

TABLE DE MATIÈRES

Remerciement.....	2
Dédicace	3
RESUME	4
Table de matières	5
Liste des figures.....	10
Liste des Tableaux.....	11
Liste des images.....	11
Introduction Générale.....	12

Chapitre I : Généralités sur les composants électriques

I.1 Introduction	14
I.2 Définition.....	14
I.3 Principe de fonctionnement.....	15
I.4 La technologie des moteurs pas à pas.....	15
I.4.1 Moteur pas à pas à réluctance variable MRV	15
I.4.1.1 Le principe de fonctionnement	16
I.4.2 Moteur pas à pas aimant permanent MP.....	16
I.4.2.1 Les moteurs pas à pas unipolaires	17
I.4.2.1.1 fonctionnement d'un moteur pas-a-pas unipolaire	17
I.4.2.2 Moteurs pas à pas bipolaires	18
I.4.2.2.1 : Fonctionnement à pas complet.....	18
I.4.3 Moteur pas à pas hybride	18
I.4.3.1 Définition	18
I.4.3.2 Fonctionnement d'un moteur pas à pas hybride.....	19
I.5 Comparaison entre les trois types de moteurs pas à pas	19
I.6 Caractéristiques générales des moteurs pas à pas	19
I.6.1 Caractéristiques statiques	19
I.6.2 Caractéristiques dynamiques	20

I .7 Alimentation des moteur Pas à pas.....	20
I .8 Les avantages et les inconvénients des moteurs pas à pas.....	21
Chapitre II: Etudes des composants électriques du circuit	
II.1 Transistors MOSFET	23
II .1.1 Historique	23
II .1.2 Définition	23
II .1.3 Les Symboles du transistor mosfet.....	23
II .1.4 Mode de fonctionnement du mosfet	24
II.1.4.1 Comportement à l'état ouvert.....	24
II .1.4.2 Comportement à l'état fermé : caractéristique statique $I_D=f(V_{DS},V_{GS})$	24
II .1.5 Caractéristiques électriques idéales du transistor MOS	25
II.2 LES Afficheurs LCD.....	25
II.2.1 Généralités.....	25
II.2.2 brochage.....	26
II.2.3 Les mémoires.....	26
II .3 Microcontrôleur, 16F876A caractéristique et brochage	27
II .3.1 Introduction.....	27
II .3.2 Définition	27
II.3 .3 La composition d'un microcontrôleur	27
II .3 .4 Microcontrôleur de la Famille Pic	28
II .3 .4.1 Généralités	28
II.3.4.2 Classification des PIC.....	29
II .3.4.3 Identification des PIC	29
II .3. 4.4 Choix du PIC 16F	29
II .3 .5 Brochage du PIC16F876A	30
I .3.5.1 MCLR	30
II.3. 5.2 OSC1 et OSC2	30
II .3. 5.3 VDD et VSS	31
II .3 .5.4 Les ports entrée/sortie	31

II.4 Protocole Bluetooth	31
II .4.1 Technologies Bluetooth	32
II.4.2 Présentation de l'interface Bluetooth	32
II .4 .3 Protocoles Bluetooth	32
II.4. 3 .1 La couche « Radio »	33
II .4. 3 .2 La couche « bande de base » (Base Band)	33
II.4. 3 .3 Le « contrôleur de liaisons » (Link Controller)	33
II.4. 3 .4 Le « gestionnaire de liaisons » (Link Manager)	33
II .4. 3 .5 L'interface de contrôle de l'hôte (Host Controller Interface ou HCI)	33
II.4. 3 .6 La couche L2CA	33
II.4.3 .7 Les protocoles	33
II .5 Système androïde	33
II.5.1 Définition	33
II.5. 2 Différentes versions de l'Androïde	34
II .5 .3 Architecture de base androïde	34
II .5. 4 Composants applicatifs	35
II.5. 4.1 Activité	35
II.5. 4.2 Service	35
II .5. 4.3 Content Provider(Le fournisseur de contenu).....	35
II .5. 4.4 Le gadget (widget)	35
II.5. 4.5 Intent	35
II.6 Comment créer une application	35
II.7 AppInventor	36
II.7.1 C'est quoi AppInventor ?.....	36
II.7.2 Comment utiliser AppInventor ?.....	36
II.7.3 Que faire avec AppInventor ?.....	36
II.7.4 Pourquoi avoir utilisé AppInventor.....	36

Chapitre III :Etude du circuit et résultats de simulation

III.1 Circuits sur ISIS	39
III .1.1Définition	39
III.2 principe de fonctionnement	39

III.2.1. command numérique	39
III .2.1.1 Introduction	39
III .2.1.2 configuration minimale du PIC	40
III .2.1.3 Fonctionnement de l’afficheur LCD	40
III 2.1.4 La commande des moteurs pas a pas	40
III .2.2 circuits de puissance	41
III. 2.2.1 Introduction	41
III. 2.2.2 Principe de fonctionnement.....	41
III. 2.2.3 Utilisation avec le moteur à courant continu	42
III .2.3 contrôle du sens de rotation :	42
III. 2.3.1 Explication	46
III .3 Les Organigrammes	46
III .3.1 Définition	46
III. 3.1.1La Fonction Main (le programme principal)	47
III .3.1.2 La Gestion de l’Afficheur LCD	47
III .3.1.3 La Gestion des boutons poussoirs	48
III .4 Liste des composants	49

Chapitre IV : réalisation pratique

IV.1 La réalisation du circuit imprimé.....	52
IV.1.1 le choix d’une technique pour fabriquer un circuit imprimé.....	52
IV.1.2 La création du typon.....	52
IV.1.3 Le réglage des dimensions du PCP.....	52
IV.1.4L’impression de la carte électronique.....	53
IV.1.5 La pose du typon.....	53
IV.1.6 Le refroidissement de la carte électronique.....	53
IV.1.7 Le retrait du typon.....	53
IV.1.8 Le séchage de la carte électronique.....	53
IV.1.9 La gravure.....	53

IV.1.10 La finition de la carte.....	53
IV.2 Le circuit imprimé de la carte principale.....	54
IV.3 Implantation des composants.....	55
IV.4 vue en 3 dimensions du circuit imprimé.....	55
IV.4.1 L'outil de visualisation en 3D.....	55
IV.5 Les photos du projet réalisé.....	56
IV.5.1 Partie puissance.....	56
IV.5.2 Partie commande.....	57
IV.5.3 La carte complète (Partie commande + Partie puissance).....	57
CONCLUSION GÉNÉRALE.....	59
BIOGRAPHIQUES.....	60

LISTE DES FIGURES

Figure I.1: Moteur pas à pas.....	14
FigureI.2: Principe de base de la création d'un champ tournant.....	15
FigureI.3 : Moteur pas à pas à réluctance variable.....	16
Figure I.4 : Moteur pas à pas à aimant permanent.....	17
FigureI.5: Moteur à aimant permanent unipolaire.....	17
Figure I.6 : Moteur à aimant permanent bipolaire.....	18
Figure I.7 : Fonctionnement à pas complet.....	18
Figure I.8: Moteur pas à pas hybride.....	19
FigureII.1 : transistor MOSET.....	23
Figure II .2 : Symboles MOSFET.....	24
Figure II .3: Caractéristiques idéales d'un transistor MOS.....	25
Figure II.4 : Afficheur LCD 2X16.....	26
Figure II .5: Schéma synoptique du microcontrôleur.....	28
FigureII.6: Brochage du 16F876A.....	30
Figure II .7 : le logo de Bluetooth.....	32
Figure II .8 : Protocoles Bluetooth.....	32
Figure II.9 : Les différentes versions d'Androïde.....	34
FigureII.10 : L'architecture Androïde.....	34
Figure II .11: photo de la structure de notre application sur Androïde.....	36
Figure II .12: Programme de simulation de notre application sur Androïde.....	37
FigureIII. 1 : Schéma synoptique du montage.....	39
Figure III. 2 : Circuit de puissance.....	42
Figure III. 3 : Brochage des boutons poussoirs avec le PIC 16F876a.....	43
FigureIII. 4 : Programme de simulation du sens de rotation.....	45
Figure III.5: Circuit électrique sur ISIS.....	46
Figure III.6 : Organigramme Programme Principale.....	47
FigureIII.7 : Organigramme de la fonction LCD.....	48
FigureIII.8 : Organigramme de la gestion des boutons poussoir.....	49

Figure IV.1: Coté cuivre du circuit imprimé (carte principale).....54

Figure IV.2: Coté cuivre du circuit imprimé (carte principale).....54

Figure IV.3 : Coté composants du circuit imprimé (carte principale).....55

Figure IV.4:Implantation des composants (carte principale).....55

Figure IV.5 :Vue du circuit imprimé en 3Dimensions.....56

LISTE DES TABLEAUX

Tableau I. 1 : le mode monophasé biphasé ou demi-pas.....16

Tableau I. 2: différentstype de moteur pas à pas.....19

TableauII. 3 :caractéristiques du PIC 16F876A.....29

Tableau III .4 : combinaisons d'états des commutateurs.....41

LISTE DES IMAGES

Image IV.6 : Partie puissance du circuit.....56

Image IV.7 : Partie puissance du circuit.....57

Image IV.8 : La carte complète du projet réalisé.....58

INTRODUCTION GÉNÉRALE

Le moteur pas à pas est l'interface idéale entre l'électronique numérique et la mécanique, il permet de convertir directement un signal électrique en un positionnement angulaire à caractère incrémental.

Malgré les différences existant entre les moteurs, le résultat recherché est la rotation de leur axe suivant un angle déterminé à chaque impulsion reçue par l'une ou l'autre bobine Cet angle varie selon la constitution interne du moteur en général est compris entre 0.0° et 90° .

Les moteurs pas à pas sont utilisés pour commander avec précision la position ou la vitesse . Constitué principalement d'un stator et d'un rotor qui tourne par incrément discret chaque incrément de rotation est provoqué par une impulsion de courant fournie à l'un des enroulements du stator.

Objectif du mémoire :

Le but de notre travail est d'une part d'étudier le fonctionnement du moteur pas à pas, d'autre part de commander ce convertisseur électromagnétique avec un dispositif numérique, L'objectif de notre travail est d'élaborer une commande, afin d'assurer un fonctionnement optimal ensuite de réaliser d'une carte électronique à base d'un microcontrôleur -PIC 16F876a, également dédié et donner un aperçu sur une technologie de réseau personnel sans fil (Bluetooth).

Organisation du mémoire :

Le travail présent dans ce mémoire sera reparti en deux chapitres :

Le premier chapitre porte sur les principaux types de moteurs pas à pas tels que les moteurs à reluctances variables, les moteurs à aimant permanent et les moteurs hybrides, et est consacré à l'étude des microcontrôleurs(PIC16F876A) et aussi expliquer le système androïde.

Le deuxième chapitre est dédié à l'étude et à l'explication du circuit de commande et de puissance .Il sera consacré à la partie commande et la mise en œuvre du contrôle du moteur pas à pas par microcontrôleur pic 16f876a, en premier lieu on va présenter d'une part le schéma électrique du montage utilisé en simulation dans ISIS et l'outil de programmation Micro C

Nous terminerons notre mémoire par une conclusion générale

Chapitre I : Généralités sur les composants électriques

Clicours.COM

I.1 Introduction:

Les nombreuses applications de la commande numérique ont remis en faveur le moteur pas à pas, appelé aussi moteur à impulsions qui était utilisé depuis déjà longtemps. Les moteurs pas à pas ont été créés en 1930. Leurs développements ont commencé au début des années soixante grâce à l'apparition des microprocesseurs, microcontrôleurs et aux énormes progrès de l'électronique à transistors de puissance.

Et encore, pendant des années, l'apparition essentielle fût celle de la commande numérique des machines outil ayant besoin d'un tel type de moteur. Puis avec le développement de la recherche spatiale, de nouveaux moteurs pas à pas firent leur apparition (dimensions réduites et pas angulaires plus petits) et furent largement utilisés grâce à leurs avantages importants de fiabilité et de simplicité de commande qu'ils offraient sur les autres moteurs.

Les moteurs pas à pas sont très utilisés dans toutes les applications mécaniques où on doit contrôler la position ou la vitesse d'un système. Ces moteurs sont par exemple utilisés en robotique (positionnement des axes, vitesse variable du robot), en informatique (déplacement d'une tête de lecture d'une lecture, déplacement de la tête d'impression, avance du papier pour les imprimantes), en astronomie (positionnement du télescope pour suivre un astre) et dans la réception TV par satellites (positionnement d'une parabole). [1]

I.2 Définition:

Le moteur pas à pas constitue un convertisseur électromécanique destiné à transformer le signal électrique (impulsion) en déplacement (angulaire ou linéaire) mécanique.

Au point de vue électrotechnique, le moteur classique ressemble à la machine synchrone, dont le stator (le plus souvent à pôles saillants) porte les enroulements de pilotage et le rotor (presque toujours à pôles saillants) est soit muni d'aimants permanents (structure dite polarisée ou active), soit constitué par une pièce ferromagnétique dentée (structure dite réluctante ou passive).

Entre le moteur et son alimentation, sont intercalés trois éléments essentiels

- une unité de calcul, qui élabore les impulsions de commande.
- un modulateur PWM, qui génère les commandes des contacteurs électroniques de commutation.
- une électronique de commutation (puissance), qui, à partir d'une alimentation, fournit l'énergie vers les enroulements appropriés du moteur [2]

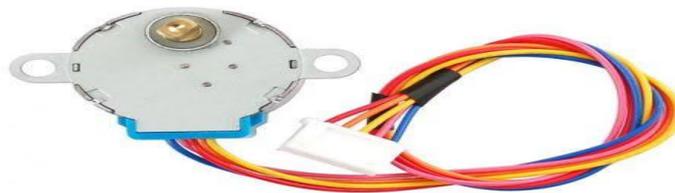


Figure I.1 : Moteur pas à pas

I.3 Principe de fonctionnement:

Le principe de base est la création d'un champ tournant, les pôles magnétiques de rotation de même nom se repoussent et les pôles des noms contraires s'attirent, le champ magnétique entraînera le rotor alimenté dans le même sens.

Ceci traduit le fait qu'on transforme une grandeur numérique en une grandeur analogique. La fréquence de rotation, ou vitesse est donc commandée par des impulsions (consigne de rotation) contrôlées elle-même par un dispositif électronique en technologie câblée ou programmée.

Théoriquement un moteur est composé d'un aimant permanent et de deux bobinages constitués chacun de deux bobines. Le passage d'un courant, successivement dans chaque bobinage, fait tourner l'aimant

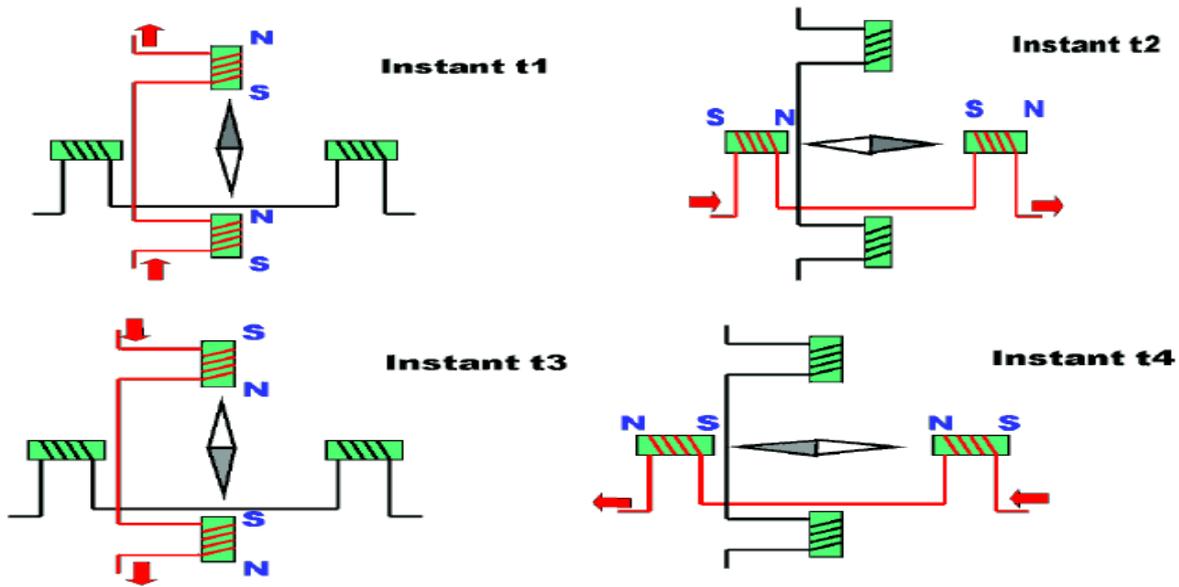


Figure I.2: Principe de base de la création d'un champ tournant

Nous avons créé un moteur de quatre pas par tour.

- Les impulsions électriques sont du type tout ou rien, c'est à dire passage de courant ou pas de passage de courant.
- L'électronique actuelle permet de piloter la chronologie de ces impulsions avec beaucoup de précision et d'en comptabiliser le nombre.
- Le moteur pas à pas et son circuit de commande permettent donc la rotation avec beaucoup de précision en vitesse et en amplitude.[3]

I.4 La technologie des moteurs pas à pas :

Il existe trois catégories des moteurs pas à pas :

I.4.1 Moteur pas à pas à réluctance variable MRV :

Les moteurs à réluctance variable fonctionnent selon un principe différent de celui des moteurs à aimant permanent. Ils possèdent bien évidemment un stator, mais ce dernier est fabriqué en acier doux non magnétique. Il n'est pas lisse et possède plusieurs dents. Ce type de moteur est représenté en (figure I.3). On peut voir, dans cet exemple, que le stator est composé de 8 plots sur lesquels enroulés les bobinages, ce qui donne 4 phases. Le rotor, quant à lui, ne comporte que 6 dents[4]

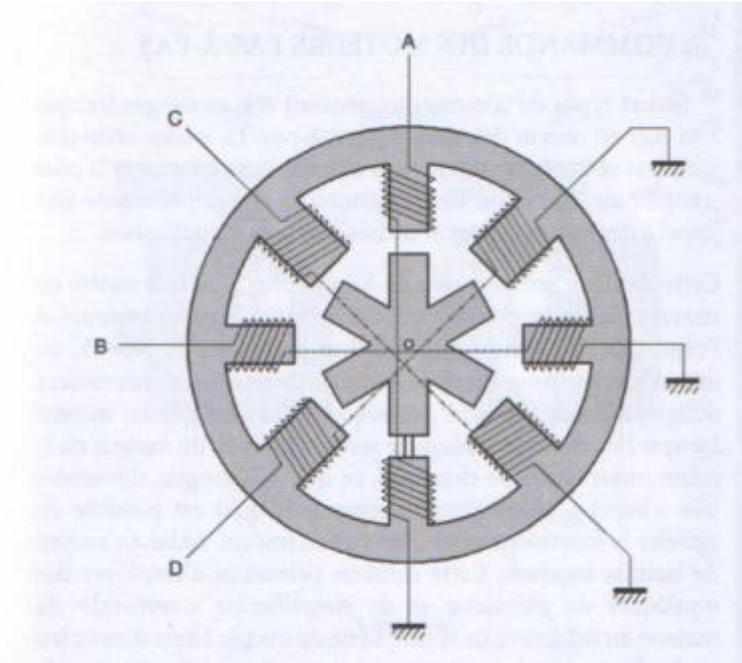


Figure I.3 : Moteur pas à pas à réluctance variable

I.4.1.1 Le principe de fonctionnement est très simple:

Le principe de fonctionnement est très simple l'une des phases alimentant deux des plots opposés du stator est traversée par un courant afin d'aligner deux des dents du rotor avec ces plots. Le mode de commande peut dans ce cas, de la même façon que pour les autres moteurs, être monophasé, biphasé ou demi-pas. Les séquences de commandes sont présentées dans le tableau suivant .[4]

Mode monophasé	Mode biphasé	Mode demi-pas
A	AC	A
B	CB	AC
C	BD	C
D	DA	BD
		D
		DA

Tableau I.1 : le mode monophasé biphasé ou demi-pas

I.4.2 Moteur pas à pas à aimant permanent MP:

Le moteur à aimant permanent est aussi appelé moteur polarisé. C'est le modèle dont le fonctionnement est le plus simple. Son rotor porte des aimants permanents qui s'alignent suivant son champ magnétique avec celui développé par le stator qui comporte deux paires de bobines. En agissant sur les bobines alimentées, et le sens des courants, on fait varier le champ créé par le stator.[1]

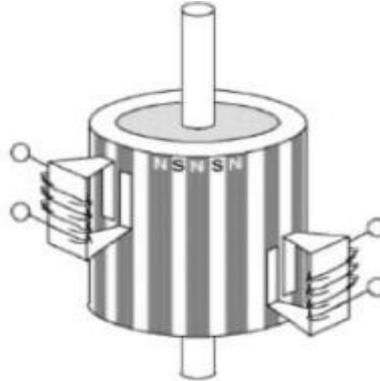


Figure I.4 : Moteur à aimant permanent

I .4.2.1 Les moteurs pas à pas unipolaires :

Les moteurs unipolaires se différencient par le fait qu'ils sont à double bobinage. Le double bobinage est utilisé pour l'inversion du flux statorique et le moteur se commande de la même manière qu'un bipolaire excepté qu'un seul transistor pour chaque enroulement suffit dans l'étage de puissance (soit quatre Darlington pour un moteur ou un réseau de 4 transistors- voir ULN 2075B). [4]

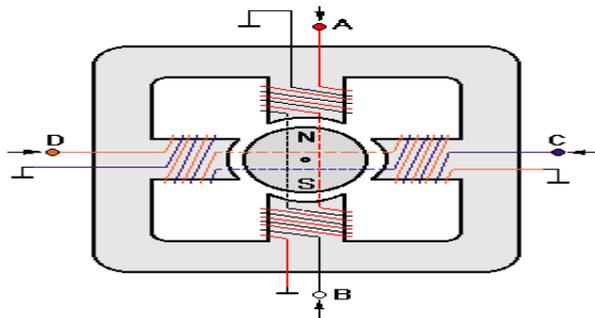


Figure I.5: Moteur à aimant permanent unipolaire

Simplement, les moteurs unipolaires sont plus chers car leur fabrication réclame un double bobinage. De plus, pour une taille donnée, ce type de moteur a un couple plus faible à cause des enroulements qui sont plus fins.

Il fut une époque où les moteurs unipolaires étaient intéressants pour les concepteurs parce qu'ils simplifiaient l'étage de commande électronique. Maintenant, grâce aux circuits de commande (push pull monolithique) du genre L298, les moteurs bipolaires sont devenus populaires et d'une utilisation courante.

Tous les moteurs à aimant permanent souffrent des oscillations (et des harmoniques qui s'ensuivent) générées par le rotor qui limitent la vitesse de rotation.

Quand des accélérations et des vitesses plus élevées sont nécessaires on utilisera de préférence les moteurs à reluctance variable. [4]

I .4.2.1.1 fonctionnement d'un moteur pas-a-pas unipolaire :

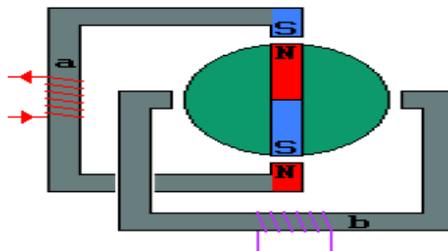
On partira du principe que la rotation d'un moteur pas à pas s'effectue en 4 étapes, dans la réalité, le moteur est constitué d'une succession d'alternances de pôles : ainsi, l'axe du modèle dont nous disposons dans notre réalisation fait un tour complet en 48 pas (un pas correspond donc à $360^\circ/48 = 7,5^\circ$).

Dans les schémas voire (**FigureI.5**), la flèche noire représente l'aiguille d'une boussole qui serait disposé en place et lieu du rotor ; elle indique l'orientation du champ magnétique (elle pointe vers le nord, qui attire donc le pôle Sud du rotor) et se décale alors d'un quart de tour à chaque étape .[4]

I .4.2.2 Moteurs pas à pas bipolaires :

Un moteur unipolaire est un moteur à deux phases statoriques avec point milieu. Chaque **demi-bobine est appelée phase**. Ce qui donne **quatre (4) phases**.

La structure interne d'un moteur de type bipolaire est représentée dans la figure ci-dessous, ce type de moteur nécessite une commande plus complexe que celle du moteur unipolaire. A chaque pas effectué, le courant devant changer de sens dans les enroulements.[3]



FigureI .6 : Moteur à aimant permanent bipolaire

I .4.2.2.1 : Fonctionnement à pas complet:

Il consiste à alimenter les enroulements en suivant la séquence A vers B/C vers D/B vers A/D vers C (BA est les mêmes enroulements que AB mais alimenté par un courantde polarité inverse). Par la suite nous simplifierons la notation pour une meilleure correspondance avec les chronogrammes des phases en indiquant uniquement la phase qui est alimentée par un courant "positif" Soit A B C D.

Cette séquence est connue sous le nom de " one phase on full step" (traduisez phase par phase ou une phase à la fois en pas entier). A tout moment il n'y a qu'une seule phase d'alimentée et nous travaillons en mode pas entier. [4]

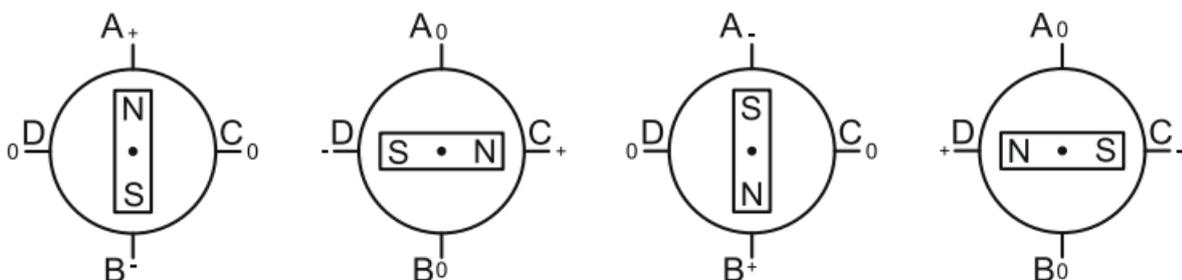


Figure I .7 : Fonctionnement à pas complet

I .4.3 Moteur pas à pas hybride :

I .4.3.1 Définition :

C'est un moteur reluctant polarisé. Il superpose le principe de fonctionnement des moteurs à aimant permanent et à reluctance variable et combine leurs avantages. Le rotor est constitué de deux disques dentés décalés mécaniquement. Entre ces deux disques est inséré un aimant permanent.[2]

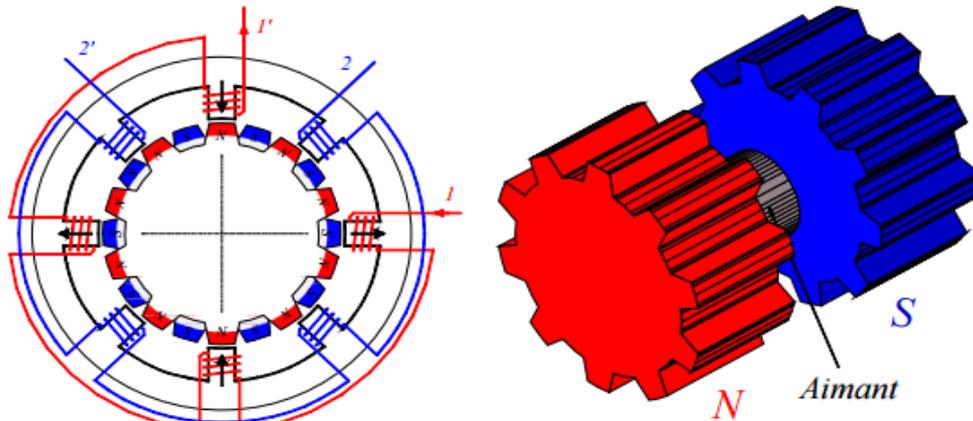


Figure I.8: Moteur pas à pas hybride

I .4.3.2 Fonctionnement d'un moteur pas à pas hybride :

En mode à pas entier, les bobines sont alimentées une par une alternativement, dans un sens puis dans l'autre. On voit bien ici l'intérêt d'avoir un rotor polarisé: quand on coupe l'alimentation de la bobine bleu et qu'on alimente la bobine rose, en l'absence de polarisation du rotor on n'aurait aucun contrôle sur le sens de rotation, les deux dents bleues et rouges étant attirées de la même façon par la bobine. Avec un rotor polarisé, on peut choisir une de ces deux dents en agissant sur le sens du courant dans la bobine.

I.5 Comparaison entre les trois types de moteurs pas à pas :

Le tableau ci-dessous présente une étude comparative des trois types de moteurs en citant les avantages de l'un par rapport à l'autre.

Type de moteur PaP	Moteur à aimant Permanent	Moteur à réluctance Variable	Moteur hybride
Résolution (nb de pas/tour)	Moyenne	Bonne	Elevée
Couple moteur	Elevée	Faible	Elevée
Sens de rotation	Dépend de : - sens du courant pour le moteur bipolaire - L'ordre d'alimentation des bobines	Dépend uniquement de l'ordre d'alimentation des bobines	Dépend de : - sens du courant pour le moteur bipolaire - L'ordre d'alimentation des bobines
Fréquence de Travail	Faible	Grande	Grande

Tableau I.2: différents type de moteur pas à pas

I.6 Caractéristiques générales des moteurs pas à pas :

Les caractéristiques sont divisées en deux, statique et dynamique :

I.6.1 Caractéristiques statiques : [1]

Ces caractéristiques concernent le moteur au repos. Elles sont essentiellement conditionnées par la construction mécanique du moteur.

➤ Angle de pas ϕ_p

C'est l'angle élémentaire effectué par le moteur lorsqu'un ou un ensemble d'enroulements sont alimentés. Cet angle est défini par les constructeurs et donné par :

$$\phi_p = (360/m) \times Nr[0] ; \text{ pour les moteurs à réductance variable}$$

m : nombre de phase

➤ La précision du pas

Ecart relative de la position réelle de l'élément mobile par rapport à une position théorique d'équilibre mesuré à vide. Cette valeur est exprimée en pourcentage du pas :

$$\text{Précision} = \text{Erreur max. d'angle} \times 100/\text{pas}$$

➤ Courant par phase

C'est l'intensité par phase à fréquence nulle (moteur à l'arrêt) qui provoque l'échauffement maximum admissible par le moteur en fonctionnement.

➤ Angle de puissance

C'est l'angle de retard de rotor par rapport au champ magnétique qui est provoqué par une charge extérieure.

➤ La résolution

C'est le déplacement minimal demandé ou l'angle du pas en degré.

I.6.2 Caractéristiques dynamiques : [1]

Le démarrage, la rotation et le freinage du moteur dépendent de la fréquence du commutateur des phases et du circuit d'alimentation pour pouvoir démarrer et l'entraîner sans perdre de pas.

Donc, il est nécessaire de connaître ces caractéristiques.

➤ Le temps de montée (T_m)

C'est le temps nécessaire pour passer d'une position d'équilibre à une autre. Il est repéré par T_m

➤ Le temps d'établissement (T_e)

C'est le temps nécessaire à établir la position du rotor autour d'une position d'équilibre avec une bande d'erreur de position donnée

➤ Fréquence maximale réversible (réponse maximale)

Fréquence maximale permettant au moteur de changer le sens de rotation en restant en synchronisation avec les impulsions d'excitation des phases.

➤ Fréquence maximale de démarrage

C'est la fréquence maximale jusqu'à laquelle le moteur peut suivre la commande au moment de démarrage ; cette fréquence est légèrement plus haute que la fréquence maximale réversible.

➤ Couple maximale dynamique

C'est le couple maximal disponible sur l'arbre lorsque les phases sont alimentées successivement.

➤ Couple de démarrage et d'entraînement

C'est le couple maximal disponible sur l'axe de moteur lors de démarrage (entraînement) à une fréquence donnée sans perdre de pas.

I.7 Alimentation des moteurs Pas à pas:

- Un moteur Pap alimenté à son courant nominal peut chauffer de 60 à 70 degrés par rapport à la température ambiante. Attention aux brûlures!
- Un moteur qui chauffe plus à l'arrêt qu'en fonctionnement est sous-alimenté (cas typique du moteur alimenté en tension, dès que la vitesse de rotation augmente). Ce n'est un problème que si la puissance disponible est insuffisante, il faut alors passer à un mode d'alimentation «en courant».
 - Ceci n'est vrai qu'en mode de fonctionnement à pas entier.
- Calcul de la résistance pour une alimentation en courant contrôlé par résistance série (choisir une résistance non inductive) :
 - $R = (V_{\text{Alim}} - V_{\text{Moteur}}) / I_{\text{Moteur}}$
 - Puissance dissipée $P = (V_{\text{Alim}} - V_{\text{Moteur}}) * I_{\text{Moteur}}$
 - Ex: un moteur prévu pour 5V, 200mA, alimenté sous 15V nécessite une résistance de $(15 - 5) / 0,2 = 50 \text{ Ohms}$, $(15 - 5) * 0,2 = 2 \text{ Watts}$. [5]

I.8 Les avantages et les inconvénients des moteurs pas à pas : [5]

Avantages

- Rotation constante pour chaque commande (précision meilleure que 5% d'un pas).
- Existence de couple à l'arrêt.
- Contrôle de la position, de la vitesse et synchronisation de plusieurs moteurs (pas besoin de contre-réaction).
- Moteur sans balais.

Inconvénients

- Plus difficile à faire fonctionner qu'un moteur à courant continu.
- Vitesse et couple relativement faible.
- Couple décroissant rapidement lorsque la vitesse augmente.
- Résonance mécanique

Chapitre II: Etudes des composants électriques du circuit

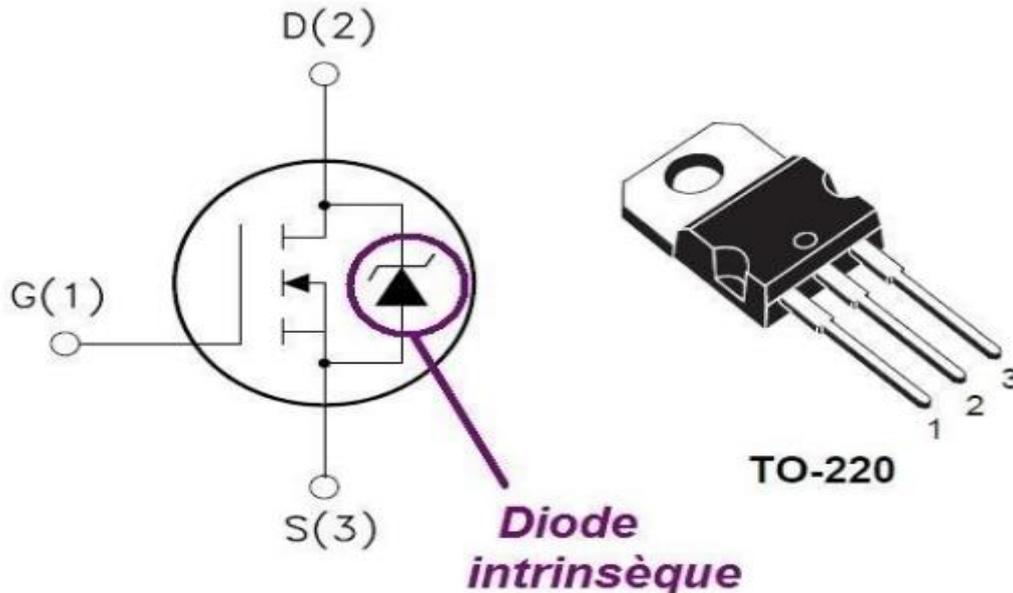
II.1 Transistors MOSFET:

II .1.1 Historique:

La théorie sur les transistors à effet de champ (*Field Effect Transistor*) a été conçue dans les années 1920 – 1930, soit 20 ans avant que le transistor bipolaire (*Bipolaire Junction Transistor*) fut inventé. A cette époque J.E. Lilienfeld (USA) propose un modèle de transistor basé sur le contrôle du courant par l'application d'un champ électrique. Par manque de matériaux semi-conducteurs appropriés, et d'une technologie immature le développement de ce type de transistor fut très lent. William Shockely proposa un premier transistor JFET en 1952. Les premiers produits industriels firent définitivement leur apparition dans les années 1970.[6]

II .1.2 Définition

Les transistors MOS sont aussi des éléments à effet de champ, mais la grille, au lieu d'être une jonction bloquée, est un isolant (oxyde). Le symbole MOS signifie *metal-oxyde-semiconductor*. L'appellation complète, un peu vieillie, est MOSFET (*metal-oxyde-semiconductorfieldeffect transistor*). On nomme aussi ce composant transistor à effet de champ à grille isolée, ce qui correspond aux initiales IGFET (*insulatedgatefieldeffect transistor*). On retrouve les deux types de canaux, N ou P. On distingue également les MOS à appauvrissement



FigureII.1 : transistor MOSET

II.1.3 : Les Symboles du transistor mosfet:

Symboles MOSFET : Source, D : Drain, G : Grille (Gate), B : Substrat (Bulk). Il existe quatre types de transistors MOS : le MOSFET canal N à enrichissement, le MOSFET canal P à enrichissement, le MOSFET canal N à appauvrissement et le MOSFET canal P à appauvrissement. La flèche figurant sur le substrat (B) des symboles IEEE indique le sens passant des jonctions substrat-source (BS) et substrat drain (BD). Ces jonctions doivent être bloquées. Les symboles simplifiés ne visualisant pas les substrats, on suppose que les jonctions BS et BD sont bloquées [8]

Clicours.COM

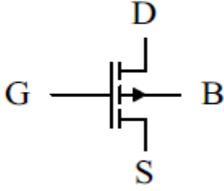
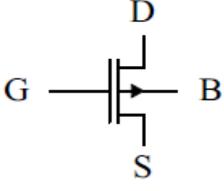
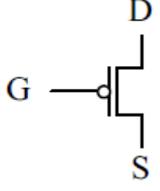
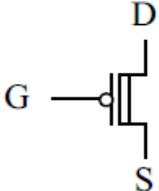
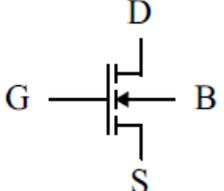
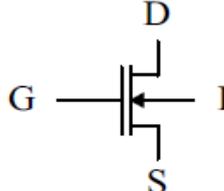
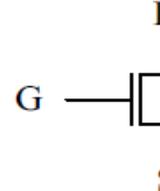
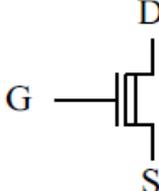
MOSFET	Symboles normalisés IEEE		Symboles simplifiés	
	à enrichissement	à appauvrissement		
Canal P				
Canal N				

Figure II. 2 : Symboles MOSFET

II .1.4 : Mode de fonctionnement dumosfet :

II.1.4.1 : Comportement à l'état ouvert:

En l'absence de polarisation positive de la Grille, le MOSFET est bloqué. La tension Drain – Source peut croître jusqu'à la tension d'avalanche $V_{BR}(SS)$. Seul un petit courant de fuite circule dans le MOSFET. Le mécanisme d'avalanche trouve son origine dans plusieurs facteurs que nous n'étudierons pas ici. [6]

II.1.4.2 : Comportement à l'état fermé : caractéristique statique

$I_D=f(V_{DS},V_{GS})$:

Lorsque la tension Grille – Source est fixée à une valeur constante supérieure à $V_{GS}(Th)$ et que la tension VDD augmente, le courant de Drain I_D augmente linéairement. Mais comme le montre la caractéristique $I_D=f(V_{DS},V_{GS})$ de la Figure 7-7, lorsque la tension VDS atteint un certain niveau, l'accroissement du courant de Drain I_D diminue pour devenir quasi constant et indépendant de VDS.

Pour comprendre ce phénomène, il faut observer la chute de tension $V_{CS}(x)$ due au passage du courant I_D dans la zone d'inversion en fonction de la résistivité du canal et de la position x . Cette tension est donnée par la relation

$$V_{CS}(x) = v_{GS} - v_{ox}(x)$$

Ou $V_{ox}(x)$ est la tension Grille – Canal en fonction de la position x et donc la valeur maximum est donnée pour $x=L$.

Pour de faibles valeurs de VDD, le courant I_D est faible et il n'y a presque pas de chute de tension aux bornes du canal sur la longueur L . $V_{ox}(0) \cong V_{ox}(L)$ ce qui provoque une largeur constante de la zone d'inversion. En augmentant VDD, le courant de Drain I_D va augmenter et la

chute de tension aux bornes de la zone d'inversion devient importante. Cette chute de tension linéique $V_{CS}(x)$ provoque une réduction de la largeur de la zone d'inversion pour des x croissants. Cet effet entraîne à son tour une augmentation de la résistance du canal et donc une diminution du courant I_D . Lorsque $V_{OX}(L)=V_{GS}-V_{CS}(L)=V_{GS}(Th)$, la largeur de la zone d'inversion atteint sa valeur minimum ce qui limite la vitesse de déplacement des porteurs minoritaires excédentaires présents .[6]

II.1.5 : Caractéristiques électriques idéales du transistor MOS :

Les transistors MOS sont caractérisés électriquement en utilisant les graphiques $I_{ds}=f(V_{gs})$ et $I_{ds}=f(V_{ds})$. Ces caractéristiques sont schématisées de manière idéale en figure I .12, ce qui nous permet de relier les différents régimes de la capacité MOS définis dans le paragraphe précédent aux modes de fonctionnement du transistor MOSFET .[9]

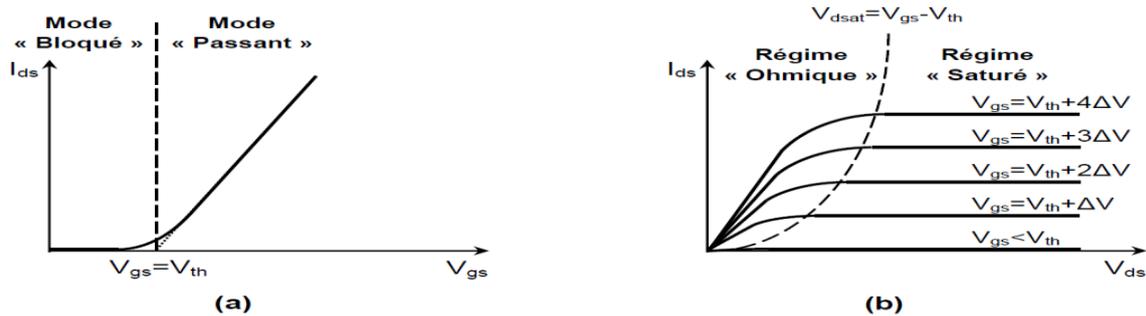


Figure II .3 : Caractéristiques idéales d'un transistor MOS.

(a) Graphique I_{ds} - V_{gs} .

(b) Graphique I_{ds} - V_{ds} . La ligne $V_{dsat}=V_{gs}-V_{th}$ sépare le régime ohmique du régime saturé.

II.2 Les afficheurs LCD

II.2 .1Généralités

Les afficheurs LCD sont devenues incontournables dans toutes applications qui demandent la visualisation de paramètres, il s'agit donc d'une interface Homme/Machine. Auparavant onéreux et difficiles à mettre en œuvre, ils sont maintenant bon marchés et l'interface parallèle au standard Hitachi permet un pilotage facile.

On rencontre aussi de plus en plus d'afficheurs pilotables avec un port série ou I2C.

Les afficheurs LCD se ressemblent tous, à part le nombre de lignes et le nombre de colonnes, le fonctionnement et le brochage est standard et identique. Un des points intéressant est de pouvoir contrôler l'afficheur en mode 8bits ou en mode 4bits.

II.2 .2Le brochage

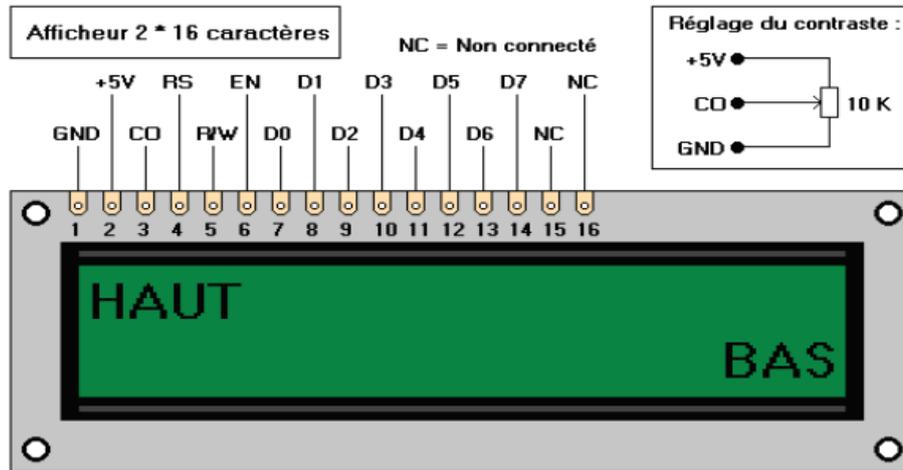


Figure II.4 : Afficheur LCD 2X16

L'afficheur LCD a 14 broches en standard et souvent 16 ,les broches 15 et 16 servent au rétro-éclairage(une option).

Broche	Nom	Description
1	Vss	Masse
2	Vdd	Alimentation 5v
3	CO	Variables de 0 à 5v permet de modifier le contraste de l'afficheur
4	RS	Indique une commande ou une donnée à afficher (0: Commande / 1: Donnée)
5	R/W	Indique une écriture ou une lecture (0: Écriture / 1: Lecture)
6	E	Indique une validation (Le niveau Haut doit être maintenu 500µs)
7	D0	Bus de donnéesbidirectionnel.
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anode rétroéclairage (+5V)
16	K	Cathode rétroéclairage (masse)

II.2Les mémoires

Les afficheur LCD possède 2 types de mémoires :

- la **DD RAM** qui mémorise les caractères affichés à l'écran ;
- la **CG RAM** qui contient la table des caractères affichables.

DD RAM (Display Data RAM)

La DD RAM commence à l'adresse 0x00 et dans le cas d'un afficheur 16x2 lignes, elle termine à 0x4F. C'est une mémoire d'affichage dont l'adresse contient le caractère affiché à l'écran à une certaine position.

La première ligne commence en 0x00 jusqu'à 0x0F incluse.
La seconde ligne commence en 0x40 jusqu'en 0x4F incluse.

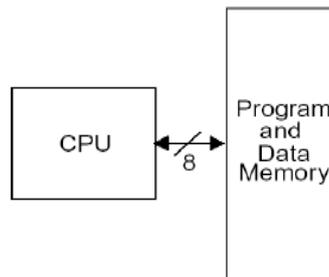
II .3 Microcontrôleur, 16F876A caractéristique et brochage:

II .3 .1 Introduction : [10]

Ces microcontrôleurs méritent bien une introduction. En effet, ils ont été conçus sur une architecture dite **HARVARD (RISC)** et non sur un modèle **VON NEUMANN (COMPLEX)**.

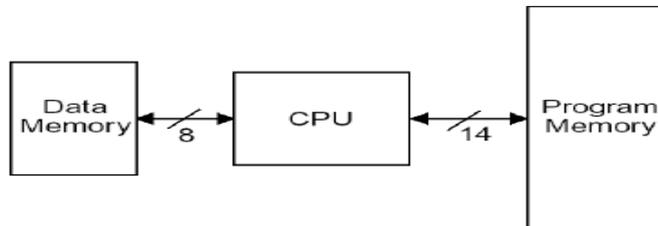
- L'architecture **VON NEUMANN** employée par la plupart des microcontrôleurs actuels (INTEL80XX, motorola HC05, HC08 et HC11, ou ZILOG Z80) est basée sur un bus de données unique. Celui-ci véhicule les instructions et les données.

von-Neumann



- L'architecture **HARVARD** utilisée par les microcontrôleurs PICS est basée sur deux bus de données. Un bus est utilisé pour les données et un autre pour les instructions.

Harvard



II .3.2 Définition :

Un microcontrôleur est une unité de traitement de l'information de type microprocesseur contenant tous les composants d'un système informatique, à savoir microprocesseur, des mémoires et des périphériques (ports, timers, convertisseurs...).

Chaque fabricant a ses familles de microcontrôleurs. Une famille se caractérise par un noyau commun (le microprocesseur, le jeu d'instruction...). [11]

II.3.3 La composition d'un microcontrôleur :

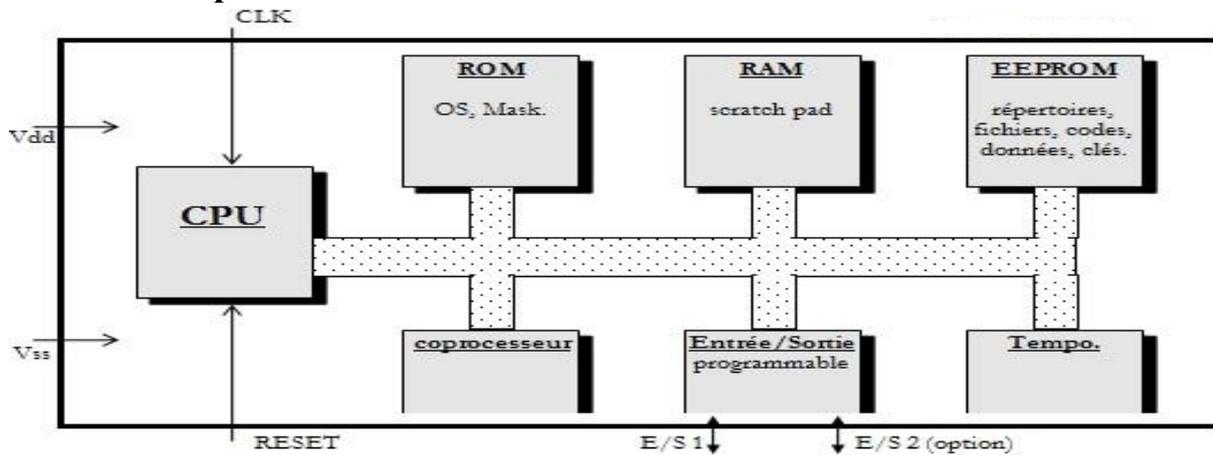


Figure II.5: Schéma synoptique du microcontrôleur

Le microcontrôleur est composé de quatre parties : [12]

Un microprocesseur (CPU) qui va prendre en charge la partie traitement des informations et envoyer des ordres. Il est lui-même composé d'une unité arithmétique et logique (UAL) et d'un bus de données. C'est donc lui qui va exécuter le programme embarqué dans le microcontrôleur.

Une mémoire de (RAM ou EEPROM) dans laquelle seront entreposées les données temporaires nécessaires aux calculs. C'est en fait la mémoire de travail qui est donc volatile.

Une mémoire programmable (ROM) qui va contenir les instructions du programme pilotant l'application à laquelle le microcontrôleur est dédié. Il s'agit ici d'une mémoire non volatile puisque le programme à exécuter est à priori toujours le même. Il existe différents types de mémoires programmables que l'on utilisera selon l'application. Notamment :

- OTPROM : programmable une seule fois mais ne coûte pas très cher.
- UVPROM : on peut la ré-effacer plusieurs fois grâce aux ultraviolets.
- EEPROM : on peut la ré-effacer plusieurs fois de façon électrique comme les mémoires flash.

La dernière partie correspond aux ressources auxiliaires. Celles-ci sont généralement :

- Ports d'entrées / sorties parallèle et série.
- Des timers pour générer ou mesurer des signaux avec une grande précision temporelle.
- Des convertisseurs A/N pour traiter les signaux analogiques.

II.3.4 Microcontrôleur de la Famille Pic :

II.3.4.1 Généralités: [13]

La société Américaine Microchip Technology a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (Peripheral Interface Contrôler), ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

Les PIC sont des composants RISC (**R**educe **I**nstructions **C**onstruction **S**et), ou encore composant à jeu d'instructions réduit [9]. L'avantage est que plus on réduit le nombre d'instructions, plus facile et plus rapide en est le décodage, et plus vite le composant fonctionne.

Les PIC existent en 3 versions :

- * les EEPROM et flash EPROM qui sont effaçables électriquement.

- * les UVPROM qui sont effaçable par une source de rayonnements ultraviolets.
- * les OTPROM programmable une seule fois

II.3.4.2 Classification des PIC : [13]

La famille des PIC de Microchip est subdivisée en 3 grandes, familles comportant chacune plusieurs références :

- * La famille **Base-Line**, qui utilise des mots d'instructions de 12 bits,
- * La famille **Mid-Range**, qui utilise des mots de 14 bits (et dont font partie les 16FXXX),
- * La famille **High-End**, qui utilise des mots de 16 bits (18FXXX).

II.3.4.3 Identification des PIC: [13]

Un PIC est généralement identifié par une référence de la forme suivante : xx(L) XXyy-zz
xx : famille du composant, actuellement « 12, 14, 16,17 et 18 ».

L : tolérance plus importante de la plage de tension.

XX : type de programme

C : EPROM ou EEPROM

F : flash

yy : identificateur

zz : vitesse maximale du quartz de pilotage.

II.3. 4.4 : Choix du PIC 16F :

Le circuit principal que nous utilisons dans le projet est le PIC 16F876A, ce circuit de Microchip est un microcontrôleur de 28 broches, de haute performance, à mémoire FLASH. Les principales caractéristiques et constituante sont présentées dans le **tableau I.3** suivant.

Caractéristique	PIC 16F876A
Fréquence Horloge	20 MHz
Mémoire programme FLASH	8 KO
Programme (instruction)	35
Mémoire données type RAM	368 Octets
Mémoire EEPROM	256 Octets
Interruption	13
Ports parallèles	A, B, C
Timers	3
Capture/Compare/PWM	2
Communications séries	MSSP / USART
Communications parallèles	–
CAN 10-bit	10-bits à 8 entrées multiplexées
RESETS(and Delays)	POR, BOR (PWRT, OST)
Quartz	8 MHz

TableauII.1 :caractéristiques du PIC 16F876A

II .3 .5 : Brochage du PIC16F876A :

Le boîtier du PIC 16F876a décrit par la figure.2 comprend 28 pins : 22 pins D'entres/sorties, 3 pins pour l'alimentation, 2 pins pour l'oscillateur et une pin pour lereset..

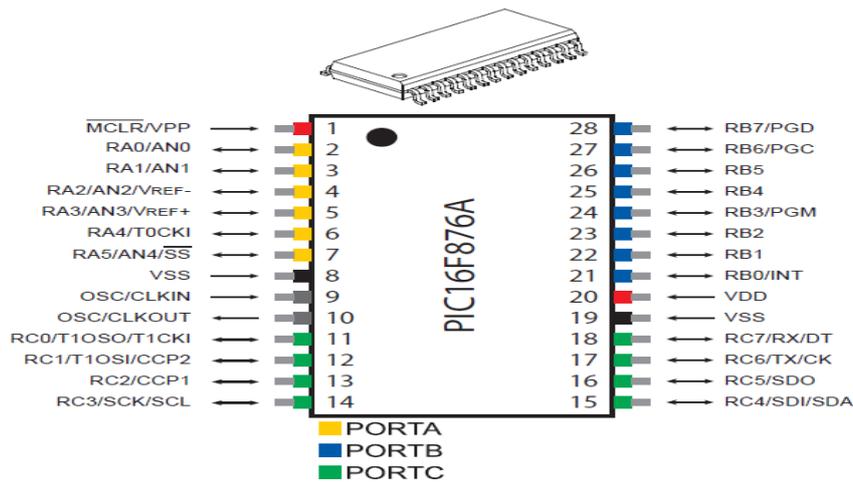


Figure II.6 : Brochage du 16F876A

II.3.5.1 MCLR: cette broche sert à initialiser le microcontrôleur PIC:[14]

- à la mise sous tension par un front montant (min 72 ms, max 72 ms +1024 x T_{osc}).

Cette initialisation est appelée POR (POWER ON RESET). Cette broche peut être simplement reliée à l'alimentation VDD si on n'a pas besoin de RESET externe.

- par la mise à l'état bas de cette broche, reset externe (EXTERNAL RESET).

Si on souhaite implanter un bouton de remise à zéro, on pourra câbler un simple réseau RC sur la broche MCLR.

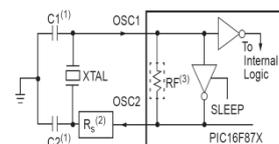
Remarque importante :

On peut se passer de circuit RC à la seule condition que le temps de montée de la tension d'alimentation VDD soit suffisamment rapide (au minimum 50 mV/ms).

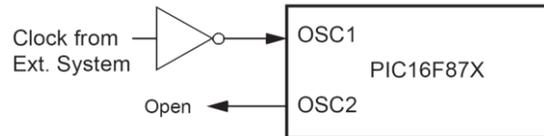
Le microcontrôleur PIC possède d'autres sources de reset mais cette fois-ci internes (WDT et BOR vues précédemment).

II.3.5.2 :OSC1 et OSC2: ces broches permettent de faire fonctionner l'oscillateur interne du microcontrôleur PIC de trois façons différentes.[14]

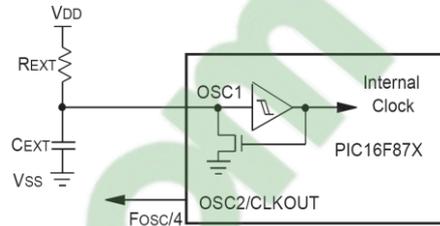
- Un quartz ou résonateur céramique permet d'obtenir une fréquence de fonctionnement très précise (voir document constructeur pour les valeurs des condensateurs C1, C2 et de la résistance R_S).



- Un oscillateur externe permet une synchronisation avec un autre circuit.



▪ Un simple réseau RC peut suffire, l'oscillateur est peu précis mais économique.



II .3.5.3VDD et VSS: broches d'alimentation du circuit (la tension VDD peut être comprise entre 4V à 5,5V).[14]

II .3 .5.4Les ports entrée/sortie :[15]

Le PIC16F876 est équipé de 22 lignes d'entrées/sorties reparties en trois ports parallèles bidirectionnels :

- 6 lignes sur le port A : RA0 à RA5,
- 8 lignes sur le port B : RB0 à RB7,
- 8 lignes sur le port C : RC0 à RC7.

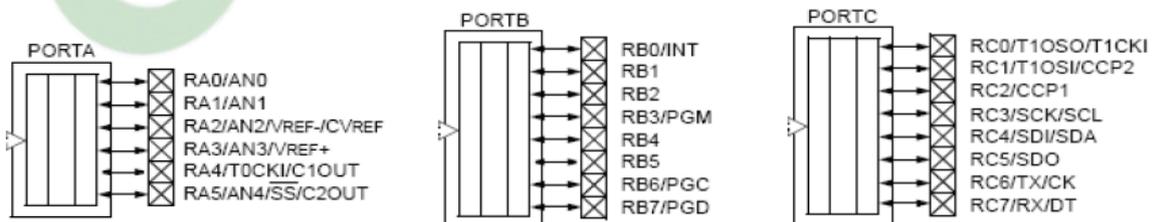
Chaque ligne peut fournir ou absorber au maximum un courant de 25 mA.

L'ensemble des ports peuvent fournir ou absorber au maximum un courant de 200 mA.

La plupart de ces lignes ont une double fonction suivant leur programmation.

La plupart de ces lignes ont une double fonction suivant leur programmation.

- Le port A (6 Bits) : I/O pur et/ou Entree du CAN et/ou Entree du TIMER0 (la broche RA4 du port A (entree du timer 0 T0CKI) est de type Drain ouvert)
- Le port B (8 Bits) : I/O pur et/ou programmation InSitu (ICSP/ICD) [Broches RB3/PGM, RB6/PGC ainsi que RB7/PGD] et/ou entée d'interruption externe RB0/INT.
- Le port C (8 Bits) : I/O pur et/ou TIMER 1 et/ou SPI/I2C et/ou USART.



II.4Protocole Bluetooth:

II.4.1 Technologies Bluetooth :

Du point de vue historique, le mot « Bluetooth™ » est tiré d'un surnom associé au roi Danois Harald Blåtand (en français « Dent Bleue ») [BLTH01]. Développé à l'origine par le constructeur Suédois Ericsson en 1994, ce n'est qu'en 1998 qu'un groupe, composé au départ de cinq sociétés dont Ericsson, décide de poursuivre le projet et de produire les spécifications Bluetooth 1. Ce groupe, nommé SIG – pour SpecialInterest Group – joue un rôle important dans le développement de cette technologie en l'intégrant à leurs différents produits et en la commercialisant. [16]



Figure II.7 : Le logo de Bluetooth

II.4.2 Présentation de l'interface Bluetooth :

Le Bluetooth est une technologie de réseau personnel sans fil(noté **WPAN** pour **Wireless Personnel Area Network**), c'est-à-dire une technologie de réseaux sans fil à faible portée(quelques dizaines de mètres). Elle permet de relier plusieurs appareils entre eux sans liaison filaire, en utilisant les ondes radiocomme support de transmission. La liaison radio fonctionne dans une bande de fréquence située autour de 2,45 GHz. Cette bande de fréquence étant libre dans la plupart des pays, ce qui permet d'utiliser les équipements Bluetooth partout dans le monde.[17]

II .4.3 Protocoles Bluetooth :[17]

Comme tout réseau, la technologie Bluetooth peut être décrite avec une notion de couche mais son modèle est différent du modèle OSI. On parle de **piles de protocoles**.



FigureII.8 :Protocoles Bluetooth

La couche interface ou HCI (Host Interface Controller) fait le lien entre les couches physiques (module) et les couches applicatives (hôte).

II.4.3 .1 La couche « Radio » :

La couche « radio » est gérée au niveau matériel. C'est elle qui s'occupe de l'émission et de la réception des ondes radio. Elle définit les caractéristiques telles que la bande de fréquence et l'arrangement des canaux, les caractéristiques du transmetteur, de la modulation, du receveur, etc.

II.4.3 .2 La couche « bande de base » (Base Band) :

La Couche « bande de base » est également gérée au niveau matériel. Cette couche définit les adresses matérielles des périphériques nommées **BD_ADDR (Bluetooth Device Address)** et codées sur 48 bits. Cette couche gère également les différents types de connexions entre les appareils : synchrones ou asynchrones.

II.4.3 .3 Le « contrôleur de liaisons » (Link Controller) :

Le « contrôleur de liaisons » gère la configuration et le contrôle de la liaison physique entre deux appareils.

II .4. 3 .4 Le « gestionnaire de liaisons » (Link Manager) :

Le « gestionnaire de liaisons » gère les liens entre les périphériques maîtres et esclaves Elle permet la mise en place des mécanismes de sécurité (l'authentification, le pairage, la création et la modification des clés et le cryptage).

II.4. 3 .5 L'interface de contrôle de l'hôte (Host Controller Interface ou HCI) :

Cette couche fournit une méthode uniforme pour accéder aux couches matérielles. Son rôle de séparation permet un développement indépendant du hardware et du software. Les protocoles de transport suivants sont supportés : USB (Universal Serial Bus), PC Card, RS-232 ou UART.

II.4.3 .6 La couche L2CA :

La couche L2CAP (Logical Link Control & Adaptation Protocol) permet d'utiliser simultanément différents protocoles de niveaux supérieurs. Un mécanisme permet d'identifier le protocole de chaque paquet envoyé pour permettre à l'appareil distant de passer le paquet au bon protocole, une fois celui-ci récupéré. Elle gère également la segmentation (et le réassemblage) des paquets de protocoles de niveaux supérieurs en paquets de liaison de 64 Ko.

II .4 3 .7 Les protocoles :

Le protocole **RFCOMM** basé sur les spécifications RS-232 permet l'émulation des liaisons séries. Il peut notamment servir à faire passer une connexion IP par Bluetooth.

Le protocole **SDP (Service Discovery Protocol)** permet à un appareil Bluetooth de rechercher d'autres appareils et d'identifier les services disponibles.

Le protocole **OBEX (Object Exchange)** permet le transfert des données grâce à grâce au protocole d'échange de fichiers IrDA.

II.5 Système androïde

II .5.1 Définition :

Androïde est un OS pour téléphone mobile et tablette tactile, promu par Google et l'Open Handset Alliance qui comprend plus de 35 constructeurs, fournisseurs de logiciel, et opérateurs. Il concurrence des plateformes telles que l'iOS d'Apple, Windows Mobile de Microsoft, RIM OS intégré dans les BlackBerry de Research In Motion, WebOS d'HP Bada de Samsung, ou encore Symbian et MeeGo de Nokia . [18]

II .5.2 Différentes versions de l'Androïde:

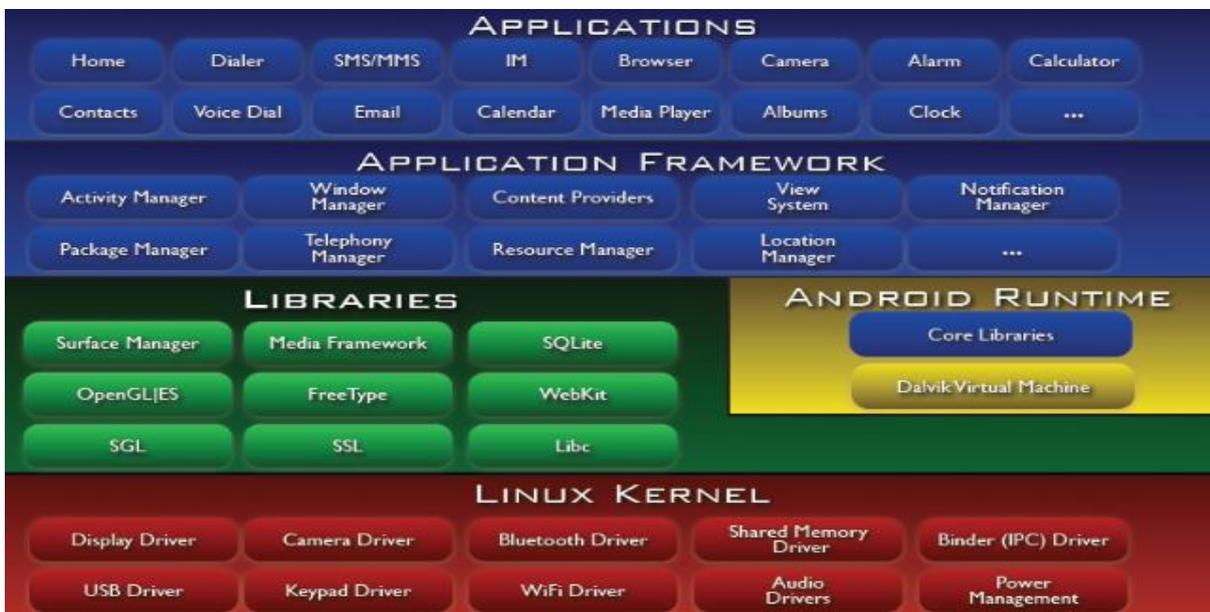


Android profite d'une structure entièrement en couches, ce qui le rend capable de développer ce qui est souvent important en termes de nouvelles fonctionnalités et améliorations, La figure suivante montre les différentes versions d'Android.



FigureII.9: Les différentes versions d'Android

II.5.3 Architecture de base android :



FigureII.10 :L'architecture Android

Le système d'exploitation Android est conçu sur la base du noyau Linux et est développé par Google . Android a une architecture en couches, comprenant la couche du noyau Linux, la couche intermédiaire et la couche d'application, qui peut fournir des services cohérents pour la couche supérieure, masque les différences de la couche actuelle et de la couche inférieure . La couche intermédiaire d'Android exécute des fonctions centrales qui peuvent être implémentées par des langages de programmation comme JAVA / C / C ++. La plupart des applications exécutées sur Android sont écrites en langage de programmation Java, puis ces multiples fichiers de classe java sont convertis au format dex par l'outil dex2jar Android. Chaque application Android est une instance indépendante à exécuter dans DVM et possède un numéro d'identification unique appelé PID. La figure 1 donne de brèves informations sur l'architecture d'un système d'exploitation Android. Dalvik Virtual Machine (DVM) [21] est le principal composant de la plate-forme Android par rapport aux autres composants d'Android. Il prend

entièrement en charge toutes les applications Java converties au format dex (DalvikExecutable). Le format dex est le format compressé du code exécutable Dalvik, qui convient à la mémoire et à la vitesse du processeur. Le code Dalvik est responsable de la ségrégation des processus et de la gestion des threads. Chaque application Androïde correspond à une instance distincte de la machine virtuelle Dalvik et peut être exécutée dans une machine virtuelle. [19]

II .5.4 Composants applicatifs : [20]

II.5.4.1 Activité : elle représente le bloc de base d'une application .elle correspond à la partie présentation de l'application et fonctionne par le biais de vue qui a-che des interfaces graphique et répond aux actions utilisateur.

II.5.4.2 Service: Un Service est utilisé pour réaliser l'application en arrière-plan, de manière invisible. C'est-à-dire, cette application peut marcher quand d'autres sont en train de s'exécuter comme les services de lecture de musique.

II.5.4.3Content Provider(Le fournisseur de contenu): Ils ont un niveau d'abstraction pour l'accès à toutes les données stocké sur le terminal ainsi le partage des données entre les applications via les Chiers, ou une base de données de SQLite. [M1]. les données sont identifiées au moyen d'URI (Unied Ressources Identier).

II.5. 4.4 Le gadget (widget): Le gadget est un composant graphique qui s'installe sur le bureau androïde. Le calendrier qui a-che de l'information ou le lecteur audio qui permet de contrôler la lecture de chiers.

II .5. 4.5 Intent : [4]

- Composant du système Androïde faisant office d'IPC (Inter Pprocess Communication) entre applications soit Activités ou Services
- Permet d'invoquer d'autres activités ou services
- Exposer des fonctionnalités aux autres activités en guise de réutilisation de code
 - L'activité Camera permet de prendre des photos
 - Le "browser" peut ouvrir une page web via une URL
 - L'application email peut envoyer des emails

II.6Comment créer une application?

1. Sélectionnez un design
Personnalisez-la en fonction de vos besoins.
2. Ajoutez les fonctionnalités désirées



Figure II.11 : photo montrant la structure de notre application sur Androïde

II.7App Inventor

II.7.1C'est quoi App Inventor?

App Inventor est une application développée par Google et est destiné aux utilisateurs d'android. C'est un logiciel en ligne qui permet de créer des applications pour appareils Android à travers une interface purement visuelle et de configurer les actions de l'application par un système de blocs logiques. Nous utilisons ce logiciel afin de crée une application pour simplifier le quotidien des personnes atteinte de diabète.

II.7.2Comment utiliser appinventor?

App inventor est très simple a utiliser, il suffit de combiner certains »bloc «(ligne de code) avec d'autres afin d'avoir un programme assez satisfaisant.

II.7.3Que faire avec appinventor?

Avec appinventor, vous pouvez crée des applications pour android que vous pourrez par la suite publié sur googleplay. Dans notre cas, nous utilisons appinventor afin de crée une application pour aider les personnes atteintes de diabète.

II.7.4Pourquoi avoir utiliséappinventor?

App inventor reste le moyen le plus simple pour créer une application pour android et surtout par des lycéens qui n'ont parfois jamais fait de programmation. Il suffit d'avoir un raisonnement logique pour utiliser appinventor.

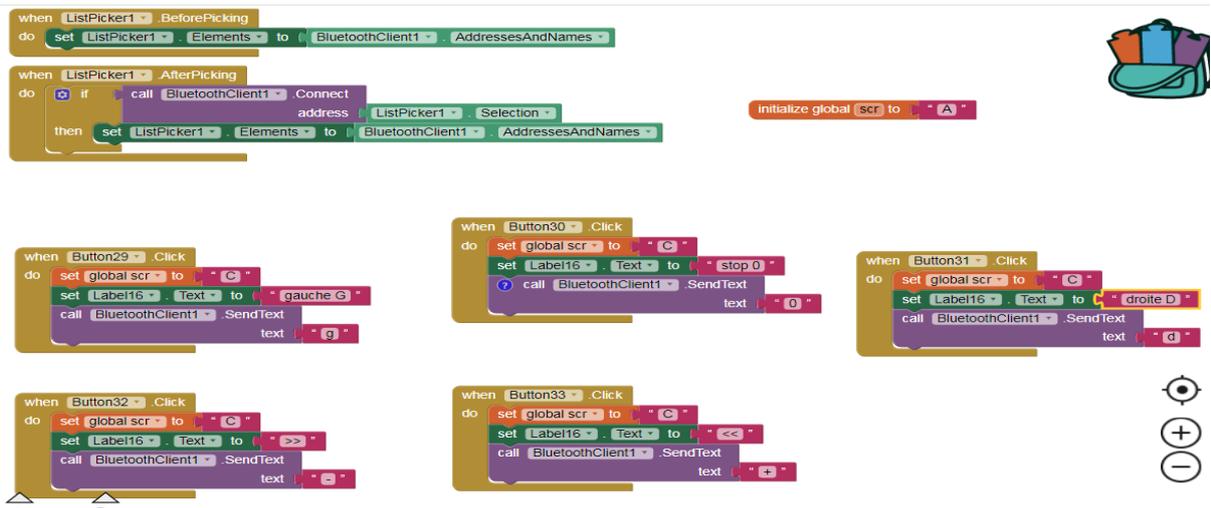


Figure II.12 : Programme de notre application sur Androide

Chapitre III: Etude du circuit et résultats de simulation

III.1 Circuits sur ISIS :

III .1.1 Définition :

PROTEUS permet de saisir les schémas électronique (ISIS), soit en page simple, soit en hiérarchique. L'environnement de saisie est entièrement paramétrable, si bien qu'il est possible de lui appliquer des skins (ZUKEN ou MENTOR par exemple). La saisie schématique est très simple et intuitive, PROTEUS est une des CAO les plus simples qui existe, tout en restant un outil très puissant pour réaliser des ensembles très complexes et atteindre un requis industriel ISO. La fonction « PAT » permet de gagner du temps dans la conception ou la vérification en faisant des recherches / modifications ou affectations conditionnelles dans le schéma.

III.2 principe de fonctionnement :

III.2.1. commande numérique:

III.2.1.1 Introduction:

Dans cette partie de notre travail nous allons réaliser le circuit qui va commander le moteur pas à pas.

Pour ce faire nous avons établi un schéma synoptique d'un montage à base d'un microcontrôleur pic 16F876A à qui nous avons ajouté une interface utilisateur et des circuits de puissance .

L'interface utilisateur est constituée de 4 boutons poussoirs pour l'introduction des commandes, et d'un afficheur LCD 2 lignes ,16 caractères qui permet de visualiser en instantanée les informations relatives à ces paramètres.

Le circuit de puissance permet l'amplification du courant de sortie du pic pour commander le moteur.

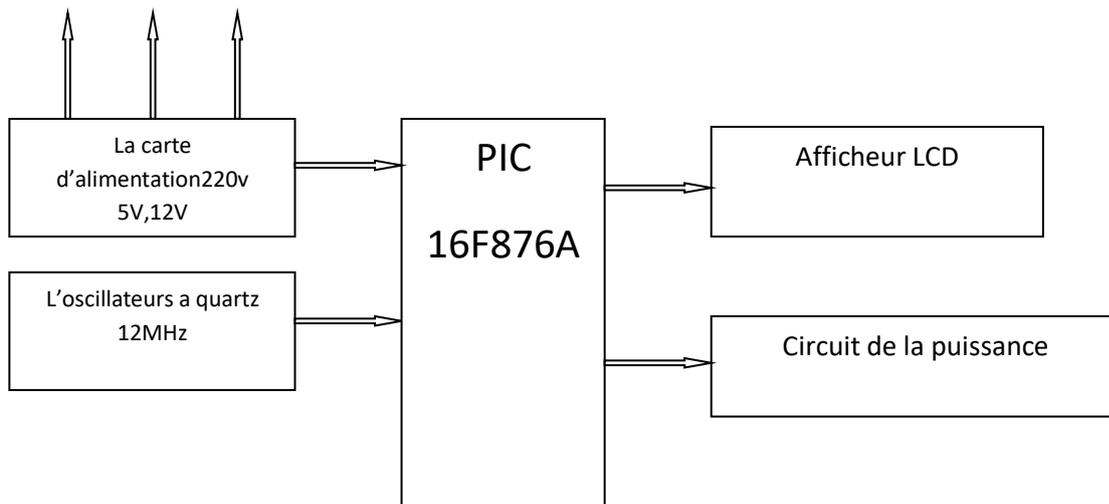


Figure III .1 : Schéma synoptique du montage.

III.2.1.2 configuration minimale du PIC :

Pour réaliser une carte a bas d'un pic 16F876A il faut au moins configurer (1) L'alimentation (2) circuit d'initialisation (3)l'oscillateur.

- (1) L'alimentation le pic fonctionne dans la plage de tension 4.5v a 6v
- (2) circuit d'initialisation étant utilisé pour effectuer un reset
- (3)l'oscillateur est le cœur du microcontrôleur c'est lui qui cadence le déroulement du programme .il doit fournir un signal carré périodique au microcontrôleur

III.2.1.3 Fonctionnement de l'afficheur LCD:

On envoie deux types d'information à l'afficheur :

- les **commandes** qui permettent de l'initialiser : positionnement du curseur, effacement écran, etc. ;
- les **données** à afficher.

L'entrée **RS** permet de spécifier si on envoie une commande ou une donnée :

- RS=0 : instruction (commande) ;
- RS=1 : caractère (donnée).

L'afficheur dispose d'une entrée **R/W** pour spécifier une lecture ou une écriture :

- R/W=0 : écriture vers l'afficheur ;
- R/W=1 : lecture de l'afficheur.

Pour valider tous les échanges sur le bus de données (D7-D0) on utilise l'entrée **E** de l'afficheur. Un **front descendant** sur cette entrée valide la donnée. En programmation, il faudra placer un court instant **E** à l'état haut puis à l'état bas.

Il est possible d'utiliser l'afficheur LCD en mode 8 bits normal ou en mode 4 bits pour économiser les broches de son µContrôleur par exemple, c'est assez pratique :

Mode 8 bits:

En mode 8 bits on place la donnée ou la commande sur le bus **D7** à **D0** et on valide avec **E**

Mode 4 bits:

En mode 4 bits on place déjà les poids forts de la donnée ou la commande sur les bits de **D7** à **D4** et on valide une première fois avec **E**. Puis on va mettre le poids faible sur les bits de **D7** à **D4** et on valide une seconde fois avec **E**.

L'envoi ou la lecture d'un octet s'effectue donc en 2 temps dans ce mode.

III.2.1.4 La commande des moteurs pas à pas :

Afin de simplifier la commande de moteur pas à pas et utiliser le minimum de circuits intégrés dans notre carte de commande, on a programmé dans le pic 16F876A les séquences nécessaires pour générer la rotation du moteur soit en mode pas entier ou bien en mode demi pas ensuite pour pouvoir faire tourner le moteur pas à pas qui nécessite une certaine puissance, on a utilisé un circuit de puissance.

III .2.2circuit de puissance :

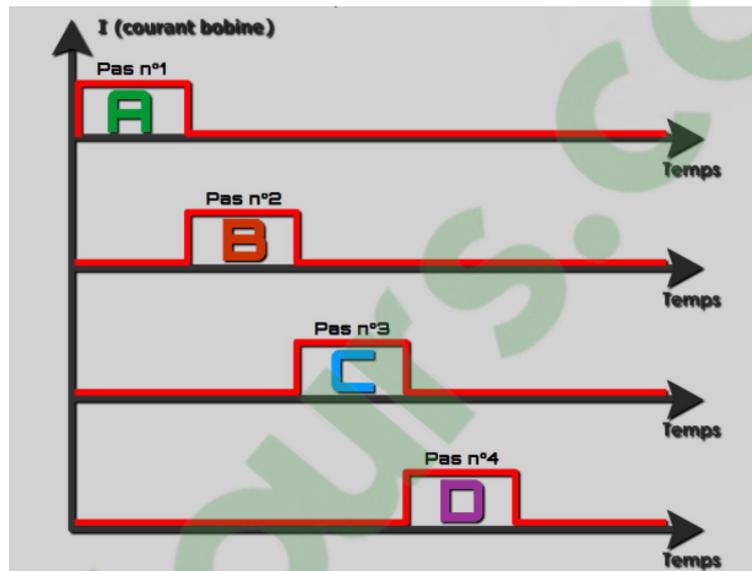
III.2.2.1 Introduction:

Le circuit intégré est un circuit de puissance qui sert à amplifier les signaux pour commander les moteurs pas à pas. Il est composé de 4 transistors MOSFET pour amplifier le courant de sortie nécessaire pour le bon fonctionnement du moteur.

III.2.2.2 Principe de fonctionner :

On utilise les sorties du microcontrôleur pour commander chaque transistor MOSFET .

Séquence alimentation des bobines pour un déplacement de 4 pas :



Il suffit de *poursuivre* en boucle cette séquence pour faire tourner le moteur.

Pour *changer* le sens de rotation du moteur, il suffit d'*inverser* la séquence précédente, on obtient donc **D - C - B - A**, afin d'*inverser* le champ magnétique générée dans les bobines, ainsi le rotor tournera en sens inverse ↔ .Notez qu'à mesure que le temps augmente, le régime moteur diminue .

Combinaisons d'états des commutateurs				Résultat à la charge
État des commutateurs				
S1	S2	S3	S4	
		X		Aucune tension aux bornes de la charge.
✓		X	✓	Courant positif à travers la charge.
X		✓	X	Courant négatif à travers la charge.
	✓		X	Charge court-circuitée.
	X		✓	

TableauIII. 1 : combinaisons d'états des commutateurs

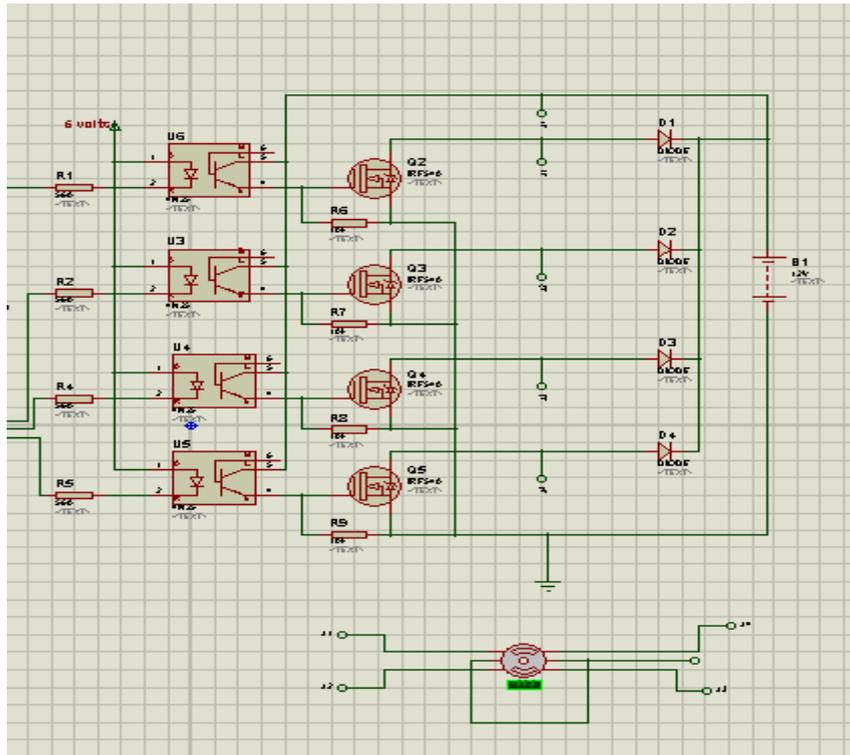


Figure III.2 : Circuit de puissance

III.2.2.3 Utilisation avec le moteur à courant continu :

Circuit de puissance permettant de réaliser 2 fonctions qui sont d'inverser le sens de rotation du moteur en inversant le courant aux bornes du moteur et la variation de la vitesse du moteur en modulant la tension aux bornes du moteur.

De plus, le circuit de puissance permet d'effectuer un freinage magnétique s'il est capable d'en dissiper la puissance générée. Cette opération s'effectue en actionnant soit les deux commutateurs supérieurs ou inférieurs en même temps, ce qui court-circuite les bornes du moteur, et le fait par conséquent freiner. Mieux encore, il est possible avec un peu d'électronique et un contrôleur perfectionné d'effectuer un freinage régénératif. Dans le cas d'une alimentation à batterie, l'énergie est renvoyée aux batteries plutôt que dissipée dans les commutateurs du pont.

III.2.3 contrôle du sens de rotation :

Notre montage comporte quatre boutons poussoirs qui ont des fonctions différentes,

- Le bouton poussoir BP1 permet de choisir le sens 1(à droite).
- Le bouton poussoir BP2 permet de choisir le sens 2(à gauche).
- Le bouton poussoir BP3 permet de faire un arrêt.
- Le bouton poussoir BP4 permet d'augmenter la vitesse.
- Le bouton poussoir BP5 permet de diminuer la vitesse.

Après avoir terminé l'étude théorique de ce projet, on est passé à la schématisation du montage du circuit au moyen d'ISIS. Le schéma obtenu est le suivant

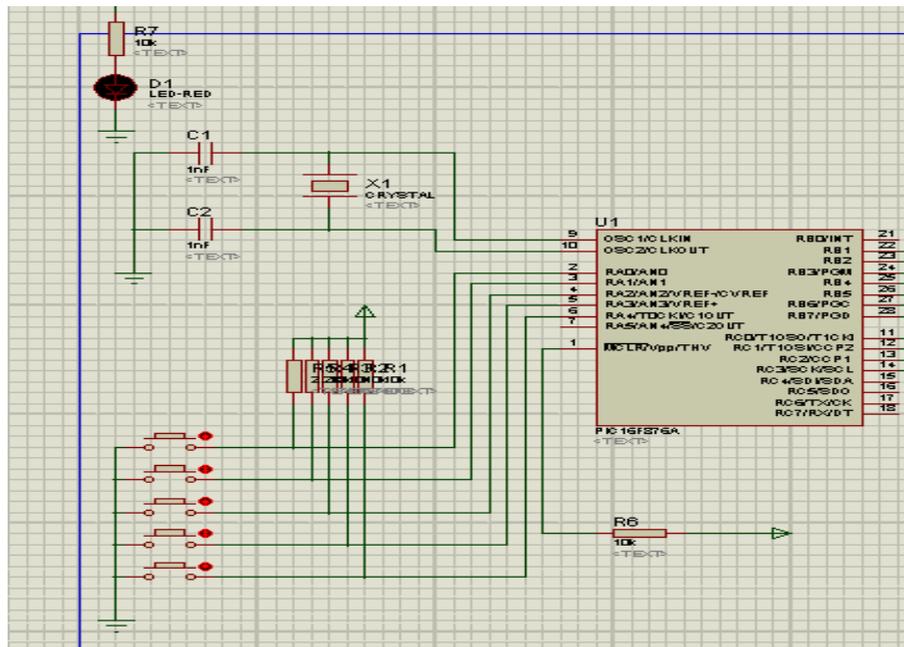


Figure III.3 : Brochage des boutons poussoirs avec le PIC 16F876a

```

1: unsigned tt1,ss1;
2: char x1;
3: // ***** teste bouton *****
4: void bouton()
5: {
6: if (porta.f0==0) // diminuer vitesse
7: {
8: tt1=tt1+1;
9: if (tt1>=10) {tt1=10;}
10: delay_ms(250);
11: }
12: else if (porta.f1==0) //augmenter vitesse
13: {
14: tt1=tt1-1;
15: if (tt1==255) {tt1=0;}
16: delay_ms(250);
17: }
18: else if (porta.f2==0) //sens1
19: {
20: ss1=1;
21: }
22: else if (porta.f3==0) //sens2
23: {
24: ss1=2;
25: }
26: else if (porta.f4==0) //stop
27: {
28:
29: ss1=0;
30: }
31: }
32: //*****bluetooth*****
33: void bluet()
34: {
35: if (UART1_Data_Ready())
36: {
37: x1= UART1_Read();
38: UART1_Write(x1);
39: if ((x1==68)|| (x1==100)) //droite D
40: {
41: ss1=1;
42: }
43: else if ((x1==71)|| (x1==103)) //gauche G
44: {
45: ss1=2;
46: }
47: else if (x1==45) //dminuter vitesse -
48: {
49: tt1=tt1+1;
50: if (tt1>=10) {tt1=10;}
51: }
52: else if (x1==43) //augmenter vitesse +
53: {
54: tt1=tt1-1;
55: if (tt1==255) {tt1=0;}
56:
57: } //arret 0
58: else if (x1==48)
59: {
60: ss1=0;
61: }
62: }

```

```

66: // ***** temporisation *****
67: void delai()
68: {
69: int i;
70: for (i=0;i<=tt1;i++)
71: {
72: delay ms(40);
73: bouton();
74: bluet();
75: }
76: }
77:

78: // ***** main *****
79: void main()
80: {
81: trisb=0b00000001;
82: trisa=0b00111111;
83: trisc=0xf0;
84:
85: UART1 Init(9600);
86: Delay_ms(100);
87:
88: ADCON1=0x06;
89: tt1=10;
90: ss1=0;
91: while(1)
92: {
93: if (ss1==0) // arret
94: {
95: portc=255;
96: bouton();
97: bluet();
98: }
99: if (ss1==1) //sens1
100: {
101: portc.b3=1;portc.b2=0;delai();
102: portc.b2=1;portc.b1=0;delai();
103: portc.b1=1;portc.b0=0;delai();
104: portc.b0=1;portc.b3=0;delai();
105: }
106: if (ss1==2) //sens2
107: {
108: portc.b3=1;portc.b0=0;delai();
109: portc.b0=1;portc.b1=0;delai();
110: portc.b1=1;portc.b2=0;delai();
111: portc.b2=1;portc.b3=0;delai();
112: }
113:
114: }
115: }

```

Figure III.4 :Programme de simulation du sens de rotation

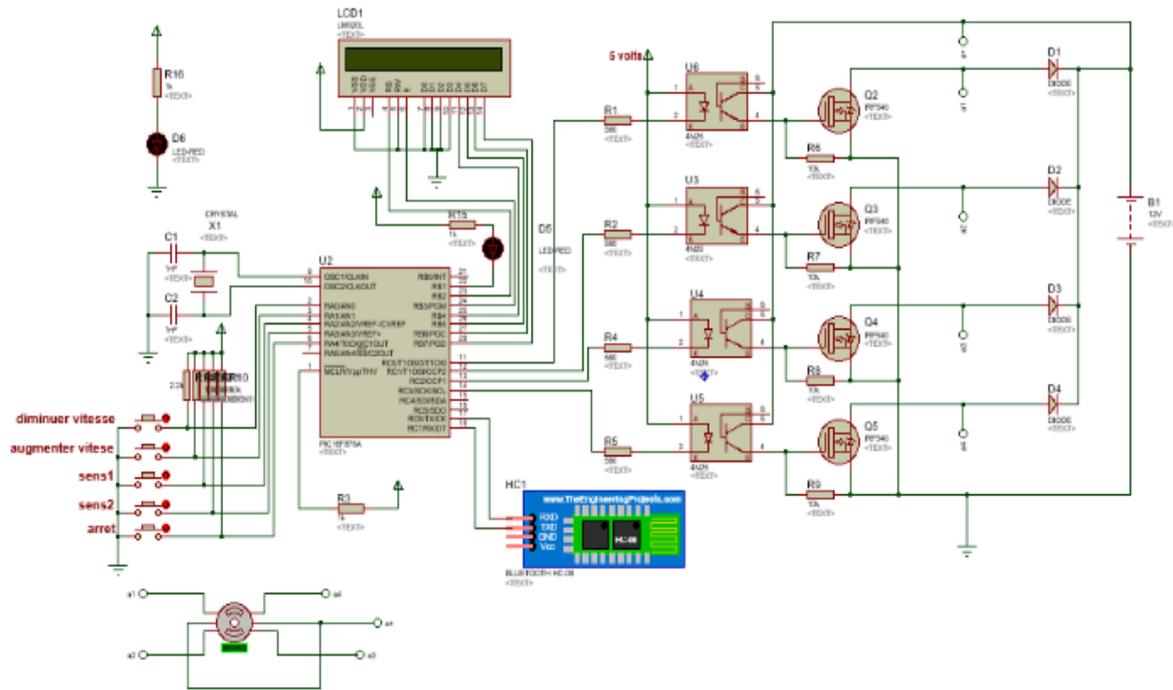


Figure III.5 : Circuit électrique sur ISIS

III.2.3.1 Explication :

Ce circuit est un montage d'un système de commande d'un moteur pas à pas composé d'un pic16F876 .

La technique de commande dans ce système est basée sur un microcontrôleur en configurant pour chaque moteur quatre broches en sortie afin d'envoyer les séquences des impulsions générées par le PIC sur les bornes du moteur pas à pas.

Chaque moteur reçoit une séquence indépendamment de l'autre à travers un amplificateur pouvant fournir un courant de sortie. Pour faciliter la communication entre l'opérateur et la machine nous avons choisi un moyen de dialogue qui est un afficheur LCD qui permet de guider l'opérateur à mettre en œuvre la machine[vitesse manuelle, origine pièce...].

III.3 Les Organigrammes :

III.3.1 Définition :

Les organigrammes ont pour but de faciliter la compréhension et le déroulement du programme, et ne représentent que les actions principales effectués par le microcontrôleur. Ils ne traduisent pas intégralement le code en C et ne reflètent pas les particularités du PIC.

Le programme peut se diviser en plusieurs fonctions :

- La fonction Main : Le Programme principal.

- Les fonctions de la gestion de l'afficheur LCD (initialisation, effacement de l'écran, commande...).
- Les fonctions du menu pour le pilotage du microcontrôleur via les 5 boutons poussoirs.

III.3.1.1 La Fonction Main (le programme principal) :

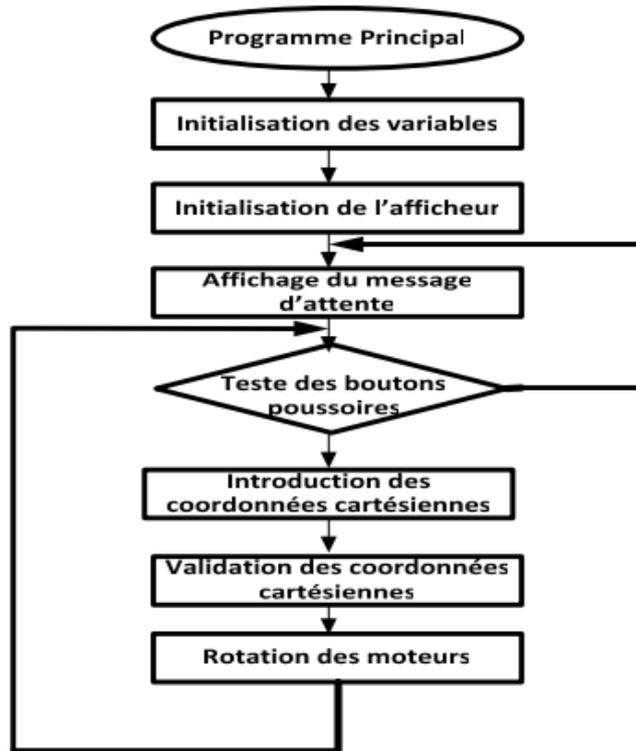


Figure III.6 : Organigramme Programme Principale

III .3.1.2 La Gestion de l’Afficheur LCD :

L’afficheur LCD nécessite une phase d’initialisation avant de pouvoir remplir sa fonction. L’afficheur est initialisé lors de l’appel de la fonction `init_lcd()` dans le programme principal. La procédure d’initialisation est la suivante :

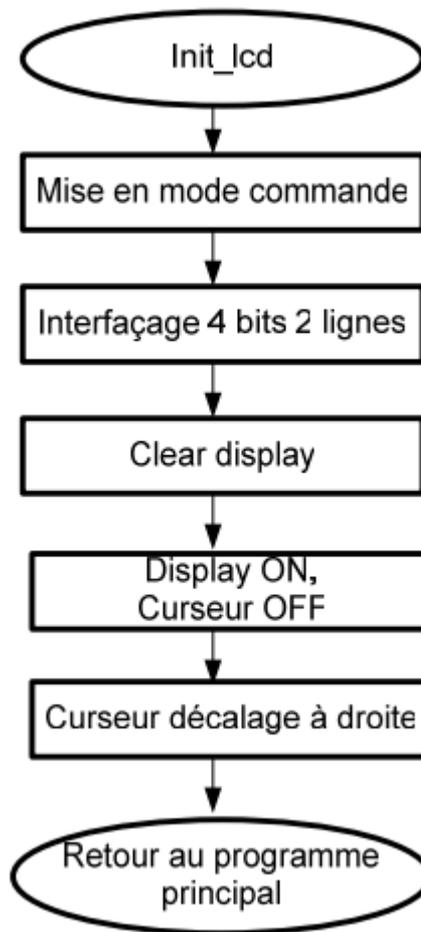


Figure III.7 : Organigramme de la fonction LCD

III.3.1.3 La Gestion des boutons poussoirs :

Ces cinq boutons poussoirs nous permettent de sélectionner et d'introduire les consignes pour faire tourner le moteur.

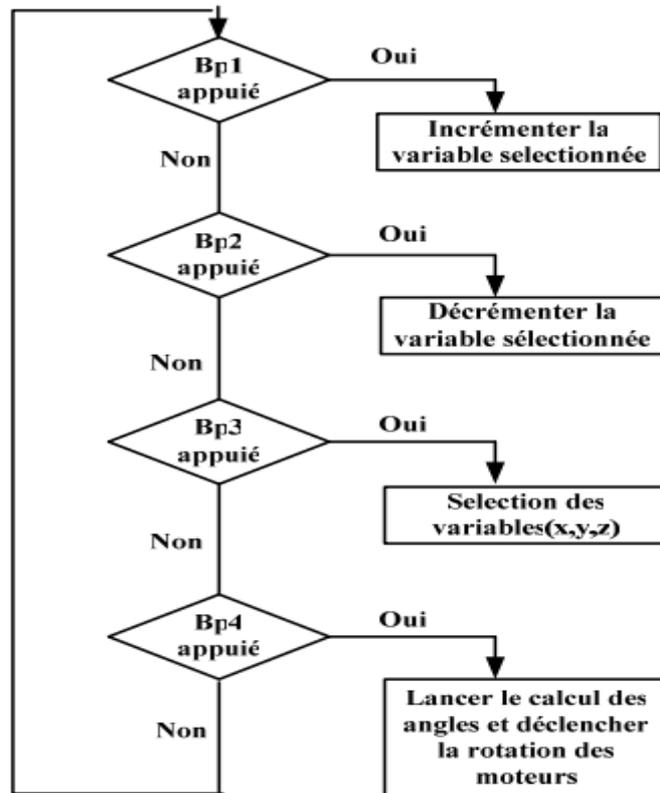


Figure III.8 : Organigramme de la gestion des boutons poussoir

III.4 Liste des composants :

Désignation	Référence	Valeur	Quantité
Résistance	R1 ,R2,R4,R5	560K Ω	4
	R6,R7,R8,R9,R10, R11,R12,R13	10 K Ω	8
	R14	2.2 K Ω	1
	R3,R16,R15	1 K Ω	3
Capacités	C1,C2	1nF	2
Crystal	X1	12MHz	1
Circuit intègre	U2	PIC 16F876A	1
	U3,U4,U5,U6	4N25	4
Diode	D1,D2,D3,D4	Diode	4
	D5,D6	LED-RED	2
Transistor	Q2 ,Q3,Q4,Q5	IRF540	4
Bluetooth	HC1	HC-06	1

Moteur pas à pas	/	/	1
Afficheur	LCD1	LM020L	1
Boutons poussoirs	/	/	5
Plaque d'essai	/	/	1
Des fils	/	/	/

Chapitre IV : réalisation pratique

IV.1 la réalisation du circuit imprimé

IV.1.1 le choix d'une technique pour fabriquer un circuit imprimé

A- La gravure chimique demande du temps et nécessite des mesures strictes de sécurité. Par ailleurs, il vous faut disposer de certains matériaux spécifiques tels que le réactif d'attaque. La gravure chimique est une technique qui convient bien à la conception de cartes électroniques simples ou pas trop complexes. Elle n'est par contre pas conseillée pour la conception de circuit imprimé avec connexions rapprochées et très fines.

B- La gravure par rayons ultraviolets demande des matériaux assez coûteux et qui en plus ne sont pas disponibles en tout lieu. En revanche, sa mise en œuvre est simple et elle garantit l'obtention de circuits à la fois fins et complexes.

C-La gravure mécanique est encore appelée «routage». Elle exige l'usage de machines spéciales pour retirer le surplus de cuivre de la carte ou créer entre les connexions des passages vides. La gravure mécanique vous reviendra chère à moins que vous n'avez pas à acheter une machine à graver. Le mieux serait définitivement d'en louer. Mis à part cet inconvénient, la gravure mécanique est l'idéal pour fabriquer un circuit imprimé en plusieurs exemplaires. De même, elle assure l'obtention de cartes électroniques de bonne qualité.

D-la gravure au laser est celle qui est le plus souvent utilisée par les grandes entreprises de production ainsi que les universités. Le principe de la gravure au laser ressemble beaucoup à celui de la gravure mécanique. La différence est que c'est un rayon laser qui fait ici le travail. De telles machines sont difficiles à trouver.

IV.1.2 La création du typon

La création du typon consiste à tracer les pistes du circuit. Dans le cas où la gravure devrait se faire au moyen d'un acide, servez-vous d'un matériau qui y soit résistant.

- Vous pouvez procéder au traçage manuel du typon en vous servant de marqueurs spéciaux que vous pourrez acquérir sans souci dans le commerce.
- Vous pouvez aussi opter pour son impression au moyen d'une imprimante laser. Et dans ce cas, c'est l'encre de l'imprimante qui sert à tracer le circuit.

La seconde méthode, bien que très attrayante, ne convient pas pour les circuits moyens ou complexes. Et il vous faut en général un logiciel spécial pour faire la conversion du schéma de principe du circuit en un schéma imprimable. Vous avez le choix entre plusieurs logiciels libres. Liquid PCB, PCB ou encore Shortcut sont quelques-uns de ces logiciels.

IV.1.3 Le réglage des dimensions du PCP

La troisième étape de la conception de circuit imprimé consiste à ajuster ses dimensions. Il vous faut d'abord l'achever sur votre ordinateur avant de régler ses dimensions grâce au logiciel. Veillez dans l'ajustement à ce que les dimensions correspondent à celles du papier.

IV.1.4 L'impression de la carte électronique

Vous passez ensuite à l'impression du PCP toujours grâce au logiciel. Servez-vous de papier glacé pour l'impression et assurez-vous que celle-ci soit réalisée en miroir. Surtout, ne touchez pas à l'encre du typon après l'impression. Vous risqueriez sinon de l'effacer.

IV.1.5 La pose du typon

À présent, vous devez mettre le recto du typon sur la face cuivrée de la carte. Veillez à ce qu'il y ait un alignement entre les deux pièces. Puis, allumez votre fer électrique et positionnez-le sur son support en attendant qu'il chauffe suffisamment.

IV.1.6 Le refroidissement de la carte électronique

Une fois que vous atteignez et ouvrez le robinet, positionnez la carte sous le filet d'eau. Vous pouvez encore, si vous préférez, plonger la plaque ainsi que le papier dans un récipient contenant de l'eau chaude. Mais, n'allez pas au-delà de 10 minutes.

IV.1.7 Le retrait du typon

Pour **retirer le typon**, vous n'aurez qu'à le décoller entièrement. Puisque l'eau aura largement agi en amont, cette opération vous sera très facile. N'hésitez pas à tremper un peu plus longtemps les endroits qui résistent. À l'issue de l'enlèvement du typon, vous disposerez d'une plaque en cuivre sur laquelle se trouve imprimé en noir le circuit !

IV.1.8 Le séchage de la carte électronique

À cette étape de conception de circuit imprimé, il s'agira de **sécher la carte**. À l'aide d'une serviette, essuyez-la soigneusement. Vous pouvez aussi opter pour une autre technique, celle consistant à l'incliner afin d'enlever les gouttes d'eau. Il ne vous faut pas plus de trente secondes pour terminer le séchage de la carte. Soyez méticuleux afin que le circuit ne s'efface pas.

IV.1.9 La gravure

L'avant-dernière étape pour fabriquer un circuit imprimé est celle de la gravure. Elle consistera en l'élimination du cuivre inutile afin de n'avoir que les pistes de connexion qui vous seront utiles. La gravure pourra se faire avec de l'acide ou au moyen d'une insoleuse UV. L'acide le plus souvent utilisé est le chlorure ferrique. Mais vous pouvez également porter votre choix sur le persulfate d'ammonium tout comme d'autres substances chimiques.

IV.1.10 La finition de la carte

Après la gravure, il ne vous reste plus qu'à procéder à la finition de votre carte :

- dans un premier temps, vous devrez percer les trous de montage. Il existe des machines spécifiquement conçues à cet effet. Mais, vous pouvez toujours vous rabattre sur une perceuse classique, si vous voulez ;
- dans un second temps, vous pourrez alors monter sur la carte les composants électroniques, puis les souder.

IV.2 Le circuit imprimé de la carte principale

Les deux faces du circuit imprimé de la carte principale, coté cuivre et coté composants sont illustrés respectivement par la figure 1.1, la figure 1.2 et la figure 2.

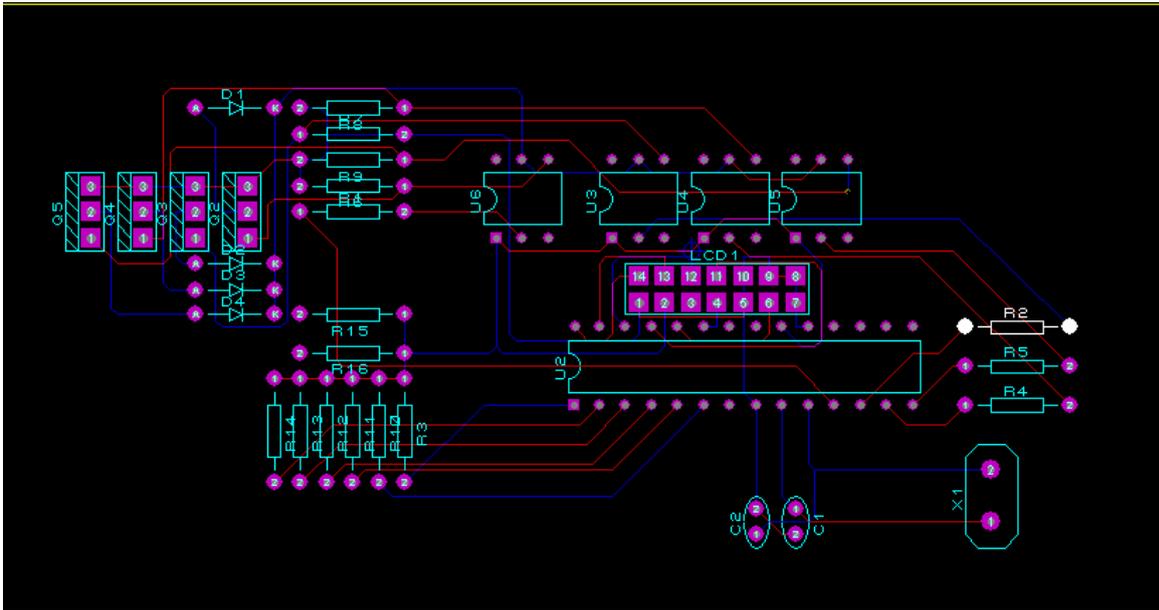


Figure IV.1: Coté cuivre du circuit imprimé (carte principale)

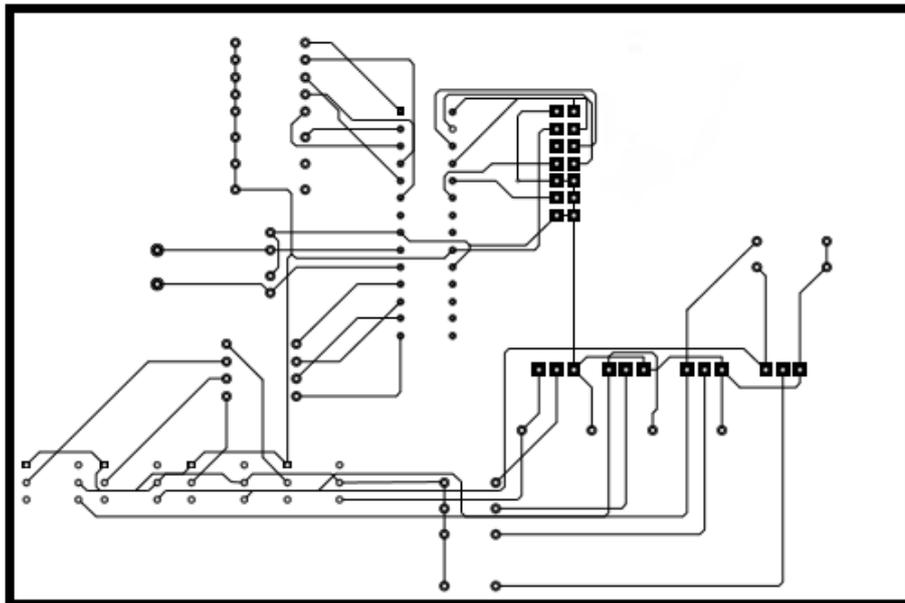


Figure IV.2: Coté cuivre du circuit imprimé (carte principale)

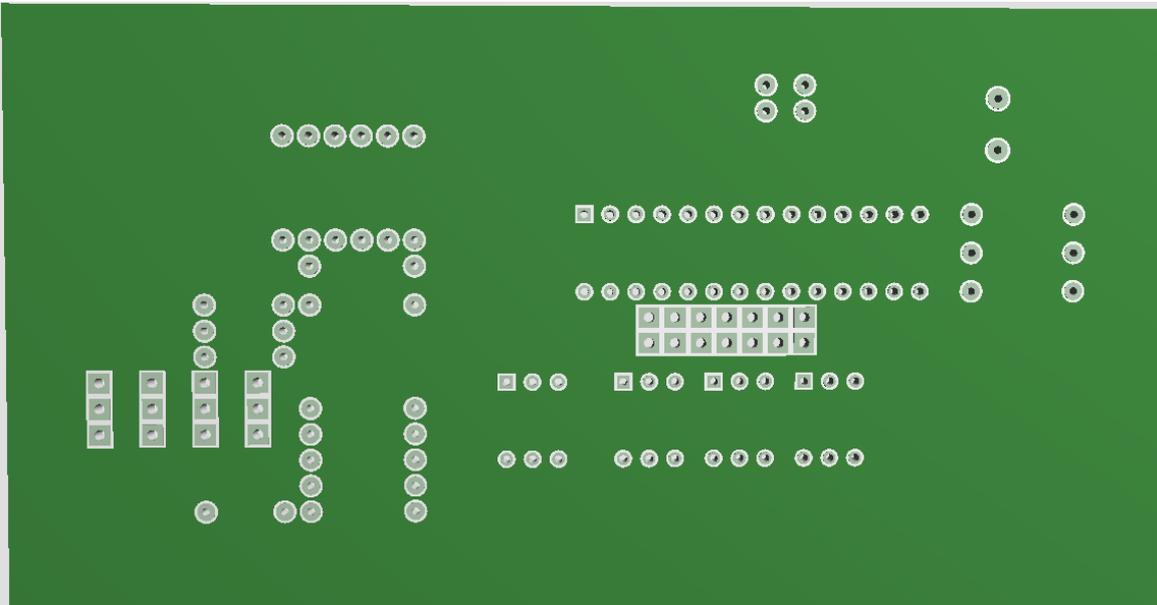


Figure IV.3 :Coté composants du circuit imprimé (carte principale)

IV.3 Implantation des composants

La figure 3 indique l'implantation de chaque composant sur la surface du circuit imprimé.

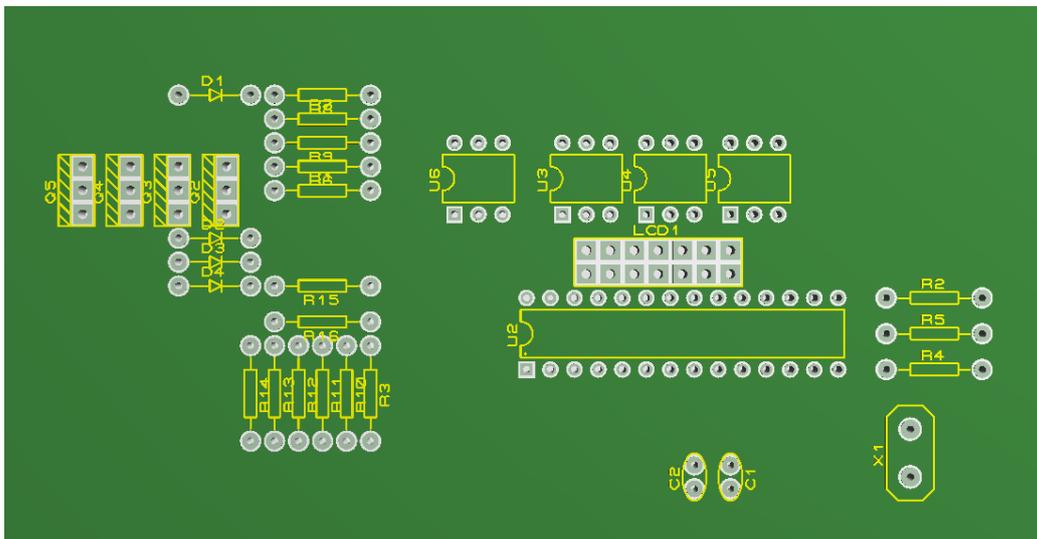


Figure IV.4:Implantation des composants (carte principale)

IV.4 Vue en 3 dimensions du circuit imprimé

IV.4.1 L'outil de visualisation en 3D

Cette outil (visionneur 3D) d'ARES permet de voir la carte telle qu'elle sera en

Réalité . Parmi les fonctionnalités principales du visionneur 3D comprennent :

- Représentation réaliste du circuit imprimé.
- Pris en compte des empreintes 3D dans les bibliothèques ARES.
- Bouton de visualisation directe des vues haut /face /bas /arrière du circuit.
- Description sous la forme de script simple pour une prévisualisation en temps réel lors de la création des empreintes 3D.

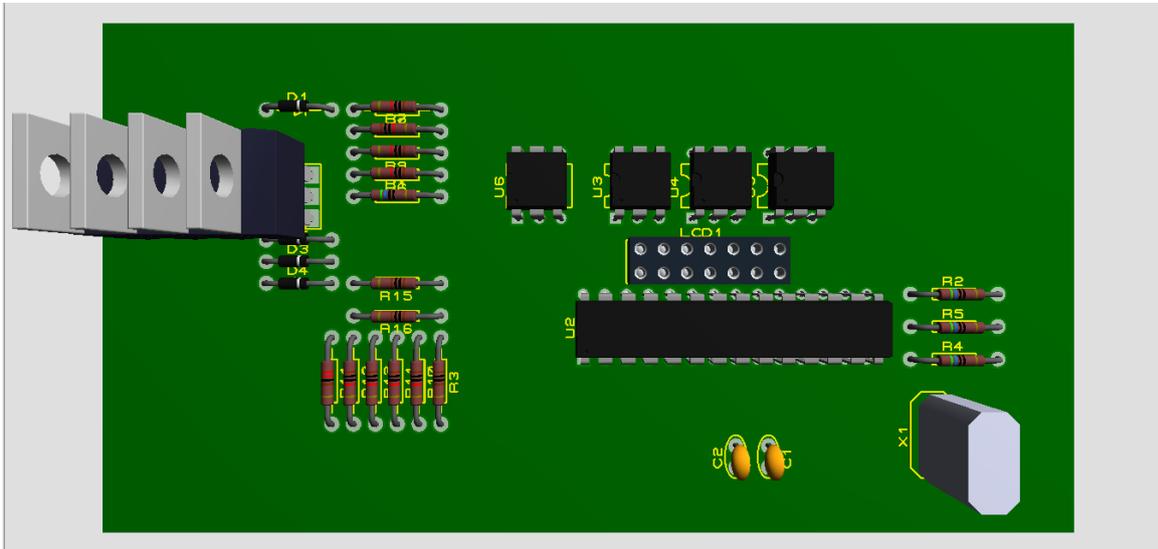


Figure IV.5 :Vue du circuit imprimé en 3Dimension.

IV.5 Les photos du projet réalisé

IV.5.1 partie puissance

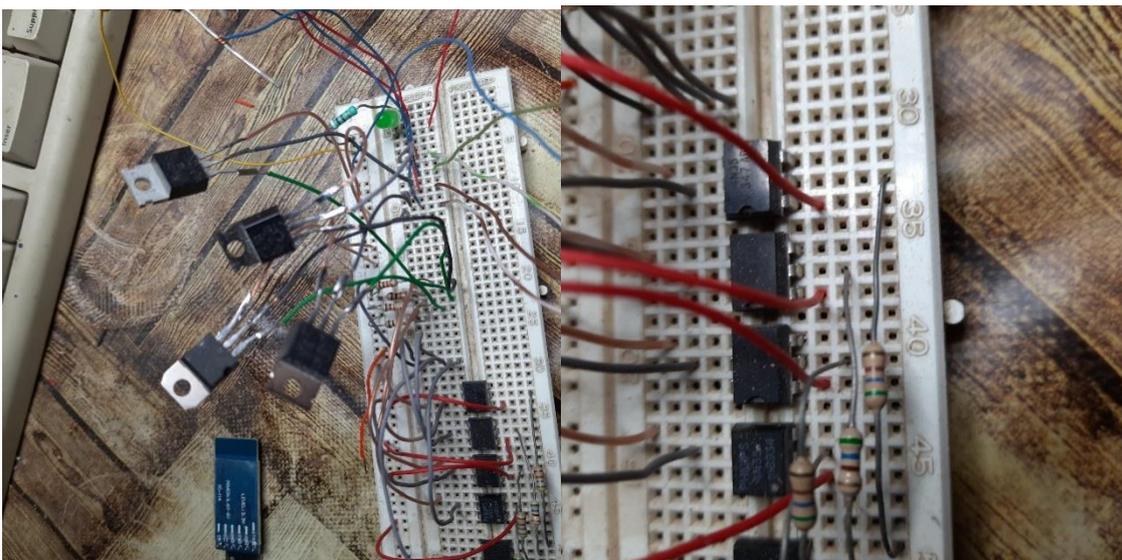


Image IV.6: Partie puissance du circuit

IV.5 .2partie commande

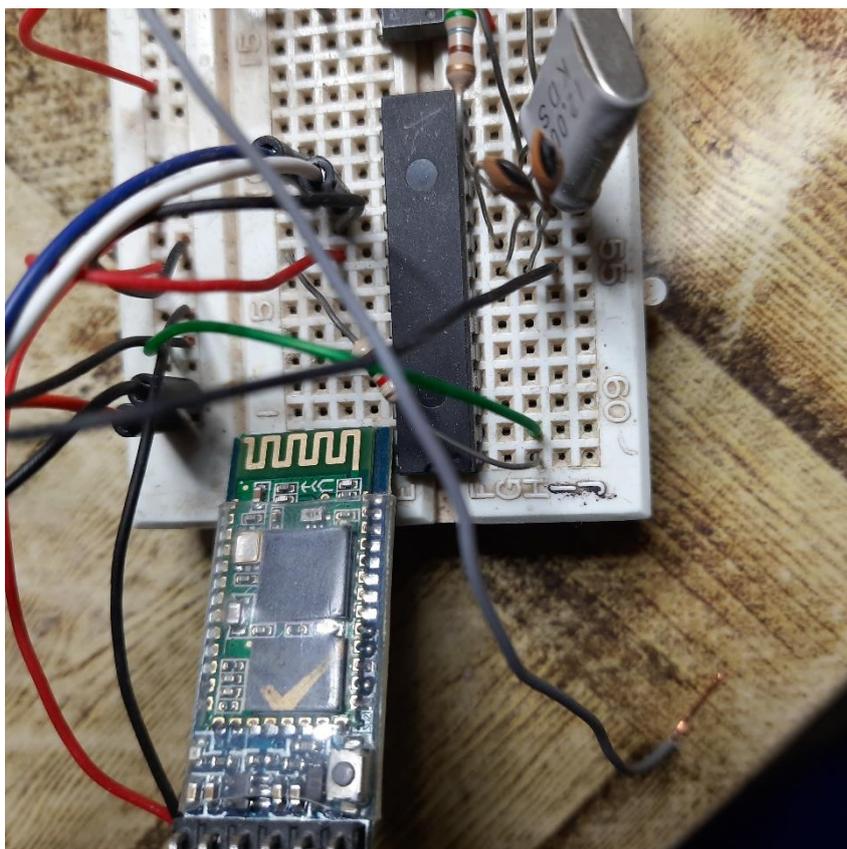


Image IV. 7 : Partie commande du circuit

IV.5 .3 la carte complète (partie commande + partie puissance)

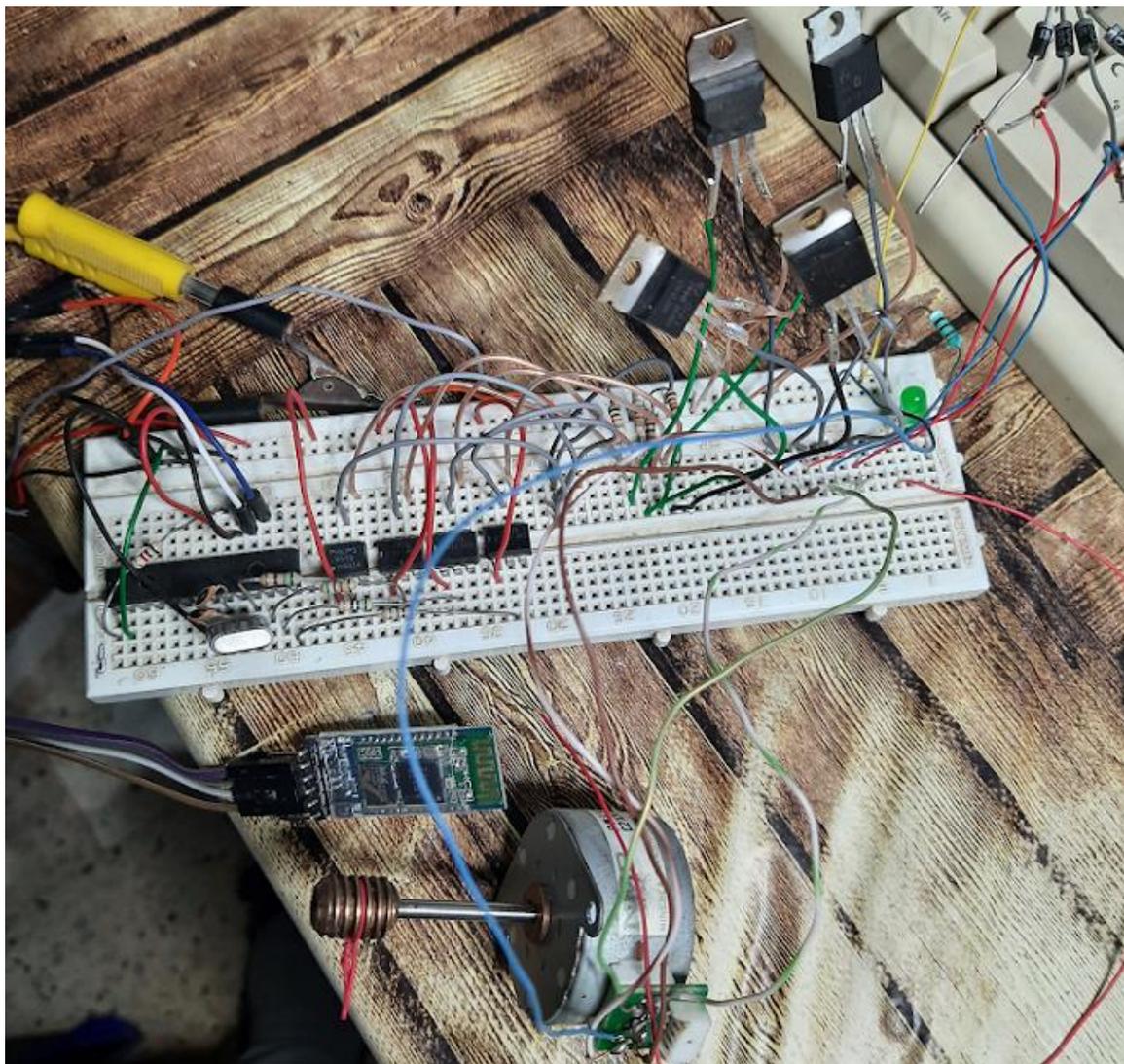


Image IV.8 : La carte complète du projet réalisé

CONCLUSION GÉNÉRALE

Au terme de ce travail, nous avons pu réaliser une carte de commande d'un moteur pas à pas architecturé autour du microcontrôleur PIC16F876A.

Ce travail nous a permis de mettre en valeur une partie importante des connaissances théoriques que nous avons accumulés pendant toute la durée de nos études. Il nous a permis également d'acquérir une expérience nouvelle dans l'électronique pratique et la programmation des PIC. Au cours de la réalisation de notre projet, nous avons utilisé des logiciels de conception, simulation et programmation des circuits électroniques. En effet, nous avons utilisé mikroC pour la programmation du microcontrôleur, PROTEUS pour le dessin du circuit électrique, le circuit imprimé et l'implantation des composants et pour la simulation de manière interactive notre projet.

BIOGRAPHIQUES

[1]Chigango ,Rouald ,’’ Etude et principe de fonctionnement des moteurs pas à pas Alimentation des moteurs pas à pas-Le rayon infrarouge’’ , Mémoire demaster ,Université Abou BekrBelkaid Tlemcen, 11-jui-2015.

[2] B.Djedouani, M.Ouldamedhaiballa ,’’ Développement d’un Banc d’essai por l’étude est le diagnostic des moteurs à pas Hybrides’’ , Mémoire de master, Université de Tébessa,2015.

[3] M .BADRI, ’’Etude et réalisation d’un variateur de vitesse d’un moteur pas a pas base de microcontrôleur pic 16f876’’ , Mémoire de master, Université Larbi Ben M’hidi de Oum El Bouaghi, juin 2016.

[4] Bentalebmokhtar, gasmielhadi,’’ Réalisation et commande d’une machine cnc à base des moteurs pas à pas, Mémoire de master, Université KasdiMerbah Ouargla,2016.

[5]Gerardyrant,’’ Les moteur pas à pas ’’ , Séminaire BellgardeNov 99

[6]]M. Correvon, Chapitre 7 Les Semiconducteurs De Puissance Deuxième Partie : LE MOSFET , Haute Ecole Spécialisée Du Suisse Occidentale.

[7] Aide-mémoire Composants Electroniques ,Dunod, Paris, 2005 Dunod, Paris, 2000 Fréquences, Paris, 1989 pour la première édition ISBN 2 10 048885 6,Page104.

[8] Guy Chateigner,DanielBouix, Michel Boës,JacquesVaillan ,Daniel Verkindère ,’’Manuel de Génie Electrique, Dunod, Paris, 2006 ,ISBN 978-2-10-048499-7, page 245.

[9]M.Tamoum,’’ Caractérisation Fine et Modélisation Non-Linéaire des Transistors MOSFET’’ , thèseDoctorat, Université Ferhat Abbas, Sétif , 24 / 02 / 2013.

[10] Les microcontrôleurs PIC 16F87X , Philippe LETENNEUR -GRANVILLE – 2003.

[11]D.Ahmid,B. Hammadi, ’’Carte d’acquisition de données en utilisant le microcontrôleur PIC16F876A ’’ , Mémoire de master, L’Université EchahidHamma Lakhdar d'El Oued , 26/05/2016.

[12] [Joscelin OUDRY](#), Professeur notamment de système d'exploitation pour l'école [Ingénieurs 2000](#) filière Informatique et réseaux 3eme année ([université de marne-la- vallée](#)).

[13] A. Mebareka, L. Kharboubi, "Etude Et Réalisation D'un Cardi-fréquence-mètre Portable A L'aide Du PIC16F84A", Mémoire de master, Université Abou Blekaide de Tlemcen, 25 Mai 2015.

[14] Bernard Lewandowicz 16/05/14 C_BL_Le_microcontrôleur_PIC16F876.odt

[15] Le microcontrôleur PIC 16F876A - page 9 Lycée LACHENAL - M. BERNARD - édité le 02/11/2008.

Le protocole **OBEX** (**OB**ject**EX**change) permet le transfert des données grâce à grâce au protocole d'échange de fichiers IrDA.

[16] M. Peter Daehne, Professeur HES, "Application Java pour terminal mobile utilisant le protocole Bluetooth", Carouge, 22 août 2008 Haute École de Gestion de Genève (HEG-GE) Filière Informatique de Gestion.

[17] Lycée Gustave Eiffel, STI2D – Enseignement de spécialité SIN, Bluetooth pdf.

[18] Mohamed Anouar Dahdeh, "Conception, Développement Et Intégration D'une Application Embarquée De Téléchargement Des Applications Androïde FTAB STORE", Mémoire de master, Université Virtuel de Tunis, Année Universitaire : 2010 – 2011

[19] Pooja Singh, Pankaj Tiwari, Dr. Santosh Singh, 7th International Conference on Communication, Computing and Virtualization 2016, Analysis of Malicious Behavior of Android Apps, Available online at www.sciencedirect.com

[20] ZOURANE .Akli, MAMMERI . ydia, "Conception et Réalisation d'une Application Réseau pour le Système ANDROID Cas : Gestion du stock de Cevital", Mémoire de master, Université A/Mira de Béjaïa, 2013.

[21] Androïde Introduction générale, James D o uglass Lefruit Ingénieur de Recherche, INRIA Grenoble

[Wikipédia Circuit sur ISIS](#)

[https://fr.wikipedia.org/wiki/Proteus_\(électronique\)](https://fr.wikipedia.org/wiki/Proteus_(électronique))

[Afficheurs LCD](#)

<https://sciencesappliquees.com>

-SHANE TOLMIE: Il présente dans son site beaucoup d'outils idéals pour la programmation des PIC en C :

www.microchipc.com

-BIGONOFF présente dans son site l'intégralité de ses cours pour les PIC (part1, part2, part3, part4 & part5) ainsi que beaucoup d'autres outils tous très intéressants :

<http://abcelectronique.com/bigonoff>

- PATRICE OGUIC , Moteurs pas à pas. L'USINE NOUVELLE, DUNOD, 2004.

www.lelogiciellibre.net

Circuit imprimé : Présentation et différentes étapes de conception !