



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abou bekr Belkaid – Tlemcen
Faculté de Technologie
Département de Génie Electrique et Electronique

Filière : ELECTRONIQUE

Projet de Fin d'Etudes

Master : Electronique

Option : Instrumentation Electronique

Intitulé : Commande de deux moteurs pas à pas et moteur à courant continu via le port parallèle avec Delphi

Présenté par : Cheddad Mohammed Amine

Mazouzi Omar

Jury :

Encadreur : Nemmich Ahmed

Examineur : Bechar Hassan

Examineur : Kerrous Abdelghani

Année Universitaire : 2012/2013

Table des matières

| | |
|--|----|
| Introduction général | 6 |
| Chapitre I | 8 |
| 1. Introduction : | 8 |
| 2. Description des éléments de la carte : | 8 |
| 2.1. Le microcontrôleur pic 18f2550 : | 8 |
| 2.1. 1. Identification d'un pic : | 8 |
| 2.1.2. Le Choix du microcontrôleur : | 9 |
| 2.1.3. La structure interne du microcontrôleur 18F2550 : | 9 |
| 2.1.4. Organisation du 18F2550 : | 10 |
| 2.1. 5. Les registres : | 10 |
| 2.1. 6. Étude du boot loader de micro-chip : | 10 |
| 2.2. Module USB du PIC 18F2550 : | 11 |
| 2.3. Programme du pic 18F2550 : | 11 |
| 2.3. 1. Présentation de l'environnement de programmation : | 11 |
| 3. Partie puissance : | 12 |
| 3.1. Introduction : | 12 |
| 3.2. Moteur pas à pas : | 13 |
| 3.2. 1 Définition : | 13 |
| 3.2. 2. Différent types de moteur pas à pas : | 13 |
| 3.2.3. Le mode de contrôle : | 15 |
| 3.2. 4. Caractéristiques couple et vitesse : | 16 |
| 3.2. 5. Configurations interne des bobines du moteur : | 17 |
| 3.3 Commande des moteurs pas à pas: | 17 |
| 3.3. 1. Unités de pilotage : | 17 |
| 3.3. 2. Le séquenceur : | 17 |
| 3.3.3. Les commutateurs de puissance ou l'interface de puissance : | 18 |
| 3.3. 4. Le mode de séquence : | 18 |
| 3.4 Alimentation des moteurs pas à pas: | 19 |
| 3.4. 1. Alimentation unidirectionnelle et bidirectionnelle : | 19 |
| 3.4.2. Alimentation en tension : | 20 |
| 3.4. 3. Le critère de choix d'un moteur pas à pas : | 20 |
| 3.5 Les moteurs à courant continu : | 20 |

| | |
|---|----|
| 3.5. 1. Le principe de fonctionnement des moteurs à courant continu : | 21 |
| 3.5. 2. La commande des moteurs à courant continu : | 22 |
| 3.5. 3. Interface de puissance d'un moteur CC : | 22 |
| 3.5. 4. Variation de vitesse du moteur : | 22 |
| 3.5. 5. Variation de sens du moteur : | 23 |
| 3.5. 6. Partie isolation galvanique : | 24 |
| 3. CONCLUSION : | 24 |
| Chapitre II | 25 |
| 1. Introduction : | 25 |
| 2. Généralité sur L'USB : | 25 |
| 2.1 Définition : | 25 |
| 2.2 Origine de L'USB : | 25 |
| 2.3 Avantages de L'USB : | 25 |
| 2.4 Facilité d'utilisation : | 26 |
| 3. vitesse de transfert de L'USB : | 27 |
| 3.1 Définitions des différentes vitesses : | 27 |
| 3.2 Raison des trois vitesses de L'USB : | 27 |
| 3.3 Domaine d'utilisation des différentes vitesses : | 27 |
| 3.4 Les débits de L'USB : | 28 |
| 4 Le câble USB : | 28 |
| 4.1 Définition du câble USB : | 28 |
| 4.2 Composition du câble USB | 28 |
| 5 Le Bus USB : | 30 |
| 5.1 Principe du bus USB : | 30 |
| 5.2 Topologie du Bus USB : | 31 |
| 5.3 Protocole USB : | 32 |
| 5.4 Type de paquet USB : | 32 |
| 5.5 Codage NRZI : | 33 |
| 5.6 Alimentation USB : | 33 |
| 5.6.1. Alimentation de périphérique USB : | 33 |
| 5.6.2. Avantage de l'alimentation USB : | 34 |
| 5.6.3. Différents types d'alimentation du BUS USB : | 34 |
| 5.7 Courant de veille : | 34 |
| 5.8. Différents types de transfert : | 34 |
| 5.8.1. Transfert en mode control : | 34 |

| | |
|---|-----------|
| 5.8.2. Transfert en mode interruptif : | 35 |
| 5.8.3. Transfert en mode Isochrone : | 35 |
| 5.8.4 Transfert en mode Bulk : | 35 |
| 5.9. L'énumération : | 35 |
| 5.9.1. Définition de l'énumération : | 35 |
| 5.9.2. Principe de fonctionnement de l'énumération : | 35 |
| 5.10. Les descripteurs : | 36 |
| 5.10.1. Définition d'un descripteur : | 36 |
| 5.10.2. Rôle des descripteurs : | 36 |
| 6. PID/VID : | 37 |
| 6.1 Introduction au Product ID et au Vendor ID : | 37 |
| 6.2 Le descripteur HID : | 37 |
| 6.3 Product ID et au Vendor ID (PIO / VIn) : | 38 |
| 7. Conclusion : | 38 |
| Chapitre III : | 39 |
| 1. Introduction : | 39 |
| 2.1. Partie matérielle : | 40 |
| 2.1.1 Description du circuit réalisé : | 40 |
| 2.1.2. Remarque : | 41 |
| 2.2. Partie logicielle : | 41 |
| 2.2.1. Programmation : | 41 |
| 2.2.2. L'organigramme de programme : | 43 |
| 2.2.3. L'interface Delphi | 47 |
| 2.2.4. Simulation du circuit : | 47 |
| 2.2.5. Quelque historique de mesure: | 49 |
| 2.3. Essais sur la carte : | 49 |
| 2.3.1. Fonctionnalité de l'application : | 49 |
| 2.3.2. Schéma réelle de la plaque | 50 |
| 3. Conclusion : | 50 |
| Conclusion générale : | 51 |
| ANNEXE | 53 |
| BIBLIOGRAPHIE | 60 |
| WEBOGRAPHIE | 61 |

LISTE DES FIGURES

| | |
|---|----|
| Figure 1 : brochage de PIC 18F2550..... | 9 |
| Figure 2: schéma minimaux pour USB-BOOTLOADING..... | 11 |
| Figure3: circulation du courant dans une bobine..... | 12 |
| Figure4: Moteur à aimant permanent..... | 14 |
| Figure5: Moteur à reluctance variable..... | 14 |
| Figure6: Exemple de rotor de Moteur hybride..... | 15 |
| Figure7: Moteurs hybride..... | 15 |
| Figure8: Mode unipolaire..... | 15 |
| Figure9: Mode bipolaire..... | 15 |
| Figure10: Domaines de fonctionnement du moteur pas à pas. | 16 |
| Figure11: Synoptique de la commande d'un moteur pas à pas (Boucle ouverte)..... | 18 |
| Figure12: Alimentation en tension unidirectionnelle..... | 20 |
| Figure 13 : Moteur à courant continu..... | 21 |
| Figure14 : exemple de commande par microcontrôleur..... | 23 |
| Figure15: Composition d'un câble USB..... | 28 |
| Figure 16 .Composition du couleur du câble USB..... | 29 |
| Figure17: Brochage des connecteurs USB de type A et B..... | 29 |
| Figure 18: Aspect des connecteurs USB de type 13(2.0 & 3.0)..... | 30 |
| Figure 19: Exemple de branchement respectant la topologie du Bus USB . | 31 |
| Figure 20: Principe du codage NRZI..... | 33 |
| Figure 21: communication entre le PC et le périphérique..... | 39 |
| Figure 22: interface de deux moteurs..... | 47 |
| Figure 23: circuit de commande | 48 |

| | |
|---|----|
| Figure 24: connexion du périphérique USB..... | 48 |
| Figure25 : résultat de la rotation..... | 49 |

LISTE DES ABREVIATIONS

| Abréviation | Désignation |
|--------------------|--|
| ADDR | Address |
| CPU | Central Processing Unit ou Unité Centrale de Traitement |
| EEPROM | Electrically EPROM |
| EPROM | Erasable PROM |
| HID | Human Interface Driver |
| PC | Personal Computer |
| PID | Paquet ID |
| PROM | Programmable ROM |
| RAM | Random Access Memory |
| ROM | Read Only Memory |
| USB | Universal Serial Bus |
| VID | Vendor ID |

Introduction général

La microinformatique est née avec l'apparition du premier microprocesseur sur le marché en 1974. En micro-informatique, nous vivons une grande révolution dans ce domaine, au niveau technologique et économique, car nous sommes passés du combinatoire vers le séquentiel et cela tient aux progrès des circuits intégrés, et des périphériques. Le chiffre d'affaires de la micro-informatique a dépassé celui de tous les autres secteurs de l'informatique réunis. La croissance de la micro-informatique est liée au développement rapide de l'électronique et à la numérisation des systèmes. Par ailleurs, la micro-informatique est devenue le lien principal entre l'électronique et l'informatique.

L'avancée technologique dans le monde, nous conduit vers une ère où tous les composants de l'ordinateur sont en perpétuelles révisions pour atteindre de meilleures performances d'applications. Avec les nouvelles technologies de l'information et de la communication nous assistons à un développement scientifique et technique de plus en plus modernisé où la commande est réellement une nécessité. Elle nous permet une liberté de mouvement tout en gardant la connectivité. C'est un outil pratique et performant, pouvant relayer le travail manuel en second plan et ce, en cas de danger ou une rapidité de fiabilité extrême est recherché.

Le travail entamé dans ce projet fait partie du domaine de la micro-informatique. Il consiste à réaliser un circuit à base d'un microcontrôleur PIC, il s'agit d'un périphérique à plusieurs sorties qui communique avec le PC au moyen du protocole USBHID. Pour donner à ce projet un aspect pratique on s'est proposé d'utiliser le circuit électronique à réaliser dans des applications utiles (commande d'un moteur pas à pas et moteur à courant continu)

C'est dans cet objectif que notre travail de nature technologique a été entamé. Pour achever le projet du point de vue réalisation matérielle on présente les différentes tâches qu'il faut accomplir et qui sont énumérées ci-dessous :

- ✓ Faire la liaison entre le PIC 18F2550 et le PC en utilisant l'USB.
- ✓ Faire des applications pratiques.

Pour les fonctions logicielles :

- ✓ L'apprentissage de la programmation des PIC18F2550
- ✓ Se familiariser avec le logiciel de programmation orienté objets DELPHI7 pour la réalisation des interfaces graphiques du PC.

Nous avons organisé ce mémoire de la façon suivante :

Le premier chapitre présente une étude détaillée sur tous les composants et les circuits de notre carte (PIC18F2550 moteur pas à pas moteur à courant contenu...).

Le deuxième chapitre présente une vue d'ensemble sur les bases de l'USB y compris sa topologie, le débit de données, les types de transferts des données et toutes ses caractéristiques.

La réalisation et l'implémentation de toute cette étude fera l'objet du troisième chapitre dans lequel nous illustrerons les différentes parties de chaque application réalisée.

En dernier lieu, nous concluons ce mémoire en présentant en annexe, quelques informations sur le travail établi

Chapitre I

1. Introduction :

Dans ce chapitre nous allons présenter les différentes solutions électriques adoptés avec quelques notions générales sur les moteurs pas à pas et le moteur à courant continu, le PIC 18F2550, ainsi que nous allons présenter aussi le système de commande.

Dans un premier lieu, on a essayé de se familiariser avec la description des éléments de nos cartes avec le choix de microcontrôleur puisqu'on a choisi le 18F2550 pour la raison USB.

Ensuite on a étudié le fonctionnement de notre moteur pas à pas et le moteur à courant continu, et les différents composants essentiels pour la commande, puisqu'ils jouent un rôle essentiel pour le fonctionnement de notre montage.

Finalement, la dernière partie sera consacrée à une récapitulation sur les principes de fonctionnement de notre carte et la solution électrique adoptée.

2. Description des éléments de la carte :

2.1. Le microcontrôleur pic 18f2550 :

2.1. 1. Identification d'un pic :

Pour identifier un pic, nous utilisons simplement son numéro.

Les 2 premiers chiffres indiquent la catégorie du pic, 18 indique un PIC **HIGH-END**.

Vient ensuite parfois une lettre :

L : celle-ci indique que le PIC peut fonctionner avec une plage de tension beaucoup plus tolérante.

C : indique que la mémoire programme est une EPROM ou plus rarement une **EEPROM**.

CR : pour indique une mémoire de type **ROM**

F : pour indiquer une mémoire de type **FLASH**.

À ce niveau, on rappelle que seule une mémoire FLASH ou EEPROM est susceptible d'être effacée.

Finalement nous trouvons sur les boitiers le suffixe « -xx » dans lequel xx représente la fréquence d'horloge maximale que le PIC peut recevoir. Par exemple -04 pour 4 MHZ Notons dès à présent que le PIC est des composants STATIQUES, c'est-à-dire que la fréquence d'horloge peut être abaissée jusqu'à l'arrêt complet sans perte de données et sans dysfonctionnement

Ceci par opposition aux composants DYNAMIQUES (comme les microprocesseurs de nos ordinateurs), donc la fréquence d'horloge doit rester dans des limites précises.

2.1.2. Le Choix du microcontrôleur :

Le choix de microcontrôleur est essentiel car c'est de lui que dépendent en grande partie les performances. La taille, la facilité d'utilisation et le prix du montage.

En fait ce choix est imposé dans le cahier de charge.

Le PIC18F2550 possède en plus des instructions très puissantes donc un programme à développer réduit, une programmation simple grâce au mode série. En fait la cause principale du choix du 18F2550 est qu'il dispose de l'option de gestion du bus USB pour satisfaire le besoin de notre application

2.1.3. La structure interne du microcontrôleur 18F2550 :

Nous allons maintenant s'intéresser à la structure interne du PIC 18F2550, avec lequel nous avons travaillé.

Le 18F2550 est un microcontrôleur de MICROCHIP, fait partie intégrante de la famille 18 des HIGH-END dont la mémoire programme est de type flash (f) et capable d'accepter une fréquence d'horloge jusque de 48 MHZ.

Il possède un grand nombre de caractéristiques mais la principale raison pour laquelle on l'a choisi est qu'il gère l'USB en hardware et que MICROCHIP fournit toutes les bibliothèques adéquates.

Le PIC 18F2550 est un microcontrôleur c'est-à-dire une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans l'ajout de composants externes.

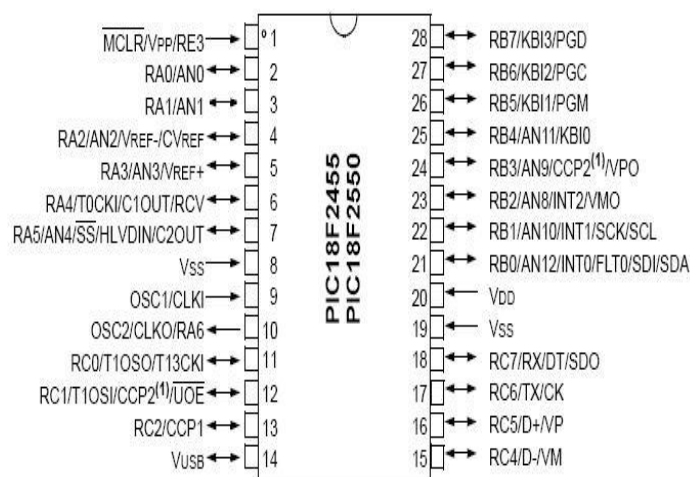


Figure 1 : brochage de PIC 18F2550

On peut distinguer sur ce schéma :

- L'alimentation : VDD (+5V) et Vss (0V)
- Les bornes du quartz (oscillateur a quartz) : OSC1 et OSC2
- L'entrée RESET (MCLR : Master CLearR)
- Les différents ports d'Entrées/Sorties : RAx, RBx, RCx

La dénomination pic est sous copyright de micro-chip, donc les autres fabricants ont été dans l'impossibilité d'utiliser ce terme pour leur propre microcontrôleur .Le PIC 18F2550 est à architecture RISC qui possède des caractéristique standards comme une ROM sur puce de code ou de programme, une EEPROM de données, et des ports entrées/sorties dont la figure 2 montre ces périphériques.

2.1.4. Organisation du 18F2550 :

- Mémoire de programme (flash) 32 K octets
- EEPROM 2048 octets
- RAM 256 octets
- Communication série avec MSSP (SPI, 12c), USART.
- Un convertisseur analogique numérique 10bits avec 13 canaux d'entrées
- Un comparateur dual analogique avec entrée multipliée
- 3 timers (timers0, timers1, timers2)
- Capture : 16bits max résolution 6.25ns.
- Compare : 16bits max résolution 100ns.
- Deux modules de génération d'impulsion à période réglable (PWM) 10bits résolution
- Un bus Can 2.0 (Controller Area Network)

2.1. 5. Les registres :

- Les registres PORTA
- Les registres PORTB
- Les registres PORTC

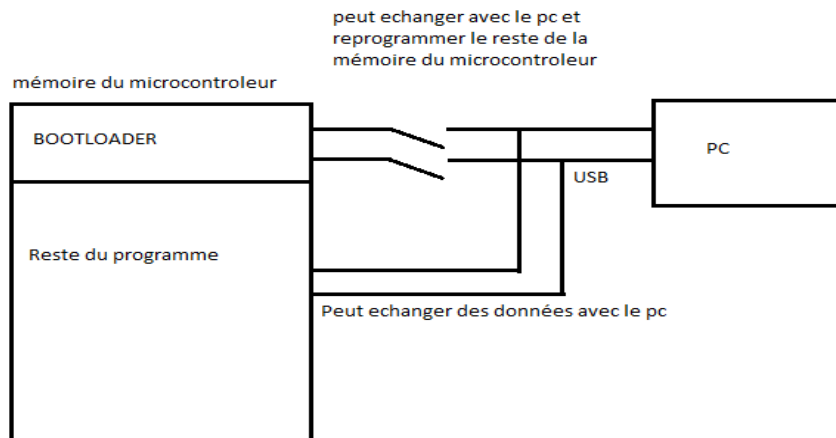
2.1. 6. Étude du boot loader de micro-chip :

Nous avons donc fait un petit montage pour essayer de programmer le PIC directement par l'USB par la technique du boot loader, nous pouvons voir ici le PIC, son quartz de 20 MHZ,

Un BOOTLOADING est un programme qui se loge en début de la mémoire programme. Si on lance le microcontrôleur normalement, il exécutera le programme situé après le BOOTLOADER.

Dans certaines conditions il peut entrer dans mode BOOTLOADER et alors nous pouvons, grâce à une interface logicielle sur le pc, reprogrammer la partie du code après le

BOOTLOADER sans passer par un programmeur externe et comme il le montre la figure suivante.



2.2. Module USB du PIC 18F2550 :

Le PIC 18F2550 possède un module USB composée de : SIE, régulateur interne, résistance pull-up, émetteur-récepteur, qui sont représentés dans le schéma de la figure 2

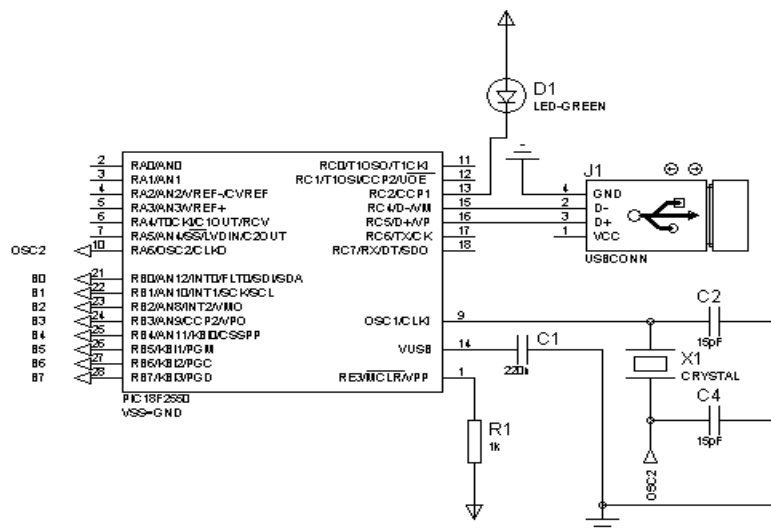
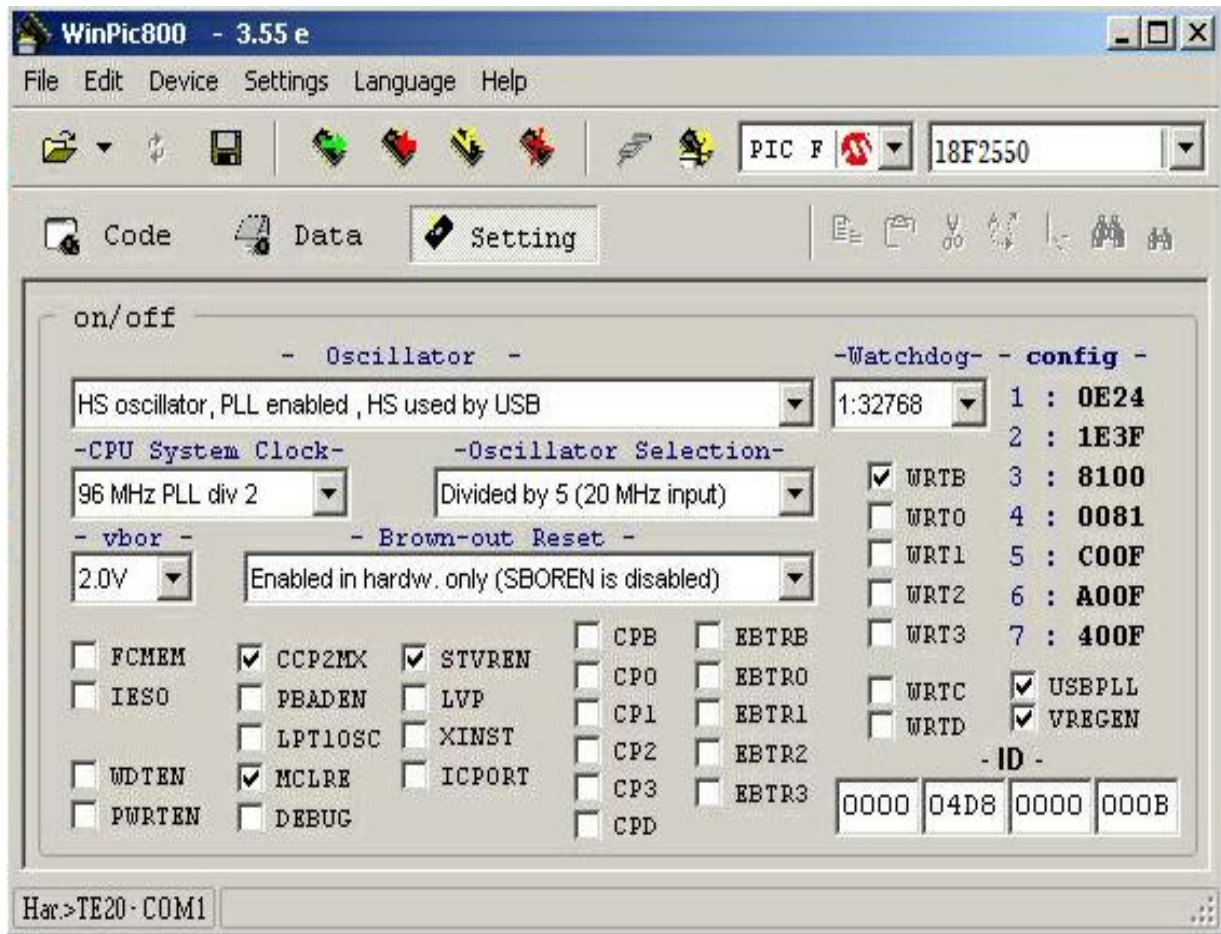


Figure 2: schéma minimaux pour USB-BOOTLOADING

2.3. Programme du pic 18F2550 :

2.3. 1. Présentation de l'environnement de programmation :

Après avoir écrit notre code en assembleur, nous le compilons et le résultat sera stocké dans un fichier .HEX à charger dans le pic grâce à un logiciel spécifique (pic flashe program) et un kit, dont la figure 6 représente le logiciel.



3. Partie puissance :

3.1. Introduction :

La circulation d'un courant électrique dans un bobinage entraîne l'apparition d'un champ magnétique, comme le détaille la figure ci-dessous dans le cas du solénoïde, et donc la présence de pôles Nord et Sud; c'est sur ce principe de base que repose le fonctionnement de tous les moteurs électriques.

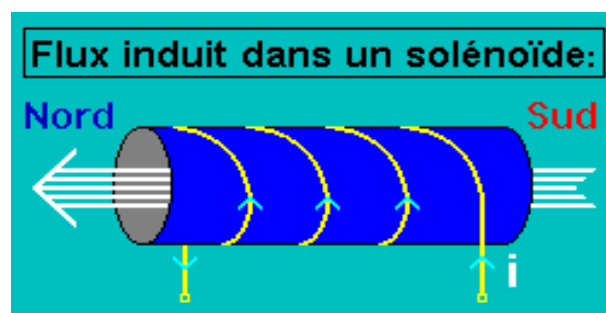


Figure3: circulation du courant dans une bobine

3.2. Moteur pas à pas :

3.2.1 Définition :

Un moteur pas à pas est une machine tournante dont le rotor se déplace d'un angle élémentaire à chaque fois que son circuit de commande effectue une commutation de courant dans un ou plusieurs de son enroulement, Il s'agit donc d'un actionneur de positionnement.

Les moteurs pas à pas sont différents par rapport aux moteurs classiques. Au lieu de leur fournir une tension continue, on peut alimenter des bobines dans une séquence précise. Grâce à ce principe, on peut déterminer l'angle exact de rotation de l'axe, de plus, en laissant une ou plusieurs bobines non alimentés, on obtient un maintien : le moteur est figé.

Chaque impulsion envoyée par le circuit de commande au module de puissance se traduit par la rotation d'un pas du moteur. L'angle de rotation minimal entre deux modifications des impulsions électriques s'appelle un pas. On caractérise un moteur par le nombre de pas par tour (c'est à dire pour 360°). Les valeurs courantes sont 48, 100 ou 200 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.

Exemple:

Moteur à 400pas= 0.9°

Moteur à 200pas= 1.8°

Moteur à 100pas= 3.6°

Moteur à 48pas= 7.5°

Moteur à 24pas= 15°

3.2.2. Différent types de moteur pas à pas :

En peut classer les moteurs pas à pas en 3 catégories:

- Moteur à aimant permanent;
- Moteur à réluctance variable;
- Moteur hybride.

3.2.2.1. Moteur à aimant permanent :

Il se compose de deux parties :

- 1- le rotor qui est la partie mobile, comporte un nombre pair à aimant permanent magnétique dans le sens radial.
- 2- Le stator qui est la partie fixe, comporte d'encoches comportant 2, 3,4 enroulements Électriques.

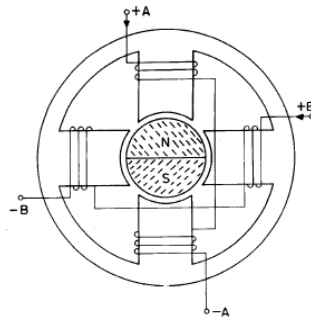


Figure4: Moteur à aimant permanent

Caractéristiques principales :

- Faible résolution : nombre de pas/tour peu important.
- Couple d'utilisation plus élevé par rapport au moteur à reluctance variable.
- Présentation d'un couple résiduel lorsque le moteur est hors tension.

3.2.2.2. Moteur à reluctance variable :

- Le stator présente un certain nombre de dents ayant un bobinage.
- Le rotor (en matériau magnétique) possède un nombre différent de dents, mais sans bobinage.
- Le rotor se positionne pour que la réluctance du circuit magnétique soit minimum.
- 12 pas par tour ou 30° par pas.
- Des séquences pour un tour complet.

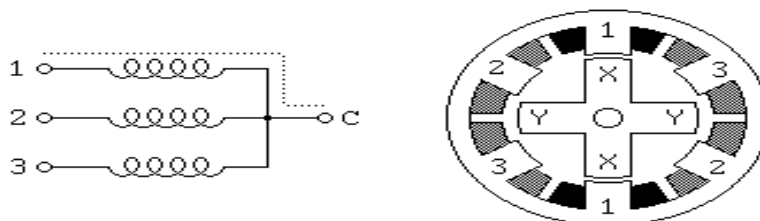


Figure5: Moteur à reluctance variable

Caractéristiques principales :

- Les fréquences de fonctionnement peuvent être élevées.
- Bonne résolution.
- Construction simple mais délicate. Couple développé.
- Absence de couple résiduel avec le moteur hors tension.

3.2.2.3. Moteur hybride :

C'est un moteur à reluctance polarisée. 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 et inséré un aimant permanent. Le stator et le rotor ont un nombre de dents différents. Quand

on alimente une paire de bobines, Le rotor place les dents nord et sud de telle façon que le flux traversant le rotor soit maximal.

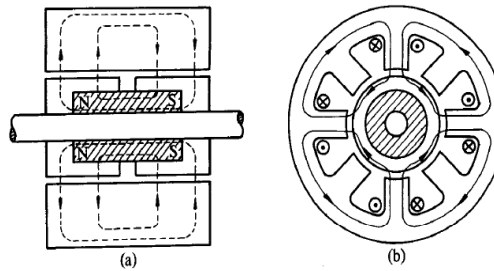
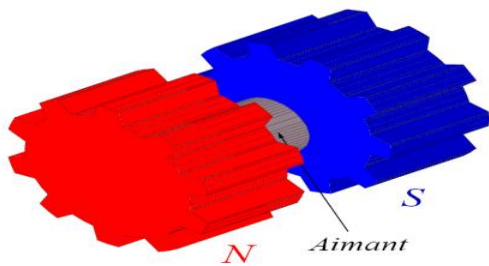


Figure6: Exemple de rotor de Moteur hybride

Figure7: Moteurs hybride

Caractéristiques principales :

- Hors tension, le rotor est maintenu en position.
- Bonne précision de la position du rotor.
- Grande vitesse de rotation.

3.2.3. Le mode de contrôle :

3.2.3.1. Le mode unipolaire :

La commande unipolaire n'utilise en effet qu'une bobine par phase. Elle se fait entre une extrémité et le point milieu de bobinage. Cela permet néanmoins de simplifier l'électronique de pilotage vu que l'on travaille dans un environnement numérique.

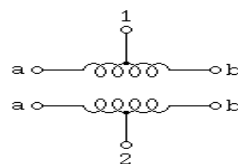


Figure8: Mode unipolaire

3.2.3.2. Le mode bipolaire :

Elle permet d'augmenter le couple de moteur, par sa position ses enroulements.

Les enroulements au stator n'ont pas de point milieu, chaque borne de chaque enroulement est alimentée successivement par une polarité positive puis négative en inversant les polarités des enroulements statique. On inverse les pôles nord et sud de stator.



Figure9: Mode bipolaire

3.2. 4. Caractéristiques couple et vitesse :

3.2.4.1. Couple d'arrêt ou couple de maintien :

C'est le couple maximum de rotation avec lequel on peut solliciter l'arbre d'un moteur pas à pas excité statiquement, sans qu'il ne se produise de modification de son angle de rotation.

3.2.4.2 Plage de démarrage :

C'est la plage dans laquelle un moteur pas à pas peut être actionné en synchronisation avec la fréquence de travail sans rampe d'accélération ou de décélération.

3.2.4.3 Fréquence limite de démarrage :

C'est une fréquence maximale avec laquelle un moteur pas à pas ne peut démarrer à la charge indiquée.

3.2.4.4. Plage d'accélération:

C'est la plage de travail dans laquelle un moteur pas à pas peut être actionné en synchronisation avec la fréquence de travail, sans qu'il ne se produise d'erreur de pas. Il faut cependant qu'il soit actionné avec une rampe d'accélération et de décélération.

3.2.4.5. Couple limite de travail ou d'entraînement :

C'est un couple de rotation maximale avec lequel on peut solliciter un arbre de rotation avant qu'il ne sorte de la cadence.

3.2.4.6. Fréquence maximale des pas :

C'est une fréquence maximale admise avec laquelle un moteur pas à pas est actionné à vide sans perte de pas. Cependant, le moteur ne peut être démarré ou stoppé avec cette fréquence sans perte de pas.

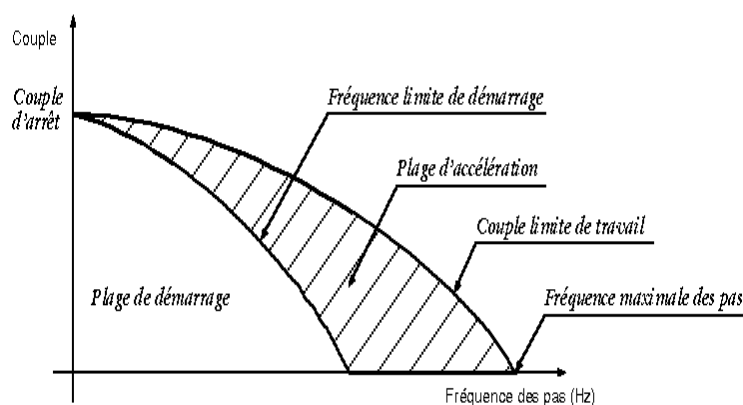


Figure10: Domaines de fonctionnement du moteur pas à pas.

3.2. 5. Configurations interne des bobines du moteur :**3.2.5.1. Les moteurs à 4 fils (bipolaire) :**

Ce moteur agit comme s'il ne possédait que 2 bobines, il est obligé d'être alimenté soit une bobine à la fois, ou les deux en même temps. A tout moment, donc le moteur a la moitié ou la totalité de ses bobines alimentées, ce qui a comme avantage de lui donner plus de force.

Par contre, il est plus complexe de contrôler un moteur bipolaire, au niveau de l'interface de puissance.

3.2.5.2. Les moteurs à 5 fils (unipolaire) :

Le moteur comporte deux bobines à point centrales, on relie le point central à l'alimentation et les autres bobines à l'interface de puissance.

3.2.5.3. Les moteurs à 6 fils :

Avec le moteur à 6 fils, on a le choix d'une commande bipolaire, ou d'une commande unipolaire. Dans le premier cas, on ignore simplement les connexions centrales, et dans le second cas, on relie les deux points centraux au (+) de l'alimentation.

3.2.5.4. Les moteurs à 8 fils :

Avec ce moteur, on a aussi le choix d'une commande bipolaire, ou unipolaire. Dans le premier cas, on ignore les 4 fils centraux, et dans le second cas, on reliant les quatre fils centraux ensemble.

3.3 Commande des moteurs pas à pas:

Le rôle de l'électronique associée à un moteur pas à pas est d'assurer les commutations nécessaires pour obtenir un déplacement d'un nombre de pas donné ou une rotation à vitesse fixée. Trois dispositifs sont nécessaires.

Le principe de commande du moteur pas à pas est en boucle ouverte

3.3. 1. Unités de pilotage :

Elle génère 2 signaux numériques. Une première sortie donne un bit qui permet de définir le sens de rotation. Une seconde sortie délivre les impulsions. Un nombre N d'impulsions correspond à un déplacement de N pas .Une répétition périodique des impulsions avec une fréquence correspondante à une rotation continue à une vitesse angulaire ; Cette fonction est souvent réalisée par un circuit programmable (PC, microcontrôleur).

3.3. 2. Le séquenceur :

Il dirige les impulsions vers les différentes phases du moteur suivant le moteur utilisé (Nombre de phase) et le mode de fonctionnement choisis (pas entier, demi pas).Ce circuit délivre sur ses sorties des signaux logiques définissant les commutations à réaliser pour chaque phase suivant les séquences précises, il s'agit d'un circuit logique câblé (bascules,...).

Cette fonction peut également être remplie par un circuit programmable.

3.3.3. Les commutateurs de puissance ou l'interface de puissance :

Il permet d'alimenter correctement les phases du moteur en fonction des informations reçues du séquenceur, il doit donc être en mesure de fournir le courant nécessaire aux enroulements du moteur, il s'agit d'une fonction analogique de puissance qui sera réalisé à partir d'un composant intégré ou discret :(Transistor ...).

Chaque phase possède une voie séparée dans le commutateur.

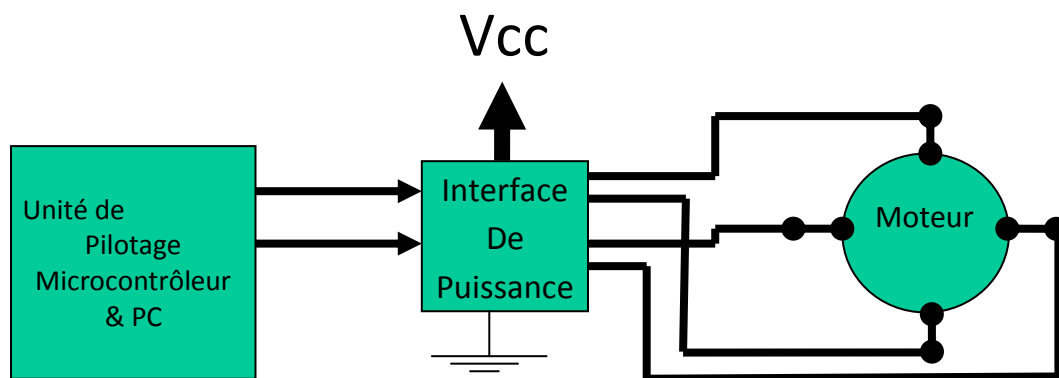


Figure11:Synoptique de la commande d'un moteur pas à pas (Boucle ouverte)

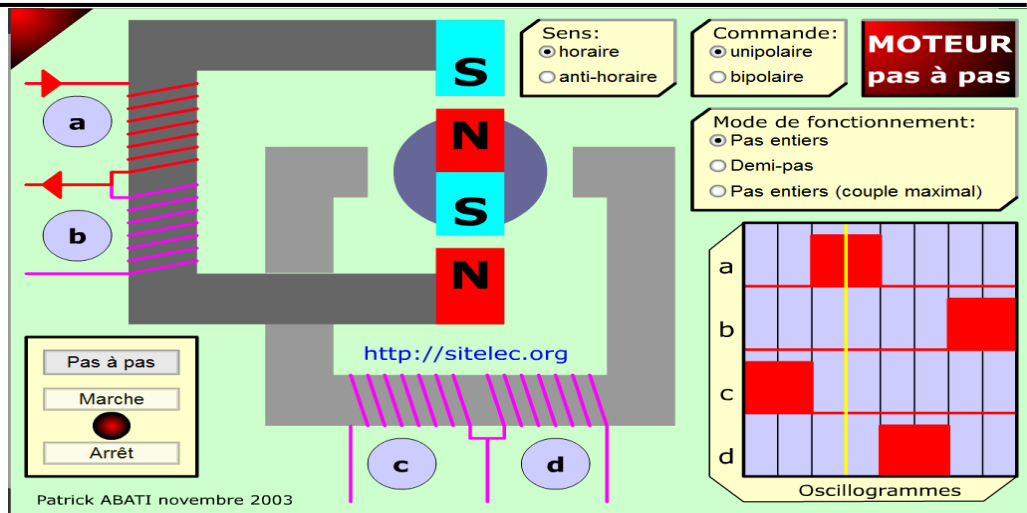
3.3. 4. Le mode de séquence :

Les modes de séquence se distinguent par leur fonctionnement interne, leur séquence générée, et par la façon dont on les commande.

3.3.4.1. Pas entier :

- **Monophasé:**

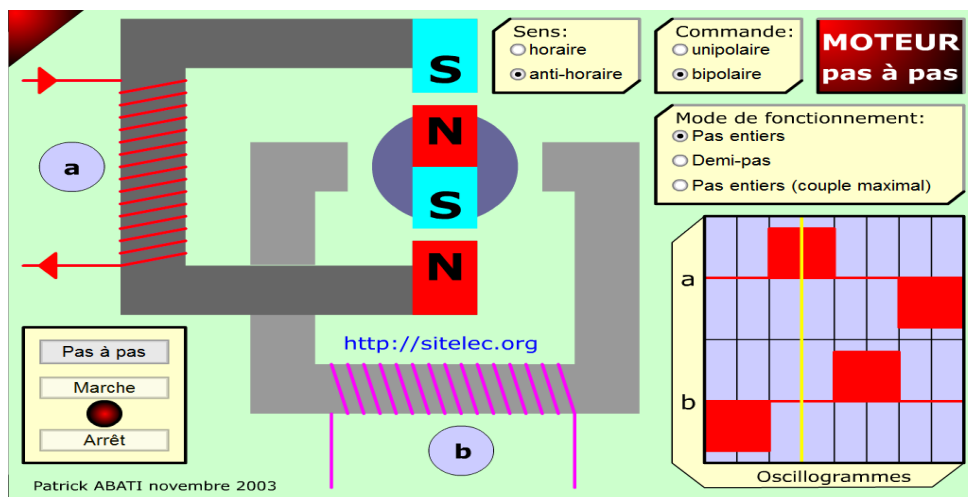
Cette séquence permet normalement d'alimenter une bobine à la fois, en se plaçant à des positions de détente.



Fonctionnement en mode monophasé

- **Biphasé :**

Cette fonction permettra au moteur de faire un pas complet où les positions d'équilibre sont à mi-chemin entre les positions de la séquence la plus simple.



2. Fonctionnement en mode biphasé

3.4 Alimentation des moteurs pas à pas:

3.4. 1. Alimentation unidirectionnelle et bidirectionnelle :

Le moteur pas à pas peut être alimenté soit en courant soit en tension. On distingue deux cas, dans le premier cas le courant dans les phases est toujours dans le même sens(alimentation unidirectionnelle), il est valide pour les moteurs à réluctance variable qui ne sont pas sensibles au sens de courant, ainsi qu'avec certains moteurs à aimant, en particulier ceux qui sont munis d'enroulement bifilaire tandis que dans le deuxième cas, il faut à chaque fois inverser le sens du courant (alimentation bidirectionnelle). Il est utilisé pour les machines à aimant ou hybrides munies d'enroulements simples.

3.4.2. Alimentation en tension :

Circuit de base

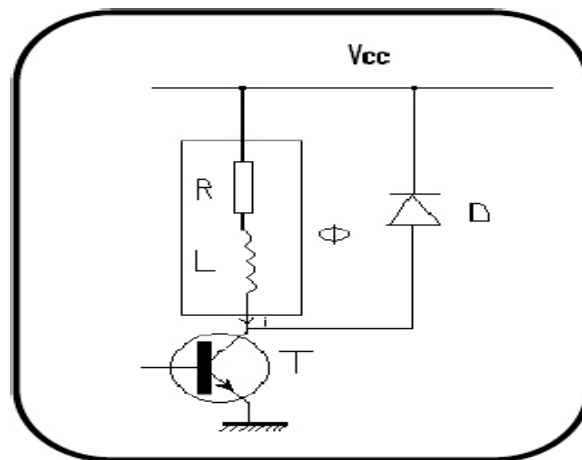


Figure12: Alimentation en tension unidirectionnelle

3.4.2.1. Alimentation unidirectionnelle:

Le schéma pour phase \emptyset comporte une alimentation continue fixe V_{cc} , un interrupteur commandé (transistor bipolaire, montage Darlington ou transistor MOS) et une diode de roue libre.

On considère tout d'abord que la force électromotrice de la rotation est négligeable. Le schéma équivalent de la phase se réduit alors à un dipôle RL série. Le séquenceur bloque le transistor lorsque la phase ne doit pas être alimentée et il le sature à l'instant de la commutation où doit apparaître le courant. Le circuit étant inductif, la variation de l'intensité ne peut pas être discontinue. La croissance est exponentielle, de constante de temps $T = R/L$

3.4.3. Le critère de choix d'un moteur pas à pas :

Le choix se fait comme suivant:

- Le mode de commande, soit bipolaire ou unipolaire.
- Le nombre de pas par tour.
- La fréquence de travail.
- La puissance du moteur.

3.5 Les moteurs à courant continu :

Un moteur électrique est un dispositif électromécanique permettant la conversion d'énergie électrique en énergie mécanique. Comme toutes les machines tournantes, les moteurs électriques à courant continu sont constitués d'un stator et d'un rotor. Le stator crée une magnétisation fixe à l'aide d'enroulements (inducteur), cette fonction peut être assurée par un courant électrique circulant dans un bobinage. Le rotor est constitué d'un ensemble de bobines reliées à un collecteur rotatif. Le collecteur rotatif permet de maintenir fixe la direction

transversale de magnétisation du rotor lorsque celui-ci tourne. Grâce à ce dispositif, les magnétisations rotoriques et statoriques, sont toujours décalées de façon optimale. Ce décalage provoque un couple selon la loi du flux maximum (un pôle nord attire un pôle sud), provoquant ainsi la rotation du rotor.

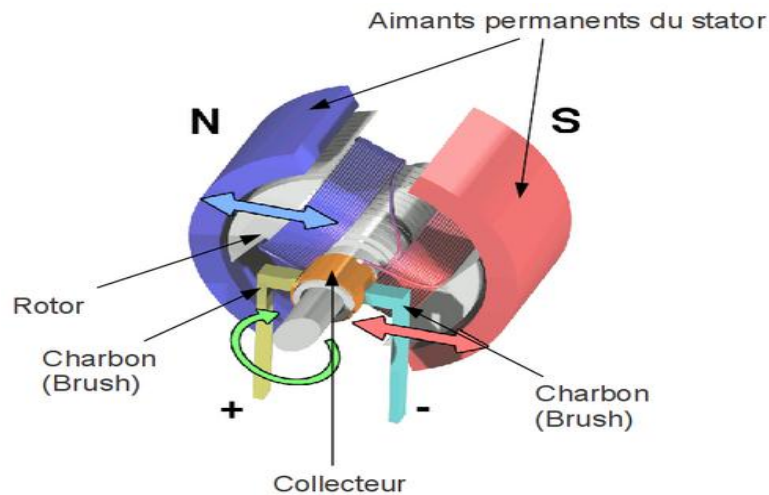


Figure 13 : Moteur à courant continu

L'avantage principal des machines à courant continu réside dans leur adaptation simple aux moyens permettant de régler ou de faire varier leur vitesse, leur couple et leur sens de rotation.

Le principal défaut de la machine à courant continu réside dans l'ensemble balais/collecteur rotatif qui s'use, et complexe à réaliser et consomme de l'énergie.

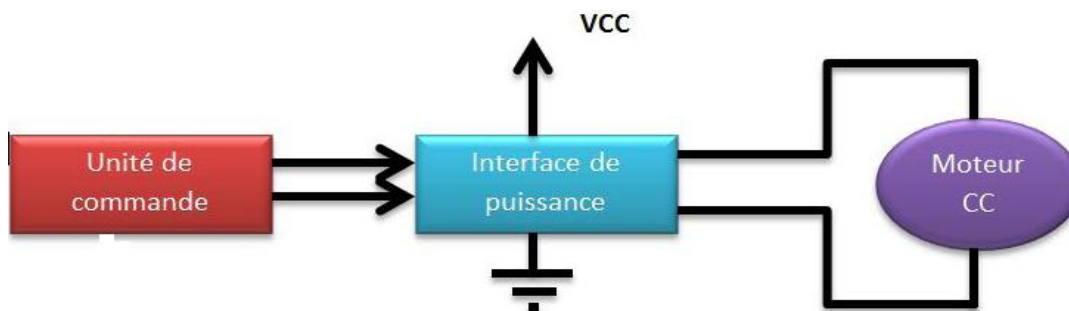
3.5. 1. Le principe de fonctionnement des moteurs à courant continu :

Lorsqu'on alimente un des bobinages du rotor, un second flux magnétique est créé, et l'interaction entre les deux flux crée un couple au niveau du bobinage alimenté due aux forces de Lorentz. Le couple créé est proportionnel aux flux magnétiques, donc aux courants dans le bobinage du rotor et éventuellement du stator, et à l'angle entre les deux flux. Pour maintenir un couple maximum, on utilise un collecteur rotatif composé de différents secteurs qui vont alimenter les différentes bobines à travers des balais. Les bobinages du rotor se déplacent dans le champ magnétique du stator. Une force contre électromotrice apparaît donc selon la loi de Lenz, proportionnelle au flux statoriques et à la vitesse de rotation. Pour assurer la circulation du courant dans les bobinages, la tension d'alimentation devra donc rester supérieure à cette FCEM.

3.5. 2. La commande des moteurs à courant continu :

Le rôle de l'électronique associée à un moteur à courant continu est d'assurer les commutations nécessaires pour obtenir une rotation à vitesse variable.

Le principe de commande du moteur à courant continu est en boucle ouverte :



3.5. 3. Interface de puissance d'un moteur CC :

Puisque le microcontrôleur ne peut pas générer des signaux de puissance capables d'alimenter les moteurs directement, un circuit d'interface de puissance est requis pour cette raison, le MOS FET irf 450 est un transistor mos utilisé pour la commande des moteurs à courant continu.

3.5. 4. Variation de vitesse du moteur :

Pour faire varier la vitesse d'un moteur on peut faire varier la tension d'alimentation à ses bornes mais dans ce cas une partie importante de l'énergie est consommée par le dispositif d'alimentation, on préfère l'alimenter de façon discontinue avec un hacheur et faire ainsi varier la tension moyenne à ses bornes. On parle alors de Modulation par Largeur d'Impulsions (MLI), ou Pulse With Modulation (PWM).

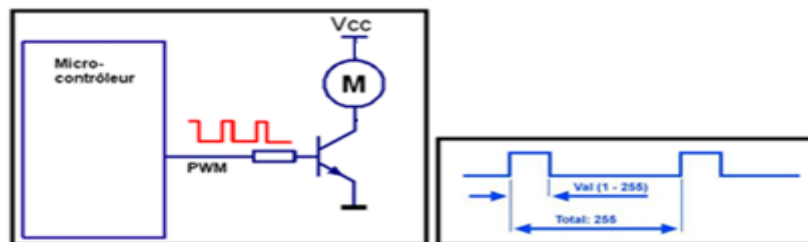
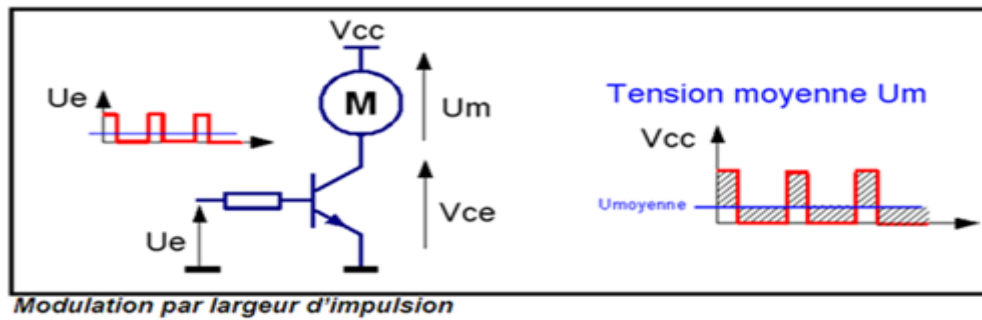
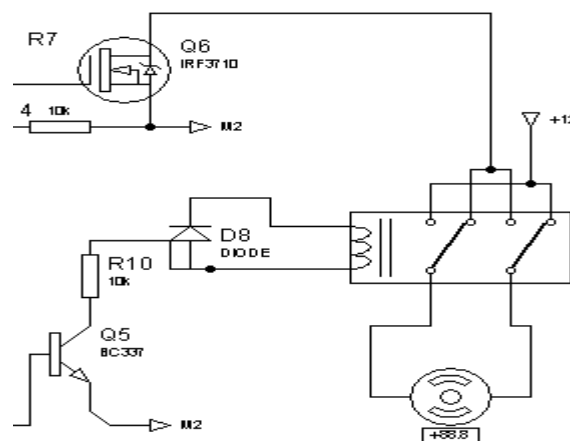


Figure14 : exemple de commande par microcontrôleur

3.5. 5. Variation de sens du moteur :

Pour pouvoir inverser le courant à partir d'une alimentation fixe unique, on utilise deux relais pour obtenir un sens inverse en saturant le bobine à partir de la PORTB 6.

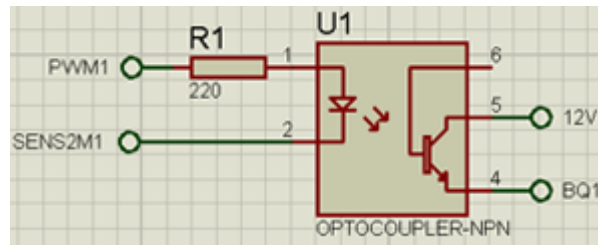
Le fonctionnement est identique à celui de l'alimentation unidirectionnelle



La sortie PWM du microcontrôleur est utilisée pour commander le transistor, La PWM permet d'obtenir un équivalent d'une variation de tension continue à l'aide d'un control TOR (tout ou rien) .Cette technique permet aux composants de puissance de moins chauffer qu'en analogique .D'autre part, les signaux numériques sont moins sensibles au parasites que les signaux analogiques et sont donc plus robustes.

3.5. 6. Partie isolation galvanique :

C'est un dispositif d'isolement qui permet la transmission de niveau électrique de manière non galvanique. Il est constitué d'une diode électroluminescente (LED) et d'un phototransistor intégré dans le même boîtier. Il réalise une conversion de "l'énergie électrique en énergie électrique, la tension d'isolement entre les deux parties de circuit peut être de l'ordre de quelques milliers de volt.



Optocoupler NPN

3. CONCLUSION :

L'étude théorique faite durant ce premier chapitre, s'appuie essentiellement sur quelques notions des moteurs pas à pas et moteur à courant continu, elle est nécessaire pour tenir compte des contraintes que toute réalisation se rapporte à ces études qui devraient être respectés.

Le principe de fonctionnement, les fonctionnalités du système conçu et tout ce qui a rapport avec la simulation du système sera étudié dans le dernier chapitre concernant les outils de développements et simulation.

Chapitre II

1. Introduction :

Dans un premier lieu, on a essayé de se familiariser avec la définition de la norme USB et ses particularités comme l'avantage et la facilité d'utilisation puis on a passé au choix de protocole de cette dernière selon notre montage en optant pour le protocole HID.

2. Généralité sur L'USB :

2.1 Définition :

L'Universal Serial Bus (USB) est une norme relative à un bus informatique en transmission série qui sert à connecter des périphériques informatiques à un ordinateur. Le bus USB permet de connecter des périphériques à chaud (quand l'ordinateur est en marche) et en bénéficiant du Plug and Play (le système reconnaît automatiquement le périphérique).

Avec près de 10 ans d'existence, la norme qui a permis d'unifier la manière dont on connecte un périphérique à un ordinateur évolue encore une fois avec la version 3.0. Après les versions Full Speed (USB 1.0), High Speed (USB 2.0), voici la version appelée super Speed (USB 3.0).

2.2 Origine de L'USB :

Le bus USB est donc réellement né de l'alliance en 1994 de sept partenaires industriels (Compaq, DEC, IBM, Intel, Microsoft, NEC et Northern Telecom). C'est eux qui ont commencé à créer la norme USB. Le bus USB a été conçu à l'origine pour faciliter les transferts de données en particulier définir une connectique (universelle) et « Plug & Play », utilisable aussi bien pour une souris que pour un modem ou un moniteur.

2.3 Avantages de L'USB :

Les avantages de l'USB sont nombreux:

- faible coût de l'interface.
- alimentation possible des dispositifs via le câble
- indépendance vis-à-vis des machines hôtes, Hot Plug Play (c'est-à-dire branchement et débranchement sans avoir besoin d'arrêter le PC). jusqu'à 127 périphériques possible.
- fiabilité et sécurité (détection et correction d'erreurs).
- plusieurs vitesses possibles et 4 types de transferts

2.4 Facilité d'utilisation :

Le principal but du bus USB est la facilité d'utilisation qui se traduit suivant de multiples critères que voici:

Tout d'abord on peut dire que le protocole USB est une norme très souple; une interface unique suffit pour commander plusieurs types de périphériques, il suffit juste de posséder le bon driver.

Ensuite, un point fort de l'USB est sa configuration automatique, on l'appelle aussi le

« Plug &Play». Cela signifie que si l'utilisateur connecte un périphérique USB, Windows détecte automatiquement ce périphérique et charge le driver approprié s'il est disponible dans les fichiers de Windows. Si ce n'est pas le cas, Windows demande d'installer le disque (CD Rom) contenant ce driver afin qu'il le copie dans son répertoire de drivers (généralement c:/windows/system32/drivers) et ceci se fait une seule fois.

Lors du prochain branchement, le périphérique USB recharge automatiquement son driver, cette étape est alors transparente pour l'utilisateur. Il faut noter aussi qu'il n'est pas nécessaire, avec le protocole USB, de lancer un fichier d'installation ou de redémarrer le PC avant d'utiliser le périphérique.

Il n'y a pas non plus dans le protocole USB à faire le choix de l'adresse du port comme par exemple pour une liaison série, une adresse dynamique est allouée à chaque fois que l'on branche un autre périphérique, la norme USB est capable de fournir 127 adresses dynamiques différentes (Codée sur 7 bits) c'est pour cela qu'on entend toujours parler de brancher jusqu'à 127 périphériques sur le bus USB.

Le PC alloue donc une série d'adresses de port et une commande d'interruption pour les interfaces USB. Contrairement aux autres périphériques qui ne sont pas USB, pour chaque ajout d'un périphérique il faut lui associer une adresse de port, ainsi qu'une carte d'extension si le PC n'en possède plus ou pas assez. Avec les périphériques USB, tous ces problèmes sont inexistantes.

Un atout majeur du bus USB est qu'il est très facile à connecter, on n'a pas besoin d'ouvrir le PC à chaque fois que l'on veut rajouter un périphérique.

On peut aussi connecter un HUB sur un port déjà existant pour pouvoir connecter encore plus de périphériques USB. Le HUB joue en quelque sorte le rôle d'une multiprise. C'est un multiplieur de port USB. Actuellement certains moniteurs, claviers ... possèdent des HUBs intégrés.

On peut évidemment reconnecter d'autres HUBs aux HUBs cité précédemment. Nous verrons par la suite les consignes à respecter pour conserver l'esprit de la « norme USB »Un autre

avantage du bus USB est la connectique. En effet les câbles possèdent deux extrémités bien différentes pour ne pas se tromper lors du branchement, de plus les câbles USB sont très compacts comparés aux câbles pour les liaisons parallèles ou séries.

3. vitesse de transfert de L'USB :

3.1 Définitions des différentes vitesses :

Voici une petite comparaison entre les débits théoriques proposés par les trois normes USB, ainsi que ceux proposés par la norme Serial ATA dans ses différentes versions.

| Débits théoriques | USB 1.0 | USB 2.0 | USB 3.0 | Serial ATA | Serial ATAI | Serial ATAIII Prochaine évolution |
|-------------------|---------|-----------|----------|------------|-------------|--------------------------------------|
| En Mbits/Gbits | 11Mbits | 480 Mbits | 5 Gbits | 1.2 Gbits | 2.4 Gbits | 4.8 Gbits |
| En mégaoctets | 1.3Mo/s | 60 Mo/s | 640 Mo/s | 150 Mo/s | 300 Mo/s | 600 Mo/s |

3.2 Raison des trois vitesses de L'USB :

Le Low Speed a été introduit pour deux raisons essentielles, le premier est que les périphériques USB ne sont pas chers, c'est à dire que tout le monde peut se le permettre.

La deuxième raison est pour avoir des souris USB plus pratiques. En effet les câbles USB pour le Low Speed n'ont pas besoin d'être blindés et, de ce fait, sont très souples.

Le Full Speed a été conçu pour remplacer les liaisons séries et parallèles.

Le High Speed est une vitesse supplémentaire qui a été introduit lors de la mise à jour de la norme USB 2.0. Elle permet de mettre en valeur la puissance du bus USB.

3.3 Domaine d'utilisation des différentes vitesses :

Les applications Low Speed concernent essentiellement des périphériques interactifs (claviers, souris, consoles), mais aussi des afficheurs, des lecteurs (de carte à puce) et des applications en automatismes (mesure, capteurs) appelées à se développer.

En mode Full Speed on va trouver la téléphonie, les modems, les disques, les imprimantes, les fax ainsi que les scanners, certains lecteurs de carte à puce et le domaine multimédia (jeux, audio, vidéo limitée).

3.4 Les débits de L'USB :

Il faut noter que dans la norme USB le débit n'est pas proportionnel à la vitesse.

L'USB « Low Speed » qui est consacré aux périphériques soit disant lents est limité à échanger au maximum 8 octets toutes les 10ms, ce qui correspond à un débit maximum de 800 octets/sec soit 6400 bps. Par certains moyens détournés il est possible d'espérer atteindre 8Ko/s, mais c'est sans garantie.

L'USB « Full Speed » qui est consacré aux Hubs et aux périphériques non lents peut échanger jusqu'à 1024 octets toutes les ms, soit un débit de 1 Mo/s.

4 Le câble USB :

4.1 Définition du câble USB :

Le câblage USB est relativement simple; il a la même structure quelle que soit la vitesse de transmission. Le câble transporte deux paires de fils:

La paire de signal destinée au transfert de données D+ et D- et une seconde paire qui peut être utilisée pour la télé alimentation GND et Vcc. La première paire est non blindée pour les périphériques lents tels que les claviers, souris fonctionnant à 1.5Mbits/s tandis que caméras, micro et autres ont recours à une paire de fils torsadée blindée pour atteindre les 12Mbits/s

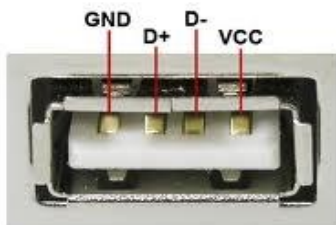


Figure15: Composition d'un câble USB

4.2 Composition du câble USB :

Chaque connecteur dispose de deux fils d'alimentation (SV et GND) et deux fils destinés au transfert de données (D+ et D-).

Une connexion entre deux PC est aussi possible par l'adjonction d'interface spéciale qui déjoue la vigilance du PC maître et transforme le second PC en «esclave» En version Low Speed le blindage n'est pas obligatoire (ce qui assure une plus grande souplesse de manipulation en particulier pour une liaison souris).

| Fonction | Couleur | Numéro de broche pour les type A et B | Numéro de broche pour les types mini B |
|--|---------|---------------------------------------|--|
| Alimentation +5v (VBUS) 500mA maximum | Rouge | 1 | 1 |
| Données (D-) | Blanc | 2 | 2 |
| Données (D+) | Vert | 3 | 3 |
| Masse(GND) | Noir | 4 | 5 |

Figure 16.Composition du couleur du câble USB

La longueur maximale autorisée par la norme est de 3m pour un câble non blindé donc généralement pour un périphérique Low USB (= 1.5Mb/s) et de 5m pour un câble blindé dans le cas d'un périphérique Full USB (=12Mb/s).

Le câble USB est composé de deux fiches bien différentes:

En amont d'une fiche appelé connecteur USB de type A, branché au host (PC).

L'extrémité aval par contre peut se retrouver en deux versions:

Connecteur USB du type B et un mini connecteur type B (appelé souvent optionnel). Ce dernier est réservé aux dispositifs de très faible dimension (ou de grande intégration) telles appareils photo numériques.

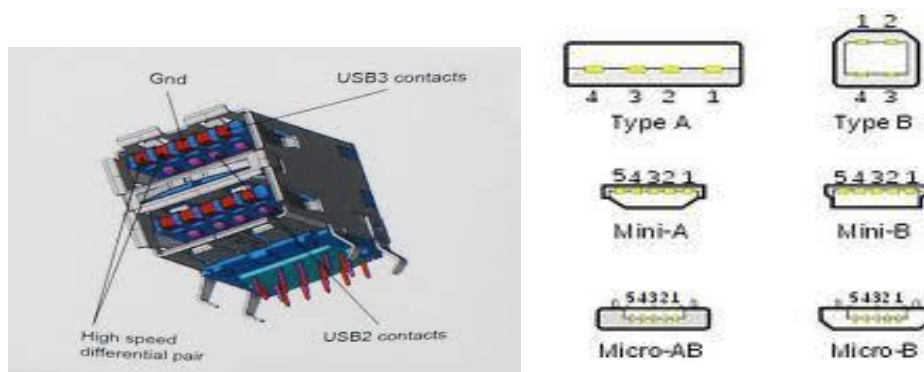


Figure17: Brochage des connecteurs USB de type A et B

Et voici L'aspect des connecteurs en trois dimensions:



Figure 18: Aspect des connecteurs USB de type 13(2.0 &3.0)

La norme qui a permis d'unifier la manière dont on connecte un périphérique à un ordinateur évolue encore une fois avec la version 3.0. Après les versions (USB 1.0) et (USB 2.0), voici la version appelée Super Speed (USB 3.0). Les périphériques compatibles se verront apposer un logo "Super Speed" pour les différencier des autres versions. Bien entendu, la principale différence est le taux de transfert proposé, qui est largement supérieur à celui des deux précédentes versions.

Avec l'USB 3.0, on annonce un débit théorique de 5 Gbits/s comparés aux 480 Mbits... De l'USB 2.0 et aux 11 Mbits de l'USB 1.0. En d'autres termes, on a multiplié les performances par dix ... en théorie bien sûr. Comme un tableau vaut mieux qu'un long discours, voici une petite comparaison entre les débits théoriques proposés par les trois normes USB, ainsi que ceux proposés par la norme Serial ATA dans ses différentes versions.

5 Le Bus USB :

5.1 Principe du bus USB :

Le bus USB est un bus fonctionnant sur la hiérarchie, commandé par un host unique.

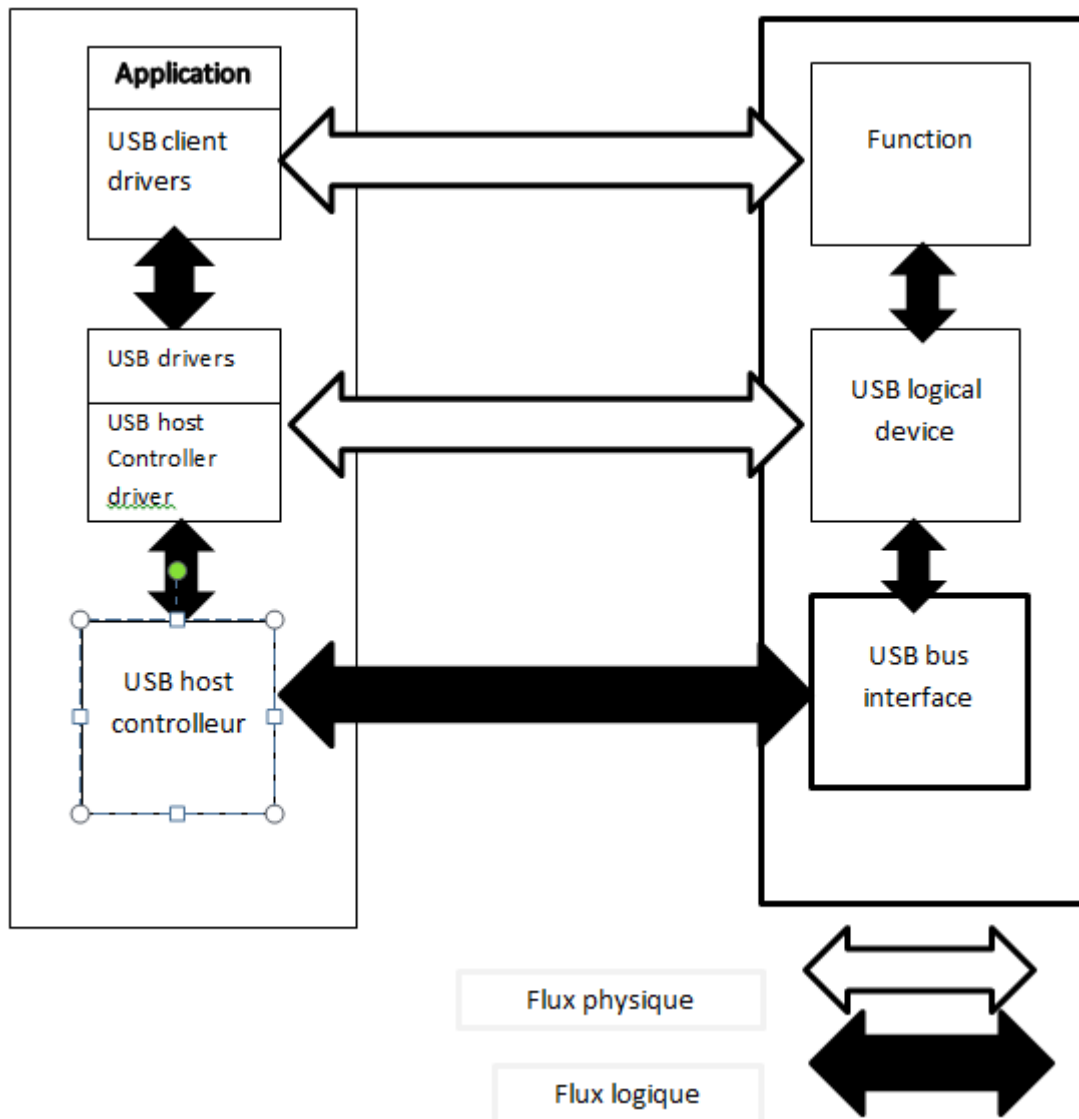
Le host utilise un protocole Maître/Esclave pour communiquer avec les périphériques USB. Cela signifie que c'est le host qui décide du transfert des données et que les différents périphériques ne peuvent pas établir de connexion entre eux tant que le maître n'a pas donné l'autorisation. On peut dire que cela peut être un inconvénient par rapport aux autres protocoles mais il ne faut pas oublier que l'USB a été conçu avec des compromis de coût et de performance. Le fait que le bus USB fonctionne avec le protocole Maître/Esclave résout implicitement ces problèmes comme par exemples les problèmes de collision ou d'arbitrage de périphériques.

5.3 Protocole USB :

Le protocole USB est, comme tous les autres protocoles un protocole à encapsulation.

Mais avant de s'intéresser à la trame proprement dite, voyons ce qu'il en est du protocole.

Le client driver communique les demandes de transfert des applications via des IRP (I/O Paquet). Puis, l'USB driver traduit chaque transfert en une suite de transactions. Ensuite l'USB Host Controller driver regroupe les transactions en trames et finalement l'USB Host Controller traduit les transactions en paquets et enchaîne les trames.



5.4 Type de paquet USB :

Contrairement à la liaison série RS232 et des interfaces séries similaires où le format des données envoyées n'est pas défini, l'USB lui est composé de plusieurs couches de protocoles bien définis. La plupart des circuits intégrés USB s'occupent de la couche inférieure, la rendant ainsi presque invisible au regard du concepteur final. Mais il est tout de même intéressant d'en connaître les grandes lignes.

L'USB a quatre types différents de paquet:

- ✓ Token (En-tête)
- ✓ SOF (Start of frame)
- ✓ Data (Optional)
- ✓ Aknowledge (Handshake)

5.5 Codage NRZI :

Pour transmettre les données, l'USB utilise le codage NRZI (Non-Retour à Zéro Inversé).

On ne va pas s'étendre longuement là-dessus, je pense que cela n'est pas indispensable pour comprendre le fonctionnement de l'USB.

Le principe de ce codage est simple, Un (1) logique est représenté par un non changement d'état en NRZI et un «0» logique est représenté par un changement d'état.

Le codage va encore plus loin et utilise le Bit Stuffing ; c'est le fait de mettre un «0» après 6 «1» logique consécutif pour forcer une transition dans le code NRZI (pour éviter les pertes de données).

Ci-dessous un exemple ce codage NRZI. Ce type de codage est uniquement utilisé pour le transport à travers le cordon USB il doit être décodé lors de la réception pour pouvoir retraiter les données

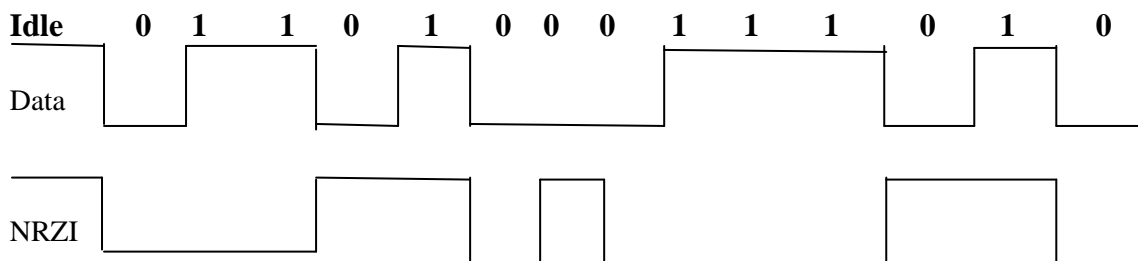


Figure 20: Principe du codage NRZI

5.6 Alimentation USB :

5.6.1. Alimentation de périphérique USB :

Pour simplifier l'explication de l'alimentation des périphériques USB, la norme a prévu deux niveaux d'alimentations, le premier niveau consomme une unité d'énergie, l'autre consomme cinq unités. Une unité vaut 100 mA... C'est à dire qu'il existe des composants qui consomment 100 mA et d'autres 500mA. Généralement les composants Low USB consomment une unité d'énergie et les composants High USB consomment jusqu'à cinq unités. Par défaut tous les composants consomment une unité et c'est par soft qu'on lui demande de consommer plus si l'application le nécessite, dans le cas d'un composant High USB bien sûr. Toutes ces informations sont contenues dans les descripteurs, le composant ne pourra jamais consommer

plus que ce qui est prescrit dans son descripteur. On peut donc en déduire de cela qu'un périphérique High power doit posséder un dispositif d'alimentation séquentiel (100 mA, puis 500 mA). Aucune fonction (Device ou hub) ne peut consommer plus de 100 mA sur le bus avant d'être énumérée. Après énumération un Device peut consommer jusqu'à 500mA pour un device « High power device » ou rester à 100 mA pour un Low Power Device. L'USB est donc assez flexible et peut supporter plusieurs types d'alimentations. Certains composants peuvent être entièrement alimentés par le bus USB. (Bus powered).

5.6.2. Avantage de l'alimentation USB :

Le fait de pouvoir alimenter un périphérique USB avec le même câble qui transporte les données a de multiples avantages. D'une part cela évite déjà à l'utilisateur de brancher le périphérique sur une prise de courant extérieure, ceci rend déjà le périphérique plus léger et moins encombrant. D'autre part d'un point de vue du concepteur, cela réduit le prix de fabrication.

5.6.3. Différents types d'alimentation du BUS USB :

- . Low-power bus-powered functions
- . High-power bus-powered functions
- . Self-powered functions

5.7 Courant de veille :

Pour limiter la consommation un mode suspend a été imaginé: toute fonction y passe après 3ms d'inactivité du bus qui la relie au hub amont (consommation inférieure à 500 μ A dont 200 réservés à la seule résistance de tirage sur D).

Le courant de veille est proportionnel à l'unité de charge, pour un appareil consommant une unité, le courant de veille est de 500 μ A, essentiellement dû aux résistances de Pull Up sur le bus.

5.8. Différents types de transfert :

5.8.1. Transfert en mode control :

Ce mode de transfert est compatible avec le Low et Full Speed USB. Il est utilisé pour les opérations d'initialisations et de configurations. Il est éventuellement utilisable pour les transferts standards. Le mode contrôle est aussi utilisé pour tenter d'obtenir un débit Low Speed acceptable, ou pour utiliser le driver de classe HID standard.

5.8.2. Transfert en mode interruptif :

Ce mode de transfert également compatible avec le Low et Full Speed USB. Il est Destiné à des échanges limités et périodiques, il garantit la fréquence de scrutati6n ainsi que la reprise sur les erreurs. li est utilisé pour des transferts à l'initiative du périphérique (asynchrones) et pour des transferts périodiques ou permanents comme les claviers.

5.8.3. Transfert en mode Isochrone :

Ce mode de transfert est uniquement compatible avec le Full USB. La bande passante est garantie (début, latence) par contre dans ce mode il n'y a pas de reprise sur erreur il est utilisé pour des transferts nécessitant un flux régulier de données comme par exemple les caméras ou les téléphones ... La bande passante réclamée et non utilisée est perdue.

5.8.4 Transfert en mode Bulk :

Ce mode de transfert est uniquement compatible avec le Full USB. Ce mode est réservé. pour les gros transferts de données (ex : imprimantes ...) Le débit est variable et dépend de la disponibilité. Ce mode assure la reprise sur les erreurs. Les échanges isochrones sont les plus privilégiés dans le sens où le host leur réserve une bande passante garantie. Celui-ci peut refuser l'accès au bus à un périphérique s'il juge que les ressources qu'il requiert ne sont pas disponibles.

5.9. L'énumération :**5.9.1. Définition de l'énumération :**

Le terme «énumération» désigne un processus USB par lequel le système identifie et configure le périphérique en lui donnant une adresse unique, C'est une gestion dynamique de la connexion et de la déconnexion des périphériques reliés à un bus USB.

5.9.2. Principe de fonctionnement de l'énumération :

En effet lors de la connexion (ou déconnexion) il y a une phase de détection et une phase d'identification effectuée par l'hôte qu'on appelle énumération. Lors de cette phase, le périphérique fournit à l'hôte une suite de descripteurs qui permettent son identification complète, Lors de cette phase d'énumération, on assigne une adresse unique (Unique ID) au périphérique, on charge le driver correspondant et on positionne le composant dans la configuration qui lui à été donnée par les descripteurs. Il n'est pas indispensable de connaître parfaitement le processus d'énumération et le système de descripteurs pour pouvoir faire fonctionner un composant USB mais il est bon d'en connaître les grandes lignes pour pouvoir, au besoin, changer les descripteurs ou bien par simple culture générale. Notons également

pour ceux qui ne le savent pas encore, que cette phase d'énumération est totalement transparente et automatique pour l'utilisateur.

Lors de l'énumération initiale (à la mise sous tension du PC) les HUBs et périphériques. Sont initialisés de proche en proche.

Ainsi le HUB racine signale que sur ses ports A et B il y'a des périphériques non initialisés. L'hôte initialise alors une liaison occupée et la place dans sa liste de scrutation, puis passe la liaison suivante.

Puis c'est au tour du deuxième HUB donc l'hôte va initialiser successivement les liaisons occupées et va les placer dans sa liste de scrutation. Puis on passera au HUB suivant s'il en a un, etc. jusqu'à ce que tous les périphériques connectés aient été initialisés.

5.10. Les descripteurs :

5.10.1. Définition d'un descripteur :

On peut définir les descripteurs comme étant des blocs d'informations pré formatés. Tout composants USB doit obligatoirement posséder les descripteurs standards.

Tous les transferts d'informations durant cette phase d'énumération se font suivant le type Control. Il va de soi que tout composant USB doit pouvoir être capable de supporter ce type de transfert.

Nous verrons par la suite que ce n'est pas le cas pour tous les autres types de transfert que nous définirons.

5.10.2. Rôle des descripteurs :

Il existe sur le marché de nombreux périphériques USB. Il a fallu lors de la création de la norme USB, trouver un dispositif pour reconnaître chaque composant USB. Cela était indispensable puisque l'USB devait être un dispositif Plug Play. Lors du branchement du périphérique, le « host » autrement dit plus communément le PC, doit reconnaître tous les périphériques qui lui sont branché. Tout le processus d'énumération se fait grâce aux descripteurs qui sont rassemblés dans IDI fichier texte (fichier assembleur par exemple):en général, un fichier assembleur, qui est ensuite programmé dans le système USB.

Lorsque l'on connecte ou déconnecte un périphérique, celui-ci fournit à l'hôte toutes les informations nécessaires à son identification, c'est à dire ces descripteurs. Ils sont très utiles pour l'hôte puisqu'il ~eut, de ce fait, connaître les caractéristiques périphériques comme par exemple la puissance utile, le type de périphérique, le dispositif de transfert des données, le module de gestion ... etc.

Généralement, clans la plupart des périphériques, toutes ces informations sont stockées clans

la ROM des composants, et lors de l'énumération, le périphérique envoie simplement ce fichier pour se faire connaître. Les descripteurs (standards) sont regroupés en 4 catégories :

- Device descriptor
- Configuration descriptor
- Interface descriptor
- Endpoint descriptor

6. PID/VID :

6.1 Introduction au Product ID et au Vendor ID :

Lors de l'installation d'un nouveau périphérique USB, les descripteurs fournissent les informations le concernant à savoir un PID (Product ID) et un VID (Vendor ID). C'est grâce à ces deux valeurs que le PC peut reconnaître l'identité du composant. Comme dit précédemment la réglementation des VID est très stricte.

Chaque fabricant possède un VID et c'est grâce à cette valeur codée sur 16 bits que l'on peut retrouver les fabricants du composant. Chaque fabricant ayant plusieurs produits à son actif, il les différencie avec le PID codé également sur 16 bits. L'allocation des PID, contrairement aux VID, est faite par le constructeur du dispositif. Il n'y a aucune contrainte administrative de la part du forum USB-IF.

6.2 Le descripteur HID :

Tous les appareils USB ont une hiérarchie de descripteurs qui détaillent pour l'hôte des informations de l'appareil. Les descripteurs les plus courants sont :

- ✓ descripteur d'appareil qui spécifie :
 - La version USB (2.0).
 - Les Vendors ID\Product ID qui sont responsable de l'identification du l'appareil. (0xQ4d8\2007), Le nombre de configuration
- ✓ Descripteur de configuration qui spécifie:
 - Le nombre d'interface pour ce périphérique.
 - La consommation en courant du périphérique
- ✓ Descripteur d'interface qui spécifie:
 - Le nombre de terminaisons pour cette interface. (2 : une entrée et une sortie)
 - La classe de cette interface. (Classe HID)
- ✓ Descripteur de classe HID qui spécifie :

-
- La version de la classe utilisée,(0x0101)
 - Le code du pays.
- ✓ Descripteur de terminaison qui spécifie :
- La direction de la terminaison (entrée ou sortie)
 - La taille maximale des (paquets)

6.3 Product ID et au Vendor ID (PID / VID) :

Lors de l'installation d'un nouveau périphérique USB, les descripteurs fournissent les informations le concernant à savoir un PID (Product ID) et un VID (Vendor ID). C'est Grâce à ces deux valeurs que le PC peut Reconnaître l'identité du composant.

Chaque fabricant possède un VID et c'est grâce à cette valeur codée sur 16 bits que Ton peut retrouver le fabricant du composant (exemple:04d8 c'est le VID de micro chip). Chaque fabricant ayant plusieurs Produits à leurs actifs, ils les différencient avec le PID codé également sur 16 bits. L'allocation des PID, contrairement aux VID, est faite par le constructeur du dispositif.

Il y a quatre moyens de mise en point d'une application USB:

- Classe HID. (Humane Interface Device) requiert un quartz de 20 MHz, aucun pilote nécessaire.
- Emulation port COM (CDC) :(Communication Déviée Caisse) Sans pilote,
- USB Low-speed : avec un quartz de 20Mhz. nécessite un pilote.
- USB full-speed : avec un quartz de 48Mhz nécessite un pilote.

7. Conclusion :

Dans ce chapitre on a décrit la communication de bus USB et ses diverses applications dans le domaine électronique en particulier celle de notre carte de commande.

Chapitre III :

1. Introduction :

Dans ce chapitre, nous expliquons en détail les différentes tâches réalisées, nous avons procédé en deux phases pratiquement conformément au cahier charges établi précédemment.

1-la première partie consiste à réaliser une carte électronique basée sur un microcontrôleur PIC 18F2550 qui permet de communiquer avec le PC via la liaison USB, en utilisant le protocole de communication HID (human Interface Device).

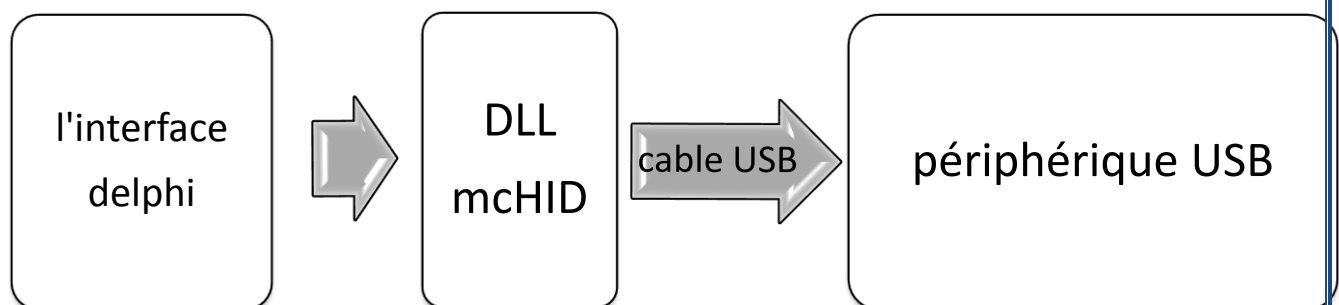


Figure 21: communication entre le PC et le périphérique

Pour exploiter notre réalisation dans une application pratique, nous avons choisi une application qui utilise une communication carte PC, dans laquelle nous avons une câble USB. Notre travail consiste alors en :

Une partie matérielle qui nous a servi à réaliser le circuit électronique.

Une partie logicielle qui consiste à développer le programme nécessaire au fonctionnement du circuit.

Pour pouvoir développer le programme nécessaire au fonctionnement nous avons utilisé des utilitaires :

1. Le logiciel MPLAB pour la programmation du PIC
2. Le compilateur MCC18 pour la programmation en langage C de la série 18Fxxxx
3. Le logiciel de programmation orienté objet DELPHI 7 pour la réalisation des interfaces PC
4. Proteus (ISIS) afin de simuler le fonctionnement du circuit
5. ARES pour réaliser le circuit imprimé.

2.Étapes de réalisation :

2.1. Partie matérielle :

Nous commençons tout d'abord par donner une approche globale sur le circuit, qui a été conçu d'une manière universelle de telle sorte à l'adapter pour chaque application, dont nous détaillerons par la suite en essayant d'illustrer ceci par des schémas.

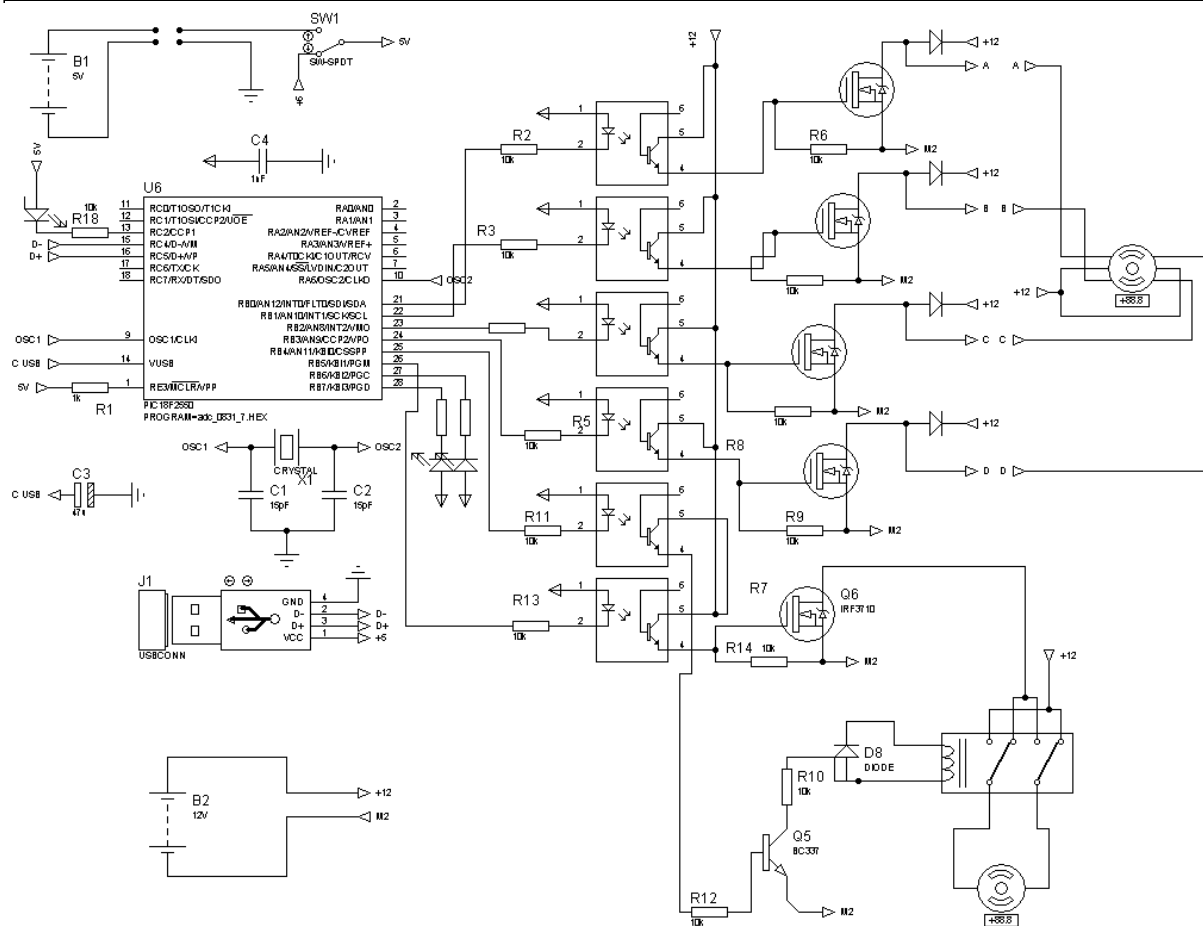
2.1.1Description du circuit réalisé :

Nous avons conçu ce périphérique avec un PIC 18F2550, afin de faire fonctionner cette carte, nous avons choisi un oscillateur à quartz de 12 MHz et nous lui avons ajouté en parallèle deux capacités céramiques $C1=C2=47PF$ pour la stabilité de oscillateur. Nous pouvons augmenter la fréquence jusqu'à 48MHz selon le type de microcontrôleur.

Notre carte est composée de trois fonctions principales :

- a. Piloter un moteur pas à pas
- b. Piloter un moteur à courant continu
- c. Commande par PC via liaison USB

Nous détaillerons la première fonction et ensuite nous expliquerons les deux applications. Le port B du PIC 18F2550 est configuré en sortie et connecté avec des résistances en série avec les LED pour visualiser l'état des sorties du PC vers la carte (RB0 à RB7) les portes RB0 à RB3 pour commander le moteur pas à pas qui sont reliés à des optocoupleur en série et à des transistor de puissance, les ports RB4 RB5 pour commander le moteur à courant continu le premier pour commander la vitesse et l'autre pour commander le sens de rotation du moteur à courant continu, les deux dernier ports RB6 RB7 pour indiquer l'état de transfert des données, les broches RC4 et RC5 du port C sont connectées aux pins D+ \ D- du connecteur USB pour le transfert des données entre le pc et la carte. Le broche RC2 est configurée en entrée et connectée à des résistances et une LED qui indique le passage de courant. Le schéma ci-dessous montre la mise en place du circuit :



2.1.2. Remarque :

Nous avons activé les résistances internes (pull up) du module USB en mettant à 1 le bit UPUEN du registre (UCFG<4>).

2.2. Partie logicielle :

2.2.1. Programmation :

Dans cette partie, nous allons faire une description du programme de chaque élément du système. Pour cela, nous avons élaboré l'ensemble des routines et sous-routines de chaque élément par des organigrammes correspondants et ce, pour une bonne compréhension et exploitation ultérieure de notre réalisation.

L'environnement de développement DELPHI :

DELPHI est un environnement de type RAD (Rapide Application Développement) basé sur le langage Pascal. Il permet de réaliser rapidement et simplement des applications Windows. Cette rapidité et cette facilité de développement sont dues à une conception visuelle de l'application. Delphi propose un ensemble très complet de composants visuels prêts à l'emploi incluant la quasi-totalité des composant Windows (boutons, boîtes de dialogue, menus, liens d'outils...). L'environnement de développement s'appuie sur un éditeur

d'interface associé à un éditeur de code source. Il doit son succès à sa facilité d'utilisation pour développer des applications graphiques.

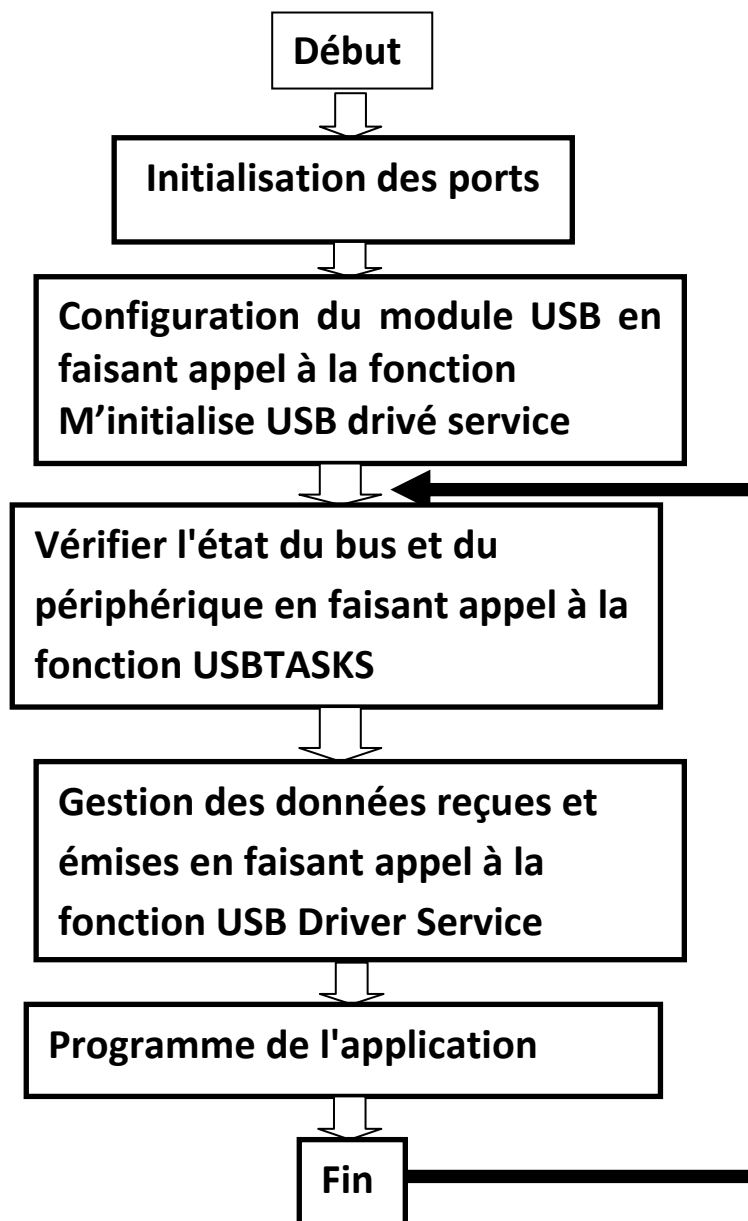
Les différentes étapes pour la construction du programme :

Pour Concevoir notre programme nous avons suivi les étapes suivantes :

- nous avons commencé par comprendre le fonctionnement du module USB du PIC 18F2550
- nous avons ajouté une partie de code nécessaire selon les besoins de notre application afin d'effectuer la communication PIC\PC dans les deux sens.

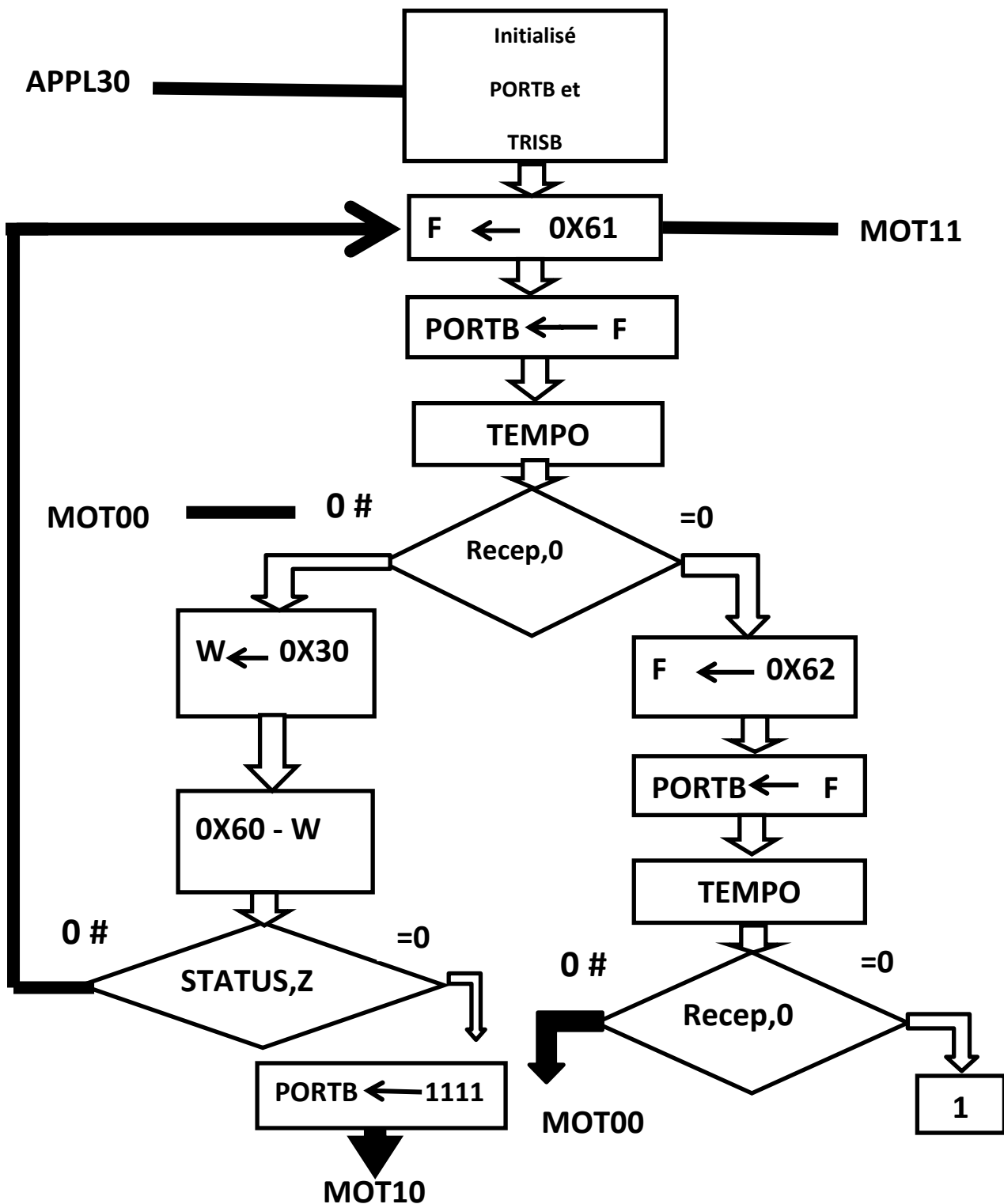
Fonctionnement du programme USB :

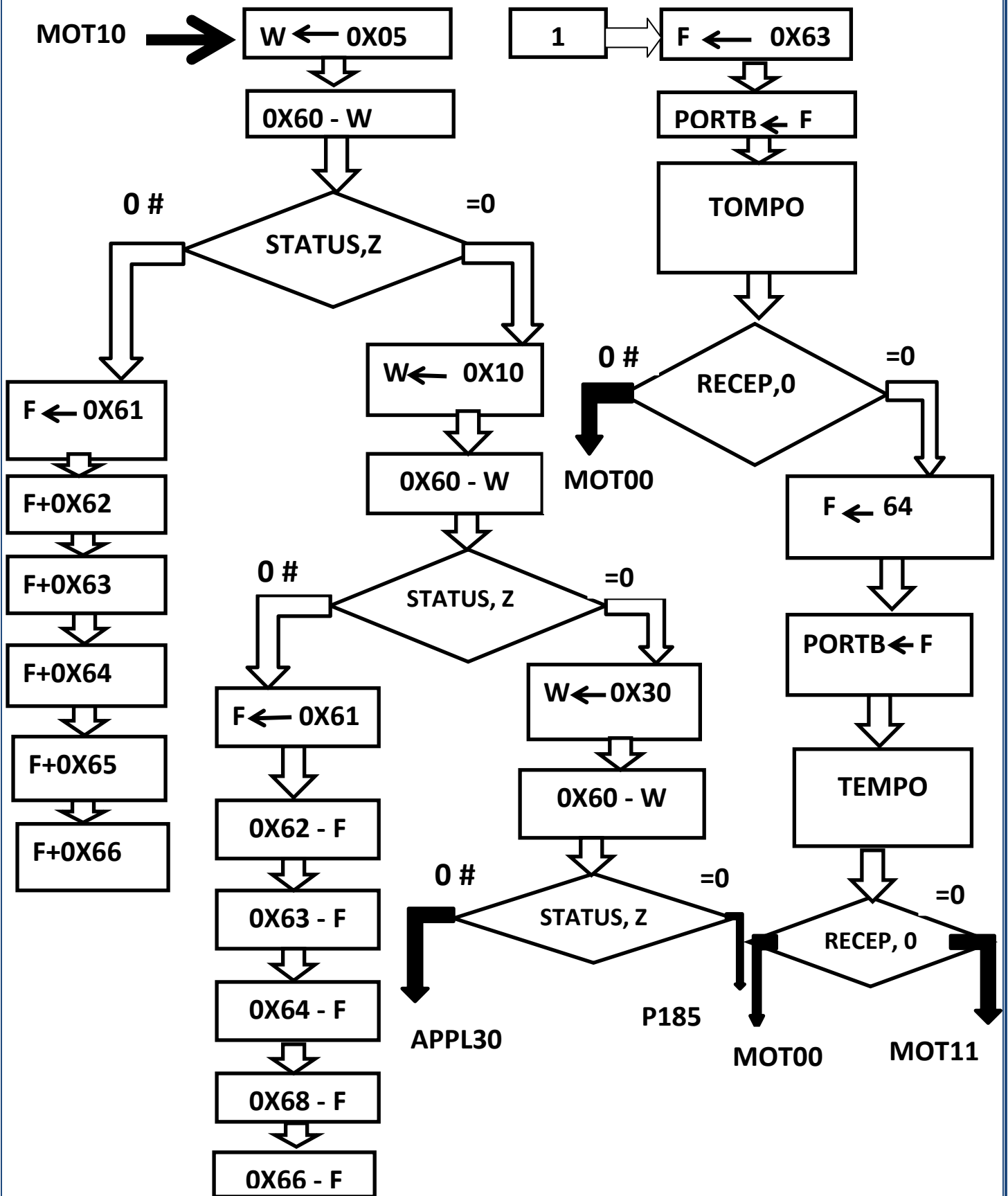
Pour que notre périphérique soit reconnu par le PC, notre programme a été conçu comme indiqué dans l'organigramme qui suit :



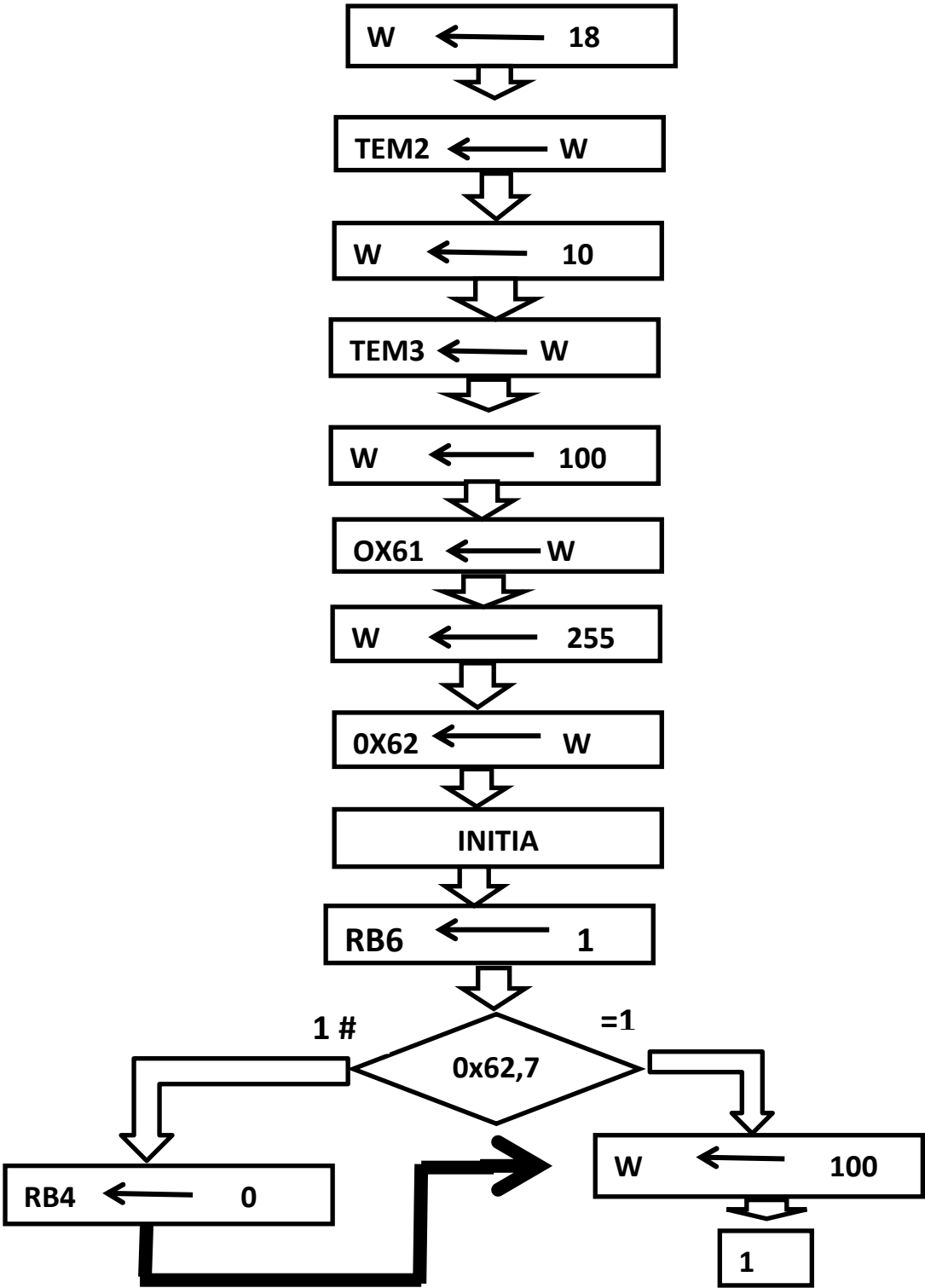
2.2.2. L'organigramme de programme :

✓ L'organigramme de moteur pas à pas :





✓ L'organigramme de moteur courant continu :



2.2.3. L'interface Delphi

Pour pouvoir interagir avec le montage développé, il est nécessaire de concevoir un programme sur le PC qui sert d'interface entre l'utilisateur et le montage réalisé. La figure ci-dessous montre l'interface de la carte réalisée :

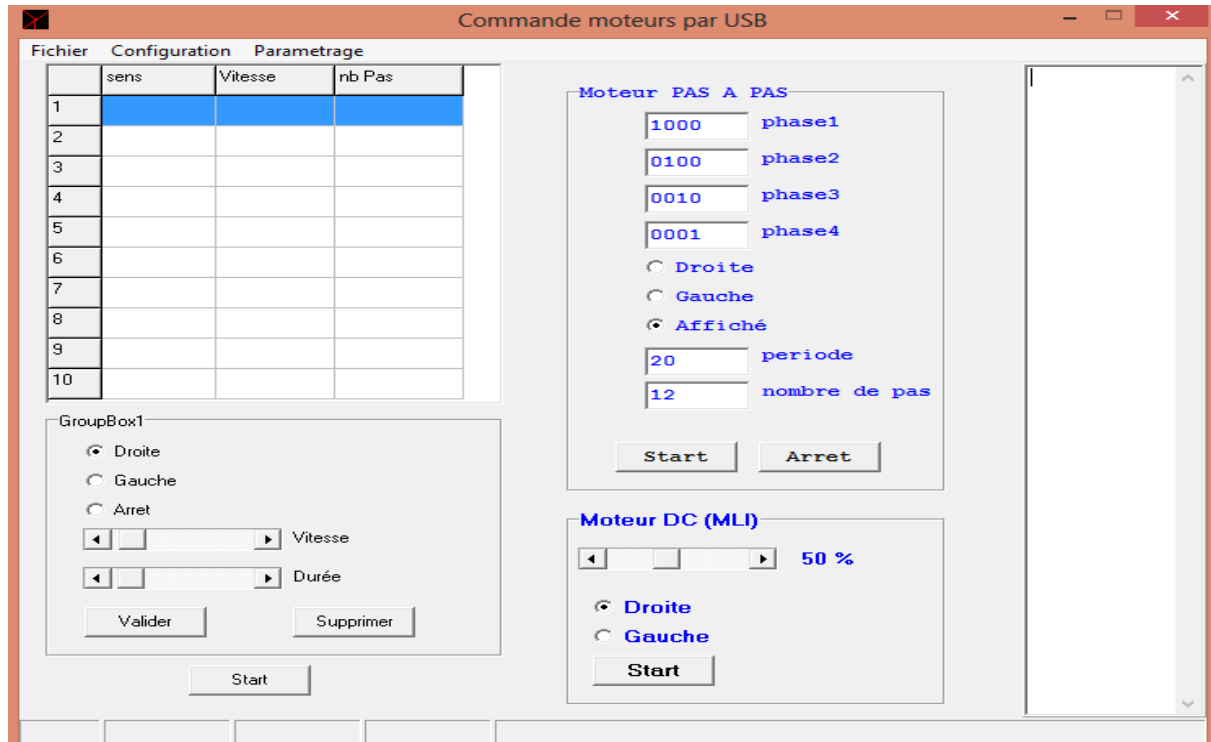


Figure 22: interface de deux moteurs

Afin d'avoir un accès physique avec le port USB de l'ordinateur nous avons utilisé la DLL mcHID qui nous permettra de :

- connecter ou déconnecter le périphérique USB, lire et écrire les données via USB, obtenir le PID et VID de notre périphérique.

- Une fois notre application créée nous appelons la fonction **connecté (application. handle)** définie dans la bibliothèque de fonction DLL mcHID, c'est pour connecter notre périphérique au port USB.

Pour envoyer les données on utilise cette bibliothèque et on fait appel à la fonction :

Fonction Write (pHandle : unit ; pData : pointer) : bool ;

2.2.4. Simulation du circuit :

Cette partie consiste à vérifier le fonctionnement du circuit selon les programmes dont nous avons parlé précédemment par le logiciel de proteus « ISIS ».

ISIS est le cœur du système de simulation proteus, c'est bien plus qu'un logiciel de saisie de schémas ; mais il peut aussi associer un environnement de dessin puissant avec la possibilité

de définir la plupart des aspects de présentation d'un montage. Pour nous, il s'agit de la simulation du circuit de l'application c'est-à-dire la connexion du périphérique par un message Windows, les connexions nous pouvons allumer les led de la carte à partir de l'interface de PC et vise versa.

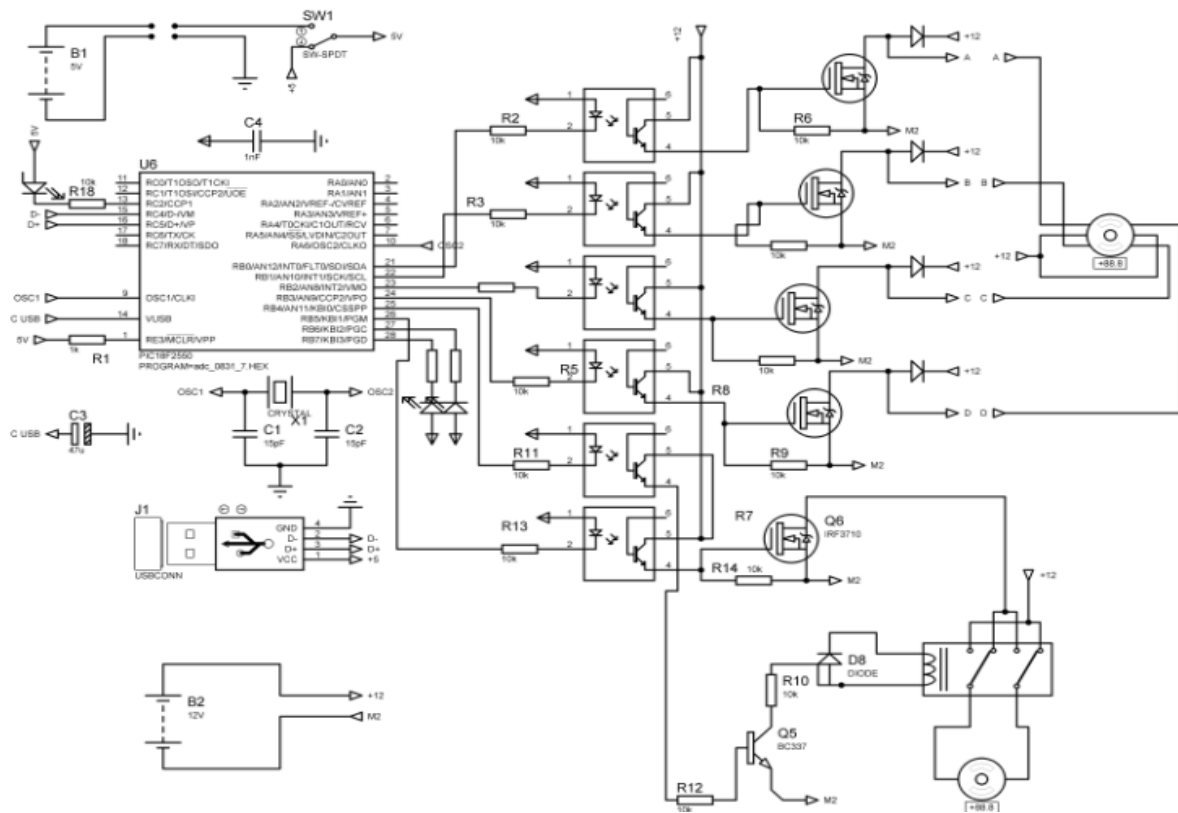


Figure 23: circuit de commande

Le PC détecte un nouveau matériel et affiche son identité en notre écran comme la montre la figure ci-dessous

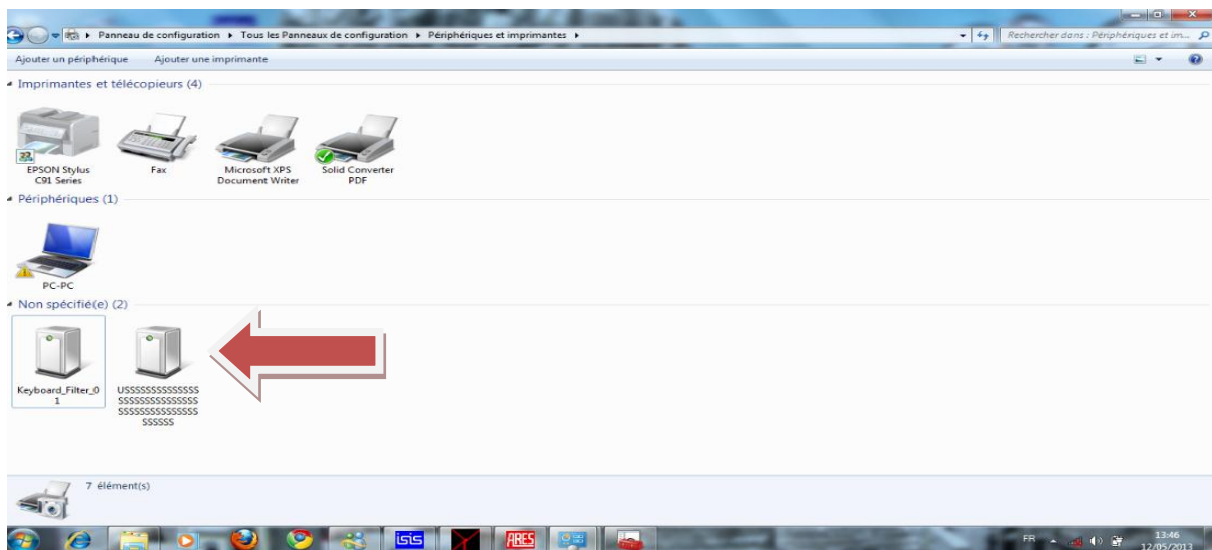
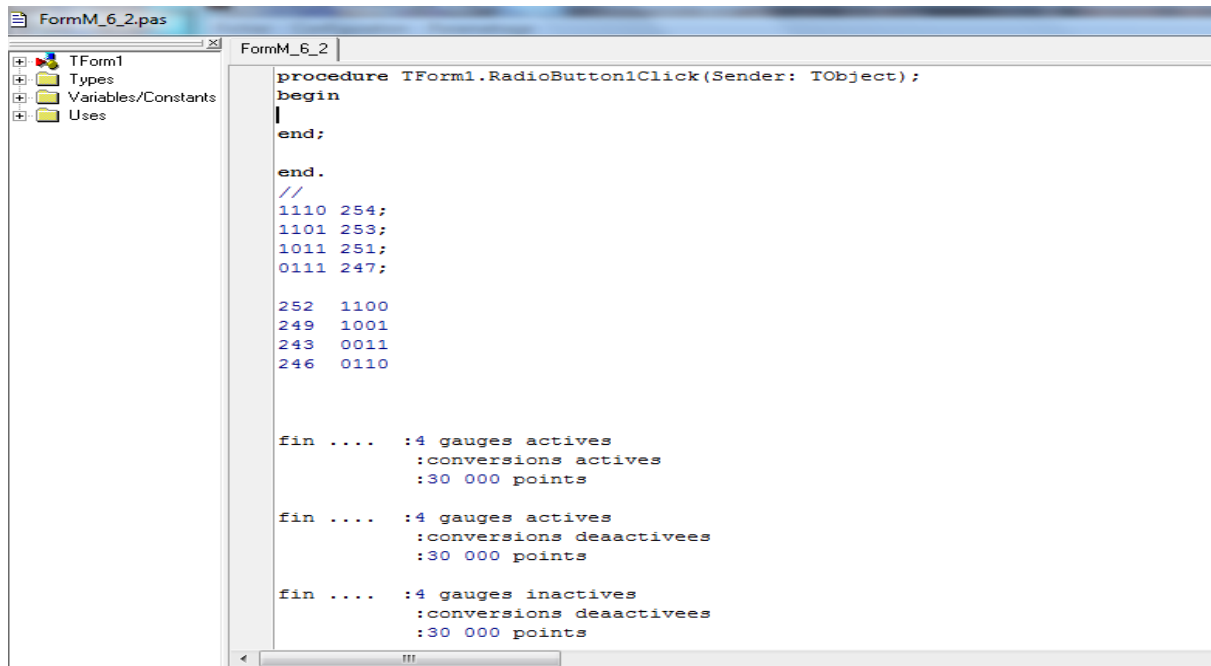


Figure 24: connexion du périphérique USB

Après avoir réussi à identifier notre périphérique qui s'affiche dans le chemin suivant :
Panneau de configuration\Matériel et audio\Périphériques et imprimantes

Puis on lance DELPHI7 pour visualiser le nombre des pas avec des angles correspondants après avoir appuyé sur l'un des interrupteurs pour choisir le sens rotation gauche ou droite.

2.2.5. Quelque historique de mesure:



```
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
|
end;

end.
//
1110 254;
1101 253;
1011 251;
0111 247;

252 1100
249 1001
243 0011
246 0110

fin .... :4 gauges actives
          :conversions actives
          :30 000 points

fin .... :4 gauges actives
          :conversions deaactives
          :30 000 points

fin .... :4 gauges inactives
          :conversions deaactives
          :30 000 points
```

Figure25 : résultat de la rotation

2.3. Essais sur la carte :

2.3.1. Fonctionnalité de l'application :

Notre carte de commande est réalisée pour faire la commande de deux moteurs (moteur pas à pas et moteur courant continu). Rappelons que cette carte a comme rôle la réception des commandes venant du logiciel DELPHI7 et les transmettre au circuit de puissance, ainsi que la transmission des informations c.-à-d. l'angle et pas de moteur et le sens de rotation du moteur.

2.3.2. Schéma réelle de la plaque :**3. Conclusion :**

La réalisation pratique de ce système nous a permis de renforcer nos connaissances théoriques après avoir pu faire fonctionner la carte correctement, aussi, elle nous a enrichi en d'autres connaissances sur ses deux parties : matérielle et logicielle. Nous avons pu maîtriser quelques logiciels qui sont très importants en électronique tels que le DELPHI, ISIS, ARES (proteus).

Conclusion générale :

Dans ce projet on a entamé la réalisation d'une interface USB 8 sorties. Le cahier de charge a été accompli en proposant et en implémentant deux applications pratiques et fonctionnelles (commande d'un moteur pas à pas et commande d'un moteurs à courant continu). Nous avons investis plusieurs domaines au cours de ce projet :

- ✓ Le domaine de la micro-informatique qui consiste à la conception des circuits électroniques à base de microcontrôleur PIC et leur interface PC.
- ✓ Le domaine de moteurs pas à pas
- ✓ Le domaine de moteurs à courant continu

La réalisation pratique de ce projet nous a permis de manipuler deux parties complémentaires matérielle et logicielle. La partie matérielle nous a servis à approfondir nos connaissances pratiques dans le domaine de l'électronique. Quant à la partie logicielle elle nous a aidé à renforcer nos capacités de programmation, telle la réalisation des programmes des microcontrôleurs et la réalisation des interfaces Pc avec le langage DELPHI7. Ceci nous a donné une connaissance approfondie et nécessaire à notre spécialité malgré le problème cités précédemment.

L'application choisie a été limitée pour ne pas s'engager dans des réalisations compliquées vue la contrainte du temps.

Ce projet nous a parmi d'entreprendre une tâche complexe (à plusieurs sous taches), de concevoir ses parties, et de faire aboutir le tout.

Ceci nous a poussé de revoir et bien assembler ce que nous avons appris lors de notre formation, et d'en rajouter ce qui concerne le projet.

Ce projet met en jeu plusieurs aspects de la construction électronique à savoir :

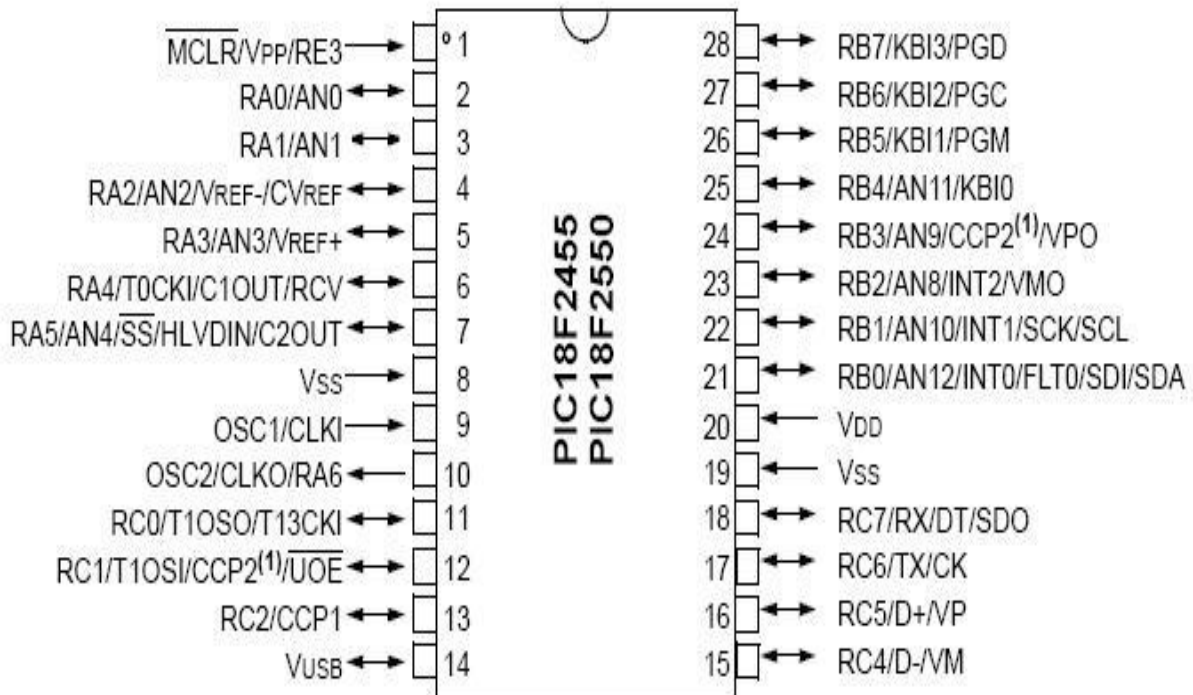
La programmation en DELPHI7, les familiarisations avec le PIC, le montage d'une carte autour du PIC, la programmation ...

Notre satisfaction est d'avoir pu réaliser le dialogue entre PC et carte.

Toutefois, il nous parait utile de relever les différentes contraintes qui nous ont gêné pour atteindre nos objectifs parmi lesquelles on peut citer :

- La durée consacrée pour le projet
- Manque de KIT spécifique pour charger le PIC 18F2550

Enfin, nous espérons que ce travail sera utile et bénéfique pour le lecteur qui trouvera dans ce mémoire l'essentiel de ce qu'il faut savoir sur la commande d'une carte basée sur le PIC 18F2550 et le protocole de communication USB.

ANNEXE**Annexe 1 : Brochage du PIC18F4550 :****Annexe 2: Spécificité du PIC18F2550 :**

MICROCHIP PIC18F2455/2550/4455/4550

**28/40/44-Pin, High-Performance, Enhanced Flash,
USB Microcontrollers with nanoWatt Technology**

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 End points (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB

-
- On-Chip USB Transceiver with On-Chip Voltage Regulator
 - Interface for Off-Chip USB Transceiver
 - Streaming Parallèle Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μ A typical
- Sleep mode currents down to 0.1 μ A typical
- Timer1 Oscillator: 1.1 μ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)

-
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
 - Compare is 16-bit, max. resolution 83.3 ns (TCY)
 - PWM output: PWM resolution is 1 to 10-bit
 - Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
 - Enhanced USART module:
 - LIN bus support
 - Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I2C™ Master and Slave modes
 - 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
 - Dual Analog Comparators with Input Multiplexing

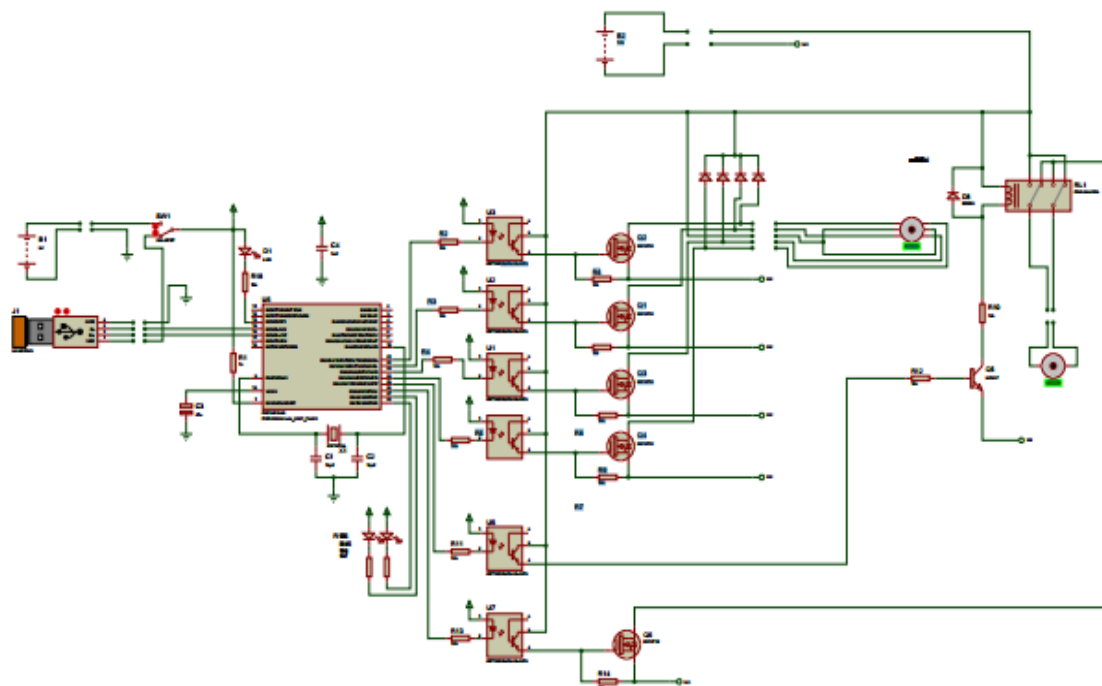
Special Microcontroller Features:

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins

- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

| Device | Program Memory | | Data Memory | | I/O | 10-Bit A/D (ch) | CCP/ECCP (PWM) | SPP | MSSP | | EAUSART | Comparators | Timers 8/16-Bit |
|------------|----------------|----------------------------|--------------|----------------|-----|-----------------|----------------|-----|------|--------------------------|---------|-------------|-----------------|
| | Flash (bytes) | # Single-Word Instructions | SRAM (bytes) | EEPROM (bytes) | | | | | SPI | Master I ² C™ | | | |
| PIC18F2455 | 24K | 12288 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F2550 | 32K | 16384 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F4455 | 24K | 12288 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |
| PIC18F4550 | 32K | 16384 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |

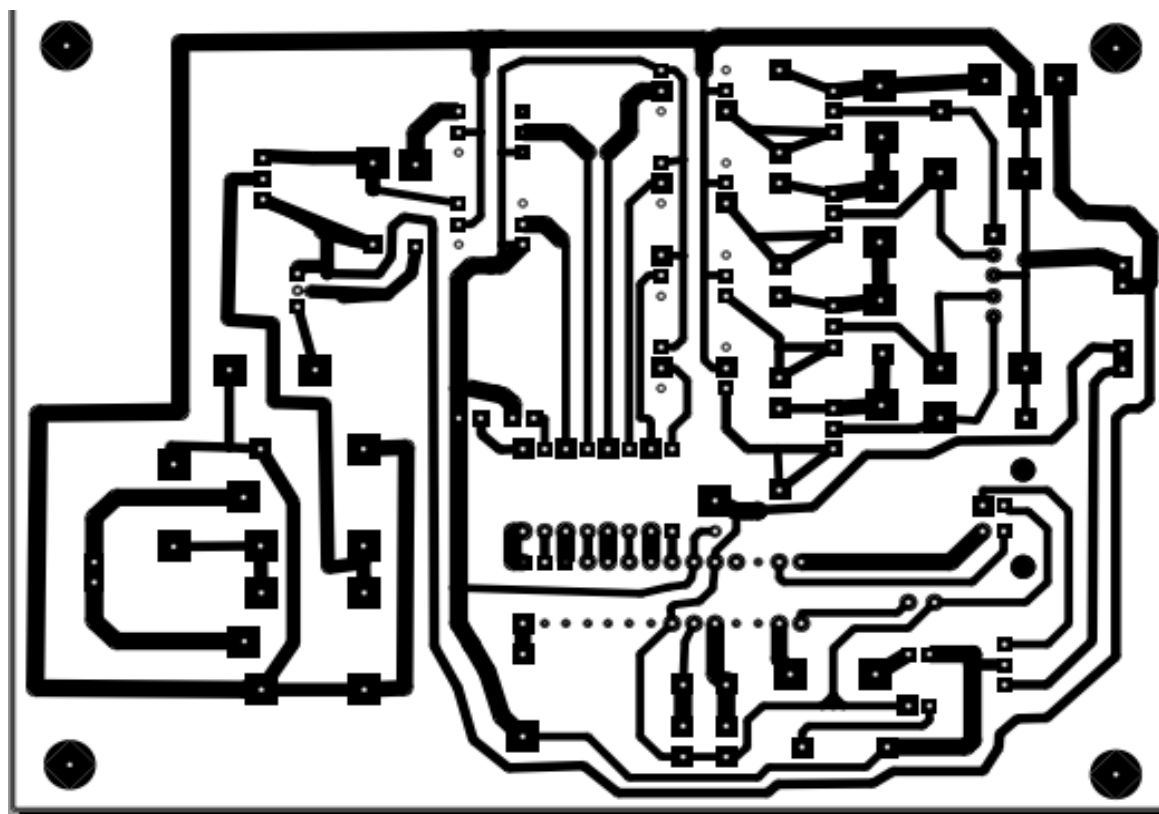
Annexe 3 : Schéma électronique de la carte d'interface :



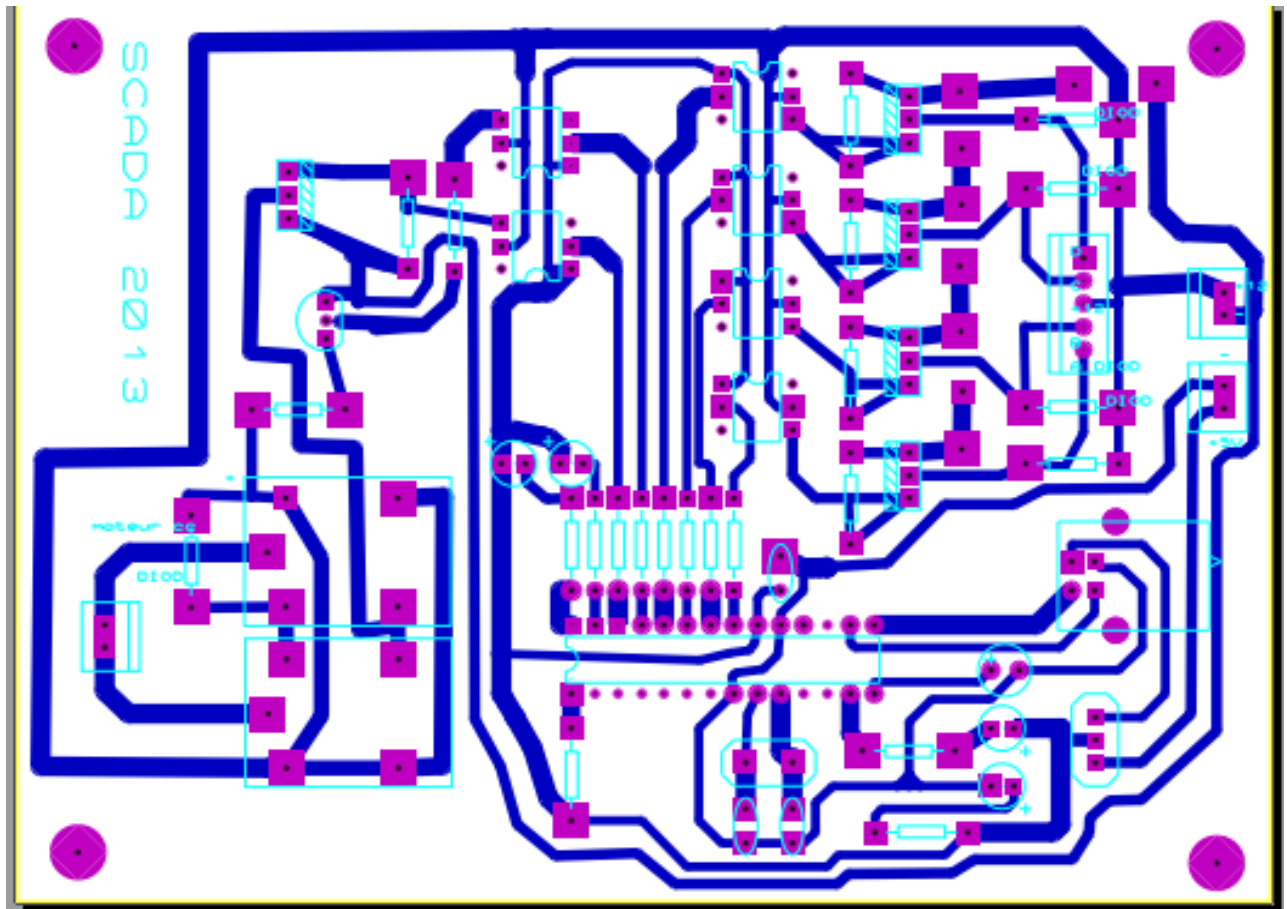
Annexe 4 : Liste des composants de la carte d'interface :

| Type | quantité | référence | Valeur |
|------------------|----------|------------------------|---------------------------|
| Résistance | 7 | R1,R2,R3,R4,R5,R11,R13 | 470Ω |
| | 1 | R15,R16,R1,R6,R7,R8,R9 | 1KΩ |
| | 1 | R14 | 1800Ω |
| | | R12 | 2200Ω |
| Diode | 5 | D2,D3,D4,D5,D8 | |
| Semi-conducteurs | 1 | V6 | Pic2550 |
| | 4 | D1,D6,D7,D9 | Led |
| Condensateurs | 2 | C1,C2 | 27pF |
| | | C4 | 1nF |
| Autres | 6 | U1,U2,U3,U4,U5,U7 | OPTOCOUPLER-NPN |
| | 4 | Q1,Q2,Q3,Q4 | IRF540(transistor mosfet) |
| | 1 | J1 | USBCONN |
| | 1 | R11 | OMI-SH-205L |
| | 1 | SW1 | SW-SPDT |

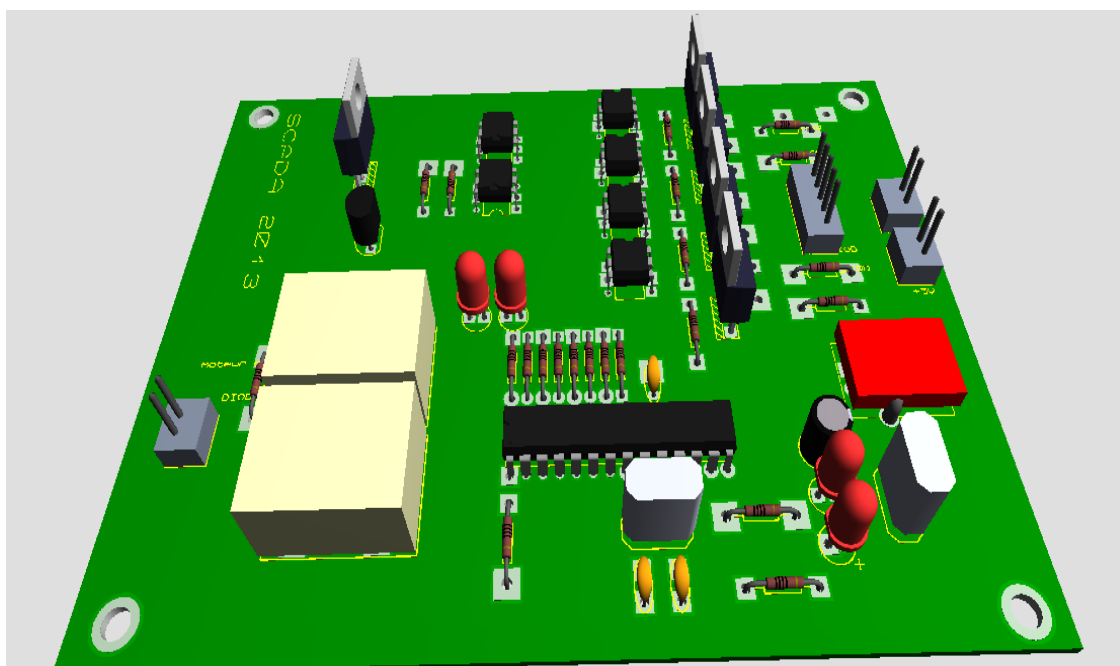
Annexe 5 : Tracés des circuits imprimés et Implantation des composants :



Implantation des composantes :



Annexe 6 : Schéma électronique de la carte d'interface 3D



BIBLIOGRAPHIE

- [1]: Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips, “USB spécification”, 2000, 662p**
- [2]: Dogan Ibrahim,” Advanced PIC Microcontroller Projects in C”, 2009, 539p**
- [3]: Don Anderson, “USB System Architecture (USB 2.0)”, 2002, 506p**
- [4]: Jan Axelson, “USB COMPLETE”, Troisième édition, 2005, 572p**
- [5]: Labcenter, “ISIS Help”, 2010**
- [6]: Labcenter, “ARES Help”, 2010**
- [7]: Microchip Technology, “MPLAB C18 C Compiler Getting Started”, 2004, 124p**
- [8]: Microchip Technology,” MPLAB C18 Users Guide”, 2004, 128p**
- [9]: Microchip Technology, “PIC18F2455/2550/4455/4550 Data Sheet”, 2004, 428p**
- [10]: Microsoft Corporation, “MSDN Library”, 1995-2000**
- [11]: Microsoft Corporation, “MSDN 2008 Library [WDK]”, 2009**
- [12]: Microsoft Corporation, “WDK Documentation”, 2009**
- [13]: Walter Oney, “Programming the Microsoft Windows Driver Model 2nd Edition”, 2003, 448p**

WEBOGRAPHIE

[14] : <http://www.abcelectronique.com/bigonoff>

[15] : <http://www.beyondlogic.org>

[16] : http://www.codeproject.com/KB/system/WDM_Driver_development

[17] : <http://www6.conestogac.on.ca/%7Eset/courses/year3/drivers>

[18] : <http://ww.datasheetcatalog.com>

[19] : http://www.intel-u-press.com/usb_dbe

[20] : <http://www.Lvr.com>

[21] : <http://membres.lycos.fr/grandzebu/electronique/usb>

[22]: <http://www.microchip.com>

[23] : <http://www.microchip.com/downloads/en/Appnotes>

[24]: <http://www.microchip.com/downloads/en/devicedoc>

[25]: <http://www.microsoft.com/whdc/ddk>

[26]: <http://msdn.microsoft.com>

[27]: <http://www.newsnespress.com>

[28]: <http://www.osr.com>

[29]: <http://pic18fusb.online.fr>

[30]: http://www.pulsewan.com/data101/usb_basics.htm

[31]: <http://www.usb.org>

[32]: <http://ww.usb-by-example.com>

[33]: <http://www.usbman.com>

[34]: <http://www.usb.org/developers/>

[35]: http://u.s.b.free.fr/pdf/L_USB_et_sa_norme_v1.pdf

[36]: http://www.wikipedia.org/wiki/usb_cours.html