figure IV.6 : Schéma bloc

Schéma détaillé du robot

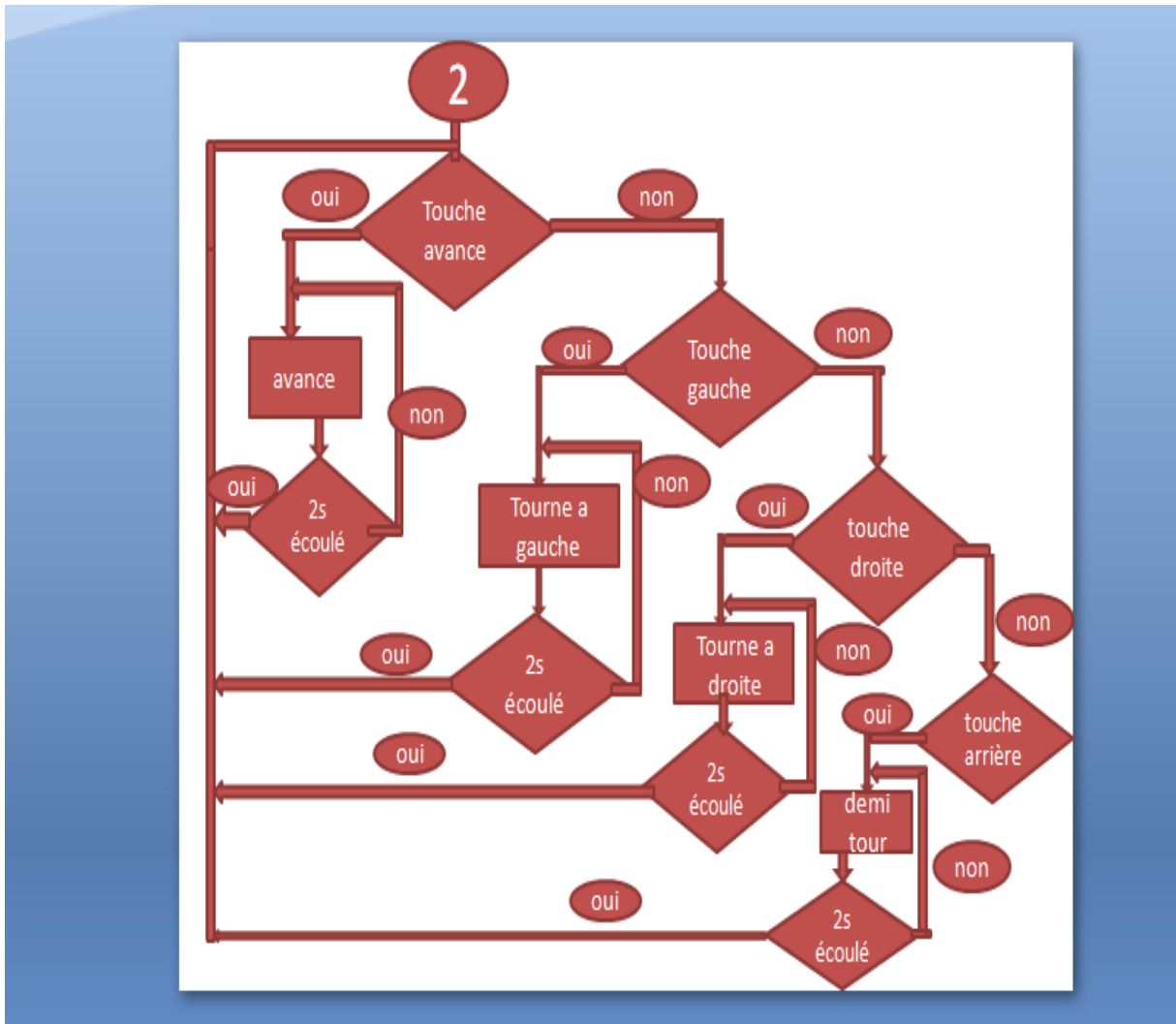


Organigramme du haut niveau du fonctionnement du robot





1-Mode Automatique



2-Mode Manuel

Figure3.6: Organigramme du haut niveau du projet

IV.4 : Montage du robot

Une fois les différentes pièces montées, le robot se présente comme sur la figure suivante :

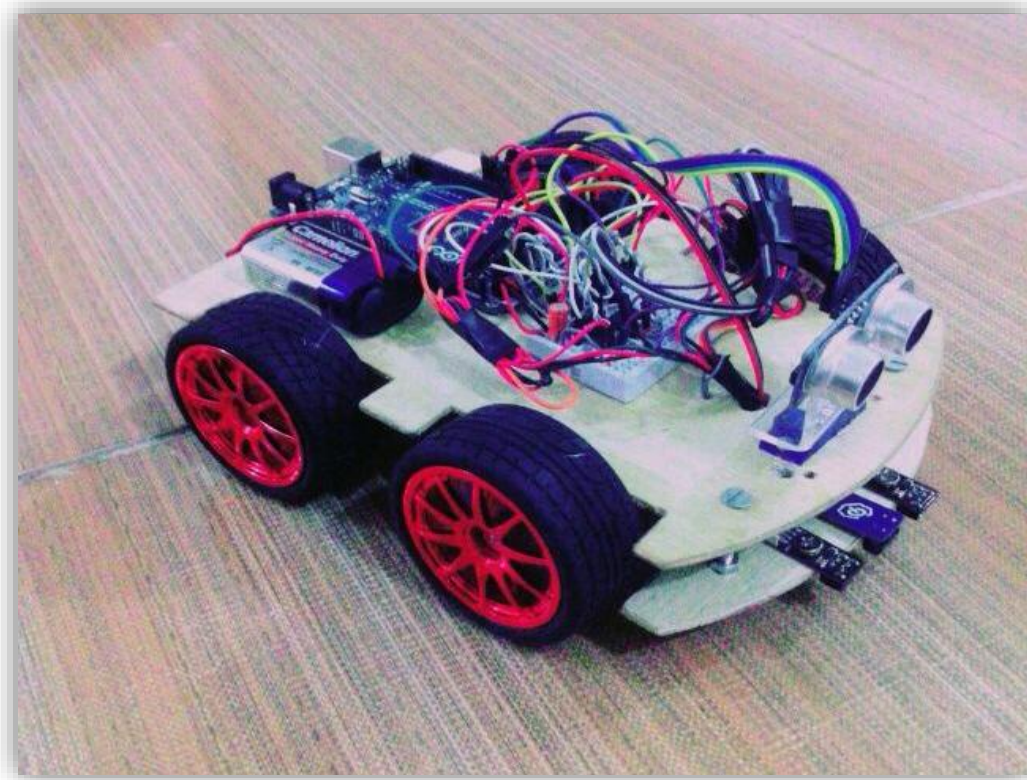


Figure IV. Montage final du robot sans les panneaux solaires



Figure IV. Montage final du robot avec les panneaux solaires

IV.5 : Conclusion

Un robot mobile autonome est une machine qui a besoin, pour sa mise en œuvre, de spécialistes dans des nombreux domaines comme la mécanique, l'automatique, l'électronique, l'électrotechnique et l'informatique. Et de même pour son indispensabilité dans les différentes applications soit: Militaire, industriel, exploration des planètes ou bien domestiques.

Conclusion générale et perspectives

L'objectif de notre travail est la réalisation d'un robot mobile type voiture avec détection d'obstacles en utilisant des capteurs de distances (ultrason) ainsi que l'application Android. Pour réaliser ce travail, on a passé par différentes étapes, l'utilisation d'un capteur ultrasonique HC-SR04 pour la détection des obstacles et le calcul de la distance entre le robot et l'obstacle, dans notre cas, l'obstacle doit être distant de quinze (15) cm pour être détecté. Le robot réalisé se déplace par quatre roues ; deux roues arrière et deux roues avant qui s'occupent du mouvement par quatre moteurs à courant continu. Nous avons utilisé ce type de moteur pour la précision dans le déplacement et la rotation du Robot. Des circuits de commande sont réalisés afin de contrôler les moteurs associés. L'ensemble de système de perception et de déplacement est commandé par un microcontrôleur qui doit en utilisant les informations actuelles, décider l'action à prendre. Pour notre cas; on a utilisé le microcontrôleur célèbre, l'arduino Uno. Les nombreuses difficultés l'indisponibilité des composants électroniques et les nombreux ralentissements rencontrés nous ont permis de comprendre nos erreurs dans notre approche et dans notre stratégie de travail. Les réflexions apportées afin de proposer des solutions convenables sont une partie du projet que nous avons particulièrement apprécié. Nous pouvons utiliser des cameras de haute qualité à cause de la perturbation de l'environnement (luminosité, les obstacles inévitables, et Utilisé une méthode de détection des objets basé sur les couleurs. Les possibilités d'optimisation du robot sont nombreuses et nous lassent alors optimistes quant à une réalisation aboutie du projet.

Annexes

Annexe 1 : Les microcontrôleurs

Annexe 1.1 : Définition d'un microcontrôleur

On peut définir un microcontrôleur comme étant un circuit intégré programmable contenant un petit système informatique. Tout comme un ordinateur, il contient un processeur, des mémoires de données et de programme, des circuits d'interfaces d'entrées-sorties pour communiquer avec le monde extérieur et toute la circuiterie électronique et logique nécessaire pour fonctionner tout cela ensemble. Il nécessite une mise en œuvre matérielle (électronique) et une mise en œuvre logicielle (informatique).

Annexe 1.2 : Généralités

Pour fonctionner, tous les microcontrôleurs, quel que soit leur type ou leur fabricant, sont en fait des systèmes à microprocesseur nécessitant certains circuits de base tels que :

Une alimentation: L'alimentation est évidemment le premier élément indispensable dans tout circuit électronique. Tous les microcontrôleurs actuels fonctionnent sous une tension unique qui varie, selon les types de circuits, de 1.8 à 6 volts environ, avec une prédilection encore aujourd'hui pour la tension 5 volts même si on se dirige peu à peu vers des versions fonctionnant sous une tension de 3.3 volts (3V3). La tension d'alimentation d'un microcontrôleur doit être stabilisée afin d'assurer un fonctionnement correct de celui-ci.

Une horloge: Le deuxième élément indispensable au fonctionnement d'un microcontrôleur est l'horloge. Son rôle est de cadencer le rythme d'exécution des instructions. La fréquence de cette horloge dépend du type du microcontrôleur utilisé et peut varier de quelques kilohertz (kHz) à plusieurs dizaines de mégahertz (MHz) (dans un PC ou un Mac, on dépasse même le gigahertz (GHz)). Plus la fréquence d'horloge est élevée, plus le microcontrôleur est rapide mais, en contrepartie, plus il consomme d'énergie.

Généralement la fréquence de cette horloge est parfaitement stable car rares sont les applications qui n'ont pas besoin de mesurer du temps ou générer des signaux aux chronogrammes précis, et cela n'est possible que si l'horloge qui pilote le microcontrôleur

est elle-même stable. Elle est généralement pilotée par un quartz ou au moyen d'un résonateur céramique, seuls composants électroniques capables de générer des signaux à une fréquence stable et précise. Il existe différents circuits pour réaliser l'horloge en externe. Le plus facile et le plus couramment utilisé est constitué d'un quartz relié avec des condensateurs de filtrage aux broches d'horloges (OSC1 et OSC2 ou CLKIN et CLKOUT) du microcontrôleur, avec une résistance (pas toujours nécessaire) à OSC2 comme présenté dans le schéma ci-dessous :

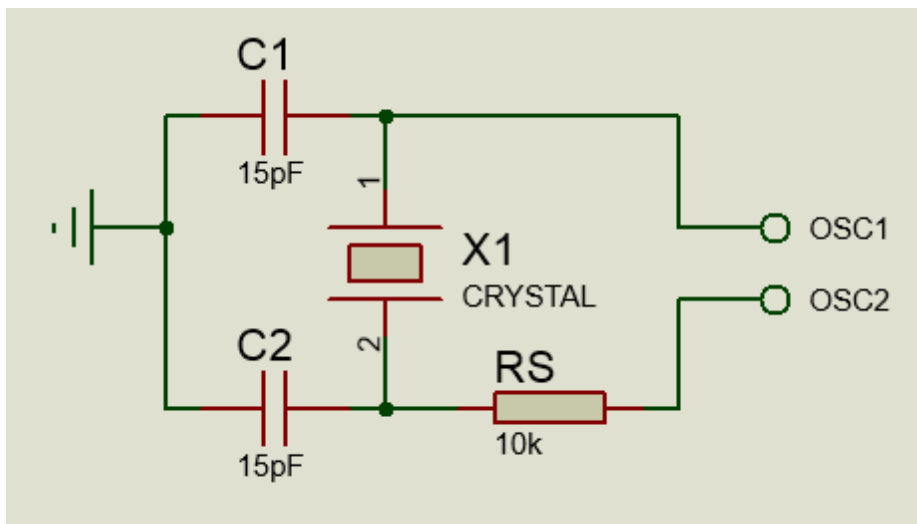


Schéma du circuit pour l'horloge externe du microcontrôleur.

Le Processeur:

appelé unité de traitement central (en anglais CPU pour Central Processing Unit). C'est le cerveau du système. Il est l'unité de traitement d'informations travaillant en trois (03) étapes : lire l'instruction, décoder l'instruction et exécuter l'instruction. Il contrôle les autres modules constituant le système à savoir les mémoires et les interfaces d'entrées sorties et travaillant à de très grandes vitesses.

La mémoire :

L'ATmega 2560 a 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATmega 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM) .

Pour info : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

Les entrées et sorties numériques :

Chacune des 54 broches numériques de la carte uno peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`. De plus, certaines broches ont des fonctions spécialisées :

- Communication Serie: Port Serie Serial : 0 (RX) and 1 (TX); Port Serie Serial 1: 19 (RX) and 18 (TX); Port Serie Serial 2: 17 (RX) and 16 (TX); Port Serie Serial 3: 15 (RX) and 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) and 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
- Interruptions Externes: Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2). Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails.
- Impulsion PWM (largeur d'impulsion modulée): Broches 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- SPI (Interface Série Périphérique): Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Uno, Duemilanove et Diecimila.

- I2C: Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils") . Noter que ces broches n'ont pas le même emplacement que sur les cartes Uno, Duemilanove ou Diecimila.
- LED: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

Broches analogiques :

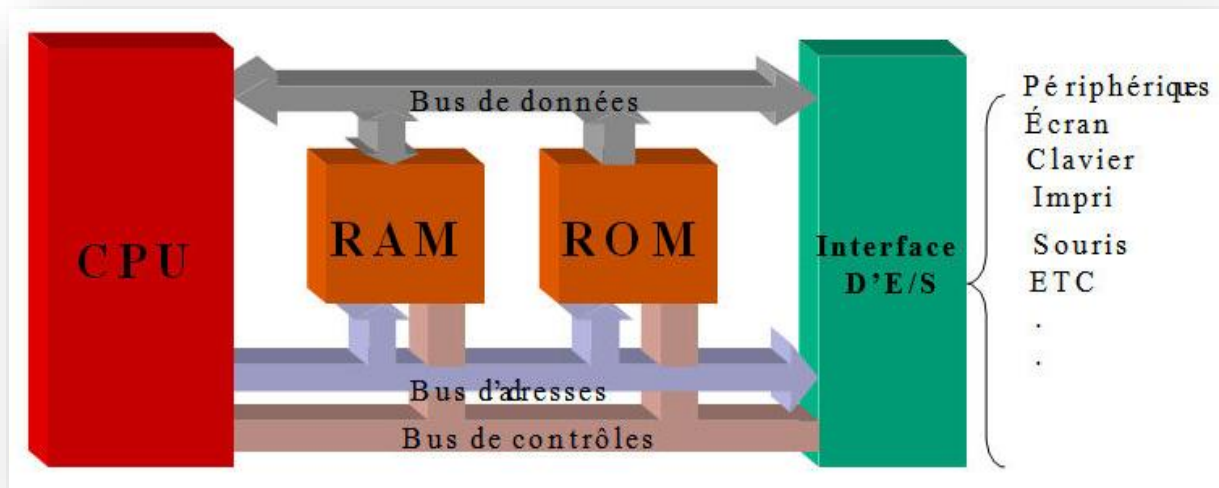
La carte uno dispose de 8 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analogReference() du langage Arduino.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques.

Les bus processeurs :

Un bus est un ensemble de pistes électroniques servant à véhiculer les informations entre le processeur et les autres modules du système (mémoires, interfaces d'E/S, etc...). De ce fait, il existe trois types de bus : le bus de données véhicule les données, le bus d'adresses véhicule les adresses et le bus de contrôle achemine les signaux de contrôle tels les signaux de validation mémoire, les ordres de lecture et d'écriture, etc... . [5]

Un processeur à n bits ($n = 4, 8, 16, 32, 64$) signifie un bus de données à n lignes



Langage de programmation

Langage de programmation arduino

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation).

Le logiciel de programmation des modules Arduino est une application Java et multiplateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande.

Le programme est constitué d'une série d'instructions saisie dans la fenêtre du logiciel. La spécificité d'Arduino se situe au niveau de la sauvegarde du code qui s'enregistre et s'exécute habituellement depuis le microcontrôleur et non pas sur votre ordinateur. Le programme se lance dès que l'appareil Arduino est mis sous tension.

Le langage de programmation utilisé est le C++, 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 maîtrisant le C ou le C++. [3]

Annexe 2 : L'Arduino

Annexe 2.1 : Qu'est-ce que l'Arduino ?

L'Arduino est une carte à microcontrôleur c'est-à-dire un ensemble constitué d'un Microcontrôleur et toute la circuiterie nécessaire à son bon fonctionnement montés sur une seule carte de taille très réduite. La carte est accompagnée d'un logiciel (EDI) multiplateforme servant à développer n'importe quelle application à implanter sur la carte avec un langage de programmation dédié que l'on peut appeler le « langage Arduino » qui est un mélange de C et de C++ restreints aux seules possibilités de la carte.

Les concepteurs de l'Arduino ont fixé leur choix sur les microcontrôleurs fabriqués par la société Atmel, et plus précisément sur la famille AVR de ce fabricant. Ce choix des AVR par les concepteurs de l'Arduino est fait sur plusieurs critères tels que l'excellent compromis prix/puissance que représentent les microcontrôleurs AVR et, même s'il ne s'agit que de microcontrôleurs à unité centrale 8 bits, ils permettent de développer des applications très intéressantes.

Le but et l'utilité

Le système Arduino, nous donnera la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, nous allons programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle

Simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique.

L'utilité est sans doute quelque chose que l'on perçoit mal lorsque l'on débute, mais une fois que vous serez rentré dans le monde de l'Arduino, vous serez fasciné par l'incroyable puissance dont il est question et des applications possibles .

Application

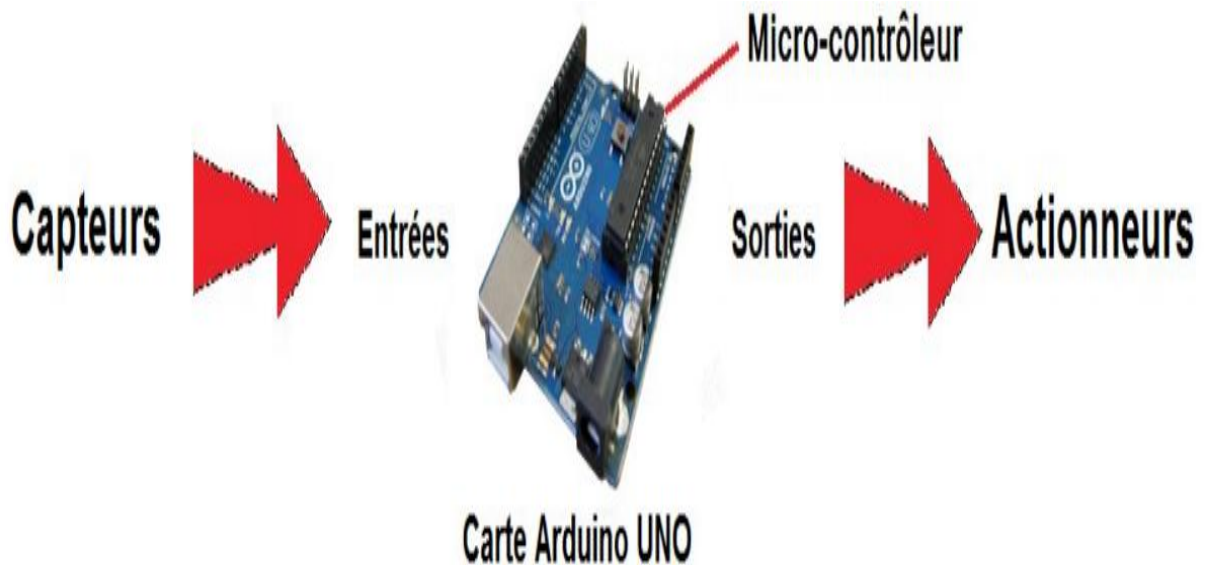
Le système Arduino, nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines

Exemple :

- contrôler les appareils domestiques
- fabriquer votre propre robot
- télécommander un appareil mobile

Annexe 2.2 : L'Arduino Uno

Une carte Arduino est une carte électronique équipée d'un microcontrôleur. Le microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable. La carte Arduino la plus utilisée est la carte Arduino Uno..



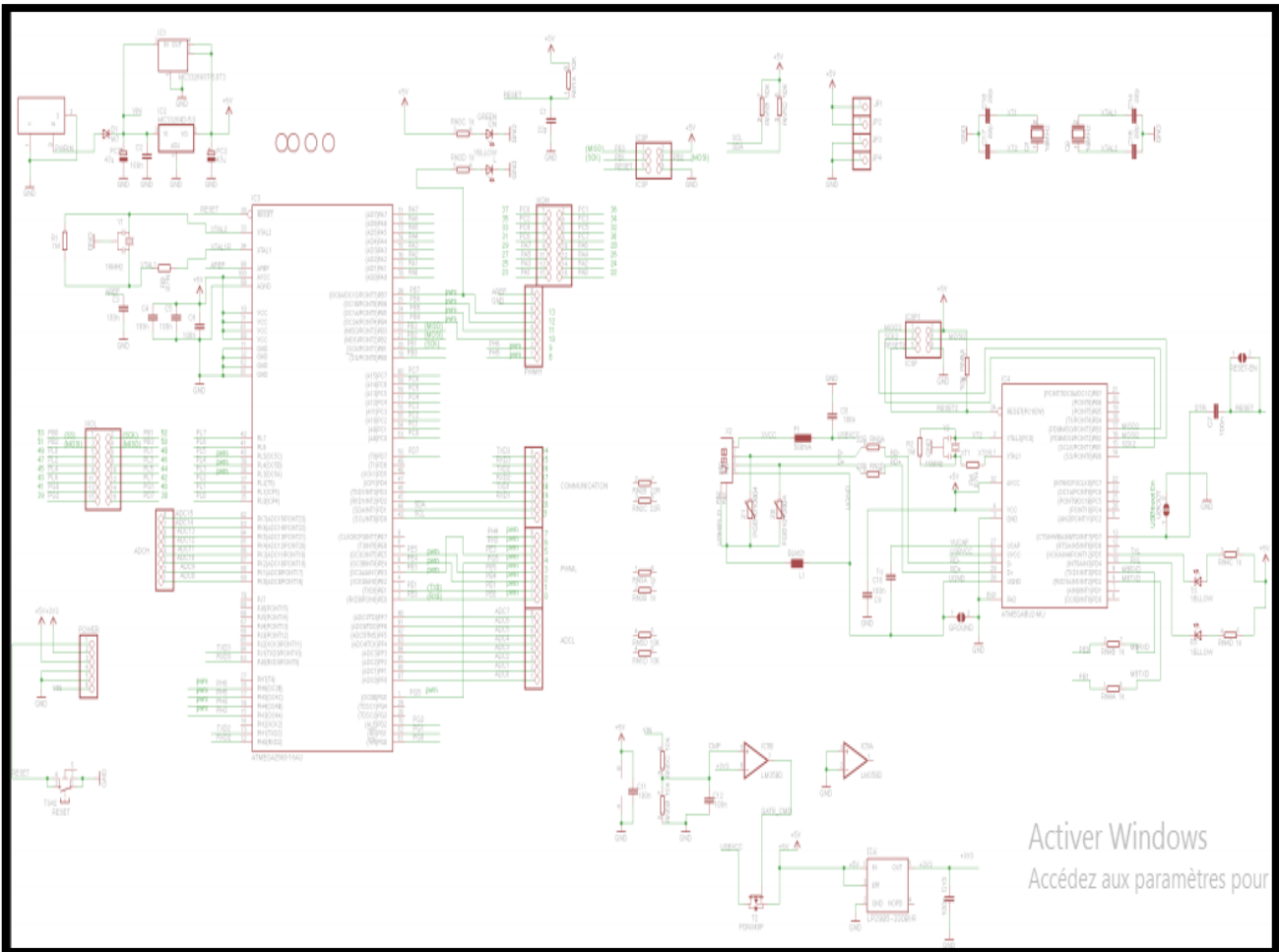
.figureA.4 : Carte arduino Uno

Caractéristique :

Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB sont utilisés par le <u>bootloader</u>
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

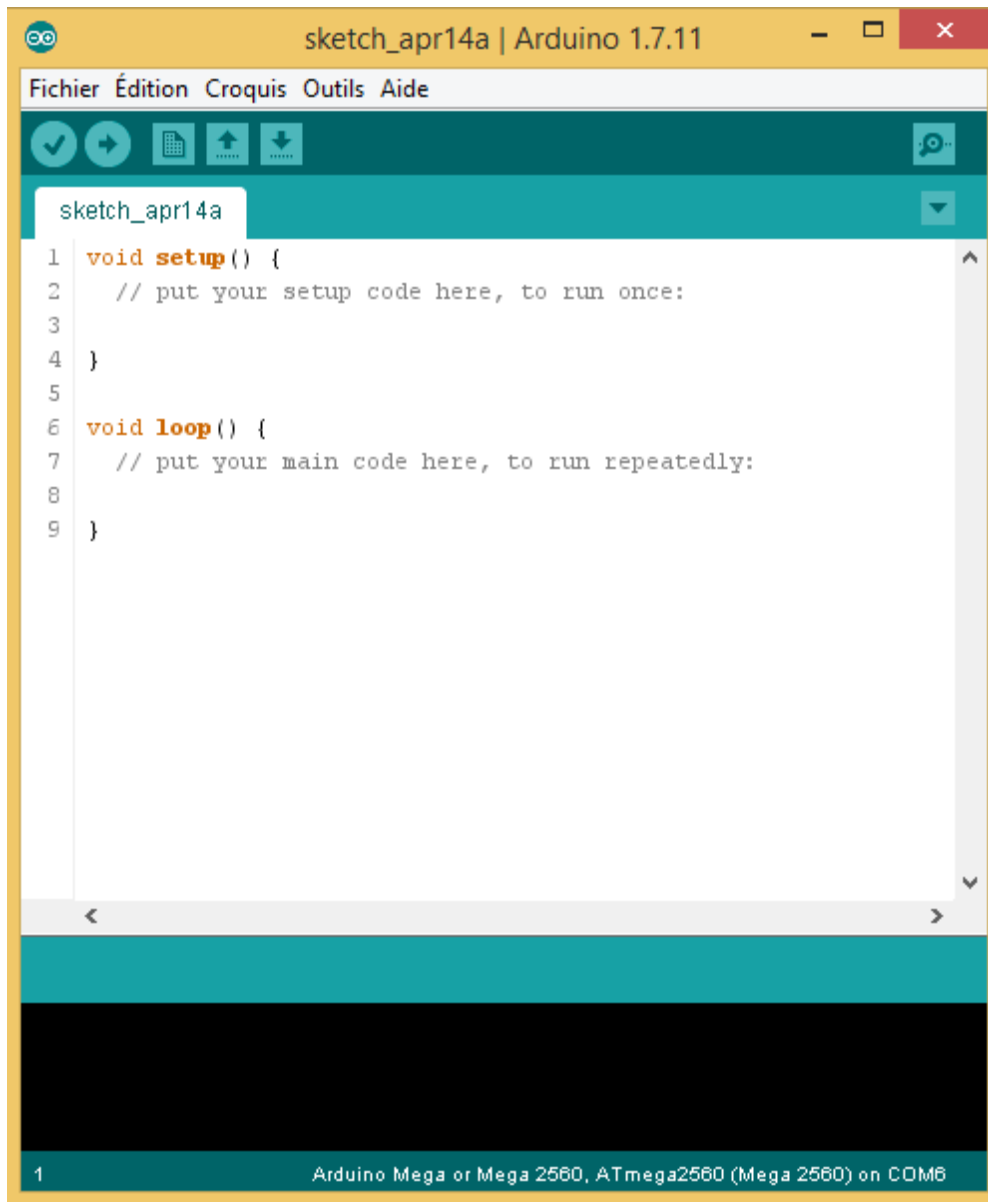
□

Circuit du schéma interne de la carte arduino Uno[1]



Annexe 2.3 : Le logiciel Arduino

Le logiciel est un environnement de développement intégré (IDE) écrit en langage Java, totalement gratuit et facilement téléchargeable sur le site officiel de l'Arduino à savoir www.arduino.cc. Le logiciel étant multiplateforme, télécharger la version correspondant à votre système d'exploitation. Une fois, le logiciel et ses pilotes, correctement installés, on peut donc programmer nos cartes Arduino. L'interface utilisateur du logiciel est présentée



FigureA.5 : interface utilisateur du langage arduino

On peut voir deux fonctions qui se présentent automatiquement dès qu'on ouvre le logiciel : il s'agit de la fonction **setup** () et de la fonction **loop** (). En effet, tout programme Arduino, appelé sketch, doit contenir obligatoirement ces deux fonctions spécifiques. Même si, elles sont vides, le programme ne marche pas si au moins l'une d'entre elles est absente.

La fonction **setup** () n'est exécutée qu'une seule fois juste après le lancement du programme. Elle contient généralement les instructions d'initialisation de certaines ressources de la carte telles que, par exemple, définition des lignes des ports parallèles soit en entrées ou en sorties, définition de la vitesse de fonctionnement du port série, etc...

Quant à la fonction **loop** (), elle est la boucle principale et se répète indéfiniment tant que l'Arduino restera sous tension. En d'autres termes, suite à un reset au moyen de son poussoir ou suite à une mise sous tension qui a pour effet de provoquer un reset automatique, l'Arduino exécute une seule fois les instructions contenues dans la fonction **setup** () puis exécute ensuite indéfiniment les instructions contenues dans la fonction **loop** () de ce même programme. Ces fonctions ne retournent pas de résultat, elles sont donc déclarées avec le mot clé «**void**».

Selon le cas, une troisième partie peut ou non être présente dans un programme Arduino mais ne contient pas d'instructions exécutables. Il s'agit de la zone de définition de constantes au moyen du mot clé «**define**» ou de «**const**», d'inclusion d'éventuelles bibliothèques utilisées par le programme au moyen du mot clé «**include**», définition de variables, etc... Cette partie, si elle est présente, se place avant la fonction **setup** (), qui elle-même se place avant la fonction **loop** (). Tout comme le langage C, le langage arduino est composé de fonctions arithmétiques et mathématiques, des opérateurs de comparaison et opérateurs logiques, les structures de contrôle (prise de décision, boucles et sauts), des fonctions de gestion du temps et des entrées/sorties et beaucoup d'autres fonctions spécifiques à l'Arduino (gestion du port série, gestion d'interruption, générateur de nombre aléatoire, jouer des notes de musique, etc...).

Bibliographie et sites internet

[1]Tavernier, Christian. *Arduino Maîtrisez sa programmation et ses cartes d'interface (shields)*. Dunod, Paris, 2011.

[2]Dorseuil A. et Pillot P., édition Dunodcollection informatique industrielle, 1991

[3]Arduino-<https://www.arduino.cc> – site officiel de l'Arduino.

[4]Microchip - <http://www.microchip.com/> - site officiel de Microchip.

[5]Technologuepro<http://www.technologuepro.com/microprocesseur/introduction-systemes-base-microprocesseur.htm> - introduction sur les systèmes à base de microprocesseur.

[6][https://www.imhof-sew.ch/.../suiveurs de ligne](https://www.imhof-sew.ch/.../suiveurs_de_ligne)

[7]<https://www.imhof-sew.ch/.../servomoteurs>

[8] <http://home.scarlet.be/lestechiniques/ultrason.htm>

[9]<https://docs.microsoft.com/en-us/windows/desktop/bluetooth/bluetooth-reference>

[10]https://fr.wikipedia.org/wiki/Machine_courant_continu

[11]<http://www.alldatasheet.com/view.jspSearchword=L298N>

[12]https://www.gotronic.fr/art-carte-arduino_uno-2560-12421.htm

[13][http://www.academia.edu-projet-panneaux solaires...](http://www.academia.edu-projet-panneaux_solaires...)

[14]<http://www.techmania.fr-decouverte.arduino>

[15][http://www.academia.edu-navigation-regulateur de charge](http://www.academia.edu-navigation-regulateur_de_charge)

[16]<http://www.batterie.com>