

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. MIRA-BEJAIA



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Faculté des Sciences et de la Technologie

Département d'Automatique, Télécommunication et d'Electronique

Mémoire du Projet de Fin d'Etudes
Pour l'obtention du diplôme de

Master

En « Electronique »
Option « Instrumentation »

Thème :

*Etude et réalisation d'un robot
mobile autonome*

Réalisé par:

KHIARI Abdelmadjid
ABDERRAHMANE Ameer

Encadré par:

HANFOUG Salah

Présenté devant le jury composé de :

TAFININE Farid (P)
SADJI Mustapha

Année universitaire : 2021/2022

Remerciements



Nous remercions d'abord dieu le tout puissant pour nous avoir accordé la santé et le désir d'avancer le long de notre parcours universitaire jusqu'à achever ce modeste travail

Nous adressons de vifs et sincères des remerciements à nos parents qui ont su nous soutenir et nous encourager sans lesquels nous n'aurions pu arriver jusqu'à l'achèvement de ce mémoire

Nous tenons à exprimer notre profonde gratitude aux camarades eu à cotés ces dernières années et certains professeurs qui ont su nous motiver et attribuer leurs confiances

Nous voudrions également remercier les membres du jury pour l'intérêt qu'ils porteront à notre travail en l'examinant et en l'enrichissant par leurs propositions

Dédicaces



Nous dédions ce travail à :

Nos très chers parents.

Nos frères et sœurs.

Nos amis.

*Ainsi qu'à toute personne qui nous a soutenus.
Et toute personne passionnée par l'électronique.*

Table des Matières

Introduction générale	1
CHAPITRE I : SYSTEMES EMBARQUES ET MICROCONTROLEURS	3
I.1. Introduction	4
I.2. Historique	4
I.3. Notion de système embarque.....	4
I.3.1. Etapes de réalisation d'un système embarque	5
I.4. Définition d'un robot.....	5
I.5. Schéma synoptique du robot.....	6
I.6. Vue d'ensemble du robot qu'on veut concevoir	6
I.7. Technologie des ultrasons et leurs capteurs	6
I.8. Définition d'un microcontrôleur	7
I.9. Famille des microcontrôleurs	8
I.10. Généralités sur les microcontrôleurs PIC.....	9
I.10.1. Définition d'un microcontrôleur PIC	9
I.10.2. Avantages du PIC	9
I.10.3. Famille des PIC.....	10
I.10.4. Identification d'un PIC	10
I.10.5. Choisir un PIC	10
I.11. Présentation du microcontrôleur PIC16F877A	11
I.11.1. Structure externe	11
I.11.2. Structure interne.....	12
I.11.3. Les Ports du 16F877A.....	13
a. Le port A.....	13
b. Le port B.....	13
c. Le port C.....	13
d. Le port D.....	13
e. Le port E.....	14
I.11.4. Les timers	14
a. TIMER0.....	14
b. TIMER1.....	14
c. TIMER2.....	15
I.11.5. Le Convertisseur Analogique Numérique	15

I.11.6. Le module CCP.....	15
I.11.7. Les interruptions	16
I.12. Conclusion.....	16
CHAPITRE II : ETUDE DE LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET	17
II.1. Introduction	18
II.2. Schéma synoptique du projet.....	19
II.3. Partie matérielle	19
II.3.1. Microcontrôleur PIC16F877A.....	19
II.3.2. Capteur ultrason HC-SR04.....	20
II.3.3. Le Pont H L293D.....	22
II.3.4. Les Capteurs infrarouge (IR).....	24
II.3.5. Alimentation	25
II.3.6. Régulateur de tension 5V	26
II.3.7. Oscillateur à quartz	27
II.3.8. Moteurs à courant continu DC.....	28
II.3.9. L'afficheur LCD 16*2	29
II.4. Partie Logicielle.....	31
II.4.1. Plateforme de programmation MikroC	31
II.4.2. Création d'un projet sous MikroC.....	31
II.4.3. Présentation de la plateforme ISIS PROTEUS.....	33
II.4.4. Les librairies dans Proteus.....	34
II.4.5. Le PICkit 2 et son logiciel PICkit V2.61.....	34
II.4.6. Téléversement du code depuis le logiciel vers le PIC16F877A	35
II.5. Conclusion.....	38
CHAPITRE III : CONCEPTION ET REALISATION DU ROBOT.....	39
III.1. Introduction	40
III.2. Description de la base	40
III.3. Organigramme de l'évitement d'un obstacle.....	41
III.4. Circuit complet du robot.....	42
III.5. La communication entre le PIC 16F877A et les autres organes du robot.....	43
III.5.1. Interfacer le PIC avec le capteur ultrason HC-SR04	43
III.5.2. Interfacer le PIC avec les capteurs infrarouges	45
III.5.3. Interfacer le PIC avec le L293D	46
III.5.4. Le contrôle des moteurs par le L293D	47

III.5.5. Interfacer le PIC avec l'afficheur LCD	49
III.6. Finalisation de la réalisation :	50
III.7. Tests et problèmes rencontrés.....	51
III.8. Conclusion.....	51

Liste des figures

Figure I.1: Etapes de réalisation d'un système embarqué.....	5
Figure I.2: Schéma synoptique du robot.....	6
Figure I.3: Emission et réception d'ondes ultrasonores.	7
Figure I.4: Constituantes d'un microcontrôleur.....	8
Figure I.5: Le PIC16F877A et ses broches.	11
Figure I.6: Structure interne du PIC 16F877A.....	12
Figure II.1: Schéma synoptique du projet.....	19
Figure II.2: Microcontrôleur PIC16F877A.....	20
Figure II.3: capteur ultrason HC-SR04.....	20
Figure II.4: Calcul de distance par ultrason.....	21
Figure II.5: Circuit intégré L293D.....	22
Figure II.6: Les Broches du L293D.....	23
Figure II.7: Capteur de proximité IR.....	24
Figure II.8: Les broches du capteur IR.....	25
Figure II.9: Batterie 9V.....	25
Figure II.10: LM7805 et ses broches.....	26
Figure II.10: Quartz HC49 8MHz.....	27
Figure II.12: Moteur à courant continu.....	28
Figure II.13: Constituantes d'un moteur à courant continu.....	29
Figure II.14: Broches de l'afficheur LCD affichant "HELLO".....	30
Figure II.15: Programmation sous MikroC.....	31
Figure II.16: Etape 1 de la création d'un projet MikroC.....	31
Figure II.17: Etape 2 de la création d'un projet MikroC.....	32
Figure II.18: Etape 3 de la création d'un projet MikroC.....	32

Figure II.19: Environnement Proteus.....	33
Figure II.20: PICkit 2.....	35
Figure II.21: Accueil PICkit 2.....	36
Figure II.22: Code Hex importé.....	37
Figure II.23: Téléversement avec succès.....	37
Figure III.1: Base du robot.....	40
Figure III.2: Organigramme du robot.....	41
Figure III.3: Circuit du robot sous Proteus.....	42
Figure III.4: Branchement du capteur ultrason au PIC.....	43
Figure III.5: Le chronogramme du fonctionnement du HC-SR04.....	44
Figure III.6: Programmation du HC-SR04	44
Figure III.7: Branchement des capteurs infrarouge avec le PIC.....	45
Figure III.8: Interfacer le PIC avec le L293D et les moteurs.....	47
Figure III.9: Interfacer le PIC avec l’afficheur LCD.....	49
Figure III.10: Vue de différents angles du robot.....	50

Liste des tableaux

Tableau 1 : Commande du moteur 1.....48

Tableau 2 : Commande du moteur 2.....48

Introduction générale

La robotique est le point de croisement des sciences, l'intersection de la pensée et de la technologie. Le produit des idées innovantes et de la facilité pour offrir au monde des machines, des robots, des outils pour l'esprit pouvant substituer des actions humaines voir les améliorer. Ainsi ce secteur d'activité a su rassembler le savoir-faire de nombreuses disciplines pour faire avancer le monde.

Aujourd'hui les facultés de perception et de raisonnement des robots continuent de progresser. Plus impressionnant, l'innovation et les statistiques laissent indiquer que ces outils pourront être des pièces encore plus indispensables à l'avenir, de ce fait ils joueront un rôle majeur dans notre vie. Nous trouvons déjà la robotique presque dans tous les domaines ; nous pouvons facilement citer des domaines d'applications comme : La robotique industrielle, la robotique domestique, des transport, médicale, scientifique, militaire et sociale.

Ces systèmes alimentés en énergie évoluent dans des environnements statiques ou dynamiques, ils sont formés d'un microcontrôleur ainsi que d'un ou plusieurs capteurs et actionneurs. Leurs conceptions se basent sur l'étude et l'analyse de leurs comportements souhaitables. Principalement grâce aux théories d'asservissement et l'exécution de taches ordonnées par le biais de programmes informatiques.

Ce travail est centré autour de la réalisation d'un robot mobile pouvant dévier et éviter les obstacles rencontrés le long de son chemin, ainsi pour ce genre de robot l'autonomie est en plus d'une qualité une indispensabilité et un critère de choix pour son utilisateur. Imaginons un instant vouloir commander un robot loin à des milliers de kilomètres. Prenons exemple sur les robots sous-marins, ces derniers atteignent jusqu'à 6000m de profondeur voir plus pour les modèles récents plongeant pendant des années, d'un autre coté les robots explorateurs de planètes se situant à plus de 200 millions de kilomètres de la terre. En tenant compte de ces exemples il serait facile de comprendre les temps de retard que peuvent prendre les signaux de commande envoyés, les informations à la réception qui serait envahi par les bruits et les parasites. On comprend nettement mieux la nécessité de s'auto-commander pour ce genre de robot.

Pour s'auto-commander, un robot doit d'abord avoir plusieurs aptitudes ; comme un humain possède des sens physiologique (la vue, l'odorat...) un robot doit lui aussi être capable de percevoir l'environnement dans lequel il aura été introduit, arriver à mesurer sa localisation, connaître ses limites et se baser sur tout cela pour savoir prendre les bonnes décisions. Quand on arrive à concevoir des systèmes aussi intelligents il

est impératif de les protéger et les préserver le plus longtemps possible et donc les faire éviter des collisions qu'ils peuvent rencontrer au cours de leurs routes, de ce fait il faut savoir les équiper de capteurs qui sauraient les orienter en mesurant les distances qui les séparent des obstacles et sur des angles différents. D'ailleurs, ces systèmes disposent de capacités d'actions, il n'est pas suffisant de détecter une possibilité de collision pour l'éviter, il est aussi nécessaire de passer des commandes pour y parvenir.

Le but de ce modeste travail est d'étudier et de réaliser un robot mobile autonome pouvant rouler en évitant tout obstacle pouvant être rencontré. Son autonomie réside dans le fait qu'il n'y a pas de besoin de le commander à distance, l'équipant ainsi d'organes de perception et de déplacement.

Ce mémoire est structuré de la manière suivante:

Au premier chapitre, on parlera de généralité sur les systèmes embarqués et les microcontrôleurs, on introduira le PIC 16F877A de Microchip Technology.

Le deuxième chapitre est formé de deux parties, la première est consacrée à l'étude et le choix des organes qui sont les composants requis pour la conception du robot, quant à la deuxième partie elle est consacrée à l'étude des logiciels requis pour la programmation et la simulation des résultats.

Pour terminer, le troisième chapitre où l'on étudiera la conception théorique (la manière dont on a interfacé chaque organe avec le microcontrôleur) ainsi que la réalisation pratique.

CHAPITRE I :

Systemes embarqués et microcontrôleurs

I.1. Introduction

Le robot que l'on veut concevoir n'est simplement qu'un système embarqué, ces derniers sont présents dans notre vie quotidienne et y jouent un rôle important. Ils nous offrent une meilleure communication à distance, un meilleur confort et plus de sécurité dans nos foyers et un déplacement plus rapide.

De ce fait dans ce chapitre nous parlerons des systèmes embarqués en général et des étapes à suivre pour les réaliser.

Comme nous désirons qu'il soit commandé par un microcontrôleur, nous expliquerons comment choisir ce dernier et ses avantages. Tout en abordant le concept de détection d'obstacle par les technologies ultrasons.

I.2. Historique

Le terme « robot » a vu le jour un certain janvier de l'an 1921 par un jeune écrivain tchèque nommé Karel Čapek dans sa pièce de théâtre RUR (Rossum's Universal Robots). Ce terme, provenant du tchèque robot qui signifie « travail forcé », désigne à l'origine une machine ayant la capacité de remplacer l'humain dans toutes ses tâches. Ainsi dans les années 1940 et 1950, les progrès de l'électronique ont permis de miniaturiser les circuits électriques notamment grâce à l'invention du transistor et du circuit intégré, ces progrès ont pu élargir les voies pour la fabrication des robots. Au prémices de la robotique, ces machines étaient considérées comme des imitations par rapport à leurs visées qui étaient centrées dans la reproduction des tâches humaines pendant un ou plusieurs cycles. Quant à aujourd'hui, les constructeurs ne tentent plus de reproduire l'aspect humain sur un robot, privilégiant avant tout sa fonctionnalité tout en visant à produire des aspects que même un humain ne peut réaliser grâce au développement de l'intelligence artificielle et la mise en point des réseaux de neurones...[1]

I.3. Notion de système embarqué

Un système embarqué est la résultante d'une combinaison d'un système électronique et informatique qui sera spécialisé dans l'exécution d'une ou plusieurs tâches bien précises.

Ainsi, le qualificatif d'embarqué leur a été attribué pour leurs intégrations dans des systèmes bien plus larges.

Cependant, ils peuvent rencontrer des contraintes variées et présenter des ressources limitées.

En résumé, c'est la fusion du matériel et du logiciel pour interagir avec l'environnement.

I.3.1. Etapes de réalisation d'un système embarqué

La réalisation d'un système embarqué, se déroule suivant un certain nombre de procédés et d'étapes que nous pouvons présenter dans l'organigramme ci-dessous:

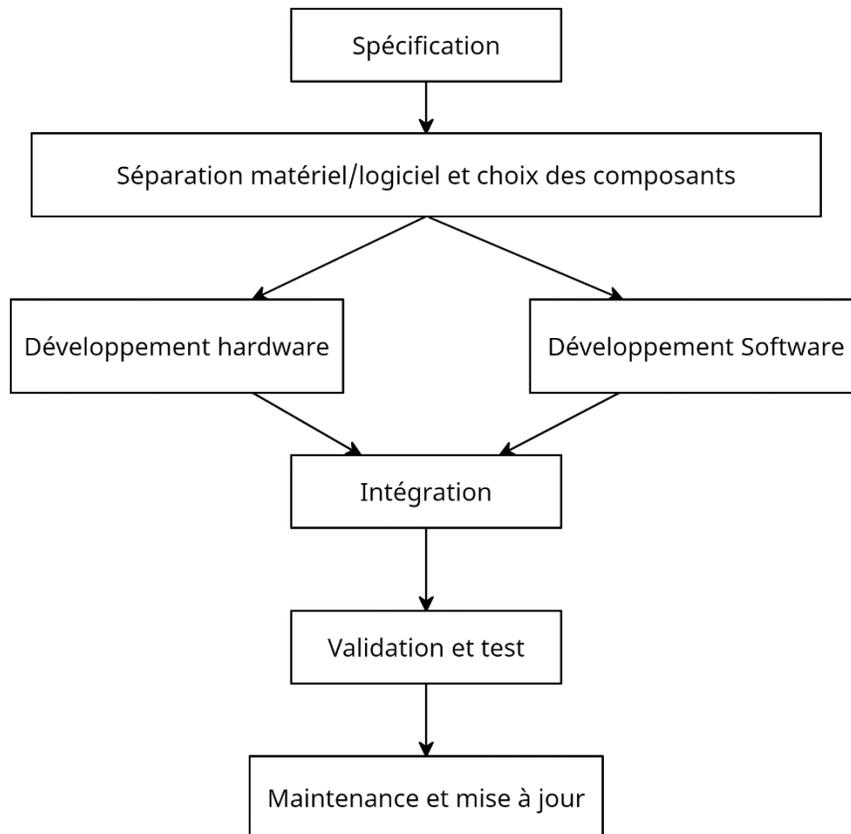


Figure I.1 : Etapes de réalisation d'un système embarqué

I.4. Définition d'un robot

Les robots sont des alliances entre la mécanique, l'électronique et l'informatique ; Ce sont donc des dispositifs "Mécatroniques". Ils sont conçus pour mettre en exécution les tâches demandées, ces dernières peuvent être pénibles, répétitives, voir risquées pour les humains. Tout cela de façon automatique et organisée.

I.5. Schéma synoptique du robot

Dans la figure ci-dessous nous vous présentons un schéma synoptique du robot que l'on veut concevoir. [2]

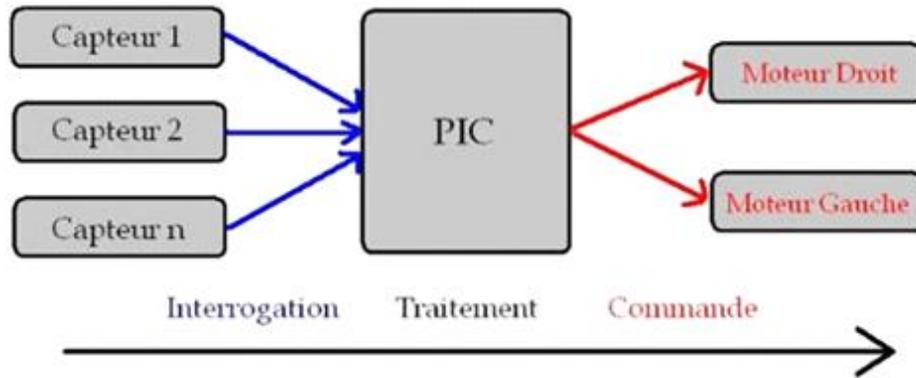


Figure I.2: Schéma synoptique du robot

I.6. Vue d'ensemble du robot qu'on veut concevoir

En entrée des capteurs enregistrant des données extérieures physiques reliés à un microcontrôleur ayant pour rôle le traitement de ces données pour commander en sortie des actionneurs (les moteurs) après avoir fait les calculs nécessaires.

I.7. Technologie des ultrasons et leurs capteurs

Le phénomène consiste à émettre et recevoir des ondes ultrasonores. Ces dernières sont bien évidemment inaudibles et impossible à être perçues par l'oreille humaine, du fait qu'elles soient supérieures à 20Khz. Ainsi, elles ne peuvent déranger les utilisateurs de ce genre de robot. L'émetteur et le récepteur (à ultrasons) sont placés côte à côte ; Le premier génère une onde ultrasonore. En présence d'un obstacle, une partie de l'onde est réfléchi vers le récepteur. En d'autres termes, L'émetteur envoie un train d'ondes qui est réfléchi sur l'obstacle pour ensuite revenir au capteur. Le temps mis pour parcourir un aller-retour permet de déterminer la distance de l'obstacle par rapport à la source. D'ailleurs, la distance trouvée doit être divisée par deux, car le signal aura parcouru le double de la distance qui les sépare pour revenir à l'émetteur (Aller-

Retour). La capacité de détection de ce type de capteur va de quelques centimètres à environ 5 mètres et voir plus.[3]

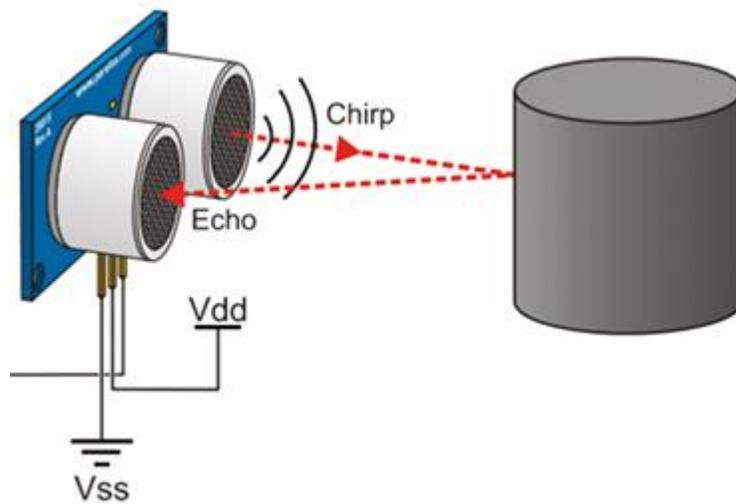


Figure I.3 : Emission et réception d'ondes ultrasonores

I.8. Définition d'un microcontrôleur [4]

Un microcontrôleur est un circuit intégré rassemblant les éléments essentiels d'un ordinateur. Ce n'est qu'une dérivation du microprocesseur, ainsi sa structure est semblable à celle de ce dernier. D'ailleurs, il est composé en plus de l'unité centrale de traitement, d'une mémoire (RAM et ROM), et d'une (ou plusieurs) interface de communication avec l'extérieur matérialisé par les ports d'entrée/sortie.

Comparé à des systèmes électroniques à base de microprocesseurs, les systèmes à microcontrôleurs permettent la diminution en taille, en consommation et en cout. C'est pour cela qu'on peut facilement les trouver présents presque partout, dans l'électroménager, les jouets, la téléphonie mobile....

Quoique les microprocesseurs demeurent beaucoup plus robustes et sont plutôt réservés pour les applications demandant beaucoup de traitement de l'information et des calculs bien complexes. Les microcontrôleurs quant à eux sont plutôt dédiés aux applications qui ne nécessitent pas une grande quantité de calculs complexes, mais qui demandent beaucoup de manipulations d'entrées/sorties.

De plus, ils sont dotés de circuits d'interface qui vont dépendent du microcontrôleur choisi à savoir, les convertisseur analogique/numérique (CAN) intégré, gestion d'une liaison série ou parallèle, un Watch dog (surveillance du programme), une sortie PWM (modulation largeur d'impulsion), les systèmes de comptage (TIMER).

CHAPITRE I : SYSTEMES EMBARQUES ET MICROCONTRÔLEURS

Voici une figure qui peut illustrer tout ce qu'on vient de décrire

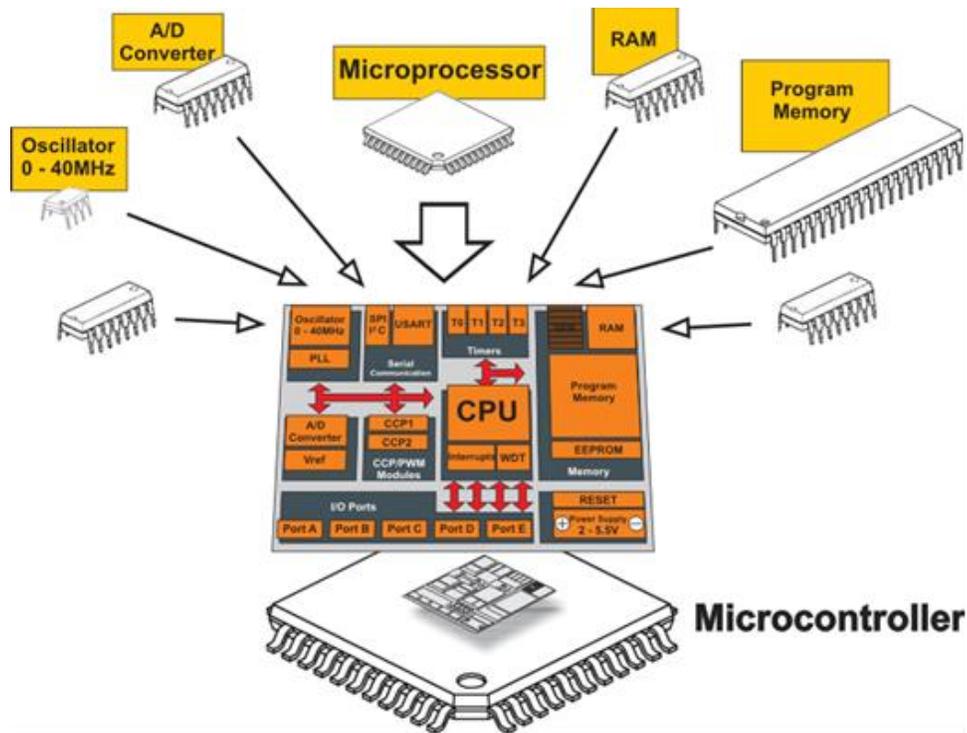


Figure I.4 : Constituantes d'un microcontrôleur

I.9. Famille des microcontrôleurs

Il existe plusieurs familles de microcontrôleurs dont les plus connues sont:

Microchip: PIC; familles 12Cxxx, 16Cxxx, 16Fxxx, 18Fxxx, ...

STMicroelectronics: famille STX

Motorola: famille 68HCxxx, ...

Atmel: AT; familles AT89Sxxxx, AT90xxxx, ...

Analog Devices: famille ADuC

Nous allons nous intéresser dans le cadre de ce projet à la famille Microchip PIC (Programmable Integrated Circuit) de moyenne gamme (MIDRANGE). Pour sa disposition au niveau de nos universités et nos acquis à travers le module "Microprocesseurs Et DSP" qui était centrée sur l'étude des microcontrôleurs PIC 16F.

I.10. Généralités sur les microcontrôleurs PIC [5]

I.10.1. Définition d'un microcontrôleur PIC

La société Américaine Microchip Technologie a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (Periphrol Interface contrôler).

Le PIC, est une unité de traitement d'information à laquelle on a ajouté des périphériques internes dans le but de faciliter l'interfaçage avec le monde extérieur sans avoir recours à ajouter des composants externes. Les Pics sont des composants "RISC" (Reduced Instructions Set Computing) ou encore composant à jeu d'instructions réduit où chaque instruction complexe peut être programmée par plusieurs instructions simples. Moins on a d'instructions, plus aisée et rapide sera le décodage, par conséquent un traitement plus rapide.

I.10.2. Avantages du PIC

- Le circuit imprimé est plus simple à tracer vu qu'il n'est plus nécessaire de véhiculer des bus d'adresse d'un composant à un autre.
- Système cohérent et plus fiable
- Moins de consommation en énergie par rapport aux autres microcontrôleurs
- Programmation plus simple.
- L'interfaçage d'un appareil analogique est facile sans aucun circuit supplémentaire.
- Moins cher que les composants semblables.
- Réalisation d'applications plus complexes que les microcontrôleurs semblables ne peuvent pas réaliser.
- Une diminution au niveau de l'encombrement du matériel et du circuit imprimé.

I.10.3. Famille des PIC

La famille des PIC est subdivisée en 3 autres familles qui sont :

- Base- line : c'est une famille qui utilise des mots d'instructions de 12 bits.
- Mid- range : c'est une famille qui utilise des mots de 14 bits (dont font partie les 16F84,16f876et 16f877).
- High- end : c'est une famille qui utilise des mots de 16 bits.

I.10.4. Identification d'un PIC

Il existe un code permettant d'identifier et de reconnaître le PIC, ce code est de la forme :
xx(L) XXyy-zz où :

- **xx** : indique la famille du PIC (12, 14, 16, 18)
- **L** : indique qu'il fonctionne avec une plage de tension beaucoup plus tolérante.
- **XX** : type de mémoire, peut être soit de type : C, CR ou F.

C : indique que la mémoire programme est un EPROM ou une EEPROM.

CR ou **F** : indique le type de mémoire ; CR(ROM) ou F (FLASH).

- **yy** : indique le modèle du PIC (son identité)
- **zz** : la fréquence d'horloge maximale supportée par le PIC.

I.10.5. Choisir un PIC

Le choix du PIC réside principalement dans l'application souhaitable à réaliser, par ailleurs il est primordial de prendre en considération les critères suivants :

- Le type de mémoire utilisée RAM, EPROM, FLASH et de sa taille.
- L'existence ou non d'une mémoire EEPROM.
- Le Nombre d'entrées/sorties pour l'application.
- La disponibilité du composant sur le marché.
- Son prix d'achat.

I.11. Présentation du microcontrôleur PIC16F877A

Le pic 16F877A est un circuit intégré contenu dans un boîtier de type « DIL 40 », il présente 40 broches, 20 de chaque côté, les broches sont virtuellement numérotées de 1 à 40 où la première broche est placée dans le coin situé à gauche de l'encoche de repérage. Le PIC 16F877-A peut supporter une fréquence de 20MHZ maximum.

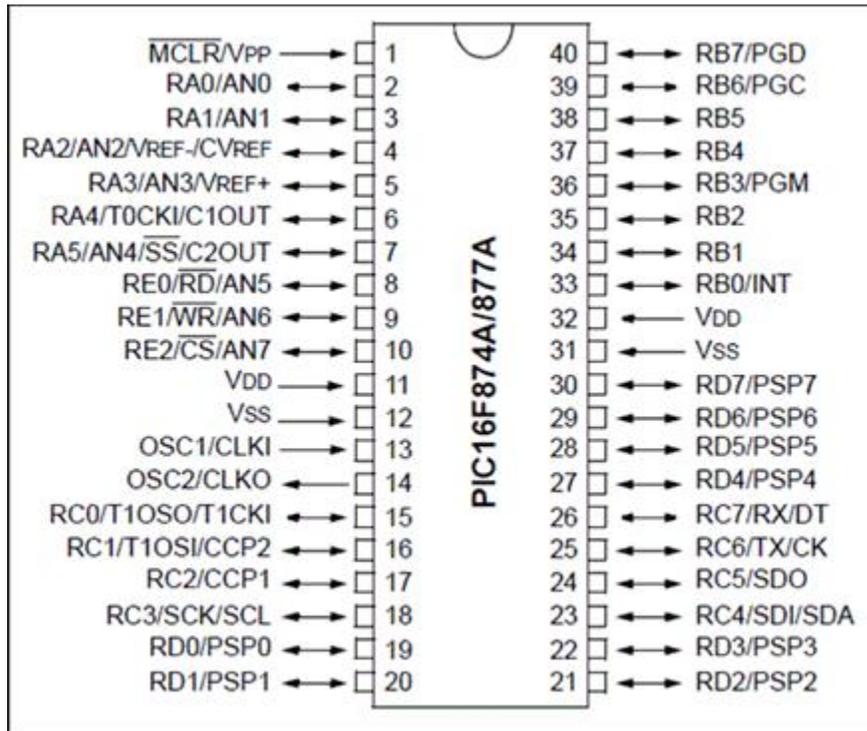


Figure I.5 : Le PIC16F877A et ses broches

I.11.1. Structure externe

Le schéma de la figure I.6 ci-dessus nous fait part de la structure externe du PIC 16F877-A Ainsi on voit qu'il dispose de :

- Quatre pins pour l'alimentation : deux connexions VDD et deux connexions VSS.
- Deux pins pour l'oscillateur (broches OSC1 et OSC2 ou CLKIN et CLKOUT), ils permettent de le faire fonctionner (l'oscillateur) soit l'interne contenu à l'intérieur PIC soit l'externe qui peut être un quartz, un oscillateur RC, un résonateur céramique
- RESET(MCLR) : qui sert à initialiser le microcontrôleur en cas de la mise sous tension et remise à zéro externe.
- Les pins restants sont les entrées/ sorties on en compte ainsi 33.

I.11.2. Structure interne[6]

À l'intérieur du 16F877A on trouve des éléments d'une structure à base d'un microprocesseur réunis dans un seul circuit intégré et ces éléments sont :

- Une mémoire de donnée (RAM et EEPROM)
- Une mémoire de programme (FLASH, ROM, OTPROM, UVPROM ou EEPROM)
- Un générateur d'horloge
- Un chien de garde
- 13 sources d'interruptions
- Des périphériques (compteur/timer, convertisseurs analogiques/numériques (10 bits) à 8 canaux, etc.).
- Une interface parallèle pour la connexion des entrées/sorties
- Une interface série (synchrone ou asynchrone)
- Des bus de données et d'adresse

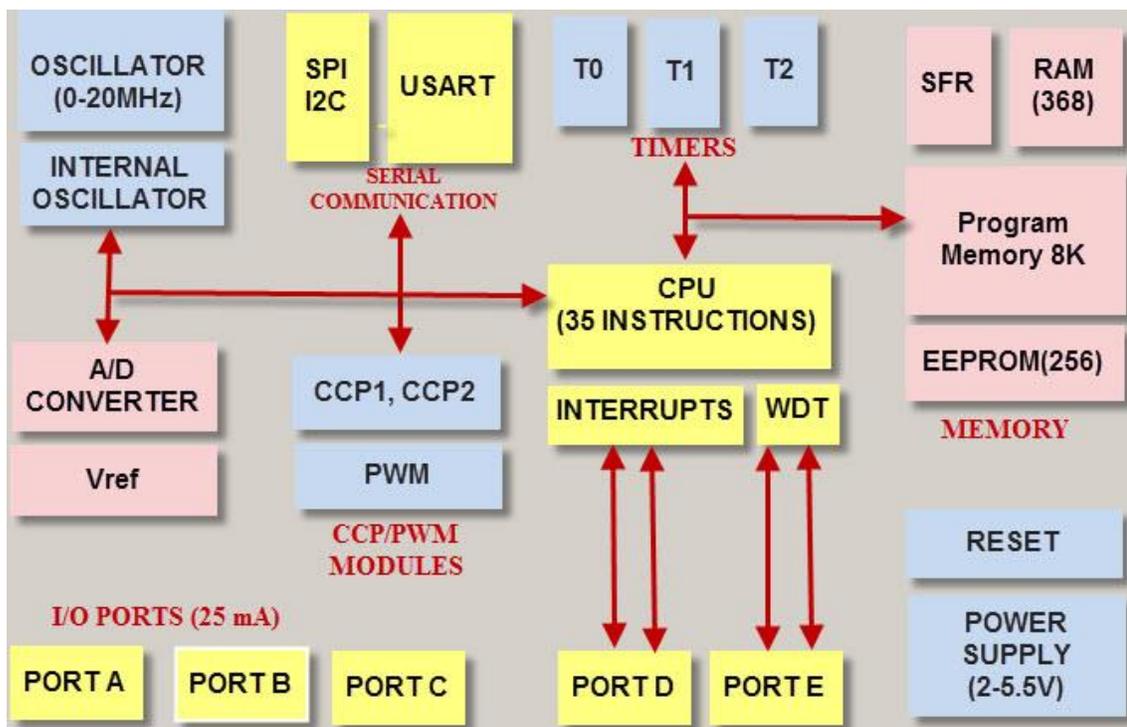


Figure I.6 : Structure interne du PIC 16F877A

I.11.3. Les Ports du 16F877A

a. Le port A

Ce port possède 6 pins d'entrées/sorties de RA0 à RA5, les 5 pins (de RA0 à RA3) et RA5 peuvent être utilisés comme des entrées sorties numériques, soit comme entrées analogiques. La broche RA4 peut être utilisée :

- Entrée/sortie numérique.
- Comme entrée horloge pour le timer TMR0.

Si le registre TRISA=1, les pins se placent en entrées.

Si le registre TRISA=0, les pins se placent en sorties.

b. Le port B

Il possède 8 pins d'entrées sorties de RB0 àRB7.

RB0 : peut-être entrée d'interruption externe

RB1 : entrée de l'USART.

RB2 : sortie ou horloge de l'USART.

RB3 : E/S du module capture/compare /PWM.

RB4 : contrôle du mode de programme.

RB6 : E/S du timer 1.

RB7 : entrée d'horloge pour timer 1.

c. Le port C

C'est un port bidirectionnel à 8 bits de RC0 à RC7. Toutes ces broches peuvent être utilisées soit comme entrées/sorties ou comme broches d'accès aux différentes ressources internes (USART, CCP, Timer 1), à l'aide du registre TRISC.

d. Le port D

C'est un port bidirectionnel à 8 bits qui ne partage avec aucune ressource interne ce port dispose d'un mode de fonctionnement particulier qui s'appelle le mode esclave parallèle (PSP).

e. Le port E

Ce port contient seulement 3 bits de RE0 à RE2, qui ne sont pas concernés de TRISE, ils sont implémentés pour d'autres fonctions.

- Les pins REX peuvent être utilisés avec la ressource PSP.
- Les pins peuvent être utilisés comme entrées analogiques.

I.11.4. Les TIMERS

a. TIMER0

Dans le passé, le TIMER0 s'appelait RTCC. C'est un compteur 8 bits (0 à 255) simple, qui compte des impulsions soit internes, soit d'une source externe. On peut par ailleurs lui appliquer une pré division programmable entre 1 et 256. On peut librement lire ou écrire dans le registre de comptage associé. On peut donc le pré charger avec une valeur, à partir de laquelle il comptera jusqu'à atteindre 255.

b. TIMER1

Le Timer1 fonctionne sur le même principe que le TIMER0, mais avec un registre de comptage plus gros : 16 bits au lieu de 8, ce qui étend notablement ses capacités de comptage. De plus, il possède un mode de fonctionnement particulier : on peut l'utiliser en association avec un des modules CCP (modules de capture et de comparaison, voir plus loin). Voyons rapidement le mode « capture » : lorsqu'un événement survient sur l'entrée du module CCP, la valeur du TIMER1 est lue.

- Le mode Timer : Dans ce mode, TMR1 est incrémenté par l'horloge système $F_{osc}/4$ éventuellement pré-divisée. Le bit de synchronisation n'a pas d'effet car l'horloge $F_{osc}/4$ est toujours synchronisée sur l'horloge système.
- Le mode Compteur : Dans ce mode, TMR1 est incrémenté à chaque front montant de l'horloge externe T1CKI (RC0) ou l'horloge dédiée générée par l'oscillateur T1OSC à condition de positionner le bit T1OSCEN à 1 et de brancher un quartz entre les broches RC0 et RC1. En mode compteur, RC0 et RC1 sont automatiquement configurées en entrée, on n'a pas besoin de configurer les bits TRISC.

c. TIMER2

Le Timer2 a un fonctionnement différent des Timer0 et Timer1. C'est un compteur 8 bits avec pré-diviseur et post-diviseur. On s'en sert pour générer des signaux carrés, ou, en association avec le module CCP, des signaux PWM. PWM étant l'acronyme de « Pulse Width Modulation » ou, en français, Modulation de Largeur d'Impulsion (MLI).

I.11.5. Le Convertisseur Analogique Numérique

ADC = Analog to Digital Converter = convertisseur analogique - numérique (CAN en français).

Le rôle du convertisseur analogique - numérique est de transformer une tension analogique en un nombre binaire (proportionnel à la tension analogique). Le microcontrôleur PIC 16F877 a l'avantage de posséder un module ADC interne, contrairement au PIC 16F84 (il faut alors adjoindre une ADC externe, ce qui complique le schéma électrique et la programmation du PIC).

Le convertisseur A/D convertit le signal analogique présent sur une de ses 8 entrées en son équivalent numérique, codé sur 10 bits. Les pattes AN2 et AN3 peuvent être utilisées comme références de tension ou comme entrées analogiques standard, les références de tension étant dans ce dernier cas prises sur les tensions d'alimentations du PIC : VDD et VSS. (VDD pour le + et VSS pour le -).

I.11.6. Le module CCP

Ces deux modules (ccp1, ccp2) peuvent fonctionner dans l'un des trois modes ci-dessous :

- Mode capture « CAPTURE » : ce mode permet en outre d'effectuer des mesures de temps.
- Mode comparaison « COMPARE » : ce mode permet de capturer la valeur du TIMER1 quand un évènement se produit sur RC1 ou RC2. Un évènement est défini comme : un front montant, un front descendant.
- Mode PWM signifie « Pulse Width Modulation », ce qu'on pourrait traduire par modulation de largeur d'impulsion, il s'agit d'un signal binaire de fréquence fixe dont le rapport cyclique peut être modulé par logiciel. Le rapport cyclique d'un signal binaire à fréquence fixe peut être défini comme étant le rapport entre le temps où il se trouve à l'état «1 » par rapport au temps total d'un cycle.

Ces modules sont associés aux broches RC2/CCP1 et RC1/T1OSI/CCP2 suivant le mode choisi.

I.11.7. Les interruptions

Une interruption provoque l'arrêt du programme principal pour aller exécuter une procédure d'interruption. A la fin de cette procédure, le microcontrôleur reprend le programme principal à l'endroit où il l'a laissé. A chaque interruption sont associés deux bits, un bit de validation et un drapeau. Le premier permet d'autoriser ou non l'interruption, le second permet au programmeur de savoir de quelle interruption il s'agit. Ses deux bits se mettent à 1 lorsque l'événement déclencheur d'une interruption intervient et que cette dernière est validée.

Le microcontrôleur dispose de plusieurs sources d'interruptions

- Une interruption externe, action sur la broche INT/RB0.
- Débordement du TIMER0.
- Changement d'état logique sur une des broches du PORTB (RB4 à RB7).
- Une interruption d'un des périphériques (PEIE).
- Fin de programmation d'une case mémoire de l'EEPROM.
- Changement d'état sur le PORTD (PSPIE)
- Fin de conversion analogique numérique (ADIE).
- Interruption du registre de capture et/ou de comparaison 1 (CCPI1E).
- Interruption du registre de capture et/ou de comparaison 2 (CCPI2E).

I.12. Conclusion

Dans ce chapitre, nous avons décrit ce qu'est un robot, en passant par la définition des systèmes embarqués, leurs caractéristiques principales et leurs étapes de conception.

Par ailleurs nous avons pu aborder et décrire la visée du projet en centrant le chapitre autour du microcontrôleur et particulièrement le microcontrôleur PIC 16F877A de Microchip Technology dû au fait que ce dernier représente le cerveau du robot et le commandant de ses organes. On a d'ailleurs parler de ses constituantes internes et de ses 40 broches.

Après avoir vu les systèmes embarqués et l'importance des microcontrôleurs, il nous reste plus qu'à passer au choix des composants requis et des plateformes de travail pouvant offrir une avant-première à travers leurs outils de simulation.

CHAPITRE II :

**ETUDE DE LA PARTIE
MATERIELLE ET
LOGICIELLE DU
PROJET**

II.1. Introduction

Nous estimons que la plupart des gens, des établissements et des industries s'intéressent de plus en plus à la robotique mobile.

Nous aimerions réaliser un robot dont la plage d'utilisation pourrait être variée ; Il est simple à comprendre que nous essayons de mettre en œuvre une petite machine circulante et évitant les obstacles, mais imaginons un instant l'ampleur des fonctions que peut faire ce robot rien qu'on ajoutant un simple autre composant. Car en électronique on ne vend pas que des machines mais on vend aussi ce qu'elles peuvent réaliser. Par exemple, si on ajoutait un détecteur de gaz, les gens dormiraient mieux et seront plus à l'aise en mettant en marche le robot toute une nuit et à la moindre fuite de gaz il saurait la détecter et par ailleurs en fera signe par un buzzer ou directement vers un mobile à travers un module Bluetooth.

Par ailleurs, nous pouvons aussi fusionner la base avec un aspirateur, de ce fait on aurait un aspirateur intelligent qui saura nettoyer le sol sans que l'on dépense la moindre calorie. Ainsi la base du système que nous étudions devient nettement plus intéressante.

Ce chapitre est décomposé en deux parties dont une première consacrée à l'étude des composants et la définition du matériel nécessaire pour la réalisation des taches précises que l'on veut à partir d'un microcontrôleur pic16F877 aux commandes. Quant à la seconde, comme les systèmes embarqués sont une alliance entre l'électronique et l'informatique, elle sera consacrée au software en parlant des logiciels utilisés, de leurs environnements et leurs façons de travail.

II.2. Schéma synoptique du projet

Voici un Schéma synoptique du projet :

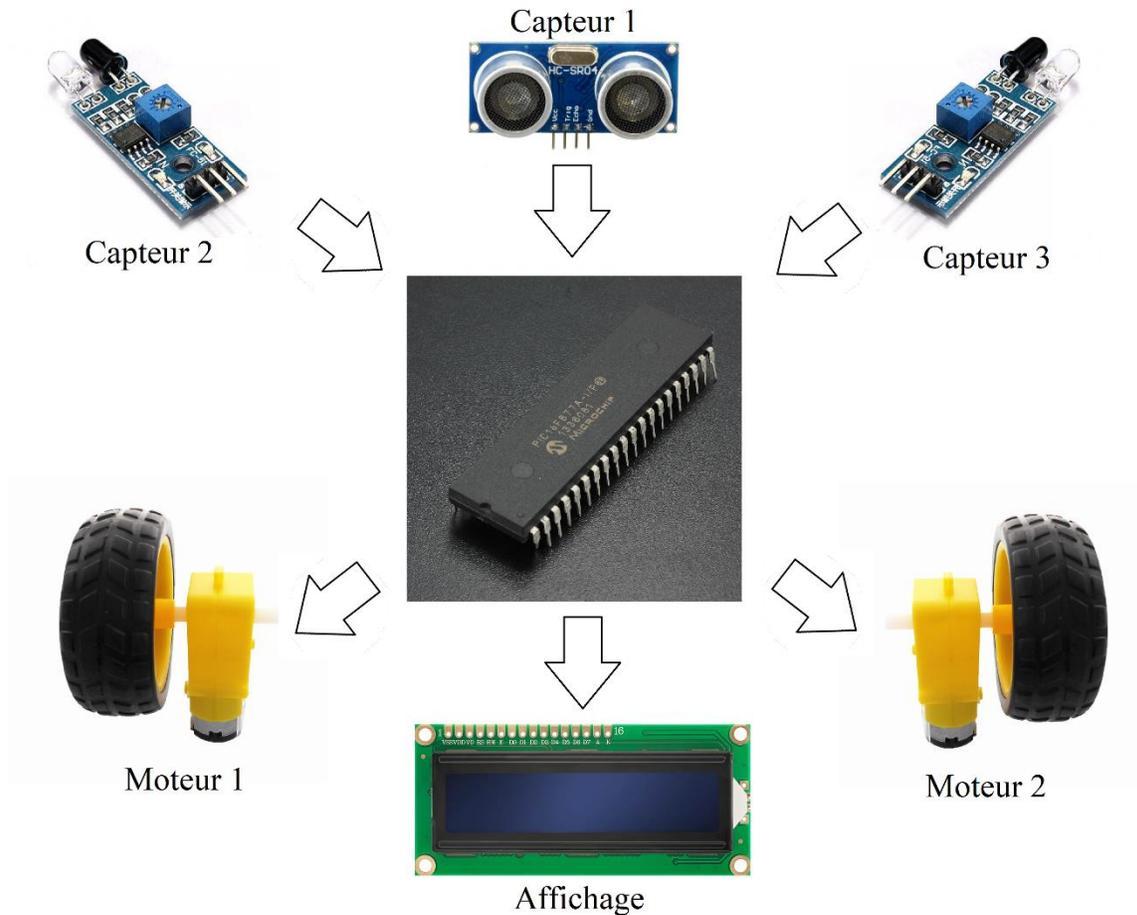


Figure II.1 : Schéma synoptique du projet

II.3. Partie matérielle

II.3.1. Microcontrôleur PIC16F877A

Bien exposé au chapitre I, le microcontrôleur PIC16F877A possède les critères suffisants pour amener à bien la réalisation de notre projet. Il a donc été évident de le choisir, déjà pour sa disposition sur le marché et notamment pour les nombreux compteurs, les modules CCP pouvant servir à la variation de vitesse...



Figure II.2 : Microcontrôleur PIC16F877A

II.3.2. Capteur ultrason HC-SR04

Le capteur de distance à ultrasons HC-SR04 se compose en fait de deux transducteurs à ultrasons. L'un agit comme un émetteur qui convertit le signal électrique en impulsions sonores ultrasonores de 40 KHz inaudible pour les humains, quant à l'autre, il agit comme un récepteur et écoute les impulsions transmises.

Lorsque le récepteur reçoit ces impulsions, il produit une impulsion de sortie dont la largeur est proportionnelle à la distance de l'objet devant.



Figure II.3 : capteur ultrason HC-SR04

CHAPITRE II : ETUDE DU LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET

Ce capteur offre une excellente détection de plage sans contact entre 2 cm et 400 cm (~ 13 pieds) avec une précision de 3 mm. Ce capteur dispose de 4 broches qui sont :

VCC : alimente le capteur à ultrasons HC-SR04.

Trig : (déclencheur) cette broche est utilisée pour déclencher des impulsions ultrasonores. En réglant cette broche sur HIGH pendant 10µs, le capteur initie une rafale ultrasonique.

Écho : Cette broche passe au niveau haut lorsque l'impulsion est transmise et reste au niveau haut jusqu'à ce que le capteur reçoive un écho, après quoi elle passe au niveau bas. En mesurant le temps pendant lequel la broche Echo reste haute.

GND : la mise a terre du HC-SR04

La distance peut être calculée par la loi : Distance (cm) = Vitesse du son (cm / µs) × Temps (µs) / 2.

Voici une figure illustrant le calcul de la distance séparant le capteur de l'objet :

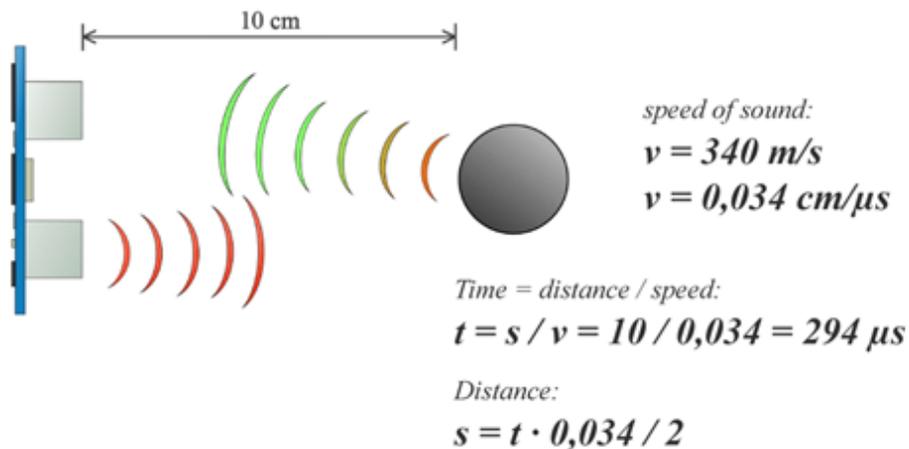


Figure II.4: Calcul de distance par ultrason[7]

II.3.3. Le Pont H L293D

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux. En raccordant les sorties de façon appropriées, il est possible de constituer deux pont-h. Il est ainsi possible de commander deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.



Figure II.5: Circuit intégré L293D

Le pont H est un assemblage de 4 transistors monté de telle façon que le courant puisse passer soit dans un sens, soit dans l'autre au travers de la charge (un moteur continu par exemple). En inversant le sens du courant dans le moteur, ce dernier changera de sens de rotation à partir du microcontrôleur nous ne pouvons pas connecter un moteur directement car le premier ne peut pas donner suffisamment de courant pour entraîner les moteurs CC. Le pilote de moteur est un dispositif d'amélioration du courant, il peut également agir comme dispositif de commutation en commandant les moteurs ainsi que le sens de rotation. Par conséquent, nous insérons le pilote de moteur entre le moteur et le microcontrôleur.

Le pilote du moteur prend les signaux d'entrée du microcontrôleur et génère la sortie correspondante pour le moteur. Le L293D est l'un des pilotes les plus populaires sur le marché. Il y a plusieurs raisons qui font du L293D le pilote préféré des utilisateurs, telles que le prix bon marché (par rapport aux autres pilotes),

CHAPITRE II : ETUDE DU LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET

la forme et la taille appropriées, le contrôle facile.... Ce circuit intégré peut configurer des moteurs avec une tension comprise entre 5V et 36V et un courant allant jusqu'à 600 mA. 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 microcontrôleur). Il peut piloter des charges inductives comme des relais solénoïdes, moteurs continus et moteurs pas-à-pas.

Caractéristiques et spécifications

- Le L293D est conçu pour fournir des courants bidirectionnels allant jusqu'à 600 mA à des tensions de 4,5 V à 36 V
- Large plage d'alimentation de 4.5V à 36V
- Courant de sortie 600mA par canal
- Il dispose de 16 Broches dont deux broches d'alimentation VS pour le circuit interne et VSS pour les moteurs

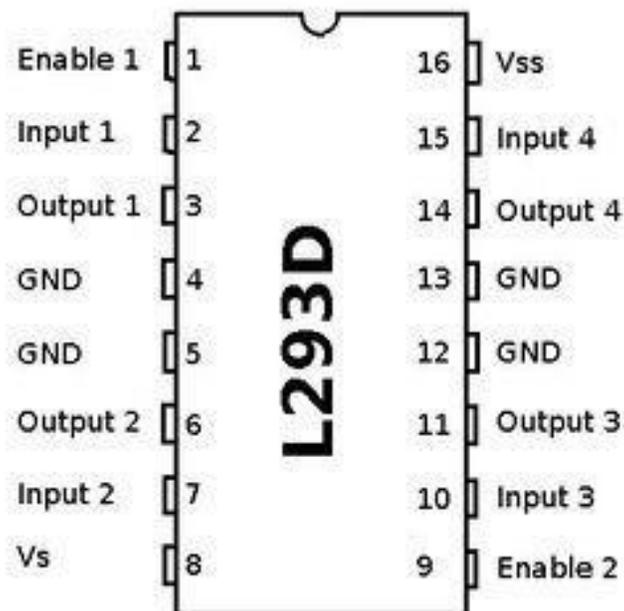


Figure II.6: Les Broches du L293D

II.3.4. Les Capteurs infrarouge (IR) FC-51

IR en abrégé infrarouge détecte la présence d'un objet en émettant un faisceau de lumière infrarouge. Il fonctionne de la même manière que les capteurs à ultrasons, bien qu'au lieu d'utiliser des ondes sonores, l'IR est transmis. Ils se composent d'une LED IR qui fait l'émission et d'un détecteur de lumière pour la détection de la réflexion. Lorsqu'un objet s'approche du capteur, la lumière infrarouge de la LED se réfléchit hors de l'objet et est détectée par le récepteur. Les capteurs IR actifs agissent comme des capteurs de proximité, et ils sont couramment utilisés dans les systèmes de détection d'obstacles (comme dans notre cas le robot).



Figure II.7 : Capteur de proximité IR FC-51

Avantage des capteurs de proximité IR

- Détection sans contact
- Application pour les utilisations diurnes et nocturnes
- Capable de mesurer la distance aux objets mous contrairement aux capteurs de proximité à ultrasons

Inconvénient des capteurs IR

- Baisse de performance sur de plus longues distances (Plage de détection : 2cm ~ 30cm)
- Affecté par les conditions environnementales

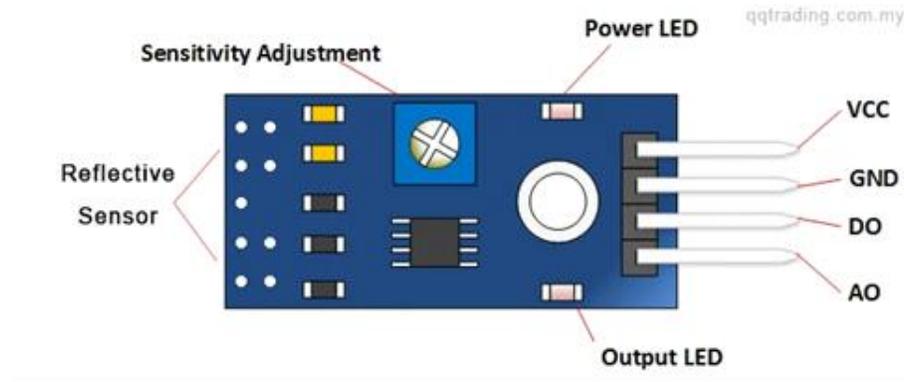


Figure II.8 : Les broches du capteur IR

VCC : Alimentation
GND : Terre
DO : Valeur numérique
AO : Valeur analogique

II.3.5. Alimentation

L'alimentation est l'une des étapes les plus importantes et les plus sensibles, en effet un mauvais branchement, un mauvais choix de tension d'alimentation peut amener à la destruction des composants électroniques. Un microcontrôleur n'a besoin que d'une tension de 5V pour bien réaliser ses tâches. Cependant, certains actionneurs ont besoin d'une tension supérieure pour bien fonctionner. Il a donc été convenu d'alimenter le circuit par une tension de 9V par une simple batterie et puis de régulariser cette dernière par un régulateur de tension de 5V pour le pic et les autres composants pouvant bien travailler sous cette dernière



Figure II.9 : Batterie 9V

II.3.6. Régulateur de tension 5V [8]

Le régulateur de tension positive permet de réguler une tension d'entrée de 7V à 35V en une tension de sortie à 5V 1A, cette tension de sortie demeurera fixe et constante même si la tension d'entrée subit des modifications.

Le cœur du LM7805 est un transistor contrôlant le courant entre l'entrée et la sortie et ainsi donc la tension à la sortie, pour arriver vers ce transistor il y'a une certaine hiérarchie suivie, un circuit de démarrage fournit un courant initial à une bande appelée bande interdite, cette dernière a pour mission la protection contre la surchauffe, contre une tension d'entrée excessive et un courant de sortie excessif. Un diviseur de tension réduit la tension pour une utilisation par la référence de la bande interdite, cette dernière donne un signal d'erreur qui est amplifié par un amplificateur d'erreur, ce signal contrôle le transistor de sortie (Cœur) et cela ferme la boucle de rétroaction négative contrôlant la tension de sortie.

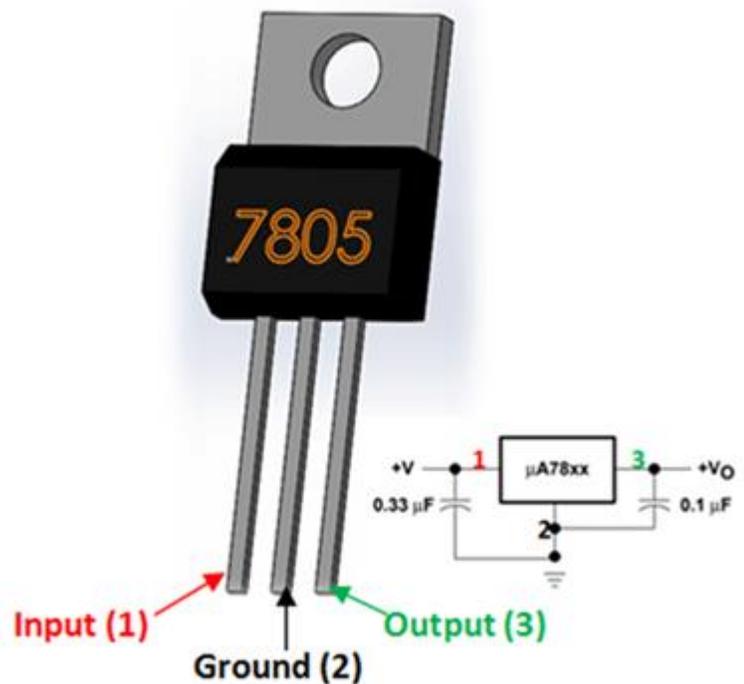


Figure II.10: LM7805 et ses broches

II.3.7. Oscillateur à quartz

Après l'alimentation le deuxième élément indispensable pour faire fonctionner un microcontrôleur est une horloge. Son rôle est de cadencer le rythme d'exécution des instructions. Les paramètres de choix de sa fréquence dépendent du type du microcontrôleur, ce dernier devient nettement plus rapide dans l'exécution des instructions quand la fréquence s'élève quoiqu'il consommera plus d'énergie. Pour réaliser l'horloge en externe comme dans notre cas par exemple, on utilise le quartz HC49 de 8MHZ relié à des condensateurs au niveau des broches OSC1/CLKIN et OSC2/CLKOUT du PIC 16F877A.



Figure II.11: Quartz HC49 8MHz

II.3.8. Moteurs à courant continu DC[9]

Inventée par "Zénope grammeé, une machine à courant continu est une machine électrique et un convertisseur électromécanique permettant donc la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique. On la qualifie de bidirectionnelle car elle peut fonctionner comme moteur en convertissant l'énergie électrique en énergie mécanique ou comme génératrice en convertissant l'énergie mécanique en énergie électrique.



Figure II.12: Moteur à courant continu

Les moteurs à courant continu sont constitués d'une partie fixe qu'on appelle " Stator", cette partie statique a pour rôle de générer un flux magnétique par des bobinages ou par des aimants permanents on qualifie ainsi le stator "d'inducteur". Le "Rotor" est la partie tournante et est relié à un collecteur rotatif. Après chaque tour, il inverse la polarité dans chaque enroulement au moins une fois afin de faire circuler un flux magnétique transversale avec le flux statorique. Le "Balais" est la constituante qui appuie sur le collecteur pour permettre le passage du courant. Ainsi le balais et le collecteur aident eux aussi dans la rotation.

En mécanique on parle de couple du moteur, cette expression veut dire la capacité du moteur à faire tourner quelque chose sur son axe, plus le couple est important plus le moteur pourra supporter une charge

CHAPITRE II : ETUDE DU LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET

plus lourde. Le couple est le produit de la force exercée par le rayon de l'action : $C = F \cdot r$. Il est donc exprimé en Newton-mètre.

On ajoute des engrenages pour les MCC pour jouer un rôle de réducteur. Ces engrenages sont des roues engrenant les unes sur les autres grâce à leurs dentures. Plus le diamètre de la roue est grand plus il y'a d'engrenage et moins est la vitesse. Plus d'égrenage amène plus de frottement ainsi plus de pertes mécaniques. En d'autres termes, le but d'ajouter les engrenages est pour avoir un couple plus important pour supporter un poids plus lourd et moins de vitesse, ce qui devient intéressant pour un robot qui a comme ordre d'éviter les obstacles.

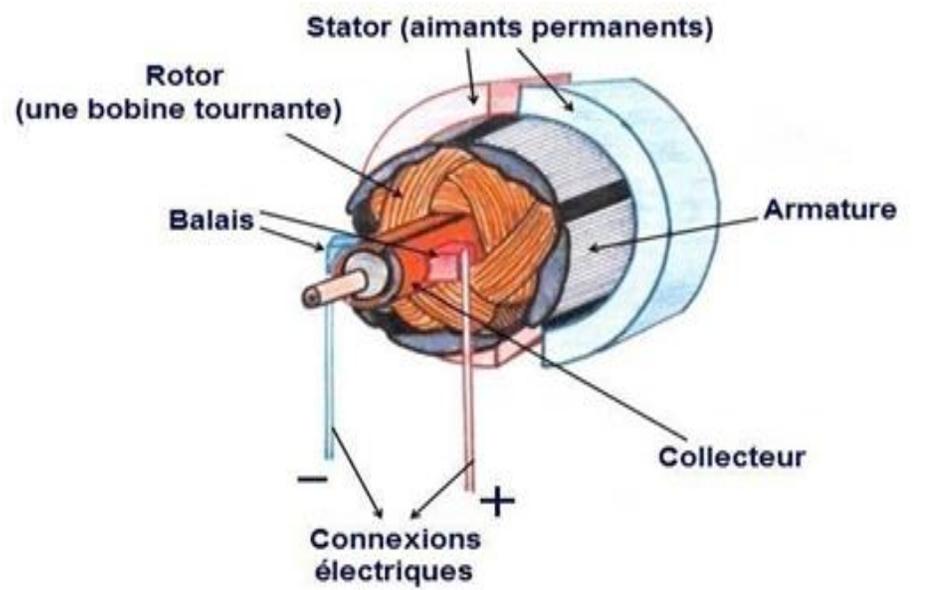


Figure II.13 : Constituantes d'un moteur à courant continu

II.3.9. L'afficheur LCD 16*2

Les afficheurs LCD (Liquid Crystal Display) sont des modules plutôt bon marché ayant pour rôle l'affichage des paramètres invisibles du circuit. Ces modules sont très utilisés dans les montages à microcontrôleur principalement pour la bonne convivialité entre ces deux dernières. C'est ainsi qu'on peut connaitre et être informés des différents changements en affichant les valeurs des variables du système. Par exemple, dans notre cas, l'afficheur LCD nous informe de la distance séparant le robot de l'obstacle grâce aux informations captées par le capteur et aux calculs du microcontrôleur.

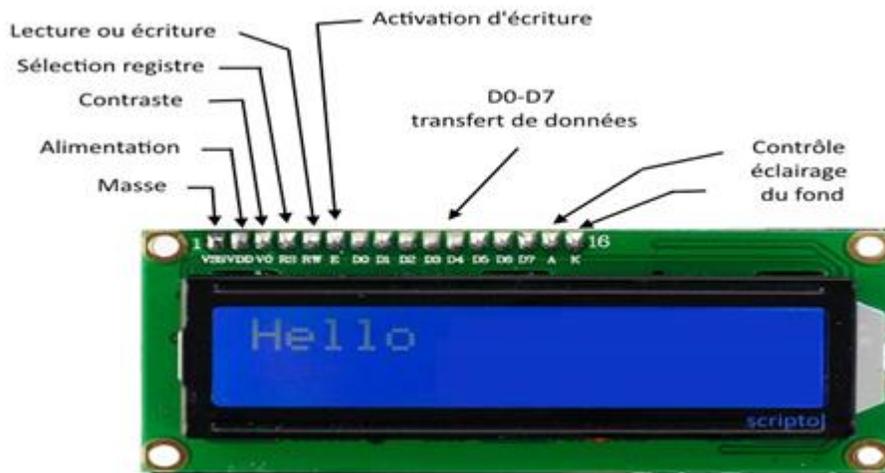


Figure II.14 : Broches de l'afficheur LCD affichant "HELLO"

- **VDD** : Alimentation (5V).
- **VSS** : La masse.
- **VO** : Réglage du contraste en utilisant un potentiomètre.
- **E** : entrée de validation (ENABLE).
- **RS** : Registre Select
- **R/W** : Read / Write, lecture (R/W=1) et écriture (R/W=0). Lorsqu'on commande l'afficheur LCD il faut se placer en écriture.
- **D7...D0** : bus de données bidirectionnel, il permet de transférer les instructions ou les données à l'afficheur LCD.

II.4. Partie Logicielle

II.4.1. Plateforme de programmation MikroC

La plateforme MikroC est un compilateur C complet destiné pour la programmation des circuits Pic de Microchip, MikroC dispose de plusieurs outils intégrés (mode simulateur, correcteur d'erreurs, gestionnaire afficheur 7 segments...) associés à sa capacité dans la gestion des périphériques pouvant être rencontrés dans l'industrie (Bus I2C, SPI, CAN, génération des signaux PWM...).

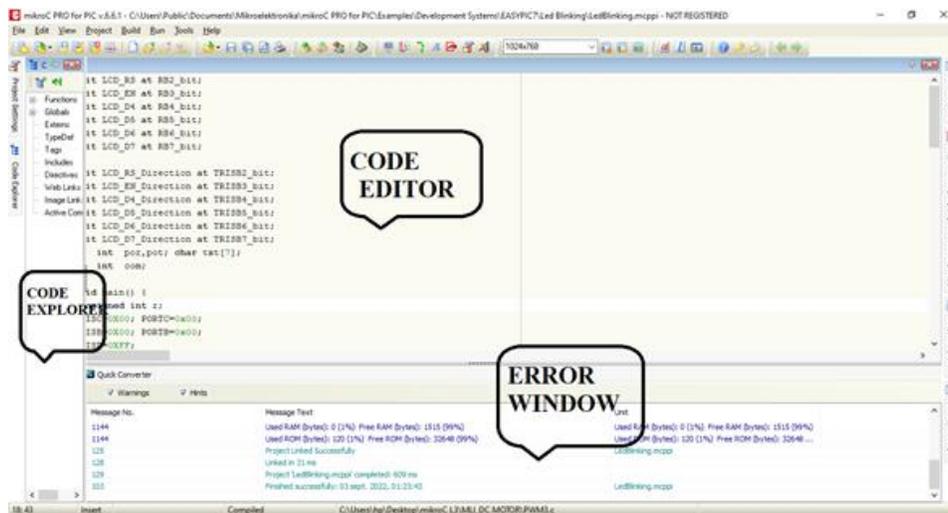


Figure II.15: Programmation sous MikroC

II.4.2. Création d'un projet sous MikroC

Pour créer un projet sous MikroC, on va sur menu, "create a new project", après y avoir cliqué dessus on tombe sur la fenêtre ci-dessous :

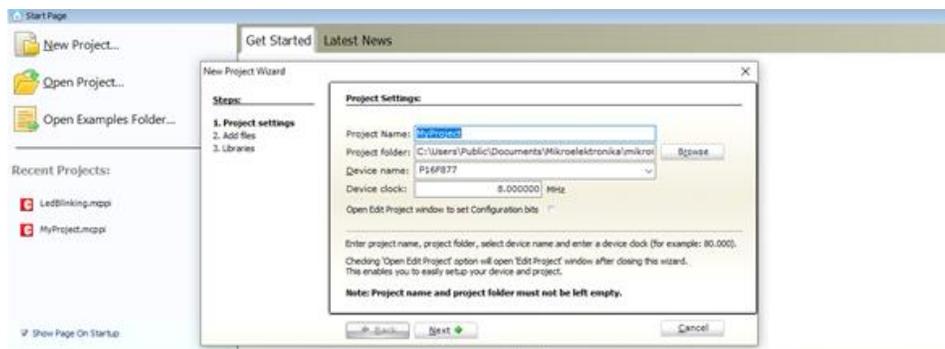


Figure II.16 : Etape 1 de la création d'un projet MikroC

CHAPITRE II : ETUDE DU LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET

On attribue un nom au projet dans la case " Project Name", on choisit l'emplacement du fichier dans la case "Project Folder" puis se fait le choix du microcontrôleur et de sa fréquence d'oscillateur dans les cases respectives "Device Name" et "Device Clock". Après avoir fait cela, on clique sur "Next". On atterrit sur la fenêtre suivante :

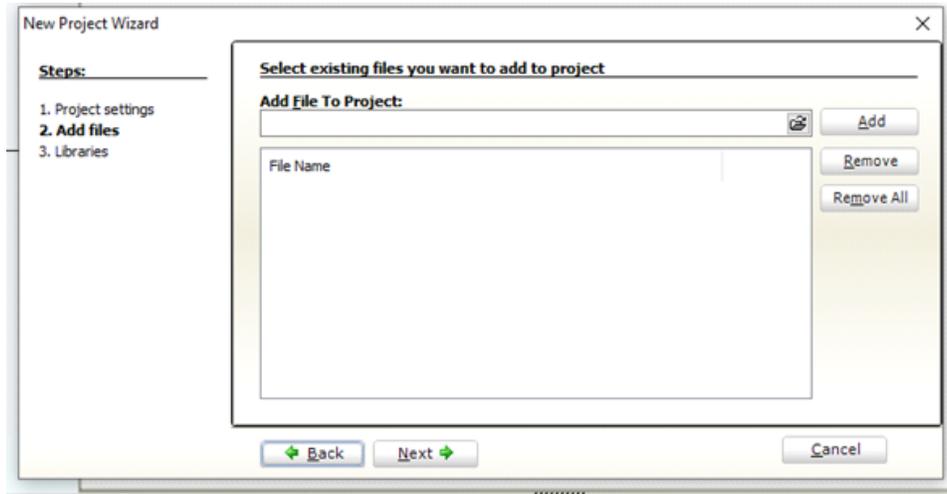


Figure II.17: Etape 2 de la création d'un projet MikroC

Ici, on choisit de lieu de l'emplacement du projet en d'autres termes, le lieu de sauvegarde. Après avoir cliqué sur "Next", on trouve une fenêtre qui suggère d'inclure les libraires (utiles lors de la saisie du code).

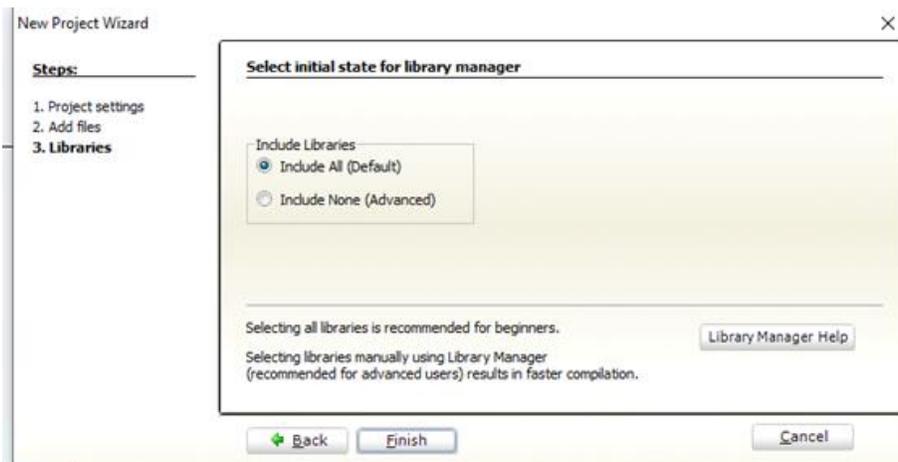


Figure II.18: Etape 3 de la création d'un projet MikroC

CHAPITRE II : ETUDE DE LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET

A la fin de ces instructions, on atterrit sur une fenêtre semblable à celle de la figure II.13 mais vierge (sans le code présenté sur la partie code editor). Il est donc facile de deviner que " Code Editor" est la partie où se passe la saisie du code. Si ce dernier présente des erreurs après avoir été compilé en cliquant sur l'icône "Build", le type d'erreurs est affiché sous forme de message dans la fenêtre en bas "Error Window".

II.4.3. Présentation de la plateforme ISIS PROTEUS

Il est très délicat de réaliser un projet électronique sans passer par l'étape de la simulation logicielle, d'ailleurs cette étape facilite la conception et nous informe de la réalisabilité théorique du projet ainsi cela peut nous épargner de perdre du temps et de protéger les composants.

ISIS PROTEUS est un logiciel de développement et de simulation à travers un environnement graphique simple. Le lancement de Proteus donne vue sur un environnement classique constitué d'une fenêtre principale et d'un ensemble de barre d'outils. Nous avons ainsi un aperçu des fonctionnalités permettant l'affichage, les options de projet et la gestion de fichiers. Cette fenêtre principale comprend une zone de travail où l'on développe et réalisent les circuits à tester. Encore, une barre d'outils principale où l'on peut trouver des icônes proposant : le type d'alimentation, les générateurs, les appareils de mesures, des options de rotation.... Au milieu, la bibliothèque d'objet qui affiche les composant en cours d'utilisation. En bas à gauche, des touches magnétoscopes pour la mise en pause, en lecture et l'arrêt de la simulation.

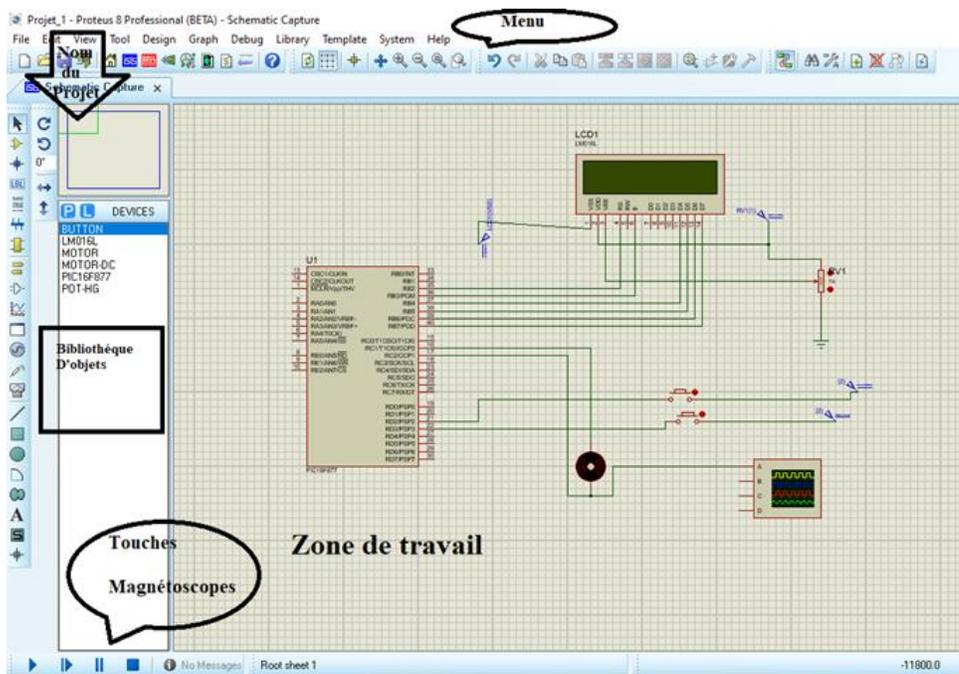


Figure II.19: Environnement Proteus

II.4.4. Les librairies dans Proteus

Le seul bémol trouvé sur Proteus réside dans le fait qu'il ne contienne pas certains composants dans sa base de données, ainsi à première vue on ne peut simuler certains circuits envisagés. Par exemple, le capteur ultrason "HC-SR04" n'en figure pas. De ce fait, il est nécessaire de chercher sur le net les bibliothèques manquantes pour chaque composant non trouvé initialement. Il existe des sites web liés à l'ingénierie électronique nous proposant des bibliothèques manquantes souvent gratuitement. Il suffit de télécharger les fichier proposés (souvent des fichiers Zip), de les compresser, on obtient après cela 3 extensions (.LIB, IDX et. HEX). A présent, pour les introduire dans Proteus, il suffit simplement de les copier et de les coller dans le dossier "Library". La dernière étape est de redémarrez Proteus, ainsi il pourra les lire et les aura déjà intégrés ces composants dans sa librairie.

II.4.5. Le PICkit 2 et son logiciel PICkit V2.61

Les pickits sont une famille de programmeurs destinés pour les microcontrôleurs PIC de Microchip Technology, Ils permettent la programmation de ces derniers ainsi que certaines mémoires EEPROM.

Le PICkit 2 est un outil de développement possédant plus de possibilités que les modèles le précédent, il peut programmer des contrôleurs PIC fonctionnant à 3V3 et 5V.

En effet, c'est une version améliorée car il a la possibilité de séparer la partie programmation de la partie débogage sur la carte d'évaluation. Ceci nous épargne de devoir insérer le PIC à programmer dans une carte de démonstration à chaque fois. Il contient un microcontrôleur PIC 18F2550 possédant un bus USB FULL SPEED. C'est une interface de programmation Windows complète, prenant en charge les familles de base (PIC10F, PIC12F5xx, PIC16F5xx), de rang moyen (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33 et PIC32 de microcontrôleurs 8 bits, 16 bits et 32 bits, ainsi que de nombreux produits EEPROM série Microchip.

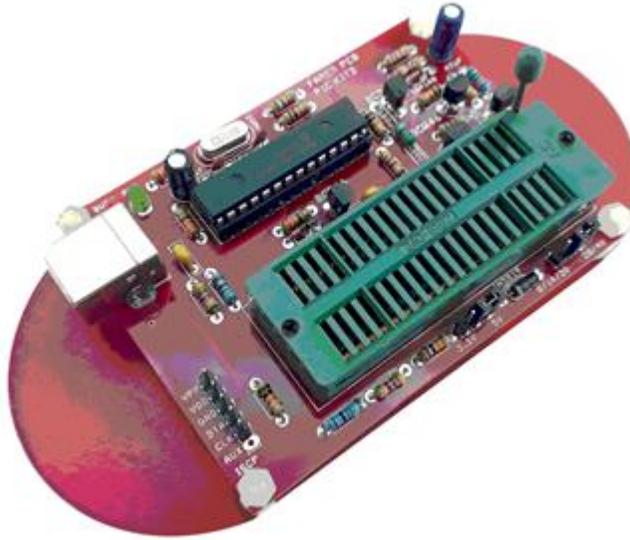


Figure II.20: PICkit 2

II.4.6. Téléversement du code depuis le logiciel vers le PIC16F877A

Le PICKIT possède un port USB, à la fois pour l'alimentation mais aussi pour la connexion à l'ordinateur, par opposition au port série traditionnel, éliminant ainsi le besoin d'une alimentation séparée. Avant, le branchement, on lance le logiciel et la fenêtre suivante apparaît :

CHAPITRE II : ETUDE DE LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET

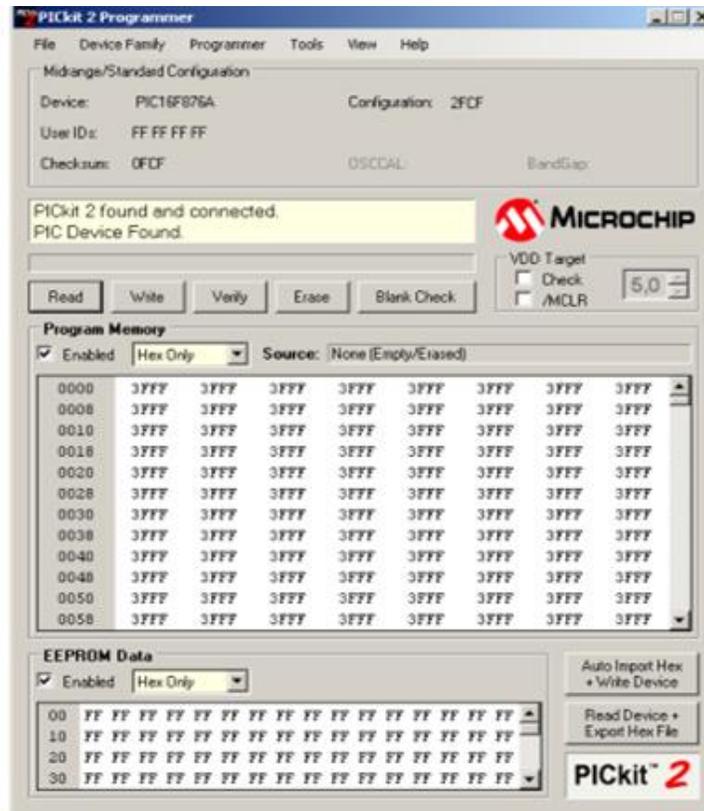


Figure II.21 : Accueil PICkit 2

On peut apercevoir le PIC utilisé et la zone de mémoire.

Après avoir importé le fichier .HEX du code que l'on veut téléverser en allant sur file > import hex, on obtient la fenêtre suivante :

CHAPITRE II : ETUDE DE LA PARTIE MATERIELLE ET LOGICIELLE DU PROJET

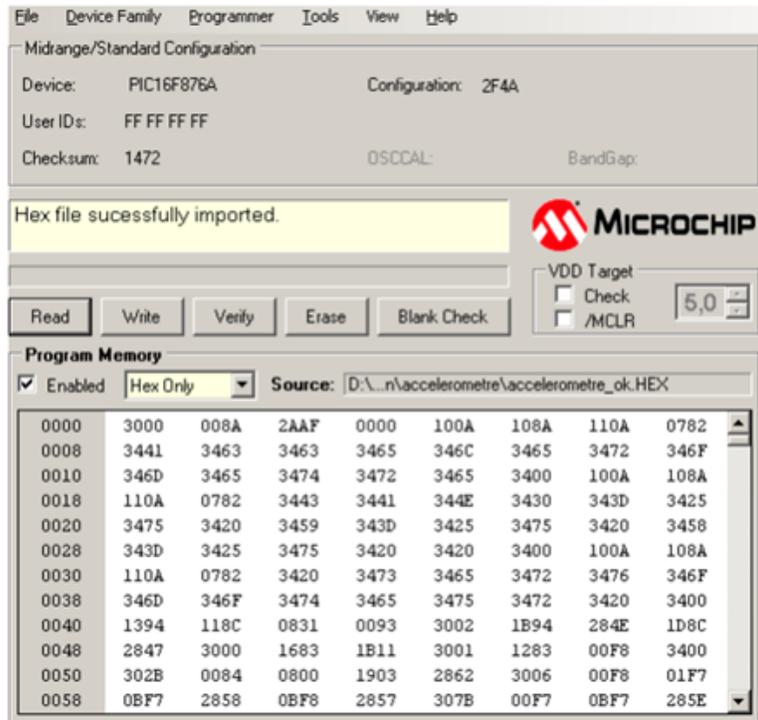


Figure II.22 : Code Hex importé

On peut voir que le code a été importé avec succès et il apparaît au niveau de program memory.

La dernière étape pour téléverser le code consiste à cliquer sur "Read" puis sur "Write". Une fois que l'on voit "Programming Successful". Cela veut dire que le programme a été téléversé vers le PIC avec succès comme dans la figure ci-dessous :

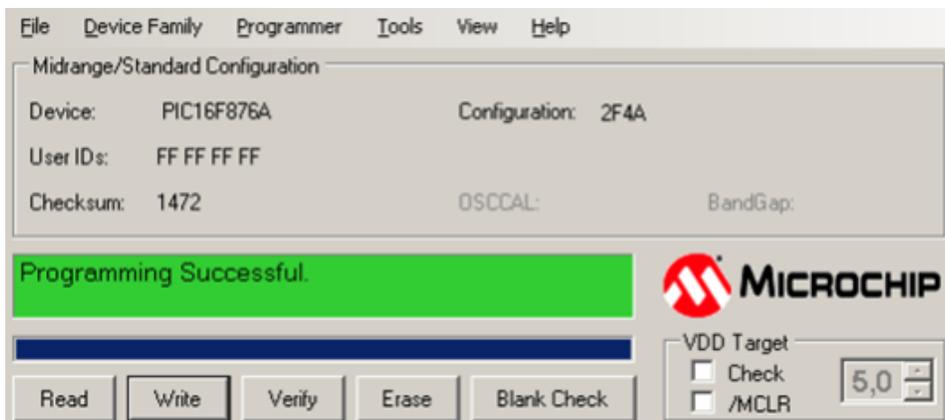


Figure II.23 : Téléversement avec succès

II.5. Conclusion

Le long de ce chapitre, nous sommes passé par une étude théorique du projet, présentant d'abord un schéma synoptique qui définit bien le sens de fonctionnement du robot.

Ensuite, nous avons présenté et expliqué le fonctionnement de chaque composant utilisé tout en exhibant leurs avantages et en soulignant les raisons de choix de ces derniers et leurs rôles dans le circuit.

Après cela, on a présenté chaque logiciel utilisé en montrant les techniques d'utilisation et certains points de vue.

Enfin, après l'étude matérielle et logicielle du projet, nous pouvons passer à la partie qui pourra le clôturer et il s'agit du dernier chapitre consacré à sa réalisation

CHAPITRE III :

**CONCEPTION ET
REALISATION DU
ROBOT**

III.1. Introduction

En pratique il y a plusieurs facteurs à tenir en compte et plusieurs défis à relever, on rencontre souvent des contraintes et des résultats différents par rapport à la simulation logicielle.

Dans le chapitre précédent, on a parlé des composants requis que l'on va utiliser. Ceci-dit il faut tenir en compte de la maquette et le choix du châssis (taille, roues), ce choix se fait par rapport au poids de cette dernière pour trouver le bon équilibre et ne pas porter atteinte à sa vitesse afin d'assurer un bon fonctionnement.

Ainsi dans ce troisième et dernier chapitre, on présentera la base du robot, l'organigramme descriptif de son fonctionnement et entamerons enfin la réalisation.

III.2. Description de la base

Pour la réalisation de ce robot, nous avons opté pour une base en bois rectangulaire d'une largeur de 75mm et d'une longueur de 135mm ayant une roue libre mise en avant au centre pour assurer la stabilité du robot du fait qu'elle soit plutôt légère. Cette plateforme est accompagnée de deux roues motrices faites en gomme pouvant être reliées chacune à un moteur à courant continu.

La figure ci-dessous illustre ce qu'on vient d'expliquer :

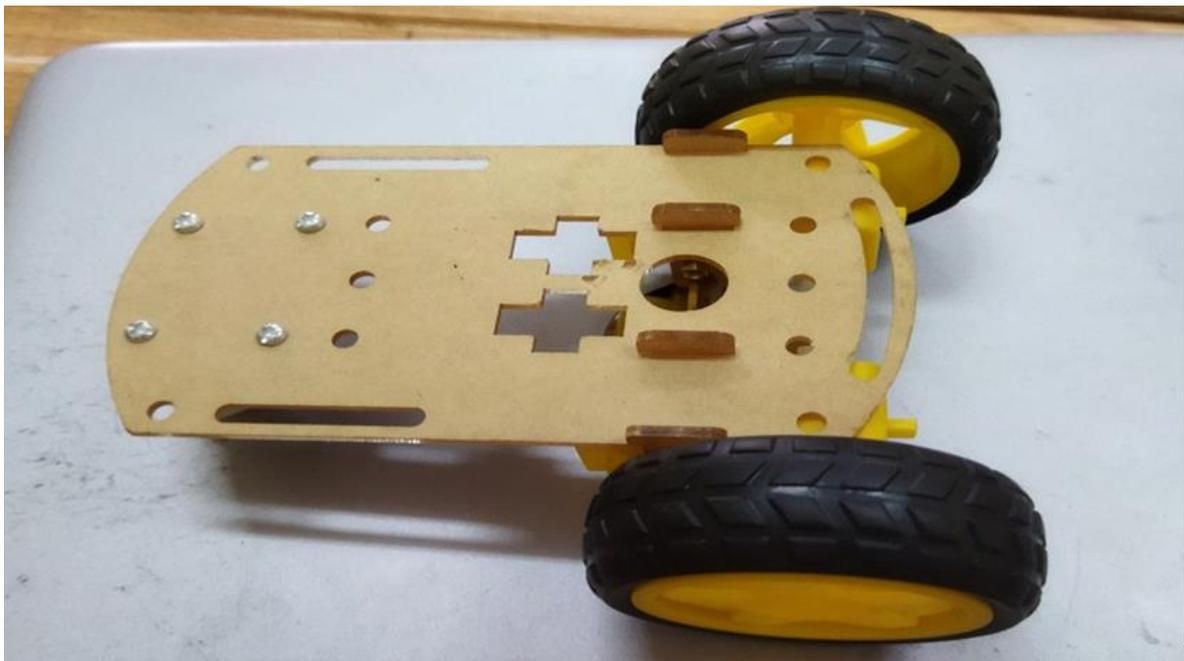


Figure III.1 : Base du robot

III.3. Organigramme de l'évitement d'un obstacle

Avant de commencer la programmation, il est nécessaire de passer par la schématisation d'un organigramme qui représentera la logique de fonctionnement du robot. Nous vous faisons part du notre dans la figure ci-dessous où : IRG, IRD et UC représente respectivement le capteur IR gauche, droit et le capteur ultrason au centre s'ils sont égale à zéro cela présente la présence d'un obstacle sinon il n'y en a pas

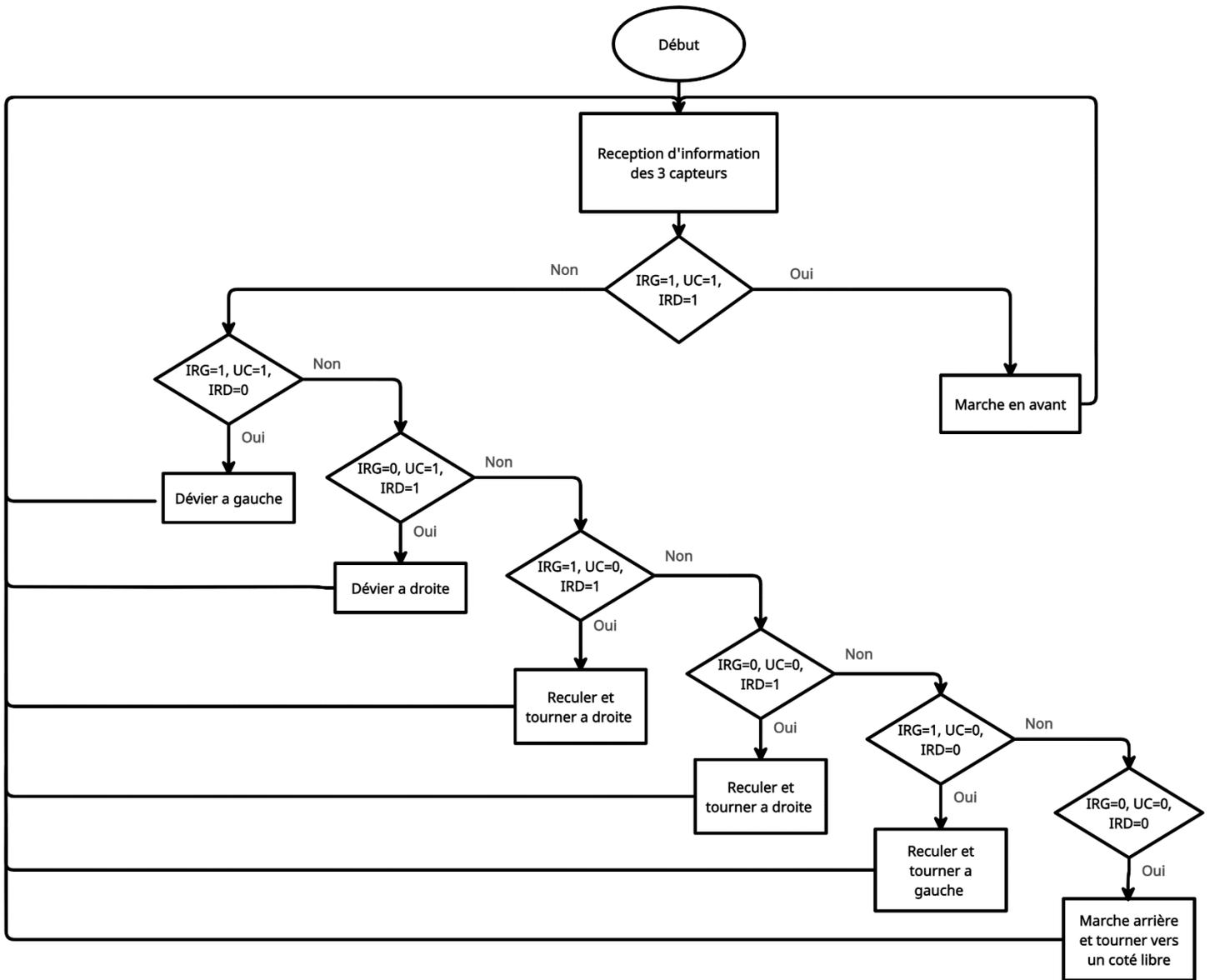


Figure III.2 : Organigramme du robot

CHAPITRE III : CONCEPTION ET REALISATION DU ROBOT

III.4. Circuit complet du robot

Dans la figure ci-dessous, nous vous présentons un aperçu du circuit complet du robot à travers le logiciel Proteus 8 par lequel nous avons effectué nos nombreuses simulations

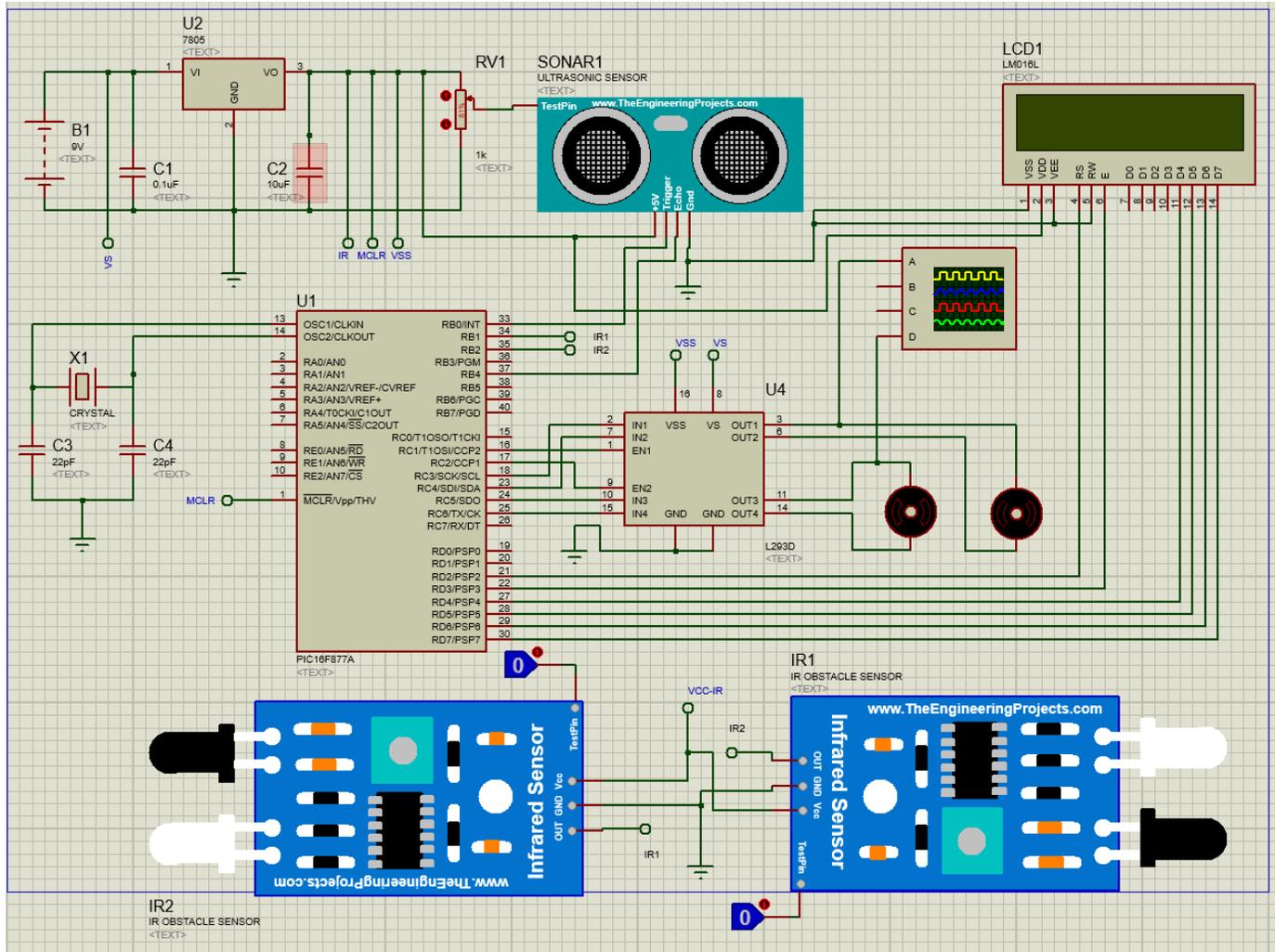


Figure III.3 : Circuit du robot sous Proteus

III.5. La communication entre le PIC 16F877A et les autres organes du robot

III.5.1. Interfacer le PIC avec le capteur ultrason HC-SR04 [10]

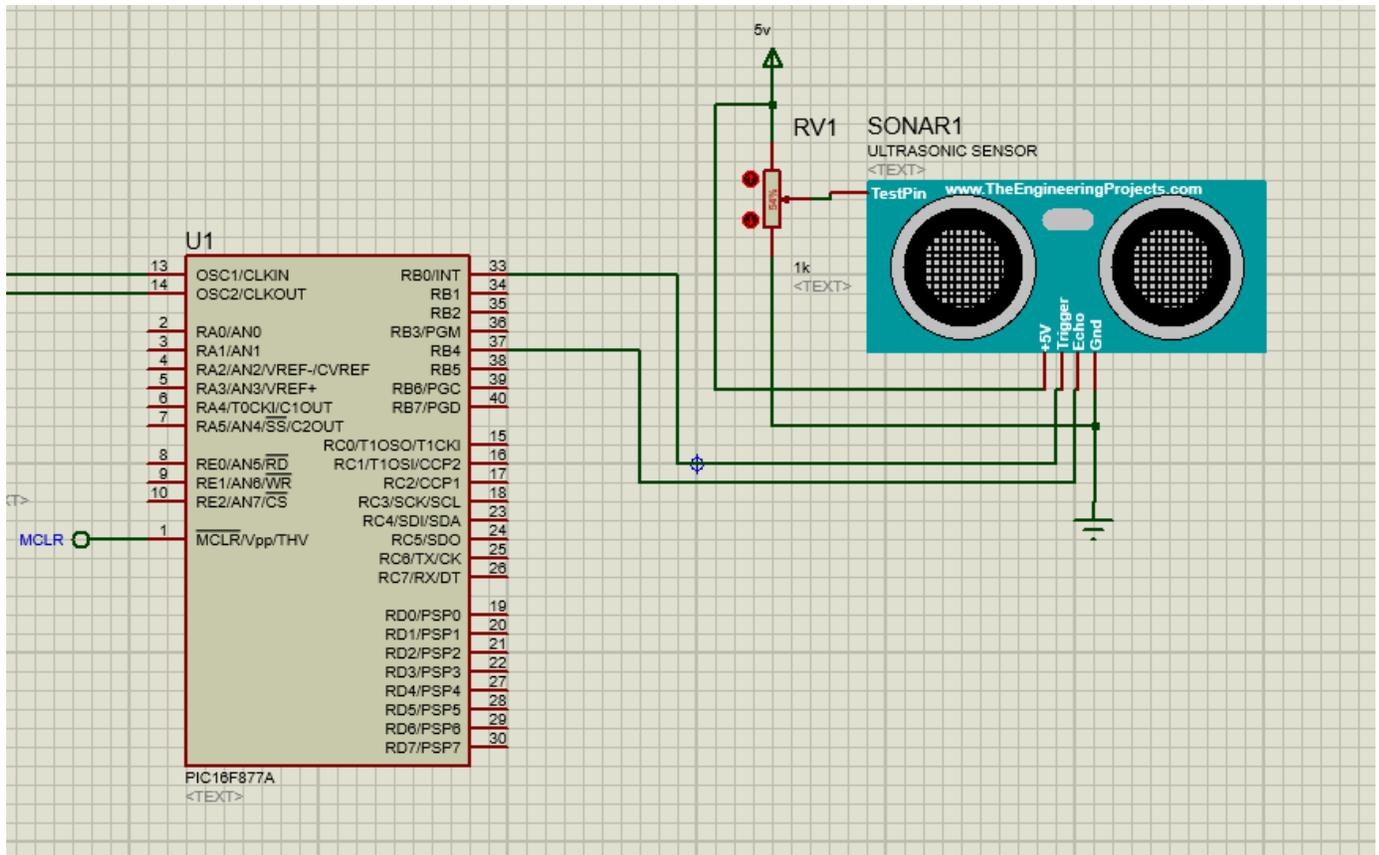


Figure III.4 : Branchement du capteur ultrason au PIC

Comme nous l'avons expliqué dans le "chapitre II", le HC-SR04 dispose de 4 broches dont Vcc qui est connectée à l'alimentation de 5V, GND à la terre. Quant au deux autres pins "ECHO" et "TRIGGER", ils sont considérés respectivement comme les entrées et sorties par rapport à la communication avec le PIC16F877A. La broche "TRIGGER" est connecté à la broche RB0 du PIC16F877A, une impulsion de 5V est déclenchée par cette dernière sur la première pendant 10 μ s, après la réception de cette impulsion le capteur émet la série de 8 impulsions ultrasoniques à une fréquence de 40Khz. Après cette émission, la broche "ECHO" connectée à RB4 est mise à 1 et donc passe à un état haut. Le compteur "TIMER1" démarre un décompte jusqu'à la réception des impulsions envoyées, cette broche d'écho restera à l'état haut jusqu'à

CHAPITRE III : CONCEPTION ET REALISATION DU ROBOT

ce que le capteur reçoit les huit impulsions qui ont été transmises au récepteur du capteur. Lorsque cela se produit, la broche d'écho reviendra à un "0" logique.

La largeur d'impulsion n'est que le temps pendant lequel l'impulsion est élevée. Maintenant, nous pouvons calculer la distance par rapport à l'obstacle. Par exemple :

Si la période de temps de l'impulsion d'écho était de 250 μ s alors la distance en cm serait calculée comme suit : $\text{distance} = (250/2) \times 0,0343 = 4,2875$ cm. On divise par deux car le signal aura fait un aller-retour donc deux fois la distance par rapport à l'obstacle. La multiplication par 0.0343 est car ce chiffre est la conversion de la vitesse du son qui est de 343 m/s en cm/s.

La Figure ci-dessous montre le chronogramme de l'entrée Trig, l'onde transmise et la sortie d'écho.

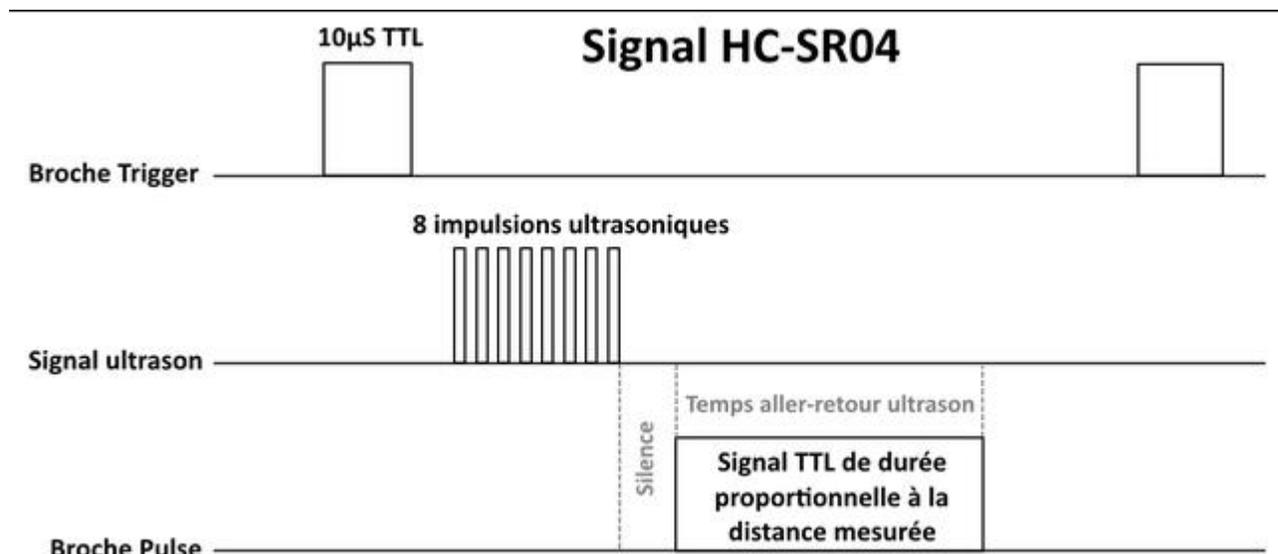


Figure III.5 : Le chronogramme du fonctionnement du HC-SR04

```
portB.f0=1;           //mettre le trigger a un logique "1"  
delay_us(10);        //attendre 10µs  
portB.f0=0;          //mettre le trigger a un logique "0"  
delay_ms(50);        //attendre 50ms  
| //lors de la reception du signal reflechi, mise a 1 de la broche echo, execution du programme "interrupt"  
distance = duration * (0.034/2); //calcul de la distance par la durée
```

Figure III.6 : Programmation du HC-SR04

III.5.2. Interfacer le PIC avec les capteurs infrarouges

Le capteur ultrason n'est bien efficace que sur un angle de 15°, au-delà de cette angle ses performances deviennent moins intéressantes. C'est principalement pour ça que nous avons utilisés les capteurs infrarouges IR comme points de repères au niveaux des côtés du robot (gauche et droite).

Nous avons donc mis un capteur infrarouge en chaque côté, le capteur IR de gauche est connecté à la broche RB2 quant à celui de droite, il est connecté à la broche RB1 du microcontrôleur. Le principe de fonctionnement de la détection d'obstacle par infrarouge est semblable à celui des capteurs ultrasons... Par exemple si on étudie le cas du capteur IR placé à gauche, ce dernier émet un faisceau de lumière infrarouge par le biais de sa "LED IR", initialement sa connexion au niveau du pic la broche RB2 est mise à l'état haut, ainsi RB2=1. S'il y'a réflexion de la lumière émise, son Détecteur IR la détectera, il en résulte ainsi une impulsion et RB2 sera mise à l'état bas (RB2=0), cela signifie la présence d'un obstacle. C'est de cette manière qu'on programme les capteurs.

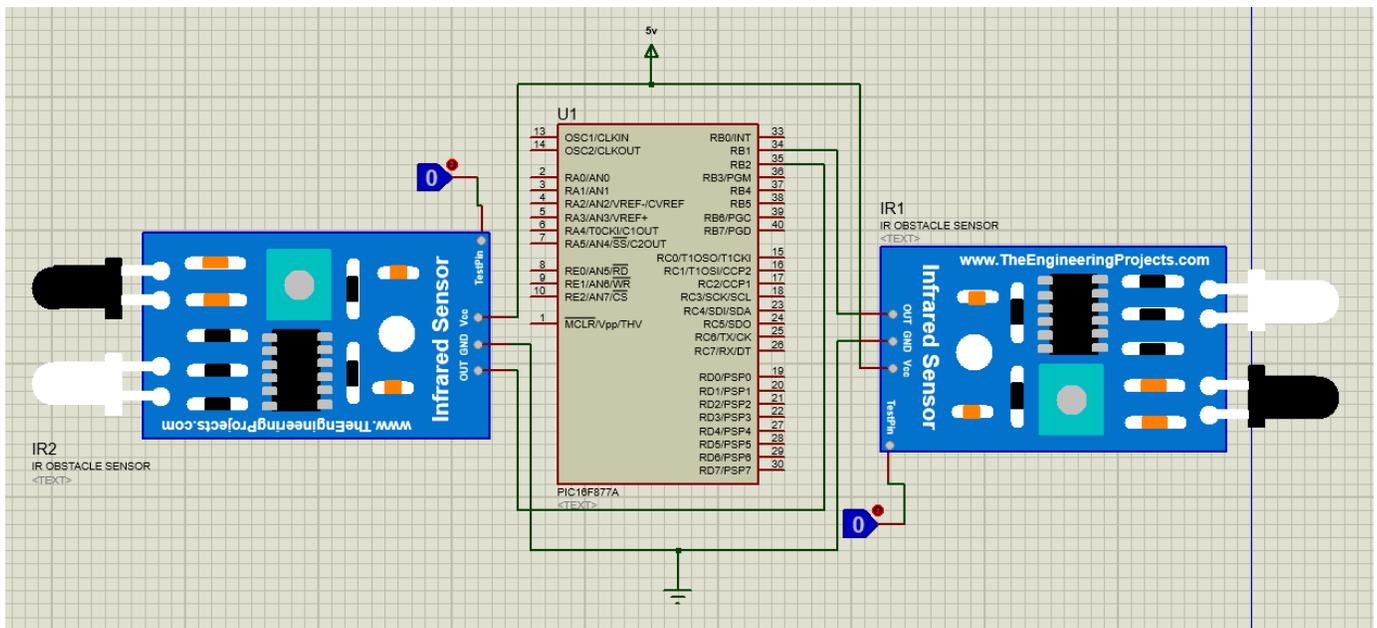


Figure III.7 : Branchement des capteurs infrarouge avec le PIC

- Si RB1=1 cela veut dire qu'il y'a un obstacle à droite et si RB1=0 cela veut dire qu'il n'y a pas d'obstacle à droite
- Si RB2=1 cela implique qu'il y'a obstacle à gauche et si RB2=0 cela veut dire qu'il n'y a pas d'obstacle à droite

III.5.3. Interfacer le PIC avec le L293D

Le double pont H est un élément indispensable pour notre projet, comme nous pouvons le savoir, théoriquement inverser une tension d'alimentation peut provoquer la rotation d'un moteur à courant continu dans le sens inverse.

Quoique, cette méthode ne peut convenir en pratique, de ce fait notre alternative pour commander les moteurs dans le sens des aiguilles d'une montre et dans leurs sens inverse a été de choisir le circuit L293D. Au niveau de ce circuit intégré, le sens de rotation du moteur est défini par le sens dans lequel le courant circule au niveau de ce dernier c'est d'ailleurs pour ça qu'il est constitué de deux transistors PNP et deux transistors NPN. En d'autres termes pour changer le sens de rotation d'un moteur il suffit de changer la polarité de son alimentation.

On voit maintenant pourquoi il nous a été préférable de choisir un circuit intégré déjà prêt au contrôle que d'élaborer son propre pont avec des transistors, principalement pour le gain en temps, le fait qu'il serait moins encombrant et sa facilité de mise en œuvre en pratique.

De plus le L293D contient des diodes de roue libre en d'autres termes, des diodes anti-retour en parallèle avec chaque transistor et placées avec une polarité inversée à celle de l'alimentation électrique dans le but de protéger les transistors et prolonger la durée de vie en renvoyant à la masse les surtensions induites par les effets transitoires ou les coupures.

Nous avons interfacé le L293D avec le PIC16F877A en reliant leurs broches respectivement de cette manière :

IN1 à RC3 et IN2 à RC4 pour commander le premier moteur

IN3 à RC5 et IN4 à RC6 pour commander le second moteur

Les deux GND à la masse

VCC1 qui est la broche n°8 à 9V pour l'alimentation unique des moteurs pour que ces derniers s'autoalimentent et n'aillent pas en priver les autres organes sachant qu'ils sont gourmands en consommation particulièrement lors de leurs démarrages.

VCC2 au 5V quoique le L293D peut supporter une alimentation jusqu'à 36V

CHAPITRE III : CONCEPTION ET REALISATION DU ROBOT

Les tableaux suivant montrent toutes les éventualités possibles pour les moteurs avec le L293D :

Input 1	Input 2	Moteur 1
0	0	Arrêt
0	1	Tourne dans le sens des aiguilles d'une montre
1	0	Tourne dans le sens inverse des aiguilles d'une montre
1	1	Arrêt

Tableau 1 : Commande du moteur 1

Input 3	Input 4	Moteur 2
0	0	Arrêt
0	1	Tourne dans le sens des aiguilles d'une montre
1	0	Tourne dans le sens inverse des aiguilles d'une montre
1	1	Arrêt

Tableau 1 : Commande du moteur 2

III.5.5. Interfacer le PIC avec l'afficheur LCD

Pour ce projet, l'afficheur LCD joue le rôle d'informateur pour l'utilisateur du robot, nous l'avons programmé de sorte qu'il nous affiche la distance séparant le robot de l'obstacle lorsque ce dernier est en marche avant. Aussi, un affichage du sens de sa tournure lors de la présence d'un obstacle sur un des cotés et un affichage « marcher arrière » lors de la présence d'obstacles en face et sur les deux cotés.

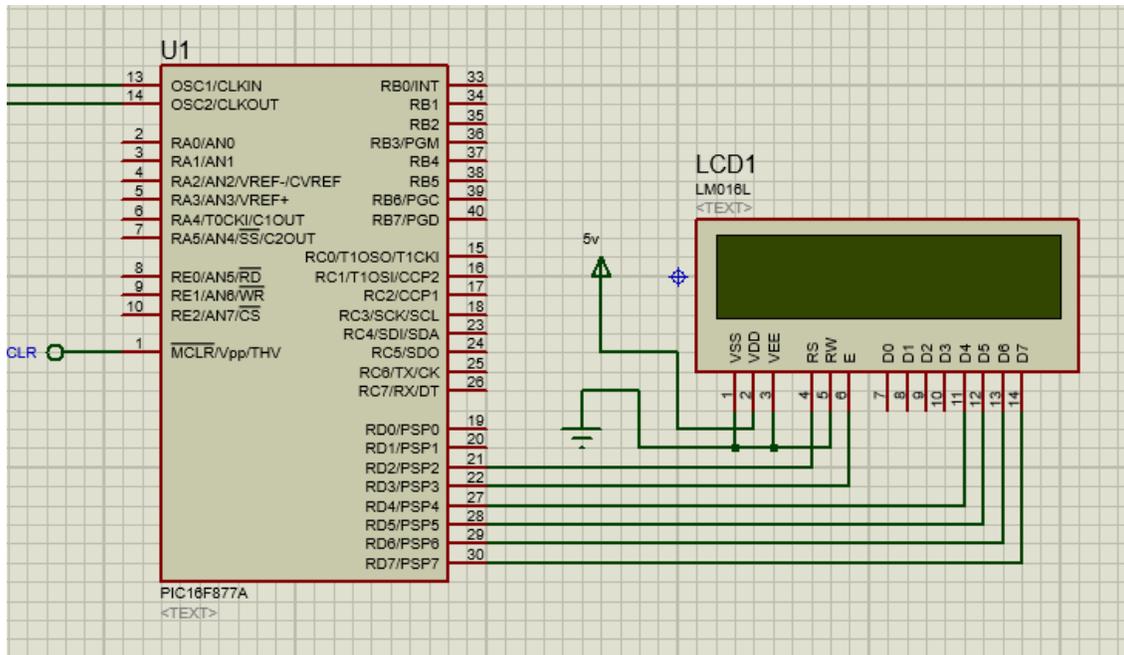


Figure III.9 : Interfacer le PIC avec l'afficheur LCD

Concernant le LCD, nous l'avons interfacé en mode 4 bits ainsi seul les quatre lignes de données supérieures sont utilisées comme bus de données pour économiser des entrées/sorties au niveau du microcontrôleur et les consacrer pour d'autres utilisations.

Le branchement s'est fait comme suit :

La broche D7 à la broche RD7 du PIC.

La broche D6 à la broche RD6 du PIC.

La broche D5 à la broche RD5 du PIC.

La broche D4 à la broche RD4 du PIC.

La broche E à la broche RD3 du PIC.

CHAPITRE III : CONCEPTION ET REALISATION DU ROBOT

La broche RS à la broche RD2 du PIC.

RW, VEE et VSS à la masse GND.

VDD à l'alimentation 5V.

III.6. Finalisation de la réalisation :

Après avoir entamé les différentes étapes de réalisation et câblage, nous vous présentons différents aperçus de la réalisation finale :

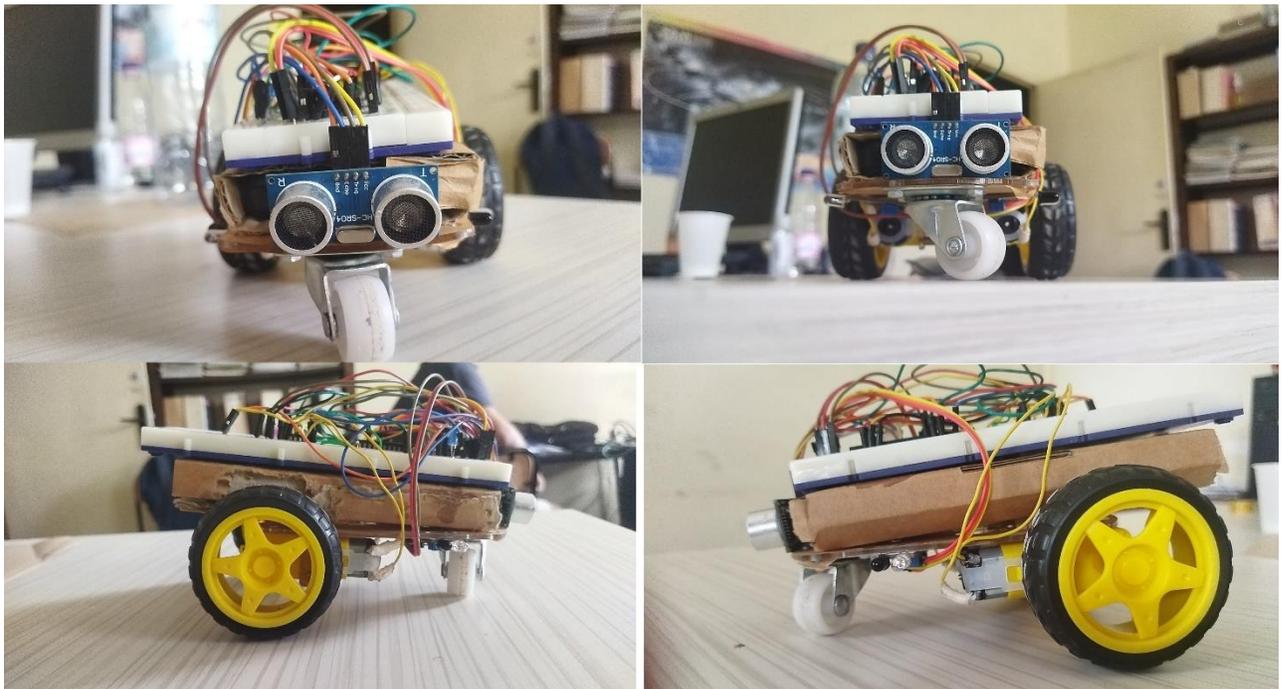


Figure III.10 : vue de différents angles du robot

III.7. Tests et problèmes rencontrés

Après le montage et le transfert du programme, le robot a présenté certaines imperfections liées à son déplacement or il avancer et recule d'une façon discontinue, parfois lors du démarrage un moteur fonctionne et l'autre non. Nous avons compris que ces problèmes étaient dû au faible courant arrivant aux bornes des moteurs car les batteries de 9V délivrent au maximum 500mA.

Alors la première idée a été d'ajouter une seconde batterie pour alimenter les moteurs séparément, ce qui a donné ses fruits. Quoique nous aimerions réaliser un robot qui peut fonctionner pendant plusieurs heures, malgré l'utilisation de deux batteries le temps de fonctionnement demeurerait réduit à une petite heure. Pour y remédier à ce problème il fallait qu'on cherche des sources d'alimentation qui pourrait nous permettre de tirer plus de courant.

Les hypothèses partagées ont été les suivantes :

- L'utilisation d'un transformateur 220V-5V 1500mA et ce qui a donné de bons résultats or la discontinuité de la vitesse de rotation des moteurs a été corrigée et le démarrage des deux moteurs bien synchronisé.
- L'utilisation d'une alimentation Power-Bank en lithium et qui délivre une tension de 5V avec un courant de 2A, ce qui a aussi bien fonctionné.

Compte tenu du fait que le robot doit être mobile, il n'était pas possible d'accepter de l'alimenter avec un transformateur car il le condamnerait à avancer d'une distance équivalente à celle du câble de ce dernier. Il a donc été évident de choisir le power Bank de plus ce dernier à une capacité de 8000mah, on peut donc le faire fonctionner pendant plusieurs heures.

III.8. Conclusion

Dans ce chapitre on a pu expliquer comment interfacier chaque organe avec le PIC 16F877A pour faire la réalisation d'un robot mobile de type tricycle où le capteur ultrason HC-SR04 détecte les obstacles en face, les capteurs de proximité IR sont jouent eux aussi le rôle d'œil mais dans les côtés. Le moyen de locomotion se résume à deux moteurs à courant continu commandés par un pont H double L293D.

Le robot arrive à éviter les obstacles à ses côtés même quand il est en marche avant ce qui veut dire même lorsqu'il n'y a aucun obstacle en face, le moyen de faire une marche arrière s'il se retrouve coincé

CHAPITRE III : CONCEPTION ET REALISATION DU ROBOT

entre des obstacles en face et de chaque cotés jusqu'à la libération d'un côté et de prendre le chemin de ce dernier. Toutes les tournures se basent sur le sens de rotation des moteurs et donc des roues. On a aussi partagé les problèmes rencontrés lors de la réalisation et notre façon de les appréhender en trouvant les bonnes solutions

Conclusion générale

La réalisation de ce projet fut un défi. Le robot se devait d'être mobile et autonome dans la poursuite de son parcours, et pour présenter de telles qualités, il nous a été imposé de chercher des sources d'alimentation embarquée qui seraient capables d'alimenter tous ses organes pendant des durées importantes, les petites batteries s'affaiblissent très vite et sont souvent pas rechargeables d'ailleurs nous avons rencontré des problèmes de fonctionnement lorsque le courant arrivant aux moteurs fut jugé insuffisant par ces derniers par conséquent ils tournaient d'une manière discontinue, le compte-rendu a été meilleur avec l'utilisation d'un Power-Bank 5V 2A de 8000mah. La partie mécanique est elle aussi très importante, il nous a fallu tenir en compte le poids global de la maquette et choisir un châssis et des roues pouvant permettre une locomotion facile et une efficacité d'exécution des manœuvres dans les endroits étroits. La partie logicielle est la partie lui offrant son autonomie en suivant la logique de programmation en planifiant lui-même sa trajectoire bien sûr grâce aux capteurs offrant les informations nécessaires et si ces derniers rencontrent des dysfonctionnements le robot ne saurait pas éviter d'éventuels obstacles.

Nous avons donc pu réussir la réalisation d'un robot mobile autonome basé autour d'un microcontrôleur PIC 16F877A fabriqué par Microchip Technology, et de trois capteurs utilisés comme des moyens de vision et de perception; le premier est le capteur ultrasonore HC-SR04 placé en avant du robot pour détecter les obstacles en face, les deux autres moyens de perception sont les capteurs de proximité IR placé l'un à la gauche et l'autre à la droite du robot principalement car la plage de détection d'obstacle par le HC-SR04 a un angle limité, nous les avons donc ajouté au lieu d'utiliser un servomoteur pouvant servir de coude en orientant le capteur dans de différentes directions pour un gain en temps de réflexion. Les mouvements du robot se font grâce aux moteurs à courant continu qui sont commandés par le circuit intégré L293D qui est un double pont H et l'élément clé pour leurs sens de rotation.

Le robot présenté offre une plage d'utilisation variée avec des possibilités d'amélioration simplement en modifiant le programme injecté à l'intérieur du microcontrôleur et en ajoutant de simples composants par exemple : on peut en faire un robot suiveur de ligne pouvant faire des tâches ménagères, un robot ayant des aptitudes pour en sortir d'un labyrinthe, un robot mobile pouvant détecter des fuites d'eau ou des fuites de gaz...etc

Nous désirons exploiter ce robot pour de futur travaux pour son amélioration de sorte qu'il arrive à reconnaître une personne ou des plaques d'immatriculation en l'équipant d'un système de reconnaissance faciale et de traitement des images. Ce genre d'amélioration peuvent l'insérer dans le milieu de la livraison qui aujourd'hui se voit de plus en plus chercher l'exploitation des robots livreurs.

Bibliographie

- [1] Karel Capek, R.U.R. Rossum's Universal Robots, Minos, coll. La Différence, 1997, 221 p.
- [2] <http://floriane.laudien.free.fr/lolo/electro.php> Consulté en mai, juin 2022
- [3] <https://djgsi974.wordpress.com/2015/03/01/gestion-du-detecteur-dobstacle-a-ultrason-ping/>
Consulté en mai, juin 2022
- [4] <https://www.efxkits.co.uk/pic-microcontroller-architecture-with-applications/> Consulté en juin 2022
- [5] <https://informatique-et-electronique.fr/index.php> Consulté en juin 2022
- [6] <https://www.watelectronics.com/pic-microcontroller-architecture-and-applications/> Consulté en juin, juillet 2022
- [7] <https://arduinolearn.github.io/ultra.html> Consulté en juin, juillet 2022
- [8] <https://www.electronicsforu.com/technology-trends/learn-electronics/7805-ic-voltage-regulator>
Consulté en juillet, aout 2022
- [9] <https://www.astuces-pratiques.fr/electronique/le-moteur-a-courant-continu-principe> Consulté en aout, septembre 2022
- [10] Hubert Henry Ward "Programming Arduino Projects with the PIC Microcontroller", 234-243