

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ IBN KHALDOUN DE TIARET.



FACULTÉ DES SCIENCES APPLIQUÉES
DÉPARTEMENT de GÉNIE ÉLECTRIQUE



Mémoire de fin d'études en vue de l'obtention du diplôme de Master

Spécialité : Génie Electrique

Option : Informatique Industrielle

Présenté par :

BAHRI Abdelkrim

GHAMAM HAMED Abdel Fattah

Sujet du mémoire

**Prototypage et mise à jour de la plateforme
ABITEC de formation des microcontrôleurs**

Soutenu publiquement devant le jury composé de :

Mr : NASRI Djilali

Président

Mr : BELARBI Moustapha

Rapporteur

Mr : SEBAA Morsli

Examineur

PROMOTION : 2015

Remerciements

Avant toute chose, nous remercions « ALLAH » le tout puissant pour la bonne santé, la volonté et la patience qu'il nous a donné tout le long de nos études.

Au terme de ce travail, nous tenons à présenter nos remerciements les plus sincères à Monsieur M.BELARBI qui a accepté de nous encadrer et qui nous a fait profiter de ses larges connaissances et ses précieux conseils au cours de notre projet de fin d'étude.

Toutefois, il faut souligner que ce travail n'aurait pu voir le jour sans l'incalculable connaissance et savoir-faire acquis dans notre honorable faculté « la Faculté des sciences de l'ingénierie de Tiaret ».

C'est donc avec une immense fierté que nous adressons nos remerciements les plus distingués à tous nos professeurs.

Qu'ils trouvent, ici, ainsi que toute personne qui a contribué à l'achèvement de ce projet, directement ou indirectement, l'expression de notre sincère gratitude.

Enfin nous exprimons nos remerciements les plus dévoués aux membres du jury qui nous ont honorés en évaluant notre travail.

Dédicaces

Nous dédions ce modeste travail à :

Nos familles.

Tous nos amis et nos camarades.

*Ainsi qu'à toutes les personnes qui nous sont
chères.*

Table des matières

| | |
|--|----|
| Liste des figures : | 7 |
| Liste des abréviations | 9 |
| Introduction générale | X |
| Chapitre I..... | 12 |
| Généralité sur les microcontrôleurs | 12 |
| I.1- Introduction | 13 |
| I.2- Les microcontrôleurs..... | 13 |
| I.2.1- Les avantages des microcontrôleurs : | 14 |
| I.2.2- Identification des PIC : | 15 |
| I.2.3- Le contenu d'un microcontrôleur : | 15 |
| I.2.4- Les microcontrôleurs de la firme micro chip : | 16 |
| I.3- Le choix d'un PIC..... | 17 |
| I.4- Les caractéristiques générales de PIC18F4550..... | 18 |
| I.4.1- Les horloges système du PIC18F4550..... | 19 |
| I.4.2- Le Reset..... | 20 |
| I.4.3- Les Ports d'entrées / sorties | 21 |
| I.4.3.1- Généralités sur les ports | 21 |
| I.4.3.2- Les principaux registres liés aux ports | 21 |
| I.4.4- Architecture interne du PIC18F4550 | 22 |
| I.4.4.1- Organisation de la mémoire | 23 |
| I.4.4.2- Les timers..... | 24 |
| I.4.4.3- Unité arithmétique et logique UAL et registre de travail W..... | 25 |
| I.4.4.4- Les convertisseurs analogiques / numériques | 25 |
| I.5- Conclusion | 26 |
| Chapitre II Les liaisons RS-232 et USB | 27 |
| II.1-Introduction | 28 |
| II.2-RS-232..... | 28 |
| II.2.1-Historique | 28 |
| II.2.2-Description..... | 29 |
| II.2.3-Utilisation | 29 |
| II.2.3.1-Haute disponibilité | 30 |
| II.2.4-Spécification | 30 |
| II.2.4.1-Portée du standard..... | 30 |
| II.2.4.2-Mécanique | 31 |
| II.2.4.3-Protocole..... | 33 |
| II.2.4.4-Électrique | 34 |
| II.2.5-Limites | 34 |
| II.3-MAX232..... | 34 |
| II.3.1-Présentation des convertisseurs rs232 max232 | 34 |
| II.3.2-Description des MAX232 et MAX 232A | 35 |

| | |
|---|----|
| II.3.3-Description du MAX232E | 36 |
| II.4-Universal Serial Bus..... | 37 |
| II.4.1-Évolution de la norme USB | 37 |
| II.4.1.1-USB 1.0..... | 38 |
| II.4.1.2-USB 2.0..... | 38 |
| II.4.1.3-USB 3.0..... | 39 |
| II.4.1.4-USB 3.1..... | 39 |
| II.4.1.5-Résumé des débits | 39 |
| II.4.2-Évolution des connecteurs USB..... | 40 |
| II.4.2.1-Les premiers connecteurs : les types A et B..... | 41 |
| II.4.2.2-Les mini-connecteurs | 42 |
| II.4.2.3-Les micro-connecteurs | 42 |
| II.4.2.4-La nouvelle norme : le type C | 42 |
| II.4.3-Applications de l'USB..... | 43 |
| II.4.3.1-Pour le transfert des données..... | 43 |
| II.4.3.2-Pour l'alimentation électrique..... | 44 |
| II.4.4-Spécifications techniques..... | 45 |
| II.4.4.1-Caractéristiques générales | 45 |
| II.4.4.2-Protocole..... | 46 |
| II.4.4.3Connexion à chaud et Plug and Play : processus d'énumération | 47 |
| II.4.4.4-Alimentation électrique..... | 48 |
| Chapitre III..... | 49 |
| Conceptions et réalisation..... | 49 |
| III.1- Introduction..... | 50 |
| III.2- Présentation des logiciels utilisés pour la conception :..... | 50 |
| III.2.1- Proteus | 50 |
| III.2.2- ISIS..... | 50 |
| III.2.3- ARES..... | 51 |
| III.2.4- Les avantage de Proteus : | 51 |
| III.3- Le logiciel pilotant le programmeur de PIC | 51 |
| III.3.1- Expliqué la fenêtre | 52 |
| III.3.1.1- Read Details. « Lire les détails »..... | 53 |
| III.3.1.2- Configuration..... | 53 |
| III.3.1.3- Programming Details « détails de la programmation » | 54 |
| III.3.1.4- The Menus « Les menus »..... | 54 |
| III.4- Développements et système de formation ABITEC « DEVELOPMENT AND TRAINING SYSTEM » | 62 |
| III.4.1- Définition : | 62 |
| III.4.2- Le circuit | 63 |
| III.4.3- Les principales composantes..... | 65 |
| III.4.4- Le principe de fonctionnement | 65 |
| III.4.5- Les avantages de la Platform ABITEC :..... | 65 |
| III.4.6- Les inconvénients :..... | 65 |

| | |
|---|----|
| III.4.7- Alimentation de la carte..... | 66 |
| III.4.8- La carte d'application..... | 66 |
| III.4.8.1- Les applications | 67 |
| III.4.8.2- Les connecteurs..... | 67 |
| III.4.9- La communication entre PC et la carte | 67 |
| III.4.9.1- Le problème..... | 68 |
| III.4.10- Réalisation de la carte de communication entre le PC et PIC16F877 | 68 |
| III.4.10.1- L'idée de la carte..... | 68 |
| III.4.10.2- Modèle d'interrupteur | 69 |
| III.4.10.3- Simulation de circuit..... | 70 |
| III.4.10.4- Schema final..... | 72 |
| III.4.10.5- Le circuit imprime | 72 |
| III.4.10.6- Schéma de circuit sous ARES | 73 |
| III.4.10.7- Le typons de la carte..... | 73 |
| III.4.11- Réalisation sur carte d'essai | 74 |
| III.4.12- Comment réaliser..... | 75 |
| III.4.12.1- L'impression du typon de la carte | 75 |
| III.4.12.2- L'insolation sous lumière ultra-violet :..... | 75 |
| III.4.12.3- Développement de la couche photosensible..... | 76 |
| III.4.12.4- La gravure | 77 |
| III.4.12.5- Décapant | 77 |
| III.4.13- Les résultats | 78 |
| III.4.14- La deuxième carte | 78 |
| III.4.14.1- Principe..... | 78 |
| III.4.14.2- Schéma | 80 |
| Pour minimiser le nombre des interrupteurs on va utiliser les switch de deux position | 80 |
| III.4.14.3- Typon de la carte | 80 |
| III.4.14.4- Schéma de circuit sous ARES | 81 |
| III.4.14.5- La carte finale..... | 81 |
| III.4.15- Les avantage..... | 81 |
| III.4.16- Les inconvénients | 82 |
| III.4.17- Conclusion | 86 |
| Conclusion générale..... | 92 |
| Bibliographie..... | 93 |
| Annexe | 94 |

Liste des figures :

| | |
|--|----|
| Figure 1 : Les PICs microcontrôleurs..... | 14 |
| Figure 2 : Architecture de harvard et von neumann..... | 14 |
| Figure 3 : Brochage du PIC18F4550..... | 18 |
| Figure 4 : Oscillateur à Quartz/Céramique [4]..... | 20 |
| Figure 5 : Architecture interne du PIC18F4550..... | 23 |
| Figure 6 : Synoptique général du CAN pour le PIC18F4550..... | 26 |
| Figure 7 : Prise femelle extrémité de câble type DE-9..... | 29 |
| Figure 8 : Un connecteur DE-9 (Improprement appelé DB-9) mâle utilisé comme port série sur un ordinateur personnel..... | 29 |
| Figure 9 : Oscillogramme de la transmission du caractère K (01001011), avec un bit de départ et un bit d'arrêt..... | 33 |
| Figure 10: MAX232 - MAX232A - MAX232E..... | 34 |
| Figure 11 : le MAX232 et MAX232A..... | 35 |
| Figure 12 : Le MAX232E..... | 36 |
| Figure 13 : Connecteur USB de type A mâle..... | 37 |
| Figure 14 : Symbole de l'USB..... | 37 |
| Figure 15 : Différents connecteurs de type USB 1 et 2..... | 40 |
| Figure 16 : Les différents types de connecteurs USB 1 et 2..... | 41 |
| Figure 17 : Fiche micro-B USB 3..... | 41 |
| Figure 18 : Ventilateurs alimentés par USB..... | 44 |
| Figure 19 : Petit accessoire affichant et enregistrant la tension (V), le courant (A) et la charge (mAh) au cours de l'alimentation d'un appareil par USB, telle que la recharge d'un téléphone..... | 44 |
| Figure 20 : Prises USB de type A et B, vue de face. USB 1 et 2..... | 48 |
| Figure 21 : PIC877DATS..... | 51 |
| Figure 22 : la fenêtre de PIC877DATS..... | 52 |
| Figure 23: Read Details. « Lire les détails »..... | 53 |
| Figure 24 : Configuration..... | 53 |
| Figure 25 : Programming Details « détails de la programmation »..... | 54 |
| Figure 26 : The Menu « Les menus »..... | 54 |
| Figure 27 : le menu charge..... | 55 |
| Figure 28 : le menu program..... | 55 |
| Figure 29 : le menu read « Lire »..... | 56 |
| Figure 30 : le menu de communications..... | 57 |
| Figure 31 : Control Switches..... | 58 |
| Figure 32 : La section principale de débogage..... | 59 |
| Figure 33 : le menu Window « Fenêtre »..... | 59 |
| Figure 34 : La fenêtre Special Function Register (SFR)..... | 60 |
| Figure 35: La fenêtre Source..... | 61 |
| Figure 36 : La fenêtre Montres..... | 62 |
| Figure 37 : Développements et système de formation ABITEC..... | 63 |
| Figure 38 : Le circuit de Développements et système de formation ABITEC / P01..... | 63 |
| Figure 39 : Le circuit de Développements et système de formation ABITEC / P02..... | 64 |
| Figure 40 : les noms de chaque composant dans la carte de ABITEC..... | 64 |
| Figure 41 : circuit d'alimentation de la carte..... | 66 |
| Figure 42 : Le bloc utilisé pour alimente la carte..... | 66 |
| Figure 43 : La carte d'application..... | 66 |
| Figure 44 : un connecteur IDC 40..... | 67 |

| | |
|---|----|
| Figure 45 : La communication entre PC et la carte | 68 |
| Figure 46 : Modèle d'interrupteur | 69 |
| Figure 47 : schéma de Simulation de circuit | 70 |
| Figure 48 : résultat de simulation | 71 |
| Figure 49 : résultat de simulation | 71 |
| Figure 50 : Schéma final de circuit | 72 |
| Figure 51 : Schéma de circuit sous ARES | 73 |
| Figure 52 : typon top copper de la carte | 73 |
| Figure 53 : typon bottom copper de la carte | 74 |
| Figure 54 : Réalisation sur carte d'essai | 74 |
| Figure 55 : Vu de dessus du typon sur la feuille acétates | 75 |
| Figure 56 : machine d'insolation..... | 76 |
| Figure 57 : La carte dans le révélateur | 77 |
| Figure 58 : Vue de dessous de la carte après le décapant | 78 |
| Figure 59 : schéma de Principe de la carte | 79 |
| Figure 60 : Schéma de circuit de communication manuel..... | 80 |
| Figure 61 : Top et bottom copper | 80 |
| Figure 62 : Schéma de circuit sous ARES | 81 |
| Figure 63 : La carte finale..... | 81 |

Liste des abréviations

PIC : Programmable Interface Controller.

RISC : Reduced Instructions Set Computer

EPROM: Erasable Programmable Read-Only Memory

EEPROM : Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et programmable

RAM : Random Access Memory ou mémoire à accès direct

RTC: Real time clock ou horloge temps réel

I2C: Inter Integrated Circuit

VDD: Voltage Drain Drain

VSS: Voltage Source Source

GND: The Ground ou la mass

MCLR: Master Clear

Ko : kilo-octets

ADC: Analog to Digital Converter

CAN : convertisseur analogique / numérique

SPI : Serial Peripheral Interface Bus

V_{reff} : tension de référence

LDR : light-dependent resistor ou cellule photoconductrice

LCD : Liquid Crystal Display ou afficheur à cristaux liquide

Introduction générale

L'évolution des besoins de l'homme pour approvisionner son environnement, d'améliorer son confort et sa qualité de vie ont mené ce dernier à rechercher des solutions en fonction des données scientifiques et technologiques. Des nouvelles exigences économiques ont incité des industriels à rechercher des moyens de satisfaire la société moderne. un état des lieux qui a donné naissance à la concurrence industrielle pour acquérir la plus grande part de de marche pour parvenir à devancer son concurrent avec un produit nouveau ou améliorer tout en réduisant le cout de reviens. Les industriels cherchent toujours de nouveau outils et moyens.

Cette course industrielle et scientifique a permis de pousser plus ce progrès technologique et électronique et en informatique et rendre possible la retro ingénierie des nouvelles technologies et cette technique s'appelle « reverse engineering ».

En général, il s'agit de décompiler ou de désassembler un programme, c'est-à-dire de prendre quelque chose de compréhensible par une machine mais pas par un être humain et d'en faire quelque chose de lisible pour un homme mais plus pour un ordinateur.

Dans la pratique, le reverse engineering consiste à analyser le fonctionnement d'un programme ou d'un périphérique « typiquement en analysant la communication qu'il a avec le reste du système », ou alors en examinant pas à pas les résultats de son exécution.

La liaison série asynchrone est couramment utilisé en informatique et en automatique (de manière plus générale en pilotage de procédés) pour traiter tout problème de transmission. Pour transporter l'information, on utilise la tension (RS232, RS422-liaison multipoint, RS485-liaison multipoint bidirectionnelle) ou le courant (boucle de courant 20mA) selon la norme EIA 232 (Electronic Industry Associate) depuis 1969 puis réévaluée en 1991.

La liaison RS232 est simple, universelle, parfaitement connue et supportée par un grand nombre de périphériques. Contrairement à la liaison parallèle, elle autorise de grandes distances mais possède des réglages plus compliqués. Ceci justifie une étude détaillée.

Les ports série / parallèle, avec leurs caractéristiques désuètes, voient leur utilité décroître au profit de récentes interfaces comme l'USB (Universal Serial Bus), le FireWire (domaine grand public) ou d'autres plus anciennes mais qui ne cessent

d'être améliorée comme le SCSI (domaine professionnel). Nous allons étudier l'interface USB qui représente une nouvelle liaison série synchrone.

Les PICs sont des composants RISC (Reduce Instructions Construction Set), ou encore composant à jeu d'instructions réduit. 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 microcontrôleurs sont plus utilisés dans l'industrie fait de lui très important dans la communication entre machines, un microcontrôleur programmable, ce qui rend nécessaire la fabrication programmeur de microcontrôleur.

Ce travail a été structuré en trois chapitres dans le présent mémoire :

Le premier chapitre présente l'étude des différents types des microcontrôleurs et le PIC18F4550 spécialement.

Le deuxième chapitre illustre l'étude de deux types de liaisons RS-232 et USB.

Le troisième chapitre présente le logiciel de plateforme ABITEC et réalisation de carte pour éviter la problème de communication entre la carte et PC.

Chapitre I

Généralité sur les microcontrôleurs

I.1- Introduction

Les microcontrôleurs sont très utilisés dans le monde de l'industrie, notamment dans les systèmes embarqués. On pourra donc les retrouver dans l'aéronautique, l'aérospatial, l'automobile, l'électronique grand public. Leur polyvalence, et leur taille les rendent intéressants pour des modules de traitement de données numériques et aussi analogiques. Ils sont certes peu puissants comparés à des processeurs dédiés, mais ils les compensent par leur prix mais surtout leur taille : un microcontrôleur peut être comparé à une carte mère d'ordinateur.

Dans ce chapitre nous présentons une description détaillée sur microcontrôleur (PICs de Microchip).

I.2- Les microcontrôleurs

Un microcontrôleur se présente comme étant 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, temporisateurs, convertisseurs, ... [1]

Chaque fabricant a ses propres familles de microcontrôleurs. Une famille se caractérise par un noyau commun (le microprocesseur, le jeu d'instruction, ...). Ainsi les fabricants peuvent présenter un grand nombre de pins qui s'adaptent plus au moins à certaines tâches. Mais un programmeur connaissant une famille n'a pas besoin d'apprendre à utiliser chaque membre, il lui faut connaître juste ces différences par rapport au père de la famille. Ces différences sont souvent, la taille des mémoires, la présence ou l'absence des périphériques et leurs nombres. [1]

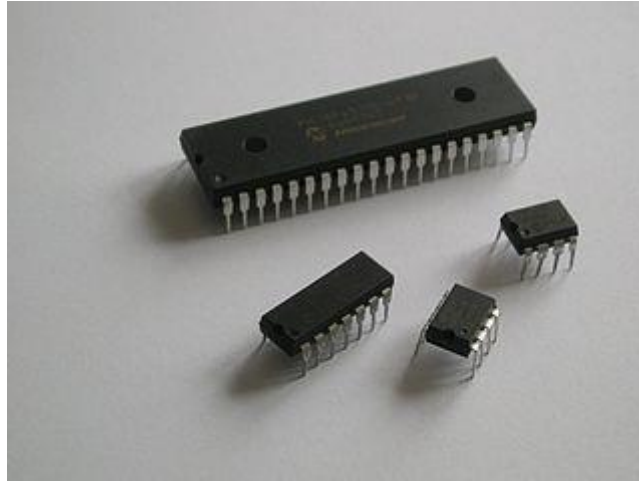


Figure 1 : Les PICs microcontrôleurs

Les microcontrôleurs PIC possèdent une architecture Harvard. À l'instar de l'architecture Von Neumann, qui utilise un seul bus pour l'échange des données sur une mémoire unique, l'architecture Harvard utilise deux bus de communication, sur deux mémoires distinctes. Soit une mémoire pour le programme et une mémoire pour les données. Voyez la différence sur la figure2.

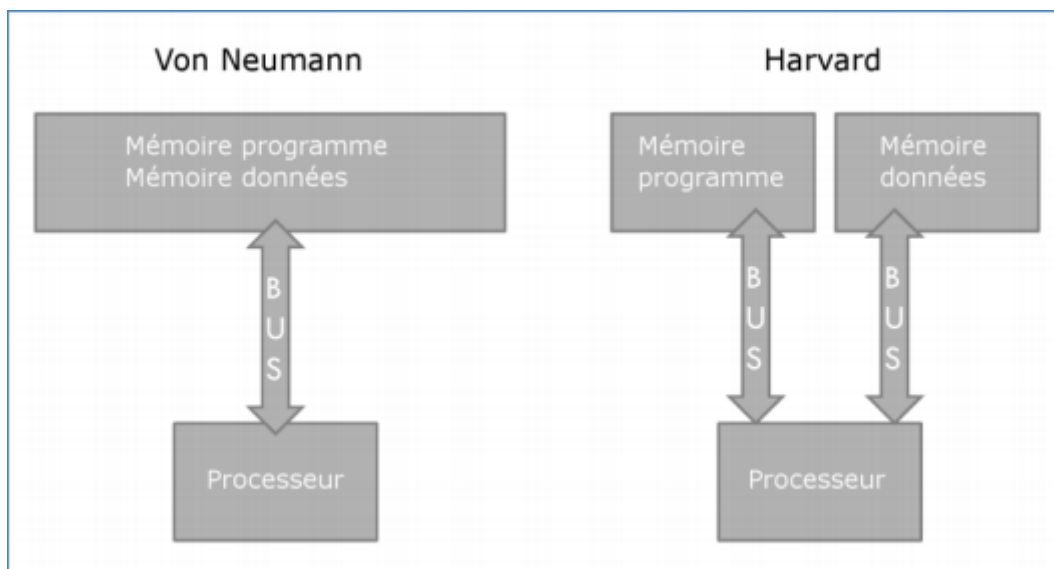


Figure 2 : Architecture de harvard et von neumann

I.2.1- Les avantages des microcontrôleurs :

L'utilisation des microcontrôleurs pour les circuits programmables à plusieurs points forts et bien réels .il suffit pour s'en persuader, d'examine la spectaculaire évolution de l'offre des fabricants de circuits intègres en ce domaine depuis quelques années. [1]

Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement de matériel et de circuits imprimé. [1]

Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de données d'un composant à un autre. [1]

Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux ; non seulement qu'il diminue les coûts de la main d'œuvre, il est moins cher que les autres composants qu'il les remplace. En plus, il a la possibilité de réaliser des applications qu'on n'arrive pas à réaliser avec d'autres composants.

I.2.2- Identification des PIC :

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 mémoire programme

C : EPROM ou EEPROM

CR : PROM

F : Flash

yy : Identificateur.

zz : vitesse maximale du quartz de pilotage.

I.2.3- Le contenu d'un microcontrôleur :

Un circuit microcontrôleur doit contenir dans un seul boîtier tous les éléments de bases, nous y retrouvons bien évidemment l'unité centrale qui est plus simplifiée par rapport à celle de microprocesseur. En contrepartie, de instructions de manipulation de bits, très utile pour faire des entrées/sorties lui ont été ajoutées. Dans certains circuits, cette unité centrale se voit dotée d'un très grand nombre des registres internes qui servent alors de mémoire vive. [2]

Nous pouvons ensuite voir la mémoire morte mais ce n'est pas une obligation. En effet dans un certain nombre de boîtiers et jusqu'à ces dernières années, cette mémoire ne pouvait qu'être programmée par masque à la fabrication de circuit. Cela imposait donc l'utilisation potentielle du microcontrôleur pour commander un nombre de pièces identiques important. [2]

Un certain de microcontrôleurs étaient, et sont toujours d'ailleurs disponibles sans ROM. Puis, les technologies d'intégration progressent, les fabricants ont appris à placer sur la mémoire programmable électriquement et effaçable aux ultraviolets (UV PROM) qui coûtent relativement cher, non à cause de la mémoire elle-même dont la technologie est maintenant facile à produire, mais plutôt à cause de la fenêtre en quartz nécessaire à son effacement, on a vu également apparaître des microcontrôleurs dits OTPROM. [1]

Dans ces circuits ou OTPROM (ONE TIME PROM), c'est-à-dire programmable une fois, la mémoire UV PROM existe toujours dans ce programme donc comme n'importe quel circuit de ce type, mais, fait de l'absence de toute fenêtre, cette mémoire n'est ensuite plus effaçable. C'est une solution intéressante pour les productions en petite série, ne justifiant pas une programmation par masque. En effet, du fait de l'absence de fenêtre, ces versions OTPROM sont disponibles en boîtier plastique très peu coûteux. [1]

I.2.4- Les microcontrôleurs de la firme micro chip :

Un PIC n'est rien d'autre qu'un microcontrôleur développé par la firme MICROCHIP. 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 nécessiter l'ajout de composants externes. Il existe plusieurs centaines de modèles regroupés dans différentes séries (pic16, pic18,...) en fonction de leurs caractéristiques. [1]

Les PICs (programmable Interface Controller) sont des composants dits RISC (Reduce Instruction Set Computer), ou encore composant à jeu d'instruction réduit, sachant que plus on réduit le nombre d'instruction plus le décodage deviendra facile et rapide, ce qui rend le fonctionnement du composant plus vite. [3]

Dans le marché, il existe deux familles opposées, les RISC et les CISC (Complex Instruction Set Computer). Chez les CISC, on diminue la vitesse de traitement mais les

instructions sont plus complexes, plus puissantes, et donc plus nombreuses. Il s'agit donc d'un choix de stratégie. [1]

La famille des Pics est divisée à l'heure actuelle en trois grandes familles :

- La famille Base -Line, qui utilise des mots d'instructions de 12bits.
- La famille Mide -Line, qui utilise des mots de 14bits.
- La famille High-End, qui utilise les mots de 16bits.

Tous les Pics Mide -Range ont un jeu de 35 instructions, il stockent chaque instruction dans un seul mot de programme, et exécutent chaque instruction (sauf les sauts) en un cycle. On atteint donc de très grandes vitesses. L'exécution en un seul cycle est typique des composants RISC.

L'horloge du pic est pré-divisée par quatre. C'est cette base de temps qui donne le temps d'un cycle. Si on utilise par exemple un quartez de 4MHZ, on obtient donc 1000000 cycles/seconde. Comme le PIC exécute pratiquement l'instruction par cycle hormis les sauts, cela nous donne une puissance de l'ordre de 1MIPS (1 millions d'instructions par seconde). Alors que les pics peuvent monter jusqu'à 20MHZ.

I.3- Le choix d'un PIC

Le choix d'un PIC est directement lié à l'application envisagée:

Il faut dans un premier temps déterminer le nombre d'entrées/sorties nécessaires pour l'application. Ce nombre d'entrées/sorties nous donne une idée sur la famille du PIC. [10]

Il faut ensuite déterminer si l'application nécessite un convertisseur Analogique/Numérique ce qui va centrer un peu plus vers le choix du PIC. [10]

La rapidité d'exécution est un élément important, il faut consulter les DATA-BOOK pour vérifier la compatibilité entre la vitesse maximale du PIC choisi et la vitesse max nécessaire au montage. [10]

La taille de la RAM interne et la présence ou non d'une EEPROM pour mémoriser des données est également important pour l'application souhaitée. [10]

La longueur de programme de l'application détermine la taille de la mémoire programme du PIC recherché. [10]

Afin de choisir un PIC adéquat à notre projet, nous avons pensé à l'utilisation du PIC PIC18F4550. [10]

I.4- Les caractéristiques générales de PIC18F4550

Les caractéristiques principales du 18F4550 sont résumées comme suit :

- 35 lignes d'entrées/sorties, répartis comme suit :
 - Un port de 7 lignes (port A) - Un port de 8 lignes (port B) - Un port de 8 lignes (port C) - Un port de 8 lignes (port D) - Un port de 4 lignes (port E) [4]
- Alimentation sous 5 Volts
- Une mémoire de programme de type flash (32Ko (32768 Octets) mots de 16 bits)
- Une mémoire RAM utilisateur de 2048 Octets
- Une mémoire EEPROM de 256 Octets emplacements
- Une interface I2C pour la gestion d'un bus à 2 fils
- Facilité de programmation comme tous les PICs[4]

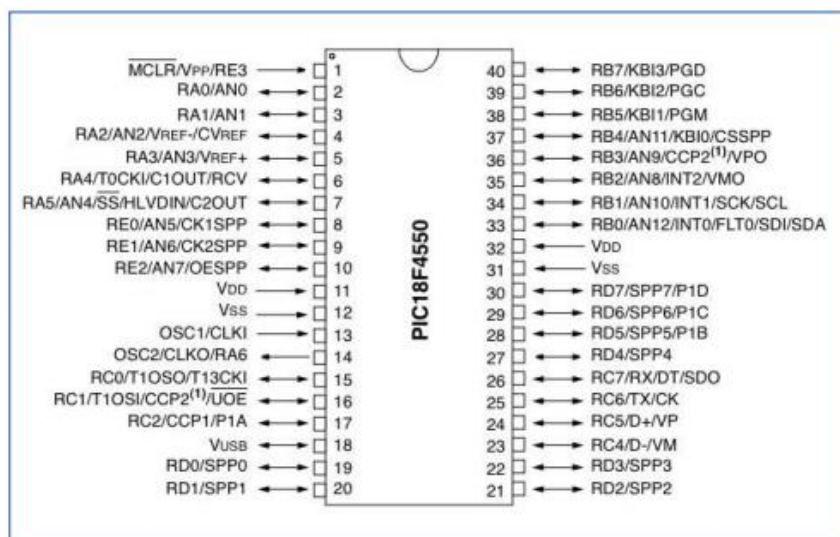


Figure 3 : Brochage du PIC18F4550

On peut distinguer sur ce schéma :

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

Tableau 1 : les caractéristiques générales de PIC18F4550.[4]

| La caractéristique | Valeur |
|-------------------------------|---|
| 48MHz | Mémoire programme FLASH |
| 32Ko (32. | Mémoire données (RAM) |
| 2048 Octets | Mémoire EEPROM |
| 256 Octets | Interruptions |
| 20 | Ports parallèles (Nombre des lignes) |
| A(7), B(8) C(8), D(8), E(4) | Timers |
| 4 (3 X1 6bit + 1X8bit) | CAN 10-bit |
| 13 | Instructions |
| 75 | Vitesse du CPU (MIPS) |
| 12 | Périphériques de communication numérique |
| 1-A/E/USART, 1-MSSP (SPI/I2C) | Capture / Comparateur / Périphériques PWM |
| 1 CCP, 1 ECCP | Comparateurs |
| 2 | USB (canaux, vitesse, respect) |
| 1, Full Speed, USB 2.0 | Plage de tension de fonctionnement (V) |
| de 2 à 5,5 | Température (C) |

I.4.1- Les horloges système du PIC18F4550

Comme la grande majorité des systèmes numériques à l'heure actuelle, les microcontrôleurs sont des composants synchrones, c'est-à-dire que toutes les opérations y sont synchronisées par une horloge commune. Dans le cas d'un microcontrôleur, le nombre d'instructions par seconde exécutées par le processeur est directement proportionnel à la fréquence de l'horloge. Ainsi il est logique de penser que l'idéal est d'utiliser l'horloge la plus élevée possible, car les performances seront ainsi maximisées. Ce raisonnement est juste, mais maladroit, car une autre chose qui est directement proportionnelle à la fréquence d'horloge est la consommation électrique du composant. [5]

Dans le PIC18F4550, en plus d'une horloge système, qui cadencera le coeur du microcontrôleur, on peut aussi avoir à fournir une horloge à certains périphériques tels que le contrôleur USB ou les Timers. [5]

Le PIC18F4550 peut utiliser 4 types d'horloge comme horloge système : - XT : Oscillateur à résonateur céramique ou à cristal (quartz) externe, jusqu'à 4 MHz. Il est nécessaire dans ce mode d'ajouter entre les pattes OSC1 et OS du composant un quartz, ainsi que deux condensateurs adaptés entre chacune de ces deux broches et la masse. - HS : Même fonctionnement que le précédent, mais pour des fréquences supérieures à 4MHz. Dans modèles d'oscillateurs HS, HSPLL, XT et XTPLL, un oscillateur en quartz ou en céramique est reliée à la OSC1 et OS repères pour établir oscillation. [5]

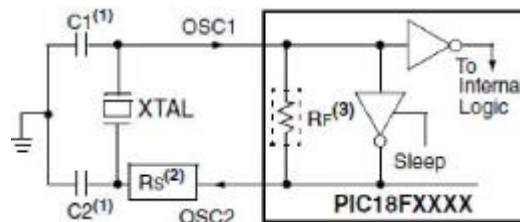


Figure 4 : Oscillateur à Quartz/Céramique [4]

- EC : Dans ce mode, il faut directement apporter à la broche OSC1 un signal d'horloge (carré) généré par un oscillateur totalement externe.

- INT : Le PIC18F4550 possède un oscillateur interne de fréquence réglable entre 31kHz et 8MHz.

Le choix de l'oscillateur se fait par des registres particuliers, les registres de configuration. Ces registres ont la particularité de ne pouvoir être écrits que lors de la programmation [5].

I.4.2- Le Reset

Le Reset d'un microcontrôleur peut être déclenché selon plusieurs événements. Ces événements peuvent être d'origine physique, tel que l'appui sur le bouton Reset du montage (broche MCLR) ou encore être dus à une baisse de tension d'alimentation. Un Reset peut également être déclenché suite à un état logiciel tel que le débordement (ou saturation) de la pile (PIC 18). Pour résumer, un Reset peut être déclenché par [5] :

- un appui sur le bouton Reset du montage (Broche MCLR = 0).
- une instruction logicielle Reset (PIC 18).
- une saturation ou un débordement inférieur de la pile (PIC 18).
- un dépassement de la temporisation du chien de garde (WDT).
- une atteinte du seuil mini de l'alimentation.

I.4.3- Les Ports d'entrées / sorties

I.4.3.1- Généralités sur les ports

La plupart des broches du PIC18F4550 sont accessibles en entrée et en sortie tout-ou-rien, c'est-à-dire qu'il peut en entrer ou en sortir un état haut logique (+VCC) ou un état bas logique (0V).

Les broches sont regroupées par ports, chaque port pouvant contenir jusqu'à 8 broches. Il est possible d'utiliser chaque patte indépendamment en entrée, en sortie ou pour d'autres fonctions. Pour localiser les broches correspondant aux différents ports, il faut se reporter au brochage du composant. Sur la plupart des broches, le premier élément de la description est de la forme Rxy, avec x une lettre, correspondant au port auquel appartient la broche, et y le numéro de la broche dans le port. Par exemple la broche 6 du composant correspond à RA4, c'est-à-dire au bit 4 du port A.

Cependant sur les broches 1 et 14 la description sous forme de port n'est pas en premier dans la liste, mais en dernier. C'est parce que ces broches ont des affectations spéciales prioritaires sur les ports. La broche 1 est par défaut la broche de RESET (MCLR), on peut l'utiliser comme RE3 en inhibant le RESET externe grâce à un bit de configuration (comme pour configurer l'horloge). La broche 14 peut être utilisée comme RA6 dans les modes d'horloge qui n'utilisent pas un résonateur externe [5].

I.4.3.2- Les principaux registres liés aux ports

Il y a trois registres que l'on utilise pour commander les ports : les registres TRISx , PORTx et LATx (avec x la lettre correspondant au port) [5].

Les registres TRISx permettent de configurer la direction (entrée ou sortie) de chaque broche du port. Un 1 sur le bit y de ce registre placera Rxy en entrée, un 0 le placera en sortie.

Pour le retenir c'est simple : 1 ressemble à la lettre « I » comme « Input », 0 ressemble à la lettre « O » comme « Output ». [5].

Par exemple pour placer RB2 en sortie, on placera le bit 2 de TRISB à 0.

Les registres PORTx permettent de placer la valeur de sortie ou de lire la valeur en entrée du port x. Lorsque la broche y du port x est placé en sortie, placer à 1 le bit y de PORTx placera Rxy à 1, idem avec 0. Lorsque la broche y du port x est placé en entrée, le bit y de PORTx est l'image de la broche Rxy. [5].

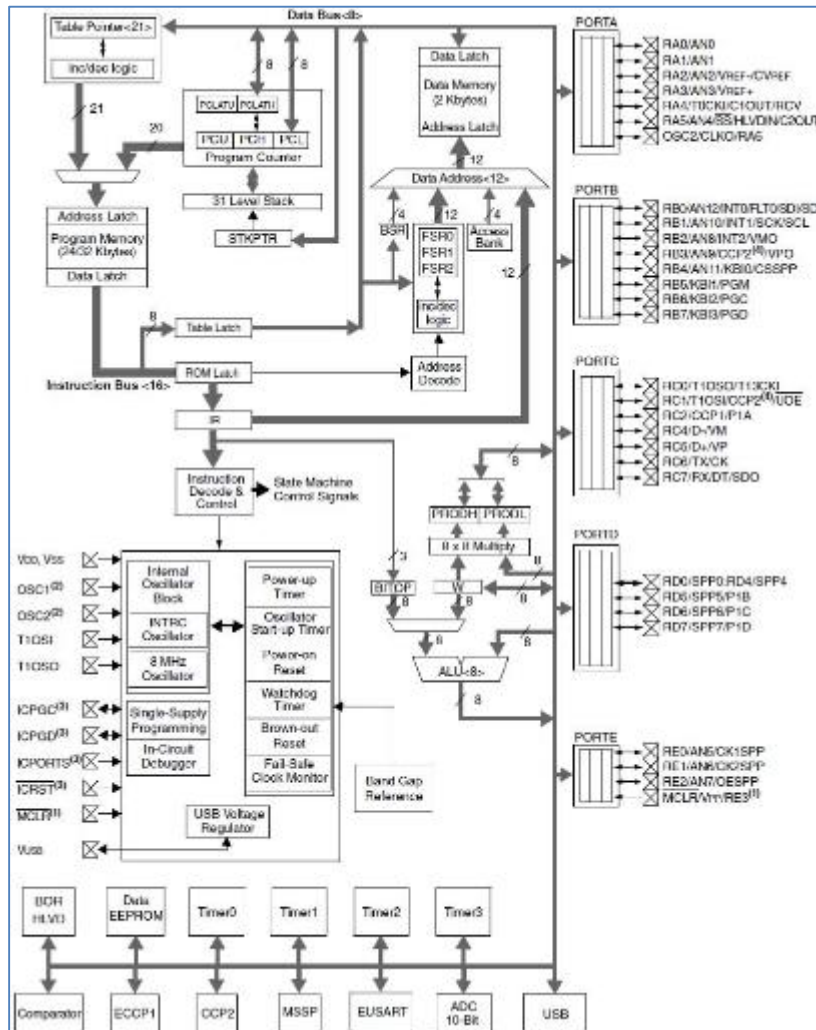
Le registre tampon de donnée en sortie appelé **LAT** pour le PIC.

Ces registres ont pour fonction d'éviter certains problèmes qui peuvent apparaître lorsque plusieurs opérations d'écriture/lecture des ports d'entrées/sorties sont effectuées successivement. [5].

I.4.4- Architecture interne du PIC18F4550

La figure (Fig 5), présente l'architecture interne du circuit. Il est constitué des éléments suivants [4] :

- Quatre ports d'entrées/sorties
- Une unité arithmétique et logique (ALU)
- Quatre compteurs (Timers) Timer0, Timer1, Timer2, Timer3
- Un compteur de programme (program counter)
- Une mémoire RAM, 2048 octets
- Une mémoire EEPROM de 256 octets de données
- Un registre contenant le code de l'instruction à exécuter (IR)
- Convertisseur analogique numérique 13 canaux 10bit



1.4.4.1- Organisation de la mémoire

Le microcontrôleur PIC18F4550 dispose de deux types de mémoires séparées [5] : - Une mémoire de programme de 32 ko (mémoire FLASH) [5].

- Une mémoire de données de 2048 octets

1.4.4.1.1- Mémoire de programme

La capacité de la mémoire de programme est de 32 ko (kilo-octets). Cependant, chaque instruction est codée sur 16 bits, ce qui signifie que chaque instruction occupera 2 octets.

Ainsi, la mémoire de programme ne pourra stocker que jusqu'à 16384 instructions au maximum [5].

1.4.4.1.2- Mémoire de données

Elle se décompose en deux parties une RAM et une zone EEPROM :

La mémoire RAM

La mémoire RAM (Random Access Memory) appelée mémoire vive, ce type de mémoire perd toutes ses informations stockées lorsqu'elle cesse d'être alimentée. La capacité totale de la mémoire de donnée est de 2048 octets. Elle est constituée de deux parties :

- Les registers SFR (Special Function Register)

Ce sont les registres de fonctionnement du PIC. L'ensemble de ces registres est souvent appelé fichier des registres.

- Les registres GPR (General Purpose Register)

Sont des positions mémoire que l'utilisateur peut utiliser pour stocker ses variables et ces données. On remarque donc que, indépendamment de leur nature, les positions de la RAM sont toujours appelé registres.

Le plan mémoire est découpé en 16 zones (bank) identiques de 256 octets. La sélection de la zone est réalisée par 4 bits du registre BSR (Bank Select Register).

Les registres internes pour les fonctions spéciales (registres généraux et configuration des périphériques ? SFR qui signifie Special Function Registers) se situent aux adresses hautes [5].

Mémoire EEPROM (Electrically Erasable Programmable Read Only Memory)

Le PIC possède une zone EEPROM de 256 octets accessibles en lecture et en écriture par le programme. On peut y sauvegarder des valeurs, qui seront conservées même si l'alimentation est éteinte, et les récupérer lors de la mise sous tension. Leur accès est spécifique et requiert l'utilisation de registres dédiés

1.4.4.2- Les timers

Le timer est un élément important dans un PIC. En effet, cette ressource permettra de réaliser de nombreuses fonctions, telle que la temporisation, la génération de signaux PWM (appelée également Modulation de Largeur d'Impulsion, MLI), ou encore les modes Compare ou Capture. Selon le type de PIC rencontré, on trouvera un certain nombre de timers

qui seront pour certains associés à des ressources internes du PIC. Un timer est en fait un compteur incrémentant au rythme d'une horloge, et ceci en ayant la possibilité de pré positionner une valeur de départ. Certains timers pourront fonctionner en mode 8 bits, 16 bits ou 32 bits. Dans le PIC18F4550 il existe quatre timers (un timer de 8 bit et le reste de 16 bits) [5].

I.4.4.3- Unité arithmétique et logique UAL et registre de travail W

L'UAL (ou ALU en anglais) sera chargée d'effectuer toutes les opérations logiques et arithmétiques de base, C'est une ALU à 8 bits qui réalise les opérations arithmétique et logique entre l'accumulateur W et n'importe quel autre registre 'F' ou constante K.

L'accumulateur W est un registre de travail 8 bits, il n'a pas d'adresse comme les autres SFR. Pour les instructions à deux opérandes, c'est toujours lui qui contient un des deux opérandes. Pour les instructions à un opérande, celui-ci peut être soit W soit n'importe quel registre F. Le résultat de l'opération peut être placé soit dans le registre de travail W soit dans le registre F [5].

I.4.4.4- Les convertisseurs analogiques / numériques

Pour utiliser un convertisseur analogique / numérique, il existe trois registres internes nommés ADCON0, ADCON1 et ADCON2, qui permettront de configurer et de sélectionner les différentes entrées analogiques (ADC pour *Analog to Digital Converter*) du PIC utilisé. Le paramétrage de ces trois registres conduit à choisir la façon dont se comporteront le ou les convertisseurs utilisés. Les entrées du PIC en relation avec le convertisseur analogique/ digital interne sont les broches AN0 à AN12. Les entrées analogiques (RA0 à RA5) transitent *via* le PORT A du PIC, les entrées AN5 à AN7 font partie du PORT E et les entrées AN8 à AN12 font partie du port B. [11]

Comme vous le voyez sur le schéma général, les bits CHS0 à CHS3 du registre ADCON0 permettent de sélectionner un canal (AN0 à AN12) qui sera en relation avec le convertisseur. Les bits VCFG0 et VCFG1 du registre ADCON1 permettent quant à eux de définir les références de tension (AN2 ou Vss pour Vref- et AN3 ou Vdd pour Vref +). Les bits PCFG0 à PCFG3 du même registre permettent de configurer chaque broche ANO à AN12 soit en entrée analogique soit en entrée / sortie logique. Le troisième registre ADCON2 permettra de sélectionner une horloge pour le convertisseur analogique (bits ADCS0 à ADCS2), ainsi qu'un temps d'acquisition avant conversion (temps nommé TAD : bits ACQT0 à ACQT2),

allant de 0 TAD à 20 TAD (soit 0 top d'horloge à 20 tops d'horloge), puis indiquera grâce au bit ADFM le format de la valeur de la conversion en sortie. [11]

Lorsqu'une conversion est lancée, un bit nommé *GO/DONE* du registre ADCON0 est forcée à «1». Dès que la conversion en cours est terminée, ce bit repasse à « 0 ». On réalise donc une boucle d'attente tant que la conversion n'est pas échue. Notez également que le bit ADON du même registre permet d'alimenter le convertisseur en 5 V : si ce bit est à « 0 », alors le convertisseur ne consomme aucun courant. Le résultat de la conversion sera stocké dans deux registres 8 bits nommes ADRESH et ADRESL. Le résultat étant sur 10 bits, il faudra choisir (ajustement à droite ou ajustement à gauche) les 10 bits parmi les 16 contenus dans ces deux registres, le bit ADFM du registre de configuration ADCON2 aura ce rôle (fig 6). [11]

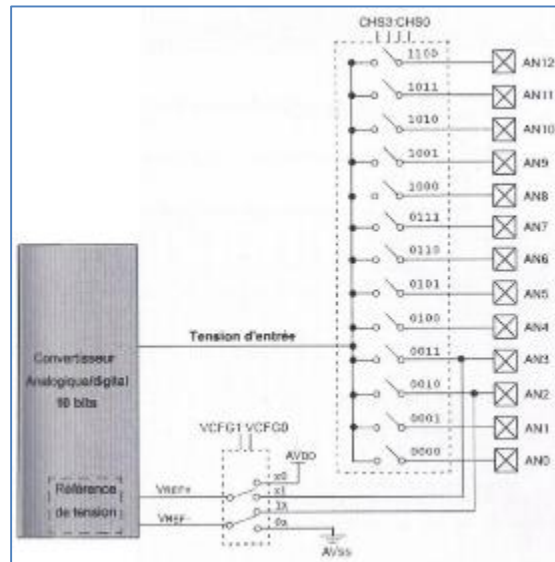


Figure 6 : Synoptique général du CAN pour le PIC18F4550

I.5- Conclusion

Ce chapitre est un portail pour notre réalisation et conception matérielle. Ca nous a permis de maîtriser les options de notre PIC et ses caractéristiques afin de les exploiter d'une manière correcte.

Chapitre II

Les liaisons RS-232 et USB

II.1-Introduction

La transmission d'informations nécessite l'utilisation d'un langage commun (écrit ou oral) ou d'un code commun. Il en est de même pour la transmission d'informations numériques entre deux postes de travail ou entre un poste de travail et un équipement, qui nécessite l'utilisation de règles communes : type de liaison, « vitesse » de transmission, format des données transmises, détection d'erreurs ...

Même si ce type de communication est peu à peu remplacé par l'USB, la liaison série de type RS232 est encore très utilisée dans l'industrie, mais aussi pour la fabrication de matériels informatiques professionnels.

II.2-RS-232

RS-232 (parfois appelée EIA RS-232, EIA 232 ou TIA 232) est une norme standardisant un bus de communication de type série sur trois fils minimum (électrique, mécanique et protocole). Disponible sur presque tous les PC depuis 1962 jusqu'au milieu des années 2000, il est communément appelé le « port série ». Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2, etc. Cela leur a valu le surnom de « ports COM », encore utilisé de nos jours. Cependant, il est de plus en plus remplacé par le port USB. [7]

Le standard RS-232 recouvre plusieurs autres standards : les recommandations UIT-T **V.24** (définition des circuits) et **V.28** (caractéristiques électriques), ainsi que la norme **ISO 2110** pour la connectique.

Les liaisons RS-232 sont fréquemment utilisées dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure, etc.). [7]

II.2.1-Historique

Le protocole d'origine, RS-232, date de 1962. Des déclinaisons ont suivi, notamment les RS-232C en 1969 et RS-232D en 1986. Peu à peu tombé dans l'obsolescence, il finit par être remplacé par les ports USB et FireWire¹ dans les années 2000. [7]

II.2.2-Description



Figure 7 : Prise femelle extrémité de câble type DE-9

La connectique de cette liaison se présente fréquemment sous la forme du connecteur DE-9 ou DB-25, mais peut aussi être d'un autre type (RJ25). Seule la version DB-25 est vraiment standardisée, la DE-9 (très souvent appelée DB-9 dans le commerce) est une adaptation d'IBM lors de la création du PC AT. La transmission des éléments d'information (ou bit) s'effectue bit par bit, de manière séquentielle. Cette transmission est décrite sur la page communication série. [7]

II.2.3-Utilisation



Figure 8 : Un connecteur DE-9 (Improprement appelé DB-9) mâle utilisé comme port série sur un ordinateur personnel

Vous pouvez voir au-dessus du port sur la photo ci-contre, le symbole pour les liaisons séries, représenté par les bits 0, 1 puis 0. [7]

Placé à l'arrière de l'ordinateur, il était souvent occupé par une souris ou un modem de type RTC, il pouvait aussi être utilisé pour le transfert des clichés numériques depuis l'appareil photo vers le disque dur du PC. [7]

Bien que ce port de communication ait tendance à être remplacé par l'USB sur les PC, il reste encore très utilisé dans l'industrie, notamment grâce à sa robustesse et à sa simplicité. Ainsi, ce port est toujours d'actualité, en particulier dans les systèmes automatisés : le transfert

de Graficets ou bien de lignes de programme pour machines-outils à commande numérique s'effectuent toujours par liaison RS-232. [7]

De même, de nombreux terminaux embarqués (qu'ils soient GPS, modems, terminaux graphiques, etc.) utilisent le RS-232 comme méthode principale de communication avec l'extérieur. Fréquemment, les périphériques réseau (routeurs, commutateurs, etc.) sont équipés d'un port RS-232 au travers duquel il est possible de les configurer. [7]

Néanmoins, les pc portables munis de ports séries deviennent de plus en plus difficile à trouver. En effet, très peu de constructeurs proposent encore ce type de ports.

En cas d'absence de port RS 232, il existe des adaptateurs USB/port série. [7]

II.2.3.1-Haute disponibilité

Pour la haute disponibilité, une liaison RS-232 est parfois utilisée : deux serveurs fonctionnent en cluster et ils se surveillent l'un l'autre via une liaison RS-232. C'est le cas par exemple de Heartbeat[7]

II.2.4-Spécification

Le standard RS-232 permet une communication série, asynchrone et duplex entre deux équipements. [7]

II.2.4.1-Portée du standard

En règle générale une jonction numérique entre un Équipement terminal de traitement de données (ETTD, en anglais DTE) et un équipement terminal de circuit de données (ETCD, en anglais DCE), située au niveau 1 du modèle OSI, se définit par trois paramètres : les circuits, les niveaux électriques et le brochage. Ce sont ces trois éléments qui sont couverts par le standard RS-232.

Plus précisément, le standard RS-232 spécifie :

- La fonction de chaque circuit de jonction. Cela correspond à la norme UIT-T V.24, dans laquelle ces circuits sont numérotés dans la série 100 (102 retour commun, 103 émission de données, 104, etc.).
- Les caractéristiques électriques de la liaison : niveaux de tension, débits possibles, etc. Cela correspond à la norme UIT-T V.28.

- Les caractéristiques mécaniques pour les connecteurs et l'identification des contacts.
Dans le cas d'un connecteur DB-25, cela correspond à la norme ISO 2110.

Par contre le standard ne définit pas :

- Le codage des caractères (ASCII, Code Baudot ou EBCDIC par exemple).
- La façon dont les caractères sont répartis en trames.
- Les protocoles de détection d'erreur ou les algorithmes de compression de données.
- Les débits de transmission : seule une limite supérieure de 20 000 bauds est indiquée.
- La fourniture de courant électrique à des périphériques. [7]

II.2.4.2-Mécanique

- Le standard spécifie initialement l'utilisation de connecteurs DB-25 mâle du côté DTE (*Data Terminal Equipment*), et DB-25 femelle du côté DCE (*Data Communications Equipment*) (modem).
- Les connecteurs DE-9 ont rapidement remplacé les DB-25, pour des raisons de taille de connecteur et d'économie de câblage, tous les signaux n'étant pas indispensables à la communication.
- Entre un DTE et un DCE, le câblage est droit : les broches sont connectées une à une de part et d'autre. Un tel câble est muni de connecteurs de genres différents (un mâle et une femelle). Ces câbles peuvent être connectés en série (comme des « rallonges »).
- Dans les configurations où deux DTE sont directement connectés (c'est-à-dire en l'absence de modem), un câble de liaison croisé dit « Null modem » doit être utilisé. Il est muni de connecteurs femelle à chaque extrémité, et possède un câblage spécifique, qui n'était pas prévu par la norme.
- Schéma usuel de raccordement d'un câble « Null modem » à 25 broches (symétrique) ; en gras les signaux croisés : **Attention, câblage à vérifier !!**

| | | | |
|--|-----|--|-------------------------|
| | Dir | | |
| | | | Dénomination (côté DTE) |

| | | | | |
|-------|-----|----|--------|--|
| 1 | --- | 1 | PG | Masse de blindage (protection électromagnétique) |
| 3 | ← | 2 | TxD | Données à transmettre |
| 2 | → | 3 | RxD | Réception des données |
| 5 | ← | 4 | RTS | Demande de transmission |
| 4 | → | 5 | CTS | Prêt pour transmission |
| 20 | → | 6 | DSR | Envoyez les données |
| 7 | --- | 7 | SG/GND | 0 Volt électrique |
| 20 | → | 8 | DCD | Détection d'un signal sur la ligne |
| 9 | --- | 9 | | + Tension |
| 10 | --- | 10 | | - Tension |
| 11 | --- | 11 | | |
| 12 | --- | 12 | SDCD | Deuxième Détection de signal sur la ligne |
| 19 | → | 13 | SCTS | Deuxième Prêt pour transmission |
| 16 | ← | 14 | STD | Deuxième Transmission des données |
| 17 | → | 15 | ST | Signal d'horloge pour transmission de données |
| 14 | → | 16 | SRD | Deuxième Réception des données |
| 15 | → | 17 | RT | Signal d'horloge pour réception de données |
| | | 18 | | DTE demande le rebouclage du DCE local |
| 13 | ← | 19 | SRTS | Deuxième Demande de transmission |
| 6 | ← | 20 | DTR | Données prêtes |
| | | 21 | | DTE demande le rebouclage du DCE distant |
| 22 | → | 22 | RI | Indicateur de sonnerie |
| | | 23 | | Signal de sélection de vitesse |
| 17,24 | ← | 24 | TT | Horloge de transmission |

| | | | | |
|--|--|----|--|---------------------------|
| | | 25 | | DCE en test de rebouclage |
|--|--|----|--|---------------------------|

Note : l'éventuel signal d'horloge émis par le terminal (DTE) en pin 24 est reçu par l'ordinateur (DCE) en pin 17. L'éventuel signal d'horloge émis par le DCE en pin 15 est reçu par le DTE en pin 17. [7]

II.2.4.3-Protocole

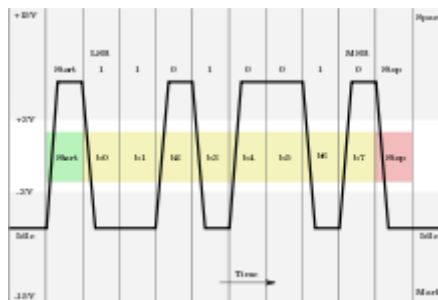


Figure 9 : Oscillogramme de la transmission du caractère K (01001011), avec un bit de départ et un bit d'arrêt.

Pour établir une communication effective via RS-232, il est nécessaire de définir le protocole utilisé : notamment, le débit de la transmission, le codage utilisé, le découpage en trame, etc. La norme RS-232 laisse ces points libres, mais en pratique on utilise souvent des UART qui découpent le flux en trames d'un caractère ainsi constituées :

- 1 bit de départ ;
- 7 à 8 bit de données ;
- 1 bit de parité optionnel ;
- 1 ou plusieurs bits d'arrêt.

Le bit de départ a un niveau logique "0" tandis que le bit d'arrêt est de niveau logique "1". Le bit de donnée de poids faible est envoyé en premier suivi des autres.

Application: pour générer un signal électrique alternatif carré (rapport cyclique 1:1) sur le port série, imprimer une suite consécutive de U (01010101) ce qui donne dans le temps 0(départ)10101010(U, du lsb au msb)1(arrêt) donc 0101010101 (01010101010101010101010101010101 = UUU) avec 8 bits de donnée, 1 bit départ, 1 bit arrêt et 0 bit de parité. Les niveaux électriques sont inversés (voir ci-dessous)

La spécification RS-232 prescrit des débits inférieurs à 20 000 bit/s. Cependant, les débits utilisés en pratique varient entre 75 bit/s et 115 200 bit/s. [7]

II.2.4.4-Électrique

Un niveau logique "0" est représenté par une tension de +3V à +25V et un niveau logique "1" par une tension de -3V à -25V (codage NRZ). D'ordinaire, des niveaux de +12V et -12V sont utilisés.

La norme V.28 indique qu'un 1 est reconnu si la tension est inférieure à -3 V, et un 0 est reconnu si la tension est supérieure à +3 V. [7]

II.2.5-Limites

Longueur maximum de câble RS232 [7]

| Débit (bit/s) | Longueur (pieds) | Longueur (m) |
|---------------|------------------|--------------|
| 2 400 | 200 | 60 |
| 4 800 | 100 | 30 |
| 9 600 | 50 | 15 |
| 19 200 | 25 | 7,6 |
| 38 400 | 12 | 3,7 |
| 56 000 | 8,6 | 2,6 |

II.3-MAX232

II.3.1-Présentation des convertisseurs rs232 max232

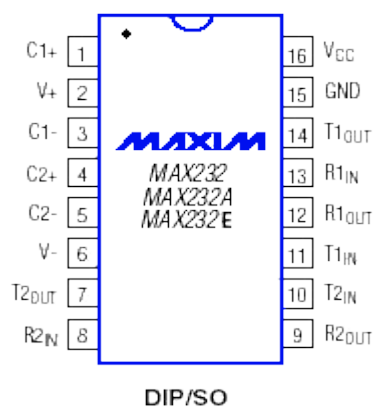


Figure 10: MAX232 - MAX232A - MAX232E

Le MAX232 est un composant créé par MAXIM que l'on trouve sous d'autres références chez d'autres fabricants. Il sert d'interface entre une liaison série TTL (0-5V) et une liaison série RS232 (+12 -12V) et ce avec une simple alimentation 5V. [8]

Rien d'original ? Pourtant le célèbre MAX232 a beaucoup évolué depuis sa création. Si le brochage est identique aux premières versions, d'autres caractéristiques ont évolué. Il existe aujourd'hui un grand nombre de versions, non décrites ici mais que vous trouverez dans le datasheet de Maxim. Sa consommation est plus faible, son débit admissible à augmenter, il est mieux protégé, plus fiable et il est fabriqué avec des boîtiers de tous types. Mais surtout, les condensateurs externes ont aujourd'hui des valeurs de capacité plus faible jusqu'à 0.1µF au lieu des 10 ou 47µF d'autrefois. Certaines versions se passent même de condensateur ! Je ne vous présenterai que la série des MAX232 - MAX232 et MAX232E[8]

II.3.2-Description des MAX232 et MAX 232A

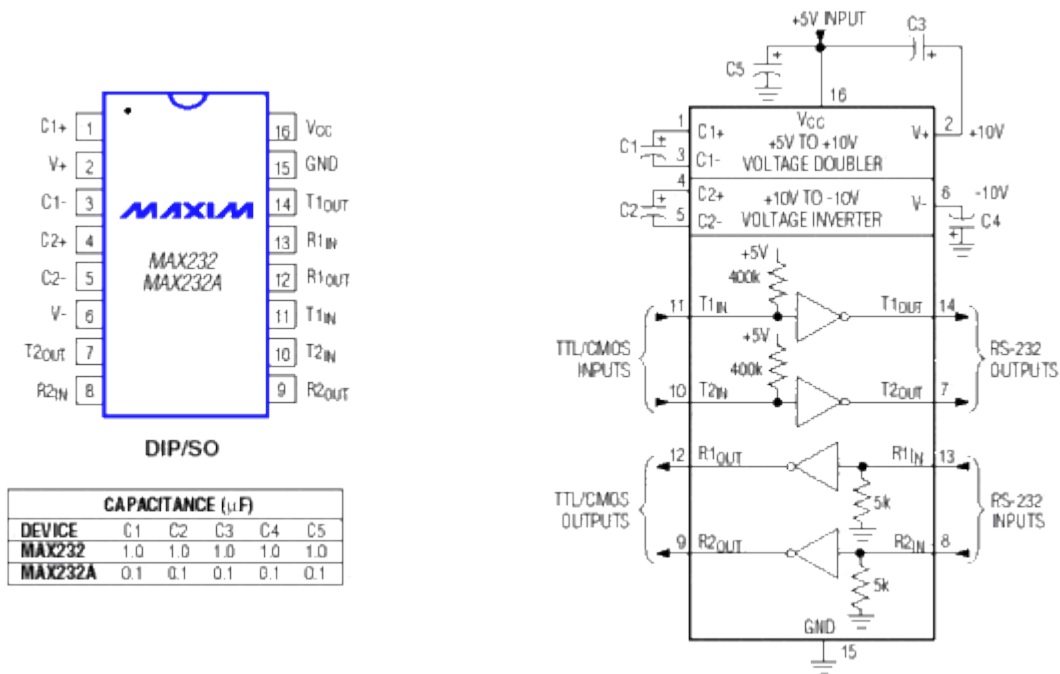


Figure 11 : le MAX232 et MAX232A

Sans trop rentrer dans les détails, le MAX232 et MAX232A sont assez proches en terme de caractéristiques électriques et se distinguent surtout par la valeur des condensateurs externes différentes et un débit plus faible pour le MAX232. Le câblage est assez simple et vous disposez de 2 drivers dans un sens et 2 dans l'autre, de quoi connecter RxD, TxD et RTS et CTS de la liaison RS232. Ce circuit convient donc dans la majorité des cas. [8]

II.3.3-Description du MAX232E

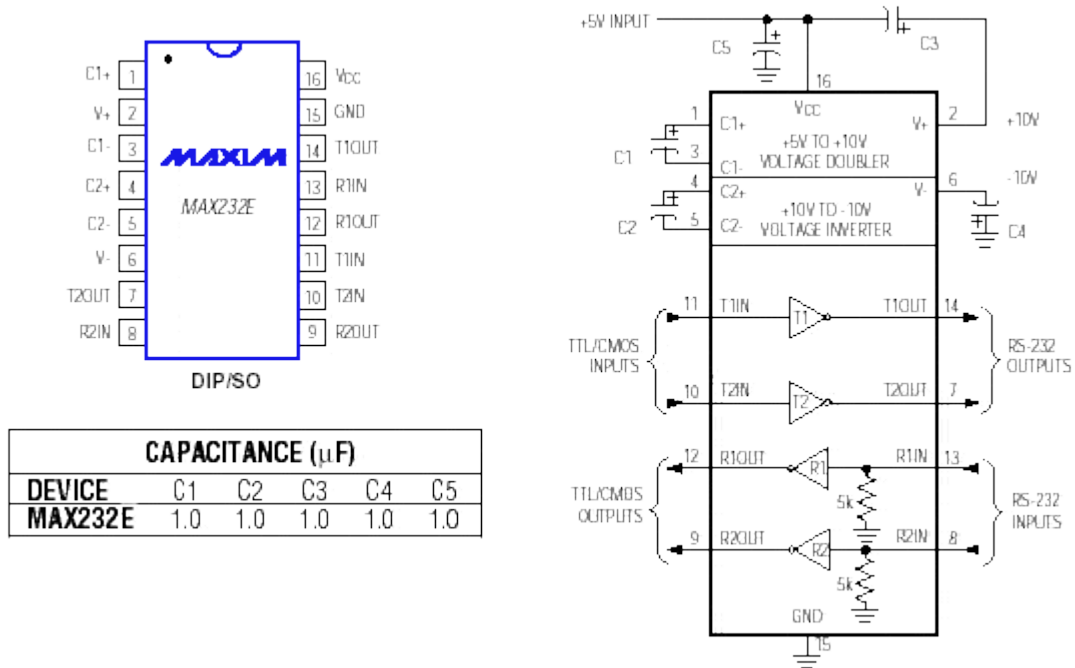


Figure 12 : Le MAX232E

Le MAX232E est compatible broche à broche avec les deux autres. Ses caractéristiques électriques sont sensiblement identiques avec le MAX232 mais il est extrêmement bien protégé des "electrostatic discharge (ESD) shocks" (décharges électrostatiques) et ce pour $\pm 15\text{kV}$. Fonctionnement garanti et protection maximum. [8]

II.4-Universal Serial Bus



Figure 13 : Connecteur USB de type A mâle.



Figure 14 : Symbole de l'USB

Le *Universal Serial Bus* (USB, en français *Bus universel en série*, dont le sigle, inusité, est *BUS*) 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). Il peut alimenter les périphériques peu gourmands en énergie (disques SSD en particulier). Apparu en 1996, ce connecteur s'est généralisé dans les années 2000 pour connecter souris, clavier d'ordinateur, imprimantes, clés USB et autres périphériques bon marché sur les ordinateurs personnels.

Les performances de l'USB, notamment concernant les débits, se sont grandement améliorées au fil des versions (USB 2.0, 3.0...). Un futur standard 3.1 a été annoncé en 2013 et ses spécifications techniques publiées en août 2014.

Pour plus de clarté, les débits dans cet article seront indiqués en octets et non en bits (pour rappel, 1 octet = 8 bits). [8]

II.4.1-Évolution de la norme USB

L'USB a été conçu au milieu des années 1990 afin de remplacer les nombreux ports externes d'ordinateurs, lents et incompatibles les uns avec les autres. Différentes versions de la norme ont été développées au fur et à mesure des avancées technologiques, chacune étant vouée à remplacer les précédentes car plus performante. [8]

II.4.1.1-USB 1.0

En 1996, la première version de la norme, l'**USB 1.0**, est spécifiée par sept partenaires industriels (Compaq, DEC, IBM, Intel, Microsoft, NEC et Northern Telecom). Mais elle reste théorique et n'a jamais vraiment été appliquée : par manque de composants, il faudra attendre la seconde version de la norme 1998, intitulée **USB 1.1**, pour que l'USB commence à être effectivement utilisé¹. Ce que nous appelons couramment "USB 1" est donc en réalité de l'USB 1.1.

L'USB 1.1 apporte des corrections à la norme 1.0 et définit également deux vitesses de communication :

Le mode lent (Low Speed) a un débit de **190 Ko/s**. Il permet de connecter des périphériques qui ont besoin de transférer peu de données, comme les claviers et souris ;

Le mode pleine vitesse (Full Speed) débite à **1,5 Mo/s**. Il est utilisé pour connecter des imprimantes, scanners, disques durs, graveurs de CD et autres périphériques ayant besoin de plus de rapidité. Néanmoins, il est insuffisant pour beaucoup de périphériques de stockage de masse (ce mode permet la vitesse « 10 X » des CD).

En août 1998, avec la sortie de l'iMac G3, Apple est le premier² constructeur à proposer un appareil disposant uniquement de ports USB en remplacement des ports d'ancienne génération, ce qui a fait décoller le marché des périphériques USB. [8]

II.4.1.2-USB 2.0

En avril 2000 est publiée la norme **USB 2.0**, qui optimise l'utilisation de la bande passante et surtout introduit un troisième débit à **60 Mo/s**, baptisé Haute vitesse (High Speed). Il est utilisé par les périphériques rapides : disques durs, graveurs... Mais au moment de sa sortie, la plupart des périphériques ont une vitesse inférieure à celle permise par l'USB 2.0.

En 2005, le **Wireless USB**, une version sans-fil de l'USB, est spécifiée par le Wireless USB Promoter Group. Elle promet 50 Mo/s à une distance de 3 m et 14 Mo/s à 10 m.

L'extension **On-The-Go** (OTG), ajoutée à la norme USB 2.0 en 2007, permet d'effectuer des échanges de données point à point entre deux périphériques sans avoir à passer par un hôte (généralement un ordinateur personnel). La norme OTG s'impose désormais comme un standard. [8]

II.4.1.3-USB 3.0

En 2008, l'USB 3.0 introduit le mode vitesse supérieure (SuperSpeed), qui débite théoriquement à 625 Mo/s. Mais ce nouveau mode utilisant un codage des données de type 8b/10b, la vitesse de transfert réelle est de seulement 500 Mo/s. L'USB 3 délivre une puissance électrique de 4,5 watts.

Les nouveaux périphériques disposent de connexions à 6 contacts au lieu de 4, mais la compatibilité ascendante des prises et câbles avec les versions précédentes est assurée. En revanche, la compatibilité descendante est impossible, les câbles USB 3.0 de type B n'étant pas compatibles avec les prises USB 1.1/2.05, mais il existe des adaptateurs.

Début 2010, l'USB 3 est introduit dans des produits grand public. Les prises femelles correspondantes sont signalées par une couleur bleue. Apparition aussi des prises femelles USB rouges, signalant une puissance électrique disponible supérieure, et appropriée au chargement rapide de petits appareils y compris (à condition de le paramétrer dans le BIOS ou l'EFI) lorsque l'ordinateur est éteint. [8]

II.4.1.4-USB 3.1

Un standard 3.1 à 10 Gb/s est annoncé en août 2013 ; ses spécifications techniques sont finalement publiées par le consortium *USB Implementers Forum* en août 2014.

L'**USB 3.1** promet des débits doubles de ceux de l'USB 3.0, soit **1,2 Go/s**. Le nouveau standard (câbles, interface) est rétro compatible avec l'USB 3.0 et l'USB 2.0. Toutefois la connectique change, elle est plus fine et n'impose pas de sens de branchement. Pour tout de même permettre la connexion vers des connecteurs USB 2.0 et 3.0, le standard permet la possibilité d'avoir des adaptateurs passifs (à l'inverse des adaptateurs Lightning, le connecteur réversible qu'Apple a lancé avec l'iPhone 5 en 2012), pour garder une taille réduite et un coût de fabrication mesuré. Cette nouvelle connectique se nomme *Type C*.

Le 9 mars 2015, Apple présente le MacBook, le premier ordinateur équipé d'un port USB Type C, mais ne bénéficiant toutefois que du débit de l'usb 3.0 (5 Gb/s) au lieu de celui de l'USB 3.1(10 Gb/s). Le connecteur utilisé est un port USB 3.1 Gen 1. [8]

II.4.1.5-Résumé des débits

Lorsque l'on parle d'un équipement USB, il est nécessaire de préciser la version de la norme (1.1, 2.0 ou 3.0) mais également la vitesse (*low/full/high/super speed*). Une clef USB

spécifiée en USB 2.0 n'est pas forcément High Speed si cela n'est pas précisé par un logo « High Speed ».

Le bus USB reste plus lent que certaines interfaces internes comme PCI ou AGP ou SATA / e-Sata, mais il comble son retard au fil des versions, au moins en termes de débits théoriques. Ainsi, si l'USB était dix fois plus lent en version 2.0 (60 Mo/s) que le SATA III (768 Mo/s), sa version 3.0 fait presque jeu égal (600 Mo/s) et la version 3.1 la surclasse largement (1,2 Go/s).

| Débit théorique des normes USB | | | | | | |
|--------------------------------|-------------------------|-----------------------|-----------------------|-----------------------|----------------------|-----------------------|
| Version | USB 1.0 | USB 1.1 | USB 2.0 | Wireless USB | USB 3.0 | USB 3.1 |
| Année | 1996 | 1998 | 2000 | 2005 | 2008 | 2013 |
| Débit | 1,5 Mbit/s 0,19 Mo/s | 12 Mbit/s 1,5 Mo/s | 480 Mbit/s 60 Mo/s | 480 Mbit/s 60 Mo/s | 5 Gbit/s 600 Mo/s | 10 Gbit/s 1,2 Go/s |

Ces débits ne sont toutefois que théoriques. Afin de trouver des valeurs concrètes, divisez la vitesse par 2 en utilisation optimale, 4 en utilisation normale, voire 15 avec beaucoup de périphériques USB simultanément utilisés. [8]

II.4.2-Évolution des connecteurs USB



Figure 15 : Différents connecteurs de type USB 1 et 2

De gauche à droite :

- micro-B mâle ;
- UC-E6 propriétaire (non USB) ;
- mini-B mâle ;
- A femelle ;
- A mâle ;
- B mâle. [8]

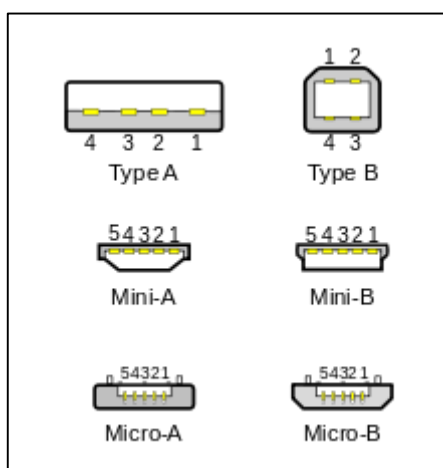


Figure 16 : Les différents types de connecteurs USB 1 et 2.



Figure 17 : Fiche micro-B USB 3.

II.4.2.1-Les premiers connecteurs : les types A et B

Le bus USB ne permet pas de relier entre eux deux périphériques ou deux hôtes : le seul schéma de connexion autorisé est un périphérique sur un hôte. Pour éviter des branchements incorrects, la norme spécifie deux types de connecteurs : le **type A**, destiné à être situé sur l'hôte, et le **type B**, destiné à être situé sur le périphérique.

Un hub USB peut comporter à la fois un connecteur B, qui permet de le relier à l'hôte, et des connecteurs A, qui permettent d'y relier des périphériques. Les appareils (hôte et périphériques) sont équipés de connecteurs femelles. Les câbles de connexion ont *toujours* une extrémité de type A mâle, et une extrémité de type B mâle, ce qui garantit le respect de la topologie du bus. Il peut aussi exister des câbles de prolongation équipés de connecteurs de même type mais de genres différents. [8]

II.4.2.2-Les mini-connecteurs

Chaque type (A ou B) existait dans les deux genres (mâle ou femelle), ce qui fait qu'il existait au départ quatre connecteurs. En octobre 2000, devant le développement des appareils compacts (téléphones portables, appareils photo numériques), une mise à jour de la norme USB 2 introduit une version miniature du connecteur B : le **mini-B**. Elle est fonctionnellement équivalente au connecteur B, mais de dimensions nettement réduites.

En décembre 2001, l'USB 2 introduit le connecteur **mini-AB**, utilisé dans le cadre de l'extension « On-The-Go »¹³. Il permet aux appareils compatibles de jouer indifféremment le rôle d'hôte ou celui de périphérique, contrairement à l'USB classique où l'hôte se branche obligatoirement sur un connecteur de type A et le périphérique sur un connecteur de type B. [8]

II.4.2.3-Les micro-connecteurs

Les appareils mobiles s'étant encore réduits, les connecteurs mini-B sont devenus à leur tour trop gros. En janvier 2007, le nouveau connecteur **micro-B** est annoncé¹⁴. Il est non seulement plus fin que le mini-B, mais également prévu pour supporter un grand nombre de cycles de connexion/déconnexion (jusqu'à 10 000), ce qui le rend particulièrement bien adapté aux appareils mobiles souvent branchés/débranchés (tablettes tactiles, smartphones, etc.).

Pour les mêmes raisons, en avril 2007, une nouvelle norme **micro-AB** vient remplacer la norme **mini-AB**, qui est officiellement dépréciée le mois suivant. [8]

II.4.2.4-La nouvelle norme : le type C

Un nouveau connecteur est introduit dans la norme le 11 août 2014 : le **type C**, destiné à remplacer tous les connecteurs précédents. Il a la particularité d'être réversible, c'est-à-dire qu'il n'a plus de sens haut/bas. Outre l'aspect pratique, il est compatible à la fois avec le standard USB 3.1 (qui porte le débit maximal théorique à 10 Gbit/s) et l'USB Power Delivery. La technologie DisplayPort lui permet également de transmettre des signaux audio et vidéo. [8]

II.4.3-Applications de l'USB

II.4.3.1-Pour le transfert des données

USB a supplanté divers bus qui équipaient auparavant les ordinateurs : port série RS-232, port parallèle, port PS/2, port joystick (ou port MIDI), port SCSI, et même des bus internes comme PCI pour la connexion de certains dispositifs (par exemple cartes son ou cartes de réception TV).

- La gamme des périphériques utilisant le bus USB est extrêmement vaste :
- périphériques d'interaction avec l'utilisateur : claviers, souris, joystick, guitare ;
- périphériques de stockage : disques durs externes, appareils photo, lecteurs multimédia, et surtout clés USB, un concept apparu spécifiquement pour le bus USB. Il s'agit de l'association d'une mémoire flash et d'une interface USB, le tout contenu dans un petit boîtier évoquant une clé par sa taille et sa forme ;
- multimédia et imagerie : imprimantes, scanners, cartes son, webcams, tuners TV, écran secondaire (intégrant son propre contrôleur vidéo), microphone ;
- adaptateurs de réseau ou de communication : Wi-Fi, Ethernet, Bluetooth, infrarouge IrDA, Modem ;
- Bus et interfaces : port série RS-232, port parallèle, port PS/2, port joystick, Bus CAN, GPIB (IEEE-488), port série RS-485.

Le bus USB est également utilisé en interne dans certains ordinateurs pour connecter des périphériques tels que webcams, récepteurs infrarouges (c'est le cas par exemple sur les MacBook Pro) ou lecteurs de cartes mémoire. [8]

II.4.3.2-Pour l'alimentation électrique



Figure 18 : Ventilateurs alimentés par USB



Figure 19 : Petit accessoire affichant et enregistrant la tension (V), le courant (A) et la charge (mAh) au cours de l'alimentation d'un appareil par USB, telle que la recharge d'un téléphone.

Le bus USB peut alimenter en énergie les périphériques, dans une certaine limite de courant consommé (2 A pour une application haute puissance, 100 mA pour une application normale). Ceci est notamment mis à profit pour permettre la recharge d'appareils portables, pour lesquels on voit apparaître des adaptateurs secteur disposant d'une connectique USB limitée à l'alimentation électrique.

La connectique USB a ainsi une diffusion au-delà des périphériques informatiques *stricto sensu*, en tant que connecteur électrique de faible puissance. Un certain nombre de gadgets alimentés par port USB qui ne sont pas des périphériques informatiques sont apparus sur le marché : lampes d'appoint, petits ventilateurs, etc.

Cependant le courant délivré par l'USB reste pour le moment trop faible pour certains périphériques, par exemple certains disques durs externes. La solution actuelle consiste à compléter l'alimentation par un branchement sur un second port USB, mais cette pratique est discutable car le périphérique monopolise à lui seul deux ports.

Ce problème devrait être résolu avec la nouvelle norme USB. En effet, un câble standard avec prises de type C (norme USB 3.1) est prévu pour une puissance électrique de 60W. Des câbles spécifiques peuvent faire transiter jusqu'à 100W. On tend vers l'utilisation d'un câble unique pour les périphériques qui assure à la fois l'alimentation et le transfert des données. Par

exemple, on peut connecter un écran à un hub USB intégré avec un seul câble USB. Et ce, sans se soucier du sens du câble ou de la prise.

L'USB est aussi devenu un moyen d'alimenter un ordinateur et pas seulement ses périphériques. En 2015, Apple sort un MacBook où l'unique port (à part le port casque) est une prise type C qui permet de le recharger. [8]

II.4.4-Spécifications techniques

II.4.4.1-Caractéristiques générales

L'*Universal Serial Bus* est une connexion à haute vitesse qui permet de connecter des périphériques externes à un ordinateur (hôte dans la terminologie USB). Il permet le branchement simultané de 127 périphériques par contrôleur (hôte). Le bus autorise les branchements et débranchements à chaud (« *Hot-Plug* », sans avoir besoin de redémarrer l'ordinateur) et fournit l'alimentation électrique des périphériques sous 5 V, dans la limite de 0,5 A, soit 2,5 W.

D'un point de vue logiciel, le bus possède une topologie arborescente (dite également en étoile) : les feuilles de cet arbre sont les périphériques ; les nœuds internes sont des *hubs* qui permettent de greffer des sous-arborescences dans l'arborescence principale. On trouve dans le commerce ces *hubs* sous forme de petits boîtiers alimentés soit sur le bus, soit sur le secteur, et qui s'utilisent comme des multiprises. Certains périphériques intègrent également un *hub* (moniteurs, claviers...). Cependant, tout bus USB possède au moins un *hub* situé sur le contrôleur : le *hub racine*, qui peut gérer les prises USB de l'ordinateur. Le nombre de *hubs* connectés en cascade est limité : *hub racine* compris, il ne doit pas exister plus de 7 couches dans l'arborescence.

À plus bas niveau, il s'agit d'un anneau à jeton (ou *Token Ring*) : chaque nœud dispose successivement du bus. Il n'y a pas de collision de paquets comme en Ethernet, mais le nombre maximal de nœuds est prédéfini. Pour cette raison, l'USB n'est pas adapté aux communications réseau : l'apparition des "modems" ADSL USB était un moyen de diffuser l'ADSL à une époque où la plupart des PC bas de gamme disposaient du port USB mais pas d'Ethernet. [8]

II.4.4.2-Protocole

La bande passante est partagée temporellement entre tous les périphériques connectés. Le temps est subdivisé en trames (frames) ou microtrames (microframes) pendant lesquels plusieurs transferts peuvent avoir lieu.

La communication entre l'hôte (l'ordinateur) et les périphériques se fait selon un protocole basé sur l'interrogation successive de chaque périphérique par l'ordinateur. Lorsque l'hôte désire communiquer avec un périphérique, il émet un jeton (un paquet de données, contenant l'adresse du périphérique, codée sur sept bits) désignant un périphérique. Si le périphérique reconnaît son adresse dans le jeton, il envoie un paquet de données (de 8 à 255 octets) en réponse. Les données ainsi échangées sont codées selon le codage NRZI. Puisque l'adresse est codée sur 7 bits, 128 périphériques (2) peuvent être connectés simultanément à un port de ce type. Il convient en réalité de ramener ce chiffre à 127 car l'adresse 0 est une adresse réservée.

USB définit quatre types de transferts :

- *transfert de commande*, utilisé pour l'énumération et la configuration des périphériques. Il convient pour des données de taille restreinte ; il y a garantie de livraison (renvoi des paquets erronés) ;
- *transfert d'interruption*, utilisé pour fournir des informations de petite taille avec une latence faible. Ce ne sont pas des interruptions au sens informatique du terme : le périphérique doit attendre que l'hôte l'interroge avant de pouvoir effectuer un tel transfert. Ce type de transfert est notamment utilisé par les claviers et les souris ;
- *transfert isochrone*, utilisé pour effectuer des transferts volumineux (bande passante garantie), et en temps réel. Il n'y a pas de garantie sur l'acheminement des données. Ce type de transfert est utilisé pour les flux audio et vidéo ;
- *transfert en masse (bulk)*, utilisé pour transférer des informations volumineuses, avec garantie d'acheminement, mais sans garantie de bande passante. Ce type de transfert est utilisé par les dispositifs de stockage.

Il est possible de structurer la communication entre un hôte et un périphérique en plusieurs canaux logiques (*pipes* et *endpoints*) pour simplifier la commande du périphérique du port USB.

- L'USB 3.1 introduit l'*Alternate Mode* (Mode Alternatif) qui est utilisé par exemple pour faire passer de la vidéo avec le protocole Display Port contrairement à des technologies existantes comme DisplayLink qui encapsulaient de la vidéo au travers du protocole USB standard. Ce mode permet aux constructeurs d'étendre l'usage de l'USB à d'autres fonctions. Pendant la négociation de protocole de l'USB-PD 2.0 (Power Delivery, rendu obligatoire avec les connecteurs Type-C), un identifiant assigné par l'USB-IF est échangé pour déterminer le mode de fonctionnement alternatif. On peut utiliser seulement les canaux supplémentaires, mais aussi ceux destinés à l'USB 2.0. [8]

II.4.4.3 Connexion à chaud et Plug and Play : processus d'énumération

Les ports USB supportent la connexion à chaud et la reconnaissance automatique des dispositifs (*Plug and Play*). Ainsi, les périphériques peuvent être branchés sans éteindre l'ordinateur.

Lors de la connexion du périphérique à l'hôte, ce dernier détecte l'ajout du nouvel élément grâce au changement de la tension entre les fils D+ et D-. À ce moment, l'ordinateur envoie un signal d'initialisation au périphérique pendant 10 ms, puis lui fournit du courant grâce aux fils GND et VBUS (jusqu'à 100 mA). Le périphérique est alors alimenté en courant électrique et peut utiliser temporairement l'adresse par défaut (l'adresse 0). L'étape suivante consiste à lui fournir son adresse définitive et à obtenir sa description : c'est la procédure d'énumération.

En effet, après avoir reçu son adresse, le périphérique transmet à l'hôte une liste de caractéristiques qui permettent à ce dernier de l'identifier (type, constructeur, nom, version). L'hôte, disposant de toutes les caractéristiques nécessaires est alors en mesure de charger le pilote approprié.

Les périphériques sont regroupés en types ou *classes* dans la terminologie USB. Tous les dispositifs d'une classe donnée reconnaissent le même protocole normalisé. Il existe par exemple une classe pour les périphériques de stockage de masse (*mass storage class*, MSC),

implémentée par la quasi-totalité des clés USB, disques durs externes, appareils photo et par certains baladeurs. La plupart des systèmes d'exploitation possèdent des pilotes génériques, pour chaque type de périphérique. Ces pilotes génériques donnent accès aux fonctions de base, mais des fonctions avancées peuvent manquer. [8]

II.4.4.4-Alimentation électrique

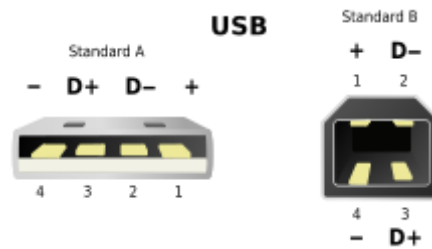


Figure 20 : Prises USB de type A et B, vue de face. USB 1 et 2.

L'architecture USB a pour caractéristique de fournir aussi l'alimentation électrique aux périphériques. Il utilise pour cela un câble composé de quatre fils pour les USB 1 et 2 (la masse GND, l'alimentation VBUS et deux fils de données appelés D- et D+) et de six fils pour l'USB 3 (séparation des données IN/OUT). Les fils D+ et D- forment une paire torsadée et utilisent le principe de la transmission différentielle afin de garantir une certaine immunité aux bruits parasites de l'environnement physique du périphérique ou de son câble. [8]

Chapitre III

Conceptions et réalisation

III.1- Introduction

Tout d'abord, ces microcontrôleurs sont vendus vierges, c'est à dire qu'ils n'ont aucun programme intégré dedans. Pour y intégrer un programme, il faut un appareil appelé un programmeur.

Il en existe de toutes sortes, les plus chers étant ceux qui permettent de programmer le plus grand nombre de modèles de microcontrôleurs. L'ennui vient justement de leur coût – trop élevé pour mes maigres moyens – et que les plus abordables sont limités et/ou disposent uniquement d'un port RS232, port absent de certains ordinateurs de nos jours (notamment les portables).

L'idéal serait de le connecter en USB, possibilité offerte par certains programmeurs figurant parmi les plus onéreux. Il existe bien des adaptateurs RS232/USB mais les programmeurs ne fonctionnent pas toujours avec. Le but est donc d'en réaliser un avec les moyens du bord. Cela apporte également l'opportunité d'étudier le fonctionnement d'un programmeur.

III.2- Présentation des logiciels utilisés pour la conception :

III.2.1- Proteus

Proteus est une suite de logiciels permettant la CAO électronique éditée par la société L'absenter Electronics.

Proteus est composé de deux logiciels principaux :

ISIS, permettant entre autres la création de schémas et la simulation électronique, et ARES, dédiée à la création de circuits imprimés.

III.2.2- ISIS

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs. Le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception . indirectement , les circuits électriques conçus grâce à ce logiciels peuvent être utilisé dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

III.2.3- ARES

Logiciel ARES est un outil d'édition et de routage qui complété parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB (Printed circuit Board) de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficiente lorsqu'elle est réalisée manuellement, ce logiciel permet déplacer automatiquement les composants et de réaliser le routage automatiquement.

Mais Malheureusement, il y'a des composantes qui ne sont pas dans les bibliothèques fournies .il faut donc dans premier temps créer une librairie qui contiendra tous les composants du projet , puis créer les composants cette dernière tâche est réellement simplifiée puisqu'il suffit simplement de créer le contour du composant et d'ajouter ensuite les différents broches autour de celui-ci , en prenant soin de respecter la nature de la broche (input, output ,power ,etc.). Ceci est capital dans la phase de vérification du schéma et de sa préparation à l'exportation vers un logiciel de routage car ce dernier pourra alors détecter d'éventuelles erreurs de connections de broches de composants.

III.2.4- Les avantage de Proteus :

- Pack contenant des logiciels facile et rapide à comprendre et utiliser.
- le support technique est performant.
- l'outil de création de prototypes virtuel permet de réduire les couts matériel et logiciel lors de la conception d'un projet.

III.3- Le logiciel pilotant le programmeur de PIC

C'est le logiciel PIC877DATS qui prend en charge ce programmeur ABTEC. Vous le trouverez avec la carte pédagogique ABITEC.



Figure 21 : PIC877DATS

Le logiciel PIC877DATS permet une gestion efficace et rapide du programmeur ABITEC :

PIC877DATS détecte automatiquement le PIC placé sur les supports du programmeur,

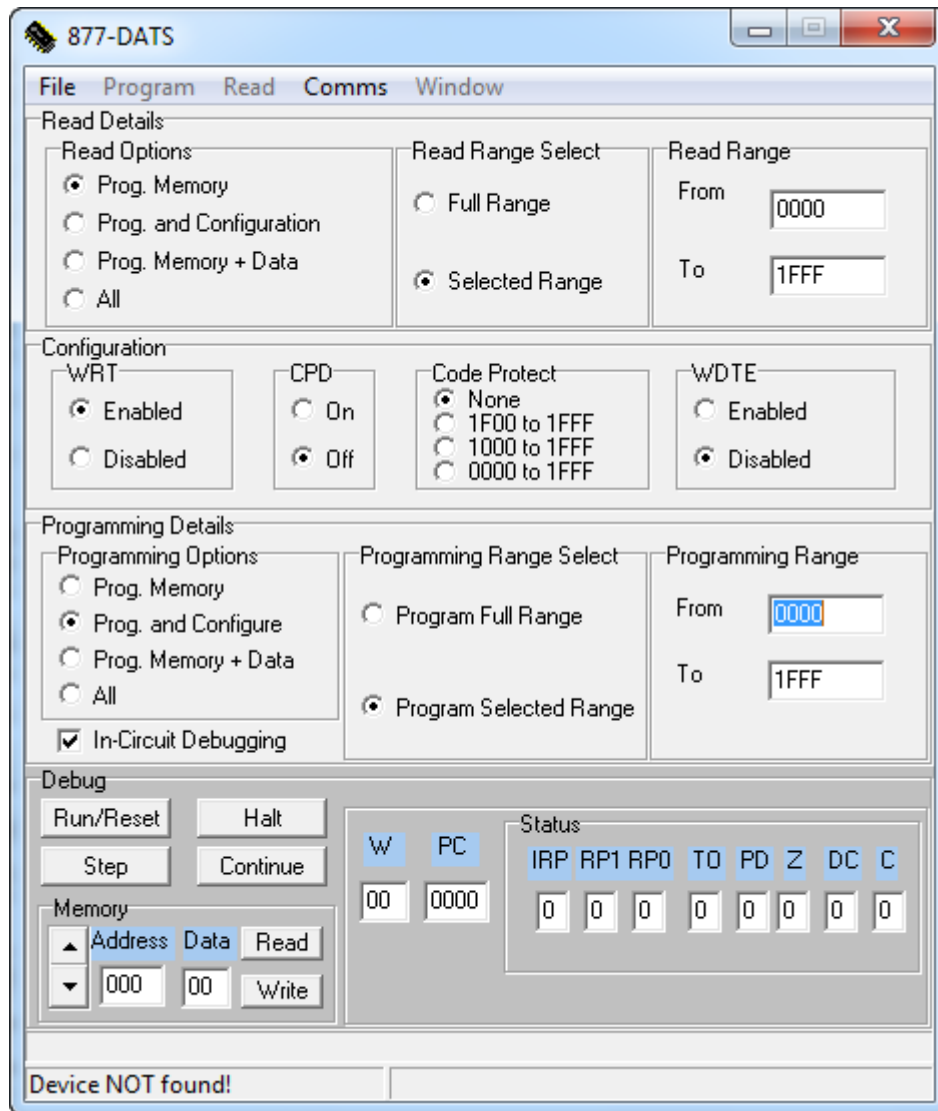


Figure 22 : la fenêtre de PIC877DATS

III.3.1- Expliqué la fenêtre

Dans la fenêtre de logiciel il y a trois parties et le menu :

- Read Details. « Lire les détails »
- Configuration. « Configuration »
- Programming Details « détails de la programmation »

III.3.1.1- Read Details. « Lire les détails »



Figure 23: Read Details. « Lire les détails »

Read Options « lire options »

Prog. Memory : « Mémoire de programme Mémoire en lecture seule »

Prog. and Configuration : « Lire la mémoire de programme et le périphérique configuration. »

Prog. Memory + Data « Lire la mémoire de programme et de la mémoire de données. »

All « Lire la mémoire de programme, la mémoire de données et configuration de l'appareil. »

Read Range « lire plage »

Full Range « Plage complète » : « Lire toute la mémoire de programme dans la plage de 0000 à 1FFFh. »

Selected Range « Plage sélectionnée » : « la plage de mémoire de programme spécifié dans le Lire la section Range. »

III.3.1.2- Configuration

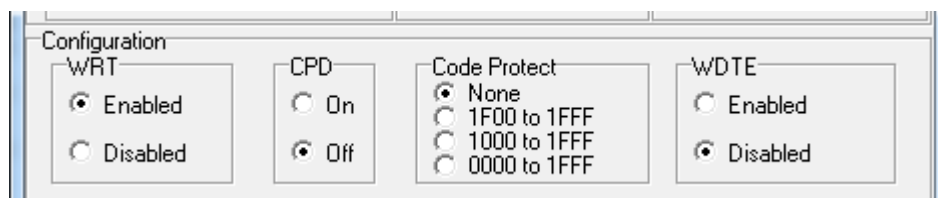


Figure 24 : Configuration

WRT « Active / désactive la mémoire programme flash écrit. »

CPD « Commutateurs EE mémoire de protection des données de code ou le désactiver. »

Code de protect « sélectionner le niveau de protection de code. »

WDTE « Active / désactive de chien de garde. »

III.3.1.3- Programming Details « détails de la programmation »



Figure 25 : Programming Details « détails de la programmation »

Ceci est très similaire à la section Lire les détails. La différence est que le dispositif est programmé au lieu d'être lue.

III.3.1.4- The Menus « Les menus »

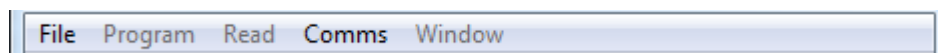


Figure 26 : The Menus « Les menus »

- File « fichier »
- Program « Programme »
- Read « Lire »
- Comms
- Window « Fenêtre »

File « fichier » :

Le menu déroulant de fichier ne comporte que deux articles:

- Load « Charger »
- Exit « Quitter »

Load « Charge »

Utilisez «charge» pour charger un programme binaire dans l'ordinateur avant la programmation.



Figure 27 : le menu charge

L'extension de fichier par défaut est HEX tel que produit par Microchips cross-assembleur.

Exit « Quitter »

Sélection Quitter ferme le programme.

Program « Programme »

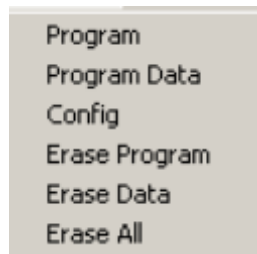


Figure 28 : le menu program

- Program
- Program data
- Config
- Erase program
- Erase data
- Erase all

Program

Mémoire de programme seulement

La mémoire de programme et de configuration

Programme et mémoire de données ou toute

Program data

Ce programme les 64 octets de mémoire de données dans le PIC16F877.

Config

Ce programme le mot de configuration dans le dispositif, comme indiqué dans la

Section Configuration.

Erase program

La sélection de cette provoque la mémoire de programme que dans le dispositif d'être effacé.

Erase Data

La sélection de cette option entraîne la mémoire de données au sein de l'appareil d'être effacé.

Erase All

Cela efface tout le dispositif, y compris la configuration, même si les bits protégés sont fixés pour protéger tous.

Read « Lire »



Figure 29 : le menu read « Lire »

- Read Prog.
- Read Data
- Read Config.
- Blank Check
- Verify

Read Prog.

Cette lire la mémoire de programme choisi par les valeurs dans le Read

Details section

Read Data

Cette lire les 64 octets de mémoire de données seulement.

Read Config.

Cette lire la configuration actuelle de l'appareil. « PIC »

Blank Check

Celui-ci vérifie que la mémoire de programme est vide.

Verify

Vérification de lire la mémoire de programme et le compare aux données contenues dans l'ordinateur

Comms « communications »

Il ya seulement un élément dans le menu de communications, qui est détecter.

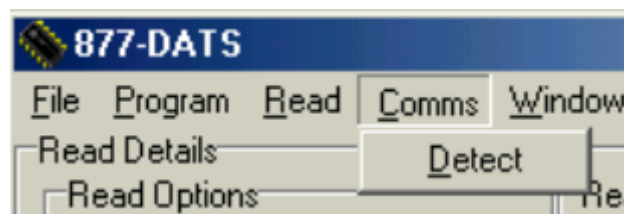


Figure 30 : le menu de communications

Cette "regarde" pour le conseil 877-DATS sur tous les ports série disponibles, si le système est trouvé le port COM utilisé est affiché dans la barre d'état contraire un dispositif introuvable message est affiché.

Control Switches « Interrupteurs de commande »

Le fonctionnement de l'unité est commandée par deux interrupteurs, interrupteur à glissière S1 (PRG / RUN) et bouton-poussoir S2 (START / STOP)

Commutateur PRG / RUN doit être dans la position PRG. Pour exécuter le programme, faites glisser le PRG / RUN Interrupteur à la position RUN, appuyant désormais le commutateur STOP/START bascule le réinitialiser ligne au PIC16F877. L'état actuel est indiqué par le RUN (vert) LED.

- In-Circuit Debugging (ICD)
ICD est implémenté en utilisant des fonctionnalités matérielles particulières construites dans le PIC16F877
 - Démarrer et arrêter les programmes à partir du PC
 - programmes en une seule étape
 - Définir des points d'arrêt
 - Examiner le contenu des registres de fichiers
 - montres de fichiers Set
 - Affichage spécial registres de fonction (SFR)

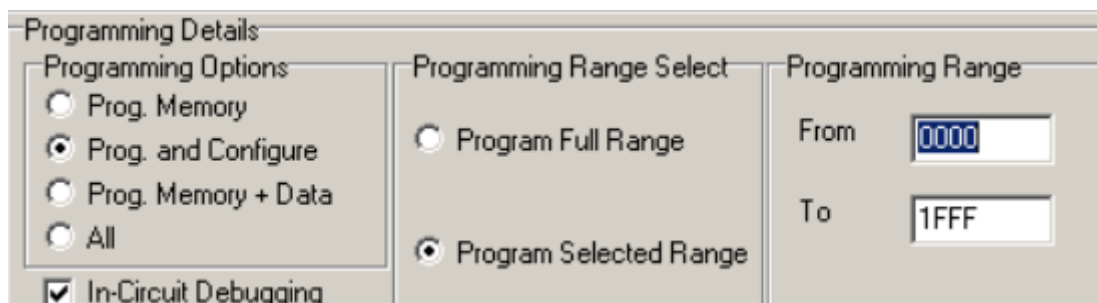


Figure 31 : Control Switches

Si ICD est activé, lorsque le PIC16F877 est programmé la séquence suivante prend lieu:

- Le programme utilisateur est chargé, la progression du chargement est indiquée sur la barre de progression
- Après le programme utilisateur est chargé le code de la ICD sera programmé et encore des progrès sera indiqué sur la barre de progression
- Le mot de configuration sera programmé (noter la configuration de l'utilisateur information est ignorée lors de l'utilisation de la ICD)

La section principale de débogage est indiquée ci-dessous:



Figure 32 : La section principale de débogage

- Run/Reset

Cliquer sur ce bouton permet de basculer la carte cible entre Run et Reset. L'état de la carte cible est indiquée par le "RUN" (verte) LED.

- Halt

Si la programmation de l'utilisateur est en cours d'exécution, en cliquant sur ce bouton interrompt l'exécution du programme et le contenu du PC

- Continue

Si le conseil de cible exécute le code de la ICD, en cliquant sur ce bouton va provoquer la carte cible de poursuivre l'exécution du programme de l'utilisateur.

- Step

Si le conseil de cible exécute le code de la ICD, en cliquant sur ce bouton va provoquer le microcontrôle pour exécuter une seule instruction

Window « Fenêtre »

Le menu Fenêtre permet à l'utilisateur d'ouvrir trois fenêtres:

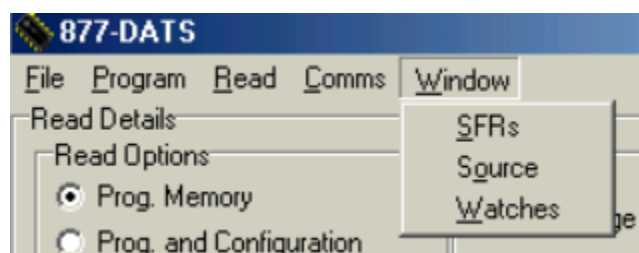


Figure 33 : le menu Window « Fenêtre »

- La fenêtre Special Function Register (SFR)
- La fenêtre Source
- La fenêtre Montres

La fenêtre Special Function Register (SFR)

Cette fenêtre affiche tous les registres de fonctions spéciales dans le PIC16F877. Les valeurs sont mises à jour uniquement lorsque le code de la ICD est en marche et le bouton 'Lire la FSR de est cliqué:

| REGISTER | VALUE | REGISTER | VALUE | REGISTER | VALUE |
|------------|-------|------------|-------|----------------|-------|
| 01 TMRO | FF | 81 OPTION | FF | 101 TMRO | FF |
| 02 PCL | 13 | 82 PCL | 13 | 102 PCL | 13 |
| 03 STATUS | 1B | 83 STATUS | 1B | 103 STATUS | 1B |
| 04 FSR | 00 | 84 FSR | 00 | 104 FSR | 00 |
| 05 PORTA | 00 | 85 TRISA | 3F | 105 | |
| 06 PORTB | 00 | 86 TRISB | 7F | 106 PORTE | 00 |
| 07 PORTC | C1 | 87 TRISC | FF | 107 | |
| 08 PORTD | 20 | 88 TRISD | 00 | 108 | |
| 09 PORTE | 00 | 89 TRISE | 07 | 109 | |
| 0A PCLATCH | 01 | 8A PCLATCH | 01 | 10A PCLATCH | 18 |
| 0B INTCON | 01 | 8B INTCON | 01 | 10B INTCON | 01 |
| 0C PIR1 | 00 | 8C PIE1 | 00 | 10C EEDATA | 00 |
| 0D PIR2 | 00 | 8D PIE2 | 00 | 10D EEAR | 00 |
| 0E TMR1L | DF | 8E PCON | 00 | 10E EEDATH | 11 |
| 0F TMR1H | 5B | 8F | | 10F EEARH | 00 |
| 10 T1CON | 00 | 90 | | BANK 2 | |
| 11 TMR2 | 00 | 91 SSPCON2 | 00 | REGISTER VALUE | |
| 12 T2CON | 00 | 92 PR2 | FF | 181 OPTION | FF |
| 13 SSPBUF | FF | 93 SSPADD | 00 | 182 PCL | 13 |
| 14 SSPCON | 00 | 94 SSPSTAT | 00 | 183 STATUS | 1B |
| 15 CCP1L | A1 | 95 | | 184 FSR | 00 |
| 16 CCP1H | 02 | 96 | | 185 | |
| 17 CCP1CON | 00 | 97 | | 186 TRISB | 7F |
| 18 RCSTA | 00 | 98 TXSTA | 02 | 187 | |
| 19 TXREG | 00 | 99 SPBRG | 00 | 188 | |
| 1A RCREG | 00 | 9A | | 189 | |
| 1B CCP2L | 22 | 9B | | 18A PCLATCH | 18 |
| 1C CCP2H | 00 | 9C | | 18B INTCON | 01 |
| 1D CCP2CON | 00 | 9D | | 18C EECON1 | 88 |
| 1E ADRESH | 00 | 9E ADRESL | 00 | 18D EECON2 | 00 |
| 1F ADCON0 | 00 | 9F ADCON1 | 00 | BANK 3 | |

Figure 34 : La fenêtre Special Function Register (SFR)

Le contenu de cette fenêtre est également mis à jour après les opérations suivantes:

- Après une opération en une seule étape
- Lorsque le programme utilisateur est arrêté
- Un point d'arrêt est rencontré

La fenêtre Source

La fenêtre Source comme son nom l'indique est utilisée pour afficher le code source de l'utilisateur. Le code source est chargé lorsque l'utilisateur charge le fichier hex pour la programmation (à partir de la "Fichier" - menu 'Load')

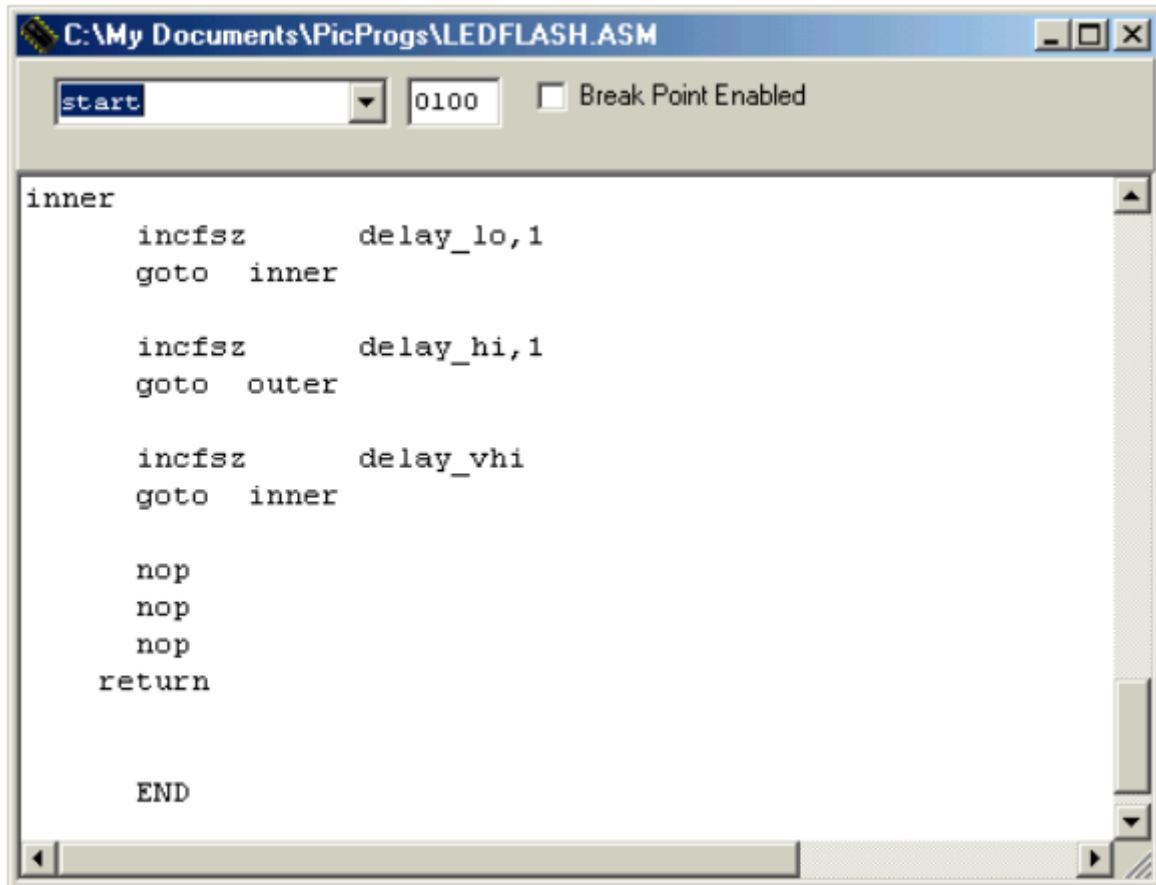


Figure 35: La fenêtre Source

La fenêtre source défilera vers, et mettre en évidence la ligne courante après les suivants événements:

- Après une opération en une seule étape
- Lorsque le programme utilisateur est arrêté
- Un point d'arrêt est rencontré

La fenêtre Montres

Cela permet à l'utilisateur de mettre en place un certain nombre de montres, dont le contenu est mis à jour après les événements suivants:

- Après une opération en une seule étape
- Lorsque le programme utilisateur est arrêté

- Un point d'arrêt est rencontré

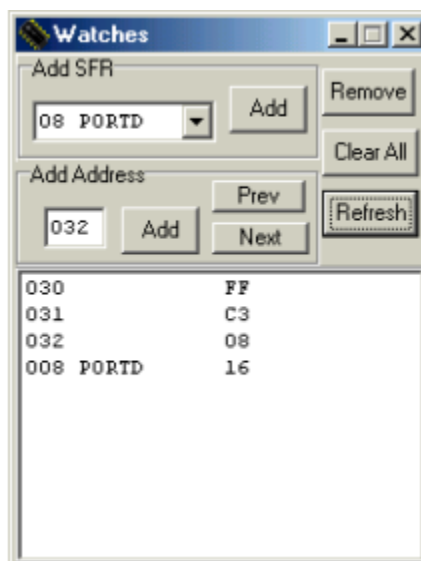


Figure 36 : La fenêtre Montres

III.4- Développements et système de formation ABITEC

« DEVELOPMENT AND TRAINING SYSTEM »

III.4.1- Définition :

Le 877-DATS développement et le système de formation permet à des programmes à développer, téléchargé et de débogage en utilisant alors (ICD) installation en circuit débogage fourni par le microcontrôleur 16F877. Lorsque la ICD est utilisé un certain nombre de «flottant» fenêtres sont disponibles pour présenter les informations de microcontrôleur tandis programmes de débogage.

Cette carte fonctionne comme un programmeur pour les microcontrôleurs 16F877 et les autres microcontrôleurs par la liaison ICSP.

Le composant principal de la programmation est PIC microcontrôleur 16F84.

Cette carte est utilisée pour les buts pédagogiques et peut également être lié à la carte des applications.

La communication entre le PC et programmeur par RS232.

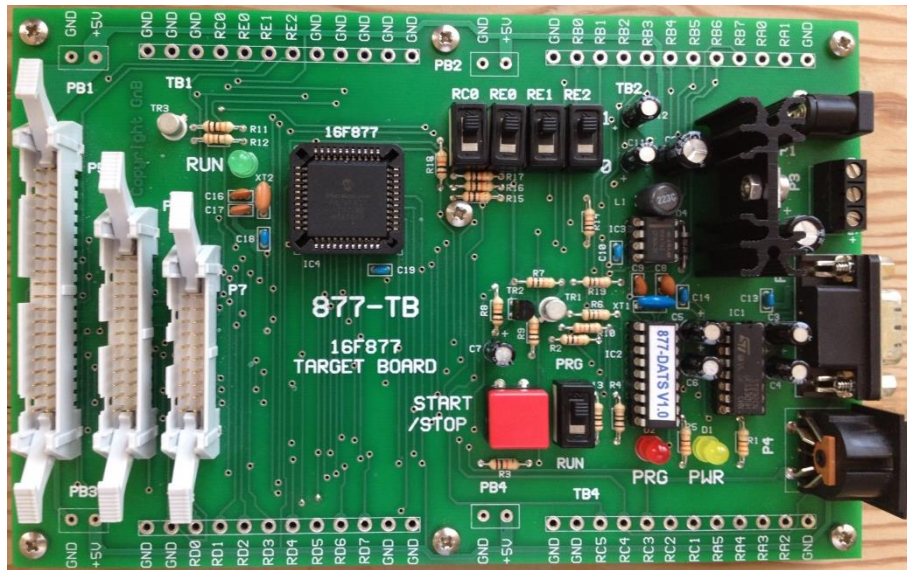


Figure 37 : Développements et système de formation ABITEC

III.4.2- Le circuit

Patrie 1

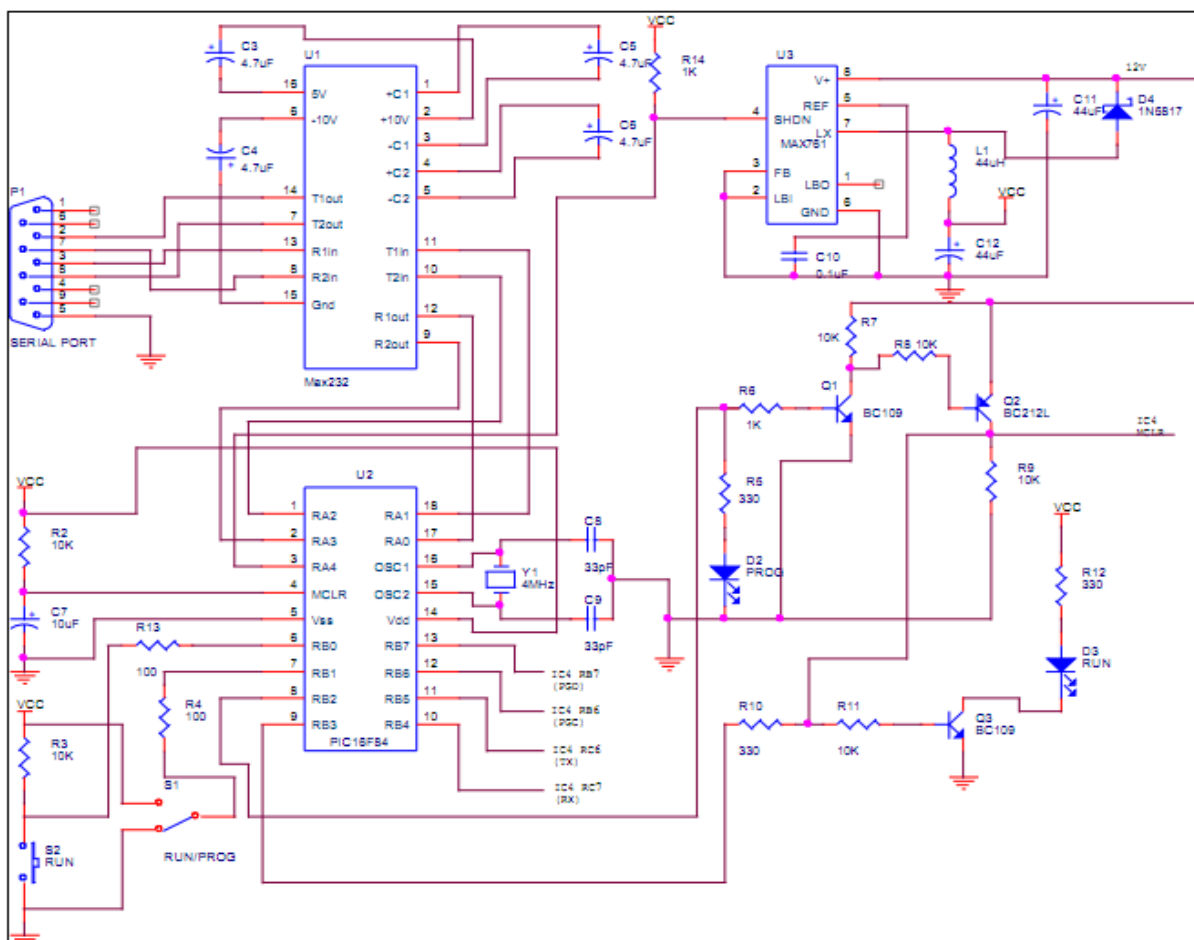


Figure 38 : Le circuit de Développements et système de formation ABITEC / P01

Partie 2 :

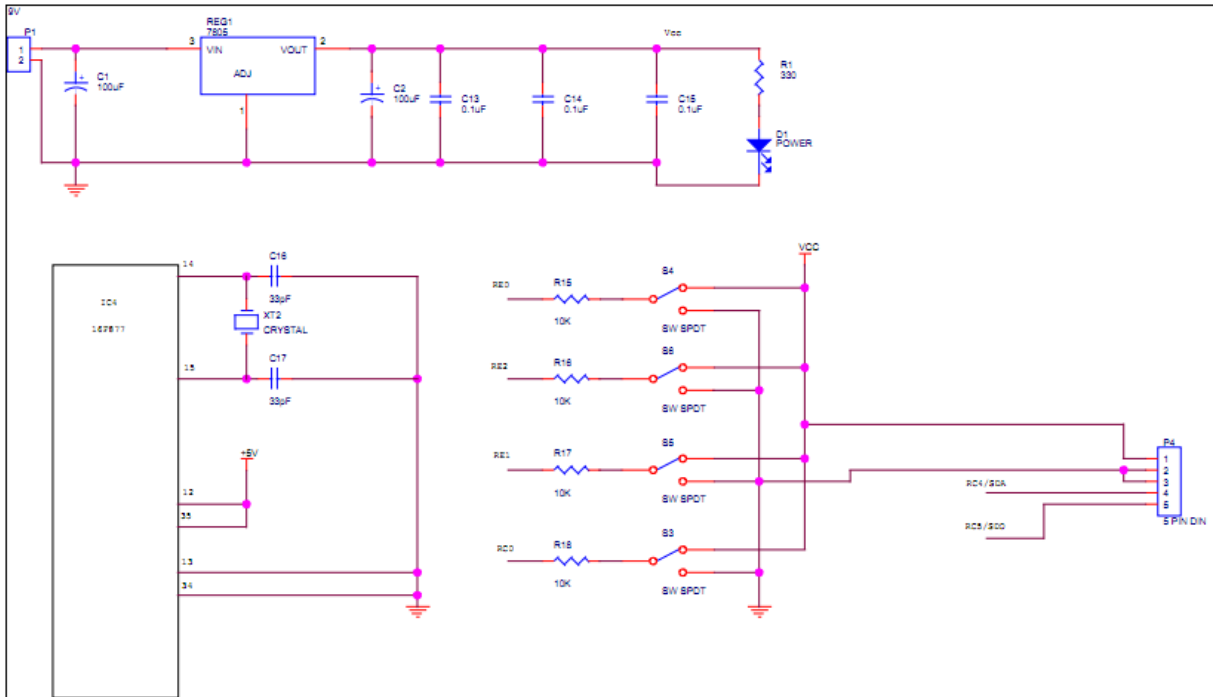


Figure 39 : Le circuit de Développements et système de formation ABITEC / P02

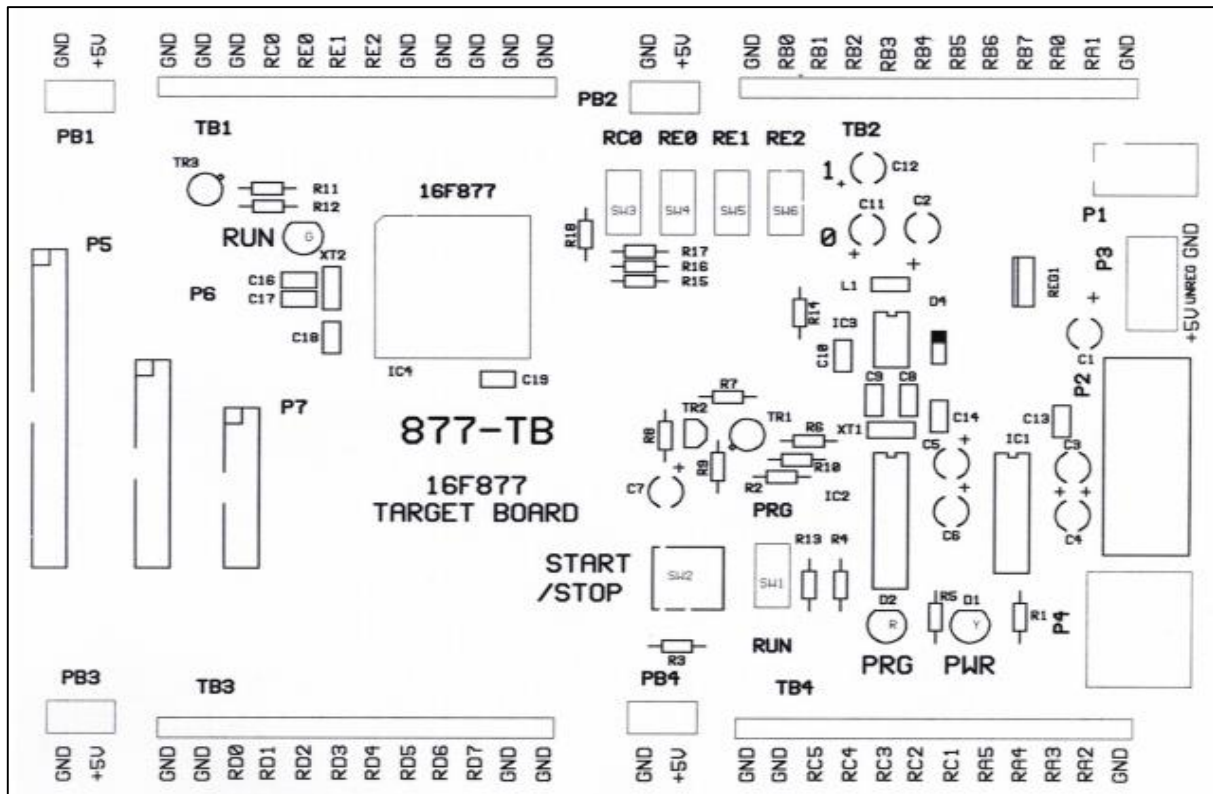


Figure 40 : les noms de chaque composant dans la carte de ABITEC

III.4.3- Les principales composantes

- PIC 16F84
- PIC 16F877
- Serial port RS
- MAX232
- MAX761
- Régulateur 7805
- Résistance, cap, LEDs, switches, transistorsetc

III.4.4- Le principe de fonctionnement

Le fonctionnement de l'unité est commandée par deux interrupteurs, interrupteur à glissière S1 (PRG / RUN) et bouton-poussoir S2 (START / STOP)

Commutateur PRG / RUN doit être dans la position PRG. Pour exécuter le programme, faites glisser le PRG / RUN Interrupteur à la position RUN, appuyant désormais le commutateur STOP/START bascule le réinitialiser ligne au PIC16F877. L'état actuel est indiqué par le RUN (vert) LED.

La carte fonctionne comme programmeur « PRG » si la tension appliquée au pin RB1 « 16F84 » est 13V, et fonctionne mode RUN si la tension appliquée au pin RB1 est 5V.

III.4.5- Les avantages de la Platform ABITEC :

- Le microcontrôleur « 16F877 » est le programmeur dans même carte
- Le bloc utilisé pour alimente la carte dans la même carte
- Passibilité pour communique le microcontrôleur 16F877 et la carte d'application par le connecteur entrée/sortie « 26 Way Connector P6, 40 Way Connector P5, 20 Way Connector P7»

III.4.6- Les inconvénients :

- La communication entre PC et la carte par la liaison RS232 et pas de liaison USB.
- Problème de communication entre PC et microcontrôleur 16F877 « on va réaliser un solution de cette problème »

III.4.7- Alimentation de la carte

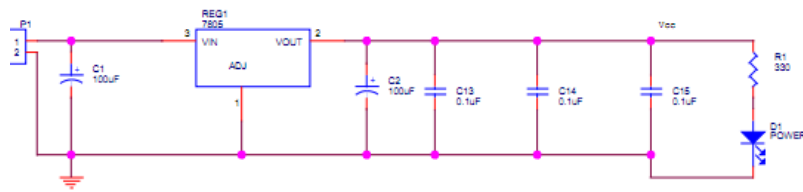


Figure 41 : circuit d'alimentation de la carte

Le bloc utilisé un régulateur de tension 5V (un LM2937 5.0 de National Semiconductor ou 7805) pour alimenter la carte

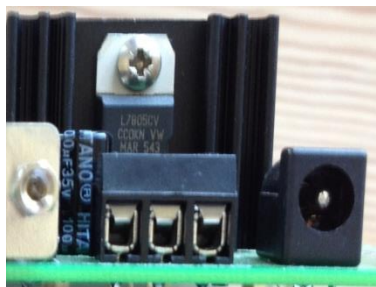


Figure 42 : Le bloc utilisé pour alimenter la carte

III.4.8- La carte d'application

La carte est un ensemble d'applications simples, connecté directement avec les 40 pins de PIC16F877 par les connecteurs.

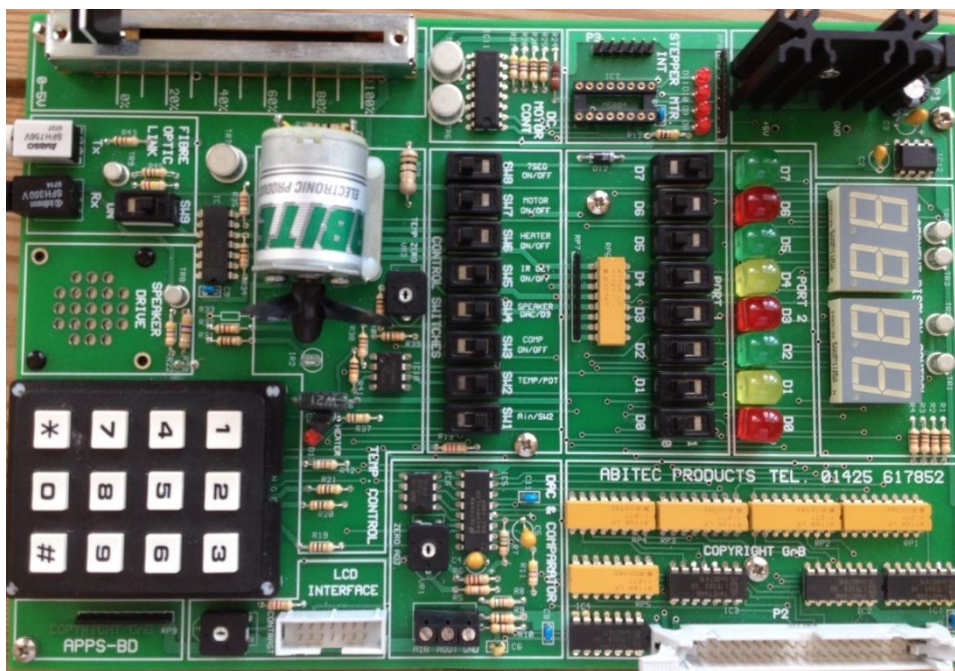


Figure 43 : La carte d'application

III.4.8.1- Les applications

- 8 LEDs
- 8 Switchs
- 2 afficheurs à sept segments
- Moteur courant continu DC
- Capteur de température à semi-conducteur.
- Circuit de commande de chauffage
- Un clavier 3 x 4 de la matrice de type téléphonique standard.
- L'émetteur et le récepteur à fibre optique.
- Une Digital 8 bits analogique à Convertisseur (DAC).
- Curseur potentiomètre fournissant une tension analogique variable.
- Haut-parleur commandé soit de former un signal numérique ou à partir de la sortie du DAC.
- Interface LCD
- Circuit de commande de moteur pas à pas.

III.4.8.2- Les connecteurs

La carte d'applications est connectée au système de contrôle par un connecteur IDC 40.



Figure 44 : un connecteur IDC 40

III.4.9- La communication entre PC et la carte

Le signal électrique représentatif des codes ASCII du message à transmettre est généré par l'ordinateur ou l'équipement émetteur sur la broche TXD (également appelée Tx) de son connecteur RS232. Du côté récepteur le signal est reçu sur la broche RXD (Rx). Pour communiquer sous forme de « chat ».

III.4.9.1- Le problème

Le problème est le manque de communication entre l'ordinateur et microcontrôleurs PIC16F877.

Il y a un problème dans les broches TX et RX entre l'ordinateur et microcontrôleurs, Pour cela, nous allons résoudre ce problème

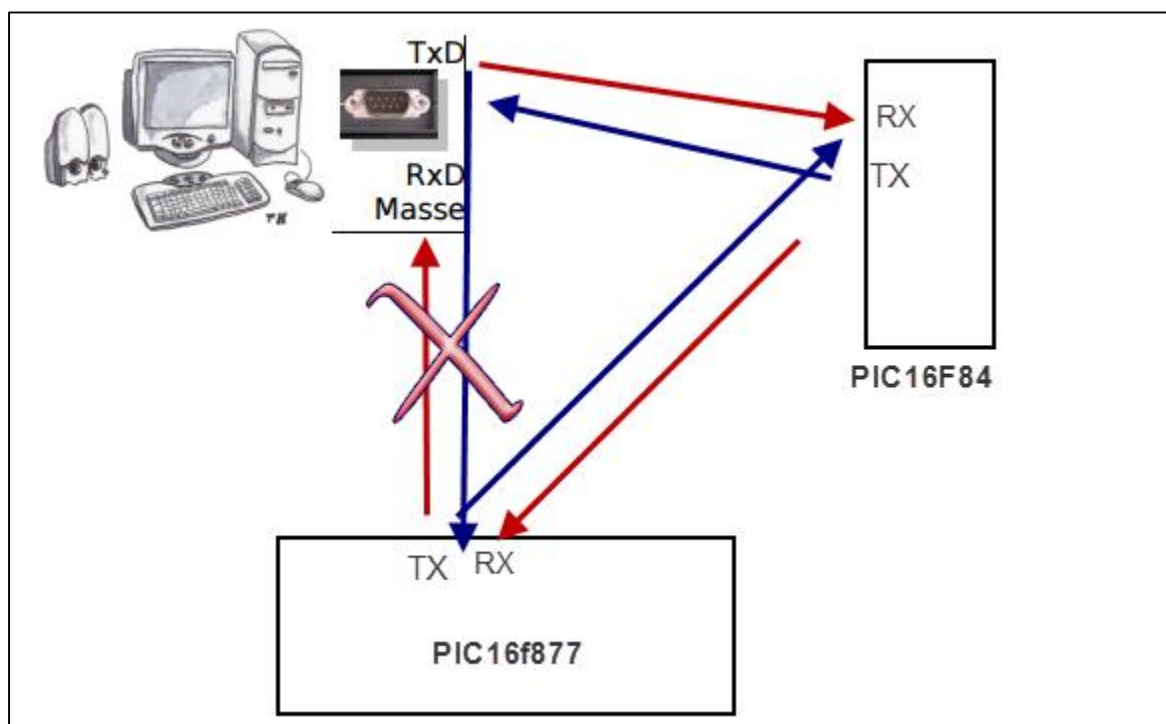


Figure 45 : La communication entre PC et la carte

III.4.10- Réalisation de la carte de communication entre le PC et PIC16F877

Pour résoudre ce problème, nous allons réaliser la carte

III.4.10.1- L'idée de la carte

Pour réaliser cette carte nous allons en utilisant six interrupteurs :

- Four de travail en même temps pour commander le TX et RX entre le PC et PIC16F84 / TX et RX entre PIC16F84 et PIC16F877 « mode programmeur »
- Deux de travail en même temps pour commander le TX et RX entre le PC et PIC16F877 « mode lire des données »

III.4.10.2- Modèle d'interrupteur

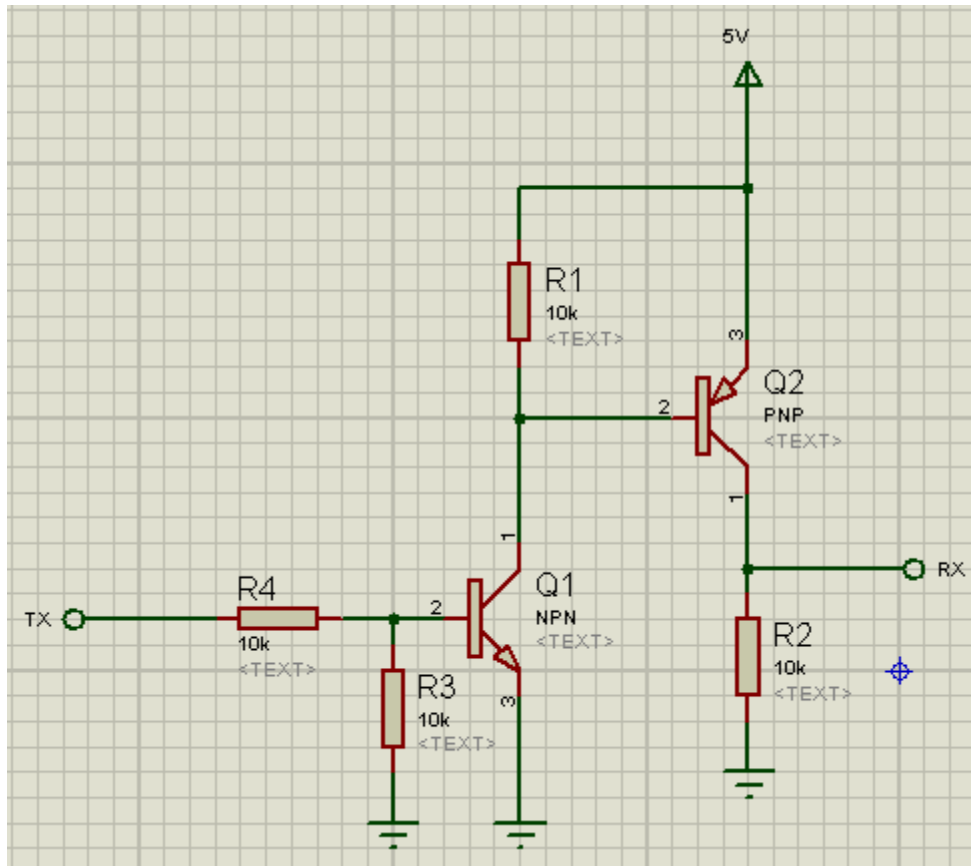


Figure 46 : Modèle d'interrupteur

Dans ce modèle utilise deux types de transistor NPN et PNP.

Le signal de transmetteur TX commende transistor NPN, et transistor NPN commende transistor PNP et le signal généré par circuit aller à émetteur RX.

III.4.10.3- Simulation de circuit

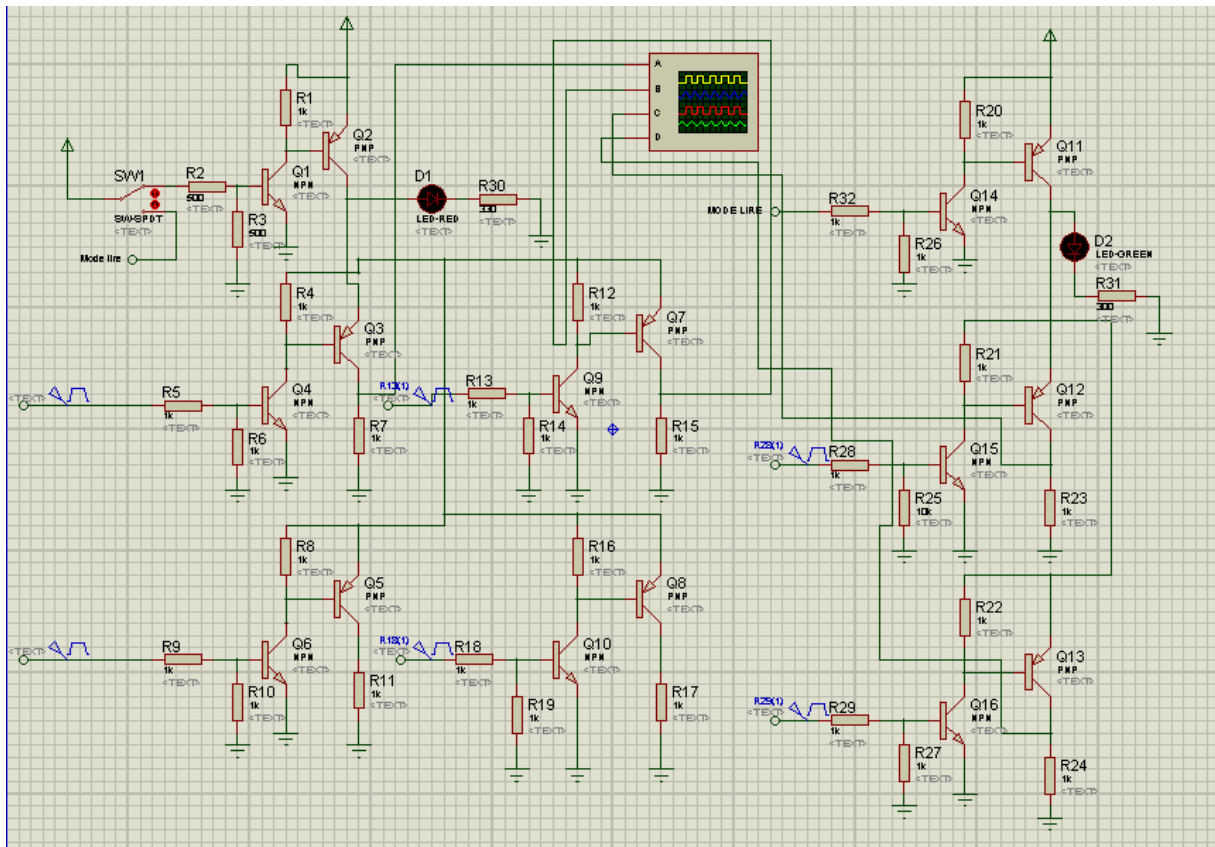
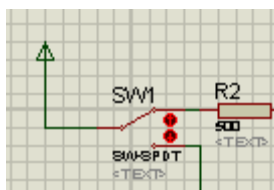


Figure 47 : schéma de Simulation de circuit

1^{er} cas « mode programmeur »



Switch dans l'état 1, en appliqué un signal carré dans tous les pins de transmettre

- Et le LED est rouge

Est le résultat dans les pins de récepteur :

- 1- TX PC vers RX PIC16F84
- 2- TX PC16F84 vers RX PC
- 3- TX PC vers RX PIC16F877
- 4- TX PIC16F877 vers RX PC

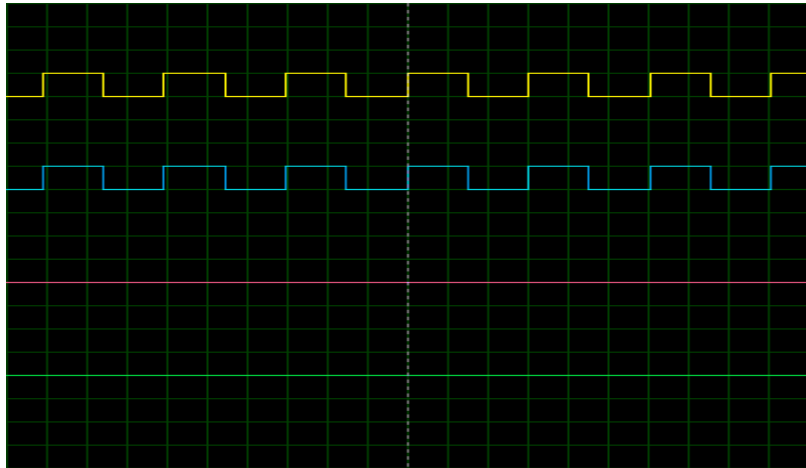
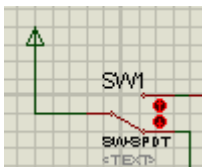


Figure 48 : résultat de simulation

2^{ème} cas « mode Lire les données »



Switch dans l'état 2, en appliqué un signal carrée dans tous les pins de transmettre

- Et le LED est vert

Est le résultat dans les pins de récepteur :

- 5- TX PC vers RX PIC16F84
- 6- TX PIC16F84 vers RX PC
- 7- TX PC vers RX PIC16F877
- 8- TX PIC16F877 vers RX PC

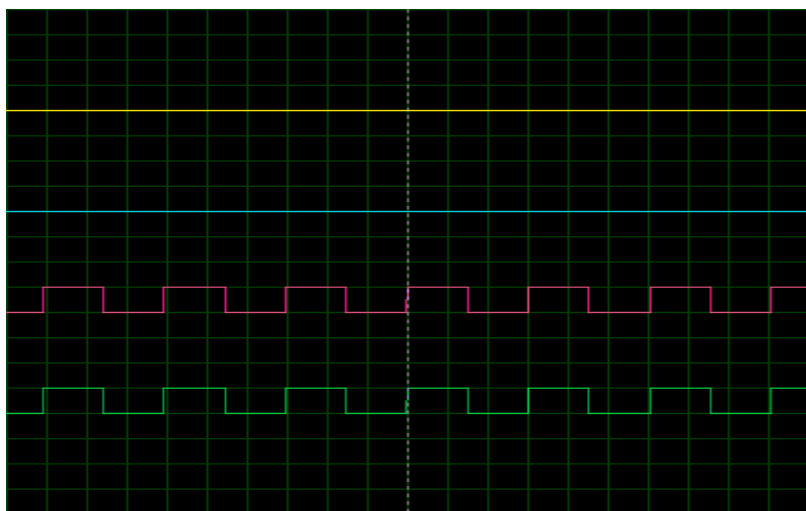


Figure 49 : résultat de simulation

III.4.10.4- Schéma final

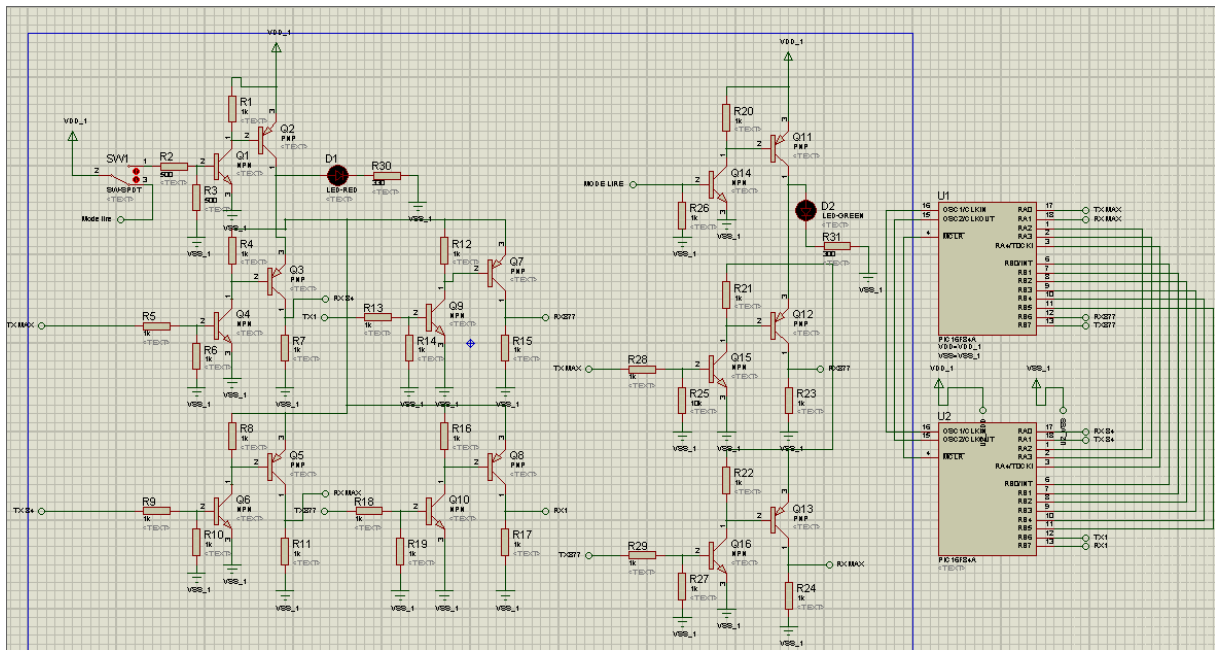


Figure 50 : Schéma final de circuit

Les composants :

R1, R4, R8, R12, R16, R20, R21, R22 : résistances de 1K

R2, R5, R9, R13, R18, R32, R28, R27 : résistances de 750 Ohm

R3, R6, R10, R14, R19, R26, R25, R27 : résistances de 1.2K

R7, R11, R15, R17, R23, R24 : résistances de 2K

R30, R31 : résistances de 330 Ohm

Q1, Q4, Q6, Q9, Q10, Q14, Q15, Q16 : transistors BC557 (NPN)

Q2, Q3, Q5, Q7, Q8, Q11, Q12, Q13 : transistors BC547 (PNP)

D1, D2 : LEDs

U1 : support de PIC16F84

U2 : PIC16F84 (PIC d'APITEC)

III.4.10.5- Le circuit imprimé

Voici une petite liste des composants dont vous pourriez avoir besoin si vous souhaitez fabriquer votre propre circuit imprimé :

- Une insoleuse UV
- Plaques de circuit imprimés
- Du nettoyant dégraissant
- Bacs plastique
- Perceuse

III.4.10.6- Schéma de circuit sous ARES

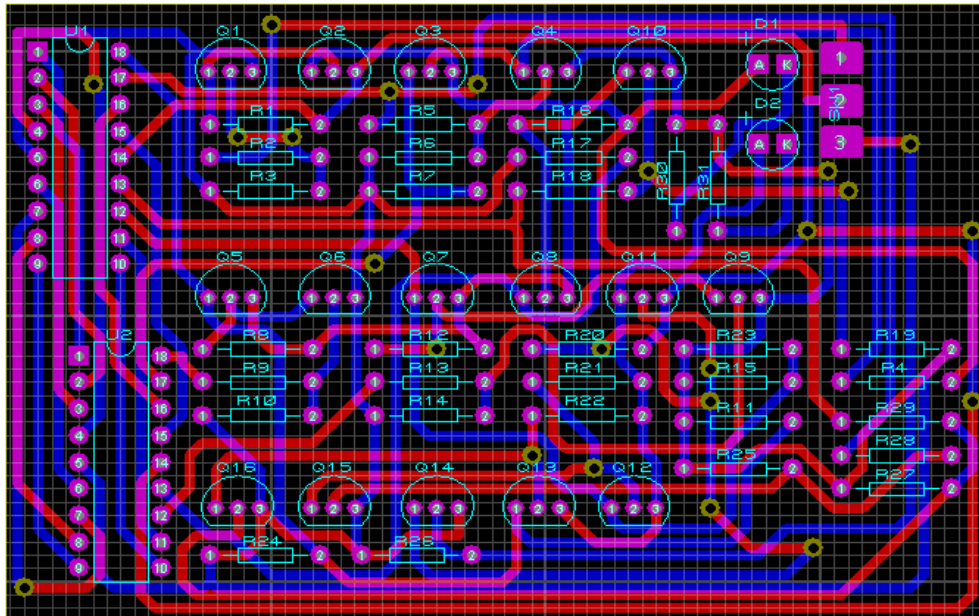


Figure 51 : Schéma de circuit sous ARES

III.4.10.7- Le typons de la carte

- Top copper

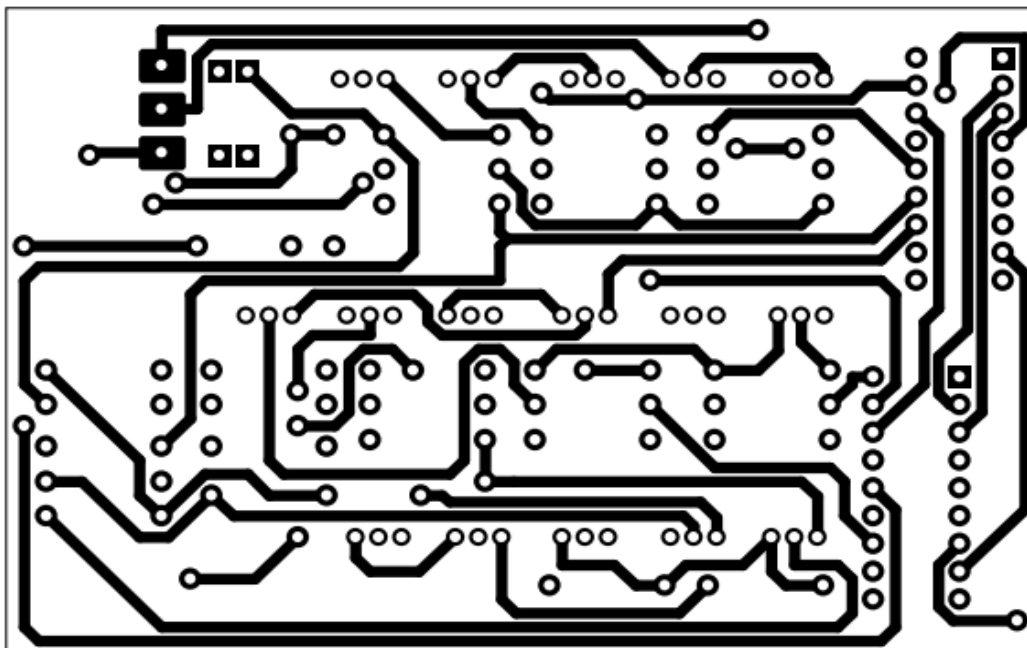


Figure 52 : typon top copper de la carte

- Bottom copper

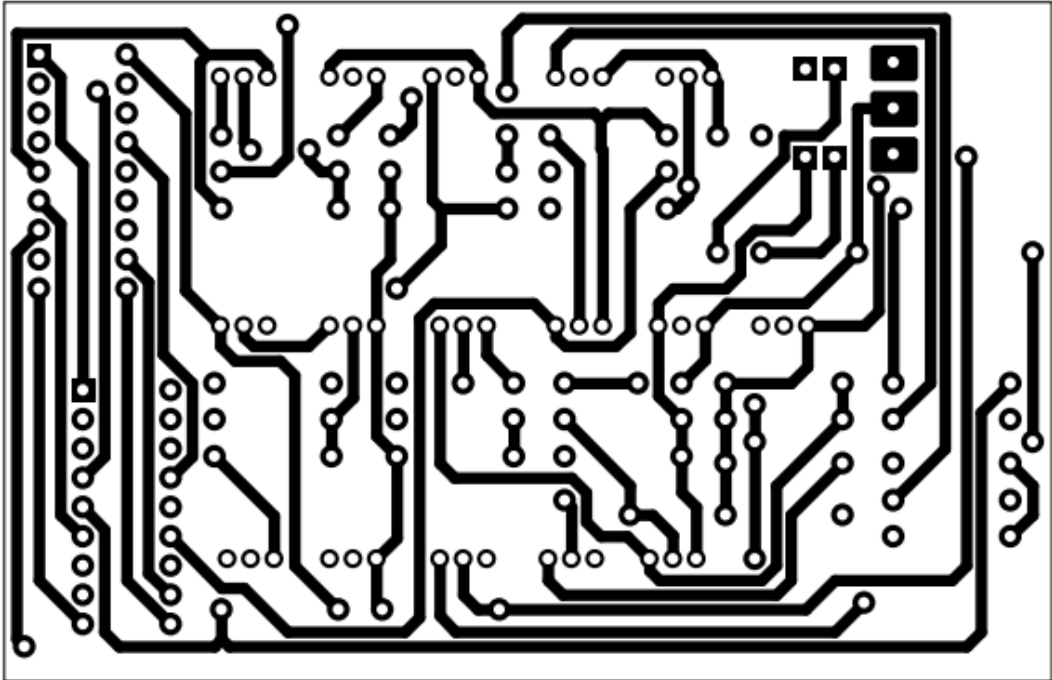


Figure 53 : typon bottom copper de la carte

III.4.11- Réalisation sur carte d’essai

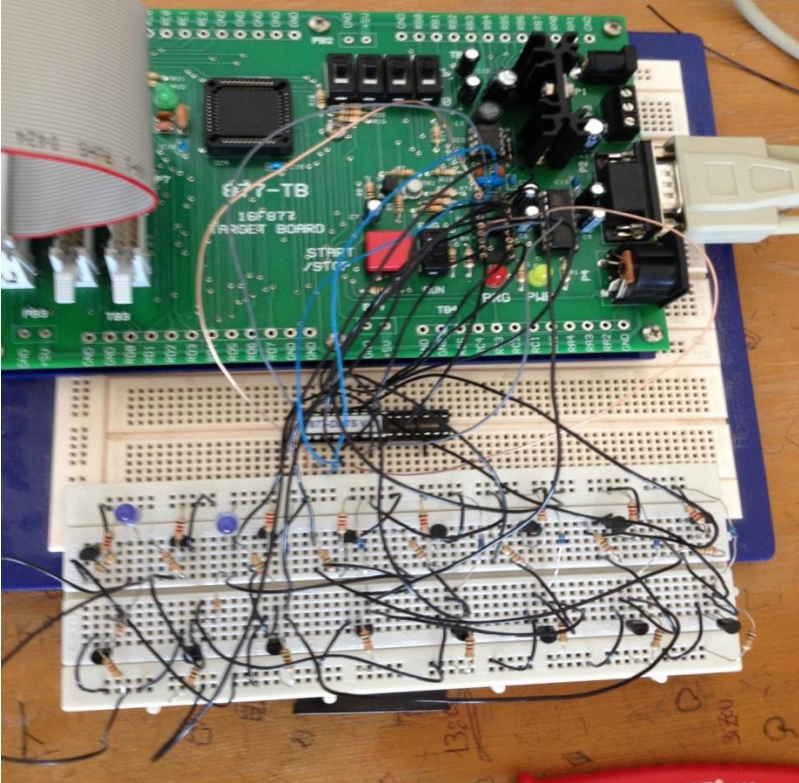


Figure 54 : Réalisation sur carte d’essai

Les composants :

- 8 transistors PNP (BC557)
- 8 transistors NPN (BC547)
- 32 résistance (8 res 1.2k, 8 res 1k, 8 res 750 ohm, 6 res 2k, 2 res 330 ohm)
- 2 LEDs

III.4.12- Comment réaliser

III.4.12.1- L'impression du typon de la carte

L'idéal est d'imprimer avec une imprimante à jet d'encre sur un papier transparent spécial, Il faut régler l'impression sur une qualité pour avoir un maximum d'encre sur le papier.

Comme il est montré dans la figure ci-dessous, notre carte électronique est composée de deux côtés, un côté de soudure et l'autre côté pour l'emplacement des composants.

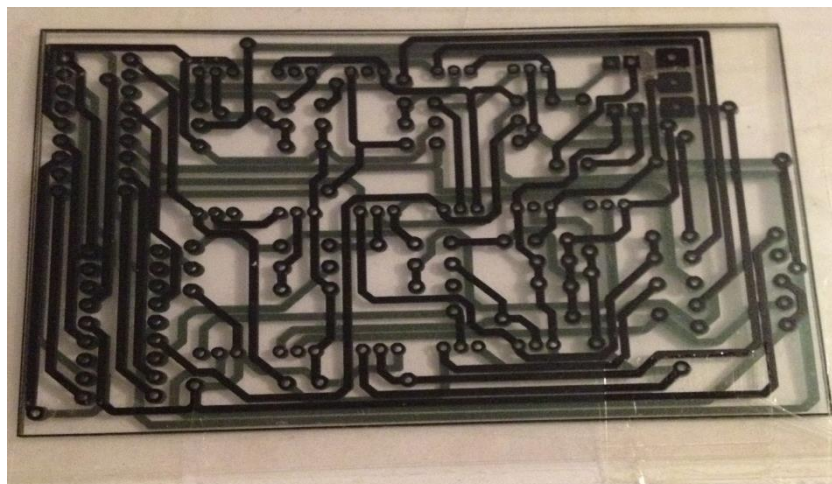


Figure 55 : Vu de dessus du typon sur la feuille acétates.

III.4.12.2- L'insolation sous lumière ultra-violet :

Nous disposons au niveau du laboratoire de développement des circuits imprimés d'une insoleuse UV deux faces dont le temps d'insolation est réglable, Ce temps est très important, s'il est très long les rayons UV passeront à travers les zones noires du typon, et inversement, s'il est très court, on aura du mal à révéler la carte, et donc le révélateur ne pourra pas décaper la couche photosensible.

Pour une bonne insolation de la couche photosensible, on suit les étapes suivantes :

- On découpe la plaque d'époxy à la taille du typon.

- On positionne le typon dans le bon sens sur la vitre de l'insoleuse (vue côté opposant).
- On retire la pellicule protectrice de la plaque.
- On pose cette plaque avec le côté de la couche photosensible sur le typon dans l'insoleuse.
- On fait l'insolation environ de 30 secondes à 3 minutes.



Figure 56 : machine d'insolation

Dès qu'on termine cette étape, on passe au révélateur pour dissoudre les zones de la couche photosensible qui a déjà été exposée à la lumière UV.

III.4.12.3- Développement de la couche photosensible

Le développement de cette couche permet de révéler les pistes de cuivre non utilisées (qui ont été touchées par la lumière UV).

Pour cela, On doit maintenant plonger la plaque qui vient d'être insolée dans une cuvette en plastique contenant le révélateur, une fois la plaque est révélée, elle est sortie du bac et rincée soigneusement à l'eau.

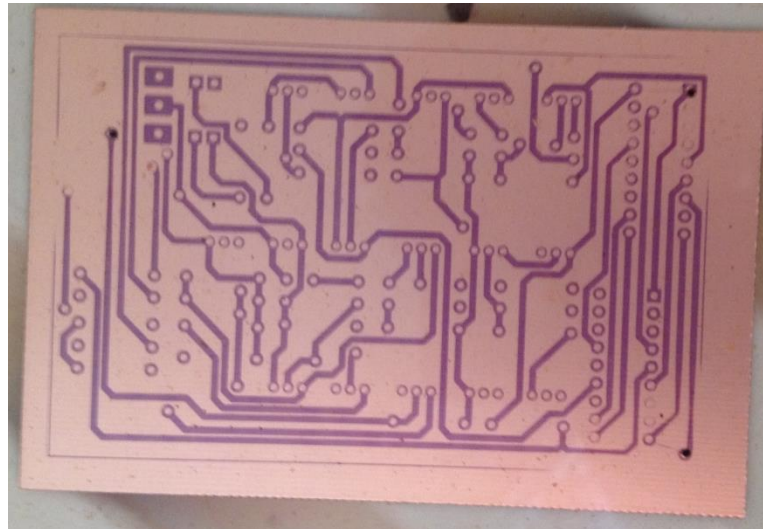


Figure 57 : La carte dans le révélateur

Après l'apparition de la couche de Cuivre autour des pistes protégées, il faut maintenant passer à l'opération gravure.

III.4.12.4- La gravure

La gravure consiste à plonger la plaque dans un bain d'acide. Elle a pour but de faire le décapage de la couche de cuivre révélée, seules les pistes protégées resteront. (Il faut faire attention pendant l'utilisation de l'acide).

Le plongement de la plaque dans le bain d'acide durera de 5 à 30min tout en suivant la gravure. La chaleur 400°C et l'agitation permettent l'accélération de cette partie. Lorsqu'on nettoie le cuivre non utilisé on sort le circuit et on le rince avec de l'eau.

III.4.12.5- Décapant

C'est la dernière étape à réaliser, le déroulement de cette étape est basé sur l'utilisation d'un dissolvant (l'acétone), il sert à frotter la couche résultante, qui couvre les pistes de cuivre. Le but est d'obtenir un circuit avec des pistes bien nettes et sans aspérités.

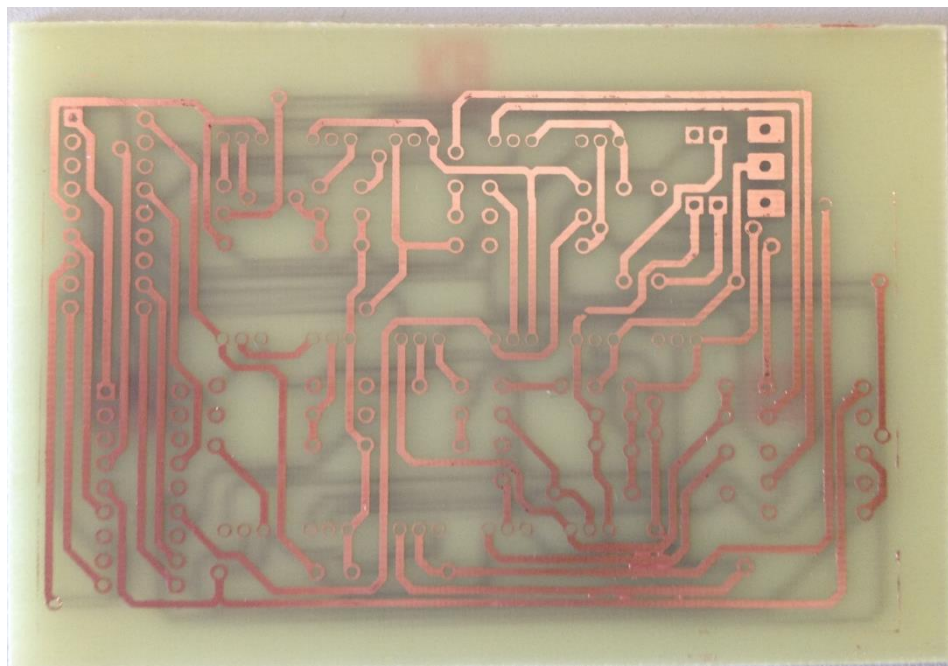


Figure 58 : Vue de dessous de la carte après le décapant.

Après toutes ces étapes, et dès que notre circuit imprimé est terminé, il nous reste plus qu'à percer la carte et souder les composants pour concevoir notre carte électronique. L'opération de perçage est l'ultime étape dans la réalisation de notre circuit imprimé, elle consiste à percer les pastilles correspondantes à l'emplacement des pâtes des composants.

Une fois toutes les pastilles percées au bon diamètre on va pouvoir souder les composants sur la plaque en se basant sur le schéma (ARES 3D), on utilise le fer à souder et l'étain c'est un métal qui se manipule facilement et sa température de fusion est assez basse.

III.4.13- Les résultats

La carte fonctionne bien, mais il y a un problème :

Grande consommation de courant et a conduit à la plus chaude de régulateur (7805), il y a possibilité court-circuit dans la carte.

Pour cela, je vais faire une deuxième carte à l'aide des switch (interrupteurs).

III.4.14- La deuxième carte

III.4.14.1- Principe

Commende les laissions entre PC et PIC16F84 et PIC16F877 par les interrupteurs manuel (switch).

Cas 1 :

Pour programme PIC16F877

Les switches 1, 2, 3,4 fermes et 5,6 ouvert.

Cas 2 :

Pour communiqué PC avec PIC16F877

Les switches 1, 2, 3,4 ouvert et 5,6 fermes.

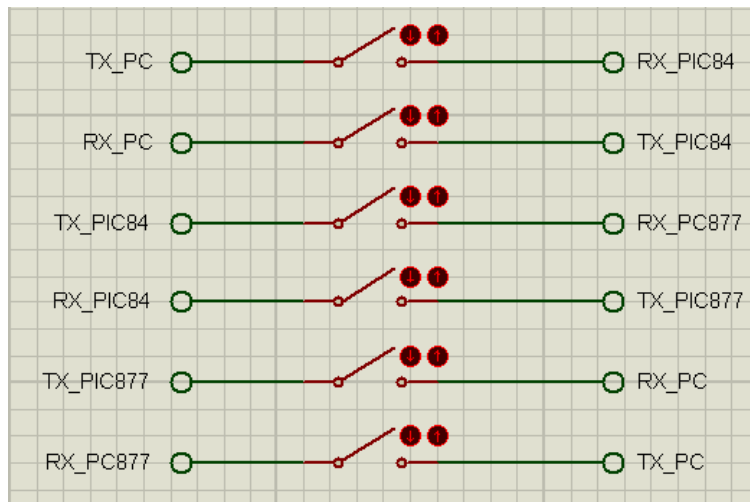
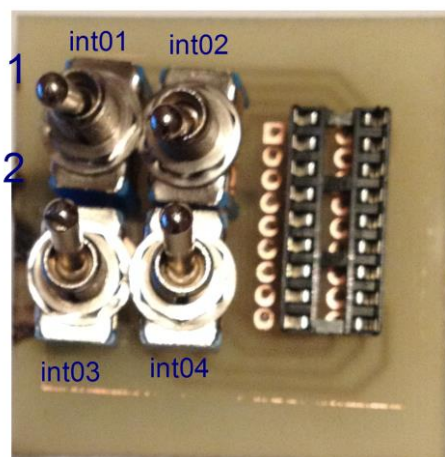


Figure 59 : schema de Principe de la carte

Les cas d'utilisation



| Mode | Int 01 | Int 02 | Int 03 | Int 04 |
|------------------------|--------|--------|--------|--------|
| Mode de programmation | 2 | 2 | 1 | 1 |
| Mode de lire de donnes | 1 | 1 | 2 | 2 |

III.4.14.2- Schéma

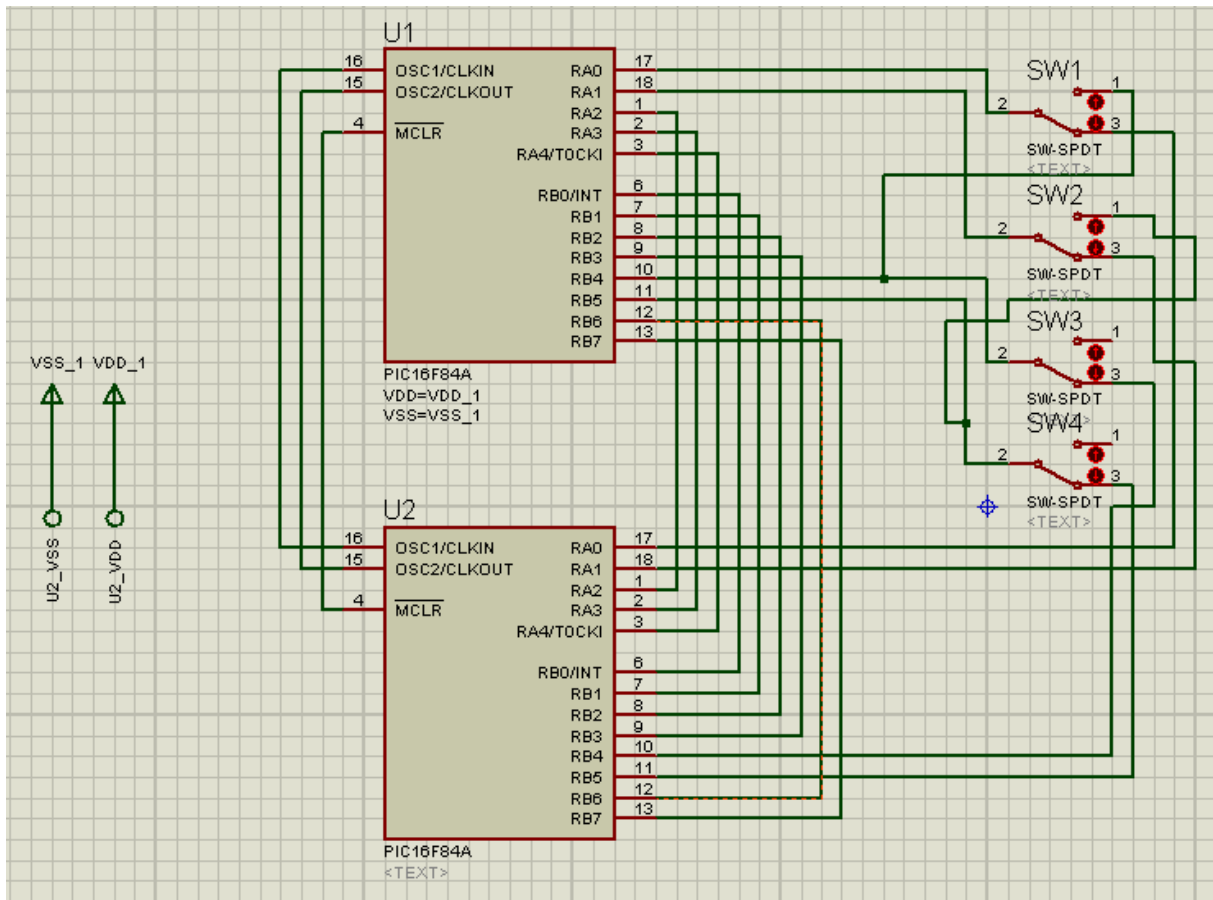


Figure 60 : Schéma de circuit de communication manuel

Pour minimiser le nombre des interrupteurs on va utiliser les switch de deux position

III.4.14.3- Typon de la carte

Top et bottom copper

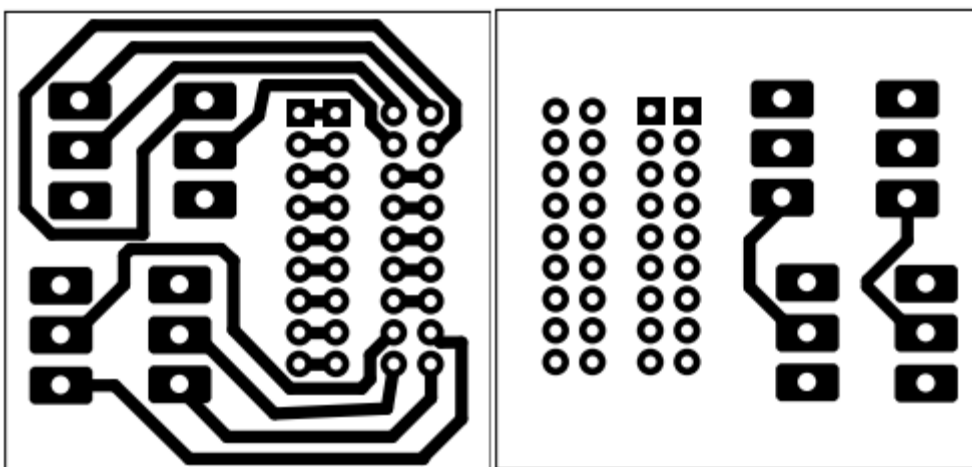


Figure 61 : Top et bottom copper

III.4.14.4- Schéma de circuit sous ARES

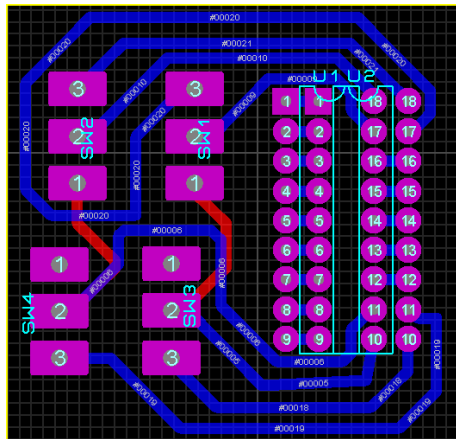


Figure 62 : Schéma de circuit sous ARES

III.4.14.5- La carte finale

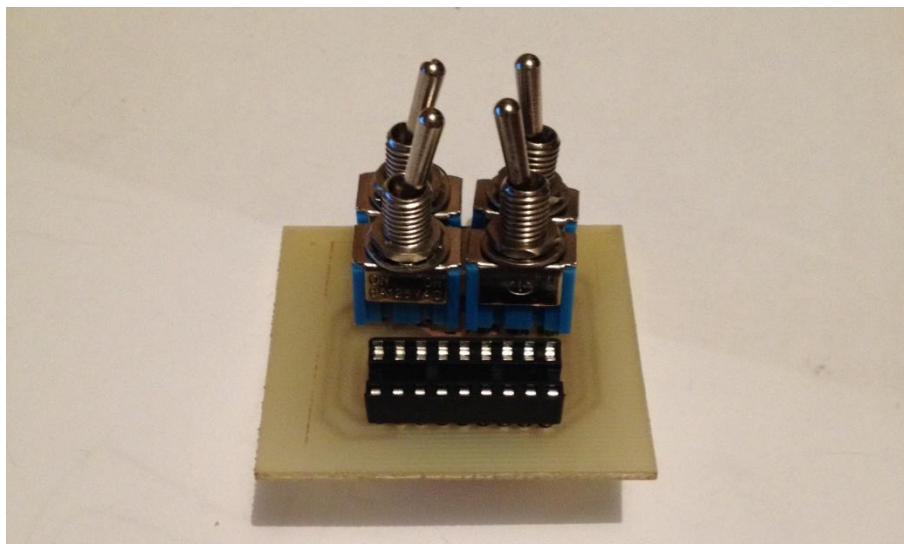


Figure 63 : La carte finale

III.4.15- Les avantages

1^{er} carte :

- Les deux fonctionnent est commandé par un seul switch.
- Pas de possibilité de court-circuit.

2^{eme} carte :

- Pas beaucoup des composants.
- Pas de problème, le plus chaud de régulateur.
- La carte est très simple.

III.4.16- Les inconvénients

1^{er} carte :

- problème, le plus chaud de régulateur.
- Beaucoup des composants.
- La carte est grande.

2^{eme} carte :

- La commande manuelle.
- Beaucoup de switches.

III.4.17- Application avec le moteur pas à pas

Le moteur pas à pas est utilisé dans plusieurs domaines d'application, comme par exemple l'automobile, le médical, le chauffage, la ventilation, la climatisation, le textile, ...etc. Ces moteurs sont conçus pour fonctionner en boucle ouverte. Comme son nom l'indique le moteur pas à pas exécute des mouvements de pas quand des impulsions de tension sont appliquées à ses phases. D'autre part, le moteur pas à pas utilise seulement 50% de son couple maximal, le reste est réservé pour vaincre toute variation de charge. La simplicité de cette commande entraîne plusieurs problèmes dans le domaine où une grande précision de positionnement est exigée. Si les mouvements ou la variation de la charge sont rapides, le moteur peut perdre des pas.

III.4.17.1- Pourquoi des moteurs Pas à Pas

• 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 de 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.

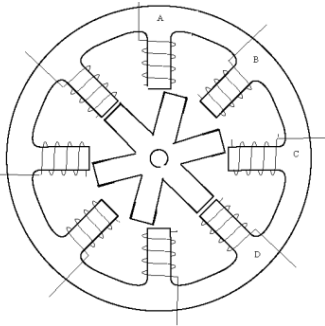
III.4.17.2- Les différents types de Pas à Pas

- Trois catégories de moteurs:

A réluctance variable:

On ne sent pas les pas.

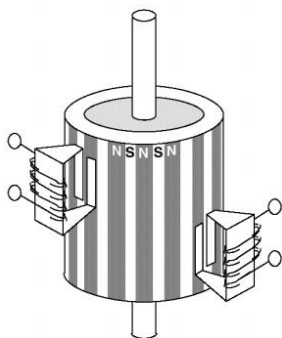
A caractéristique électrique identique, un tel moteur est moins puissant, mais plus rapide que les moteurs à aimant permanent. Sans doute les plus anciens.



A aimants permanents:

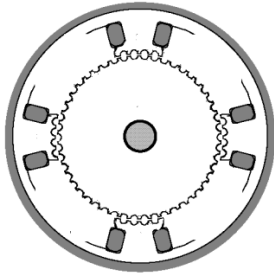
On sent les pas.

Ce sont des moteurs à faible coût de revient, et de résolution moyenne (jusqu' à 100 pas/tour).



Les hybrides:

Ces moteurs combinent les 2 technologies précédentes, et sont plus chers. Leur intérêt réside dans un meilleur couple, une vitesse plus élevée, et une résolution de 100 à 400 pas/tour.



– Les moteurs les plus courants sont ceux à aimants permanents et les hybrides.

III.4.17.3- Unipolaires et Bipolaires

• Les bipolaires:

– 4 fils, avec 2 bobines indépendantes (appelé « 2 phases »).

• Les unipolaires:

– 5 fils, 1 commun aux 4 bobines, plus un par bobine.

– 6 fils, 2 bobines avec point milieu (parfois appelé « 2 phases »).

– 8 fils, 4 bobines indépendantes, également appelé universel.

– Par abus de langage, ces moteurs sont tous appelés « 4 phases ». Ce sont les moteurs pour GP99 et BB00.

• Et les moins classiques:

– 4 ou 6 fils avec un point commun à toutes les bobines. Ce sont des moteurs unipolaires avec 3 ou 5 bobines. Ces moteurs sont appelés 3 ou 5 phases.

III.4.17.4- L'interface avec langage C#

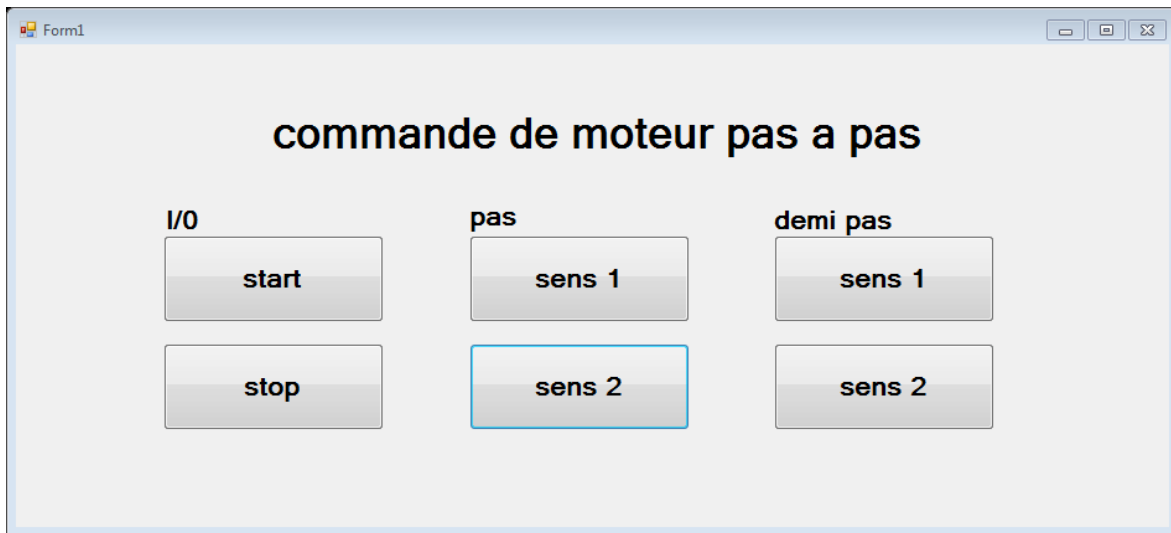


Figure 64 : L'interface avec langage C#

III.4.17.5- Programme d'interface avec langage C#

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        SerialPort port=new SerialPort( "COM1",9600,Parity.None,8,StopBits.One);
        public Form1()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            port.Open();
            port.Write("a");
            port.Close();
        }

        private void button5_Click(object sender, EventArgs e)
        {
            port.Open();
            port.Write("b");
            port.Close();
        }
    }
}

```

```

}

private void button1_Click(object sender, EventArgs e)
{
    port.Open();
    port.Write("c");
    port.Close();
}

private void button6_Click(object sender, EventArgs e)
{
    port.Open();
    port.Write("d");
    port.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    port.Open();
    port.Write("e");
    port.Close();
}

private void button4_Click(object sender, EventArgs e)
{
    port.Open();
    port.Write("f");
    port.Close();
}
}
}
}
}
}
}

```

III.4.17.6- Circuit

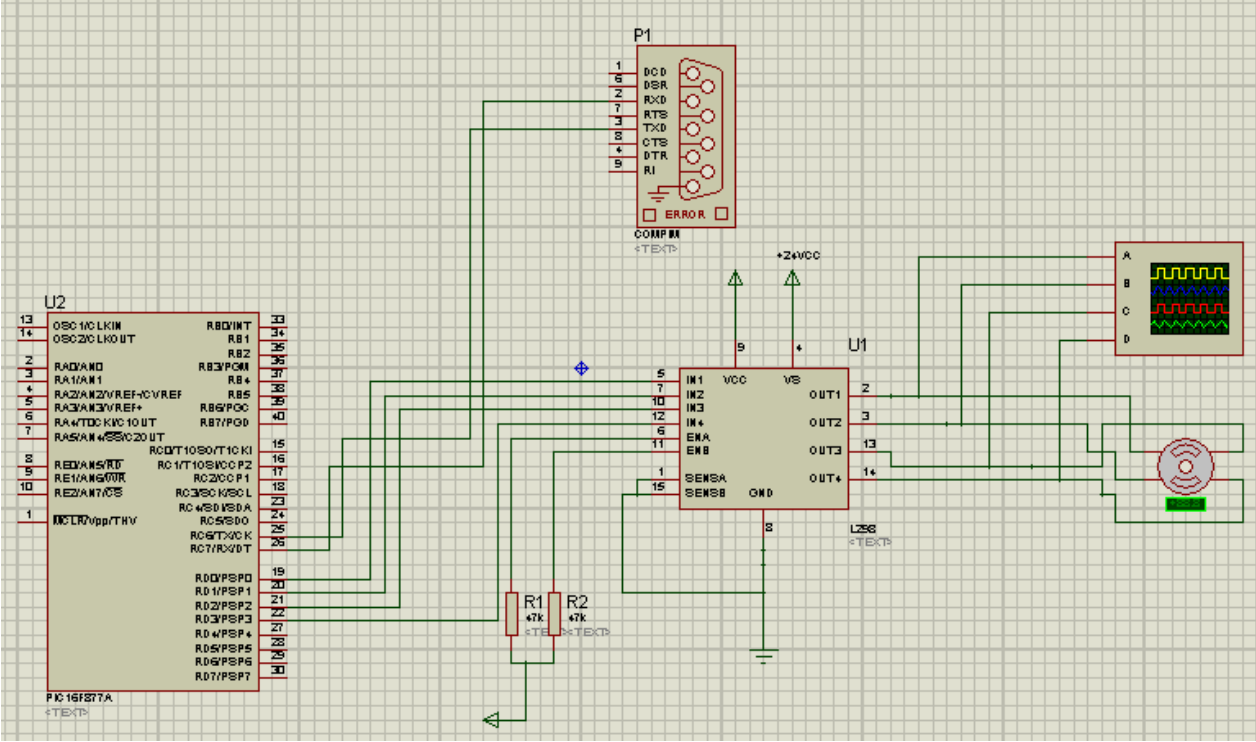


Figure 65 : Circuit de commande de mPP

III.4.17.7- Programme (Pic16F877)

```
#include "16F877A.h"

#use delay (clock=4000000)
#use rs232(baud=2400, xmit=PIN_C6, rcv=PIN_C7)
int x=1 ;
int etat=0;
char value;
#int_RDA
void isr_RDA()
{
    value=getc();
}
int sens =1;
void main()
{
    enable_interrupts(int_RDA);
    enable_interrupts(GLOBAL);
    setup_adc(ADC_OFF);
    setup_comparator(NC_NC_NC_NC);
    output_D(0);
    while(1)
    {
        while(etat==0)
        {
            if(value=='a')
            {
                etat=1;
            }
            output_D(0);
        }
        while(etat==1)
        {
            if(value=='b')
            {
                etat=0;
            }
            if((value=='a') || (value=='c'))
            {
                if(sens==2)
                {
                    output_D(0);          delay_ms(1000);
                }
                sens=1;
                output_D(6);          delay_ms(x);
                output_D(10);         delay_ms(x);
                output_D(9);          delay_ms(x);
                output_D(5);          delay_ms(x);
            }
        }
    }
}
```

```

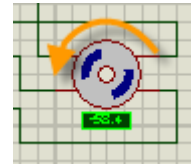
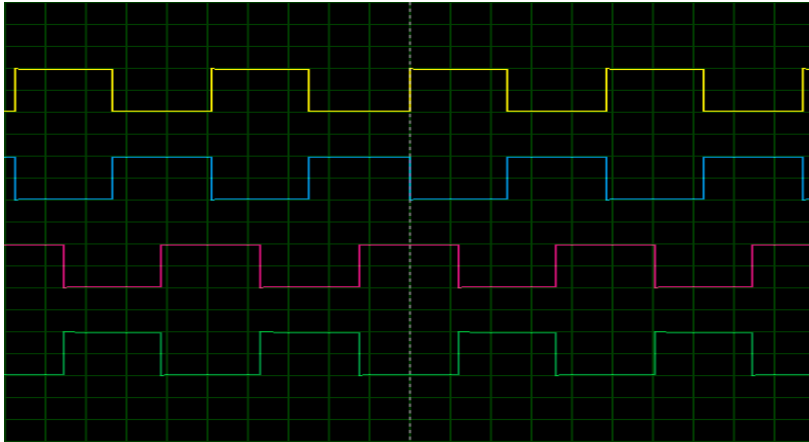
    }

    if(value=='d')
    {
        if(sens==1)
        {
            output_D(0);          delay_ms(1000);
        }
        sens=2;
        output_D(5);          delay_ms(x);
        output_D(9);          delay_ms(x);
        output_D(10);         delay_ms(x);
        output_D(6);          delay_ms(x);
    }
    if(value=='e')
    {
        if(sens==2)
        {
            output_D(0);          delay_ms(1000);
        }
        sens=1;
        output_D(4);          delay_ms(x);
        output_D(5);          delay_ms(x);
        output_D(1);          delay_ms(x);
        output_D(9);          delay_ms(x);
        output_D(8);          delay_ms(x);
        output_D(10);         delay_ms(x);
        output_D(2);          delay_ms(x);
        output_D(6);          delay_ms(x);
    }
    if(value=='f')
    {
        if(sens==1)
        {
            output_D(0);          delay_ms(1000);
        }
        sens=2;
        output_D(6);          delay_ms(x);
        output_D(2);          delay_ms(x);
        output_D(10);         delay_ms(x);
        output_D(8);          delay_ms(x);
        output_D(9);          delay_ms(x);
        output_D(1);          delay_ms(x);
        output_D(5);          delay_ms(x);
        output_D(4);          delay_ms(x);
    }
}
}
}
}

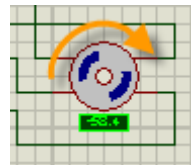
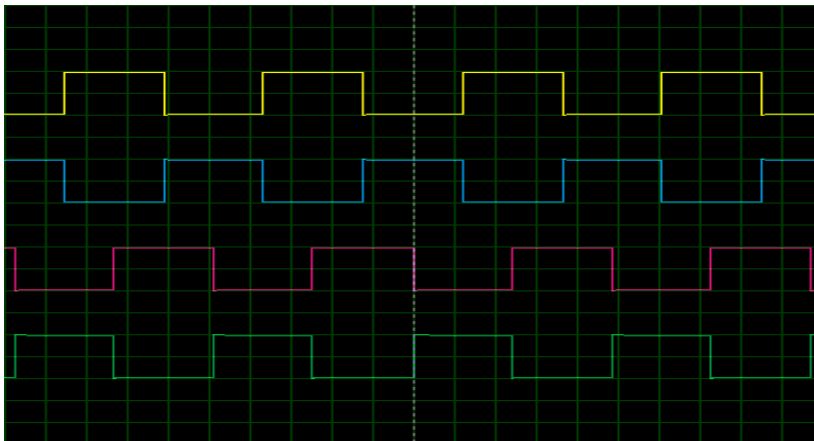
```


III.4.17.8- Les résultats de projet

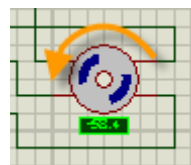
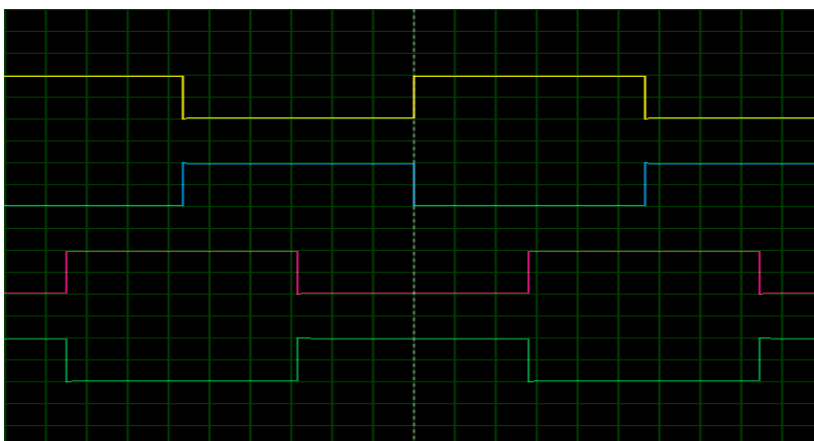
- 1^{er} cas : état pas, sens de rotation 1, vitesse fort



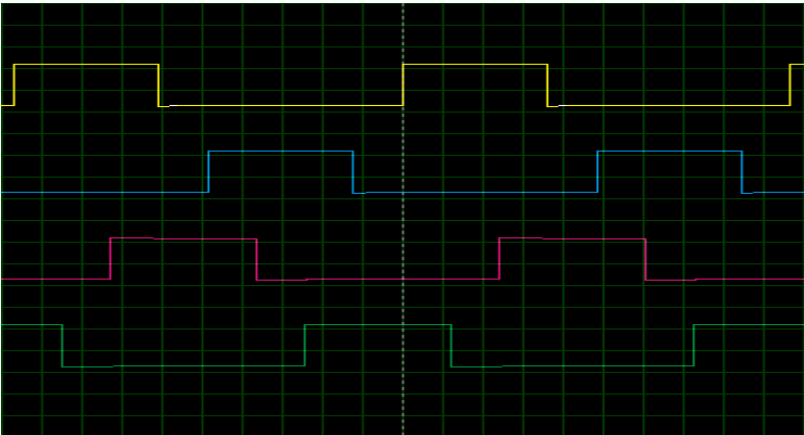
- 2^{eme} cas : état pas, sens de rotation 2, vitesse fort



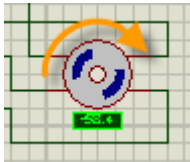
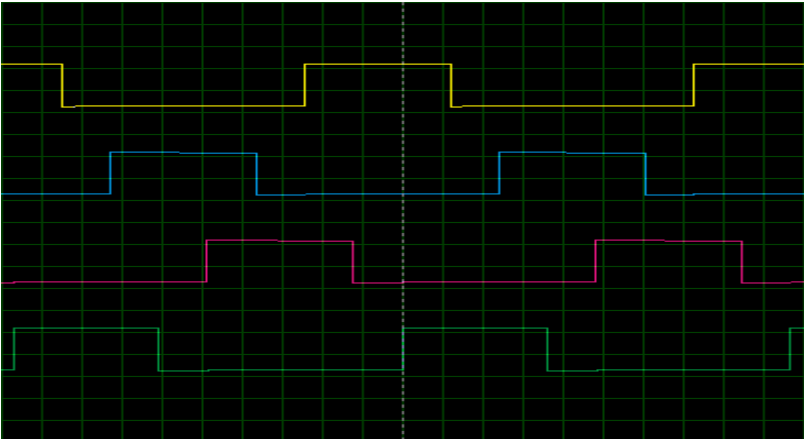
- 3^{eme} cas : état pas, sens de rotation 1, vitesse faible



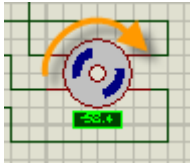
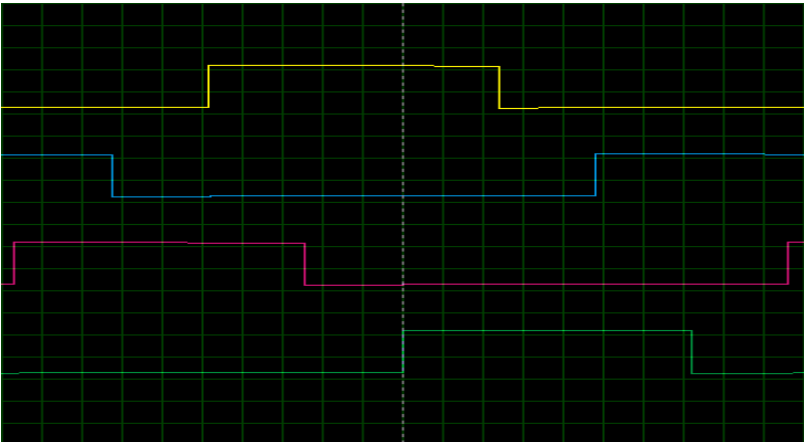
- 4^{eme} cas : état demi pas, sens de rotation 1, vitesse fort



- 5^{ème} cas : état demi pas, sens de rotation 2, vitesse fort



- 6^{ème} cas : état demi pas, sens de rotation 2, vitesse faible



III.4.18- Conclusion

Dans ce dernier chapitre, nous avons étudié la conception et réalisation détaillée de notre carte avec une bonne solution pour éviter le problème de communication entre PC et PIC16F77.

Conclusion générale

Cette étude nous donne un ensemble de solutions possibles pour éviter les problèmes de communications entre le PC et PIC16f877 dans l'appareil ABITEC 877, où nous avons traité deux solutions pour la commande la transmutions et réception des donnes.

Nous avons utilisé la première solution relier le PC et PIC16F877 dans première cas et PC et PIC16F84 dans le deuxième cas à la base de transistors, le principe de la carte est utilisé la tension de transmutions pour commende la base de transistor NPN et la tension qui passe commande la base de transistor PNP et utilise la tension qui passe comme une tension de transmutions vers le récepteur.

Nous avons utilisé la deuxième solution dans le même but mais avec un autre principe, le principe est commende l'ouverture et le fermenteur de les pistes de communication par les interrupteurs manuel, la carte fonctionne bien.

En dernier lieu, nous avons précisé les étapes à suivre pour générer le typon et la carte de notre projet.

Enfin, nous souhaitons que notre travail constitue un document pédagogique pour les futures promotions du domaine de l'électronique.

Bibliographie

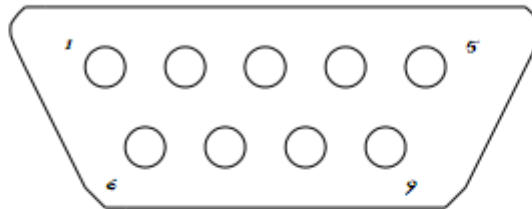
- [1] : CHRISTIAN TRAVENIER « Microcontrôleurs PIC 10, 12, 16». Editeur Dunod / 2007 - 3ème édition
- [2] : Olivier DARTOIS - Jacques DUPRAT « Microcontrôleurs Pic-Présentation Et Méthode De Programmation ». / TELN 2007
- [3] : ALAIN REBOUK « S'initier A La Programmation Des Pics ».. /Date de parution 04/11/2002 Editeur Dunod
- [4] : «PIC18F2455/2550/4455/4550 Data Sheet», 16/03/2015
- [5] : (www.audiovore.zxq.net/electronique.php), "La programmation des pic 18F", 10/03/2015
- [6] : (http://ebookbrowse.com/cours-complet-pic18f4520-v1-1-pdf-d253954337) "Informatique industrielle : le PIC 18F4520 et sa programmation en C" 30 - 32 - 28/03/2015
- [7] : <http://fr.wikipedia.org/wiki/RS-232> / 08/03/2015
- [8] : http://fr.wikipedia.org/wiki/Universal_Serial_Bus / 10/03/2015
- [9] : <http://claude.dreschel.free.fr/composants/max232/> / 11/03/2015
- [10] : Guessoumi Farid/Saadi Amel, Commande de panneau solaire à l'aide d'une carte a pic, Université Mohamed Khider, Biskra, Algerie, 2009/2010, p23-33
- [11] : Pascal Mayeux, "Apprendre la programmation des PIC High Performance par l'expérimentation et la simulation", DUNOD Editions : ETSF, Paris, France, Juillet 2010, p68 - 80

Annexe

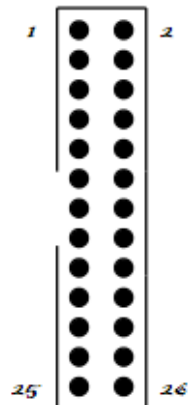
Les connecteurs de carte ABITEC et carte d'application

Serial I/O Pinout

P2 'D' Type Socket

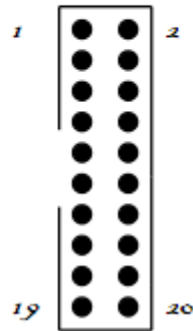


| Pin | Function |
|-----|----------|
| 1 | N/C |
| 2 | RxD |
| 3 | TxD |
| 4 | DTR |
| 5 | GND |
| 6 | DSR |
| 7 | RTS |
| 8 | CTS |
| 9 | N/C |

26 Way Connector P6

| Function | PIN No | Function | PIN No |
|----------|--------|----------|--------|
| RA0 | 1 | RA1 | 2 |
| RA2 | 3 | RA3 | 4 |
| RA4 | 5 | RA5 | 6 |
| RE0 | 7 | RE1 | 8 |
| 0 Volts | 9 | RD0 | 10 |
| RD1 | 11 | RD2 | 12 |
| RD3 | 13 | RD4 | 14 |
| RD5 | 15 | RD6 | 16 |
| RD7 | 17 | 0 Volts | 18 |
| RB0 | 19 | RB1 | 20 |
| RB2 | 21 | RB3 | 22 |
| RB4 | 23 | RB5 | 24 |
| RB6 | 25 | RB7 | 26 |

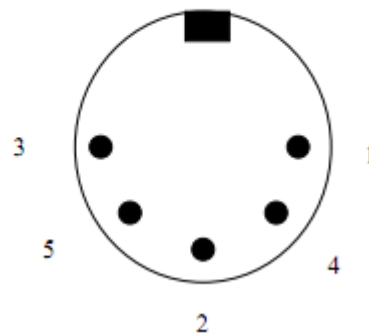
20 Way Connector P7



| Function | PIN No | Function | PIN No |
|----------|--------|----------|--------|
| RA0 | 1 | RA1 | 2 |
| RA2 | 3 | RA3 | 4 |
| RA4 | 5 | RA5 | 6 |
| RE0 | 7 | RE1 | 8 |
| RE2 | 9 | RC0 | 10 |
| RC1 | 11 | RC2 | 12 |
| RC3 | 13 | RC4 | 14 |
| RC5 | 15 | RB0 | 16 |
| 0 Volts | 17 | RB1 | 18 |
| RB2 | 19 | RB3 | 20 |

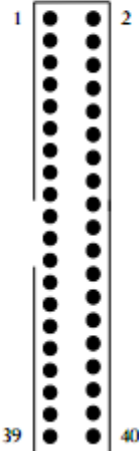
Connector P4

| Pin | Function |
|-----|----------|
| 1 | +5V |
| 2 | Ground |
| 3 | Ground |
| 4 | RC4/SDA |
| 5 | RC5/SAO |



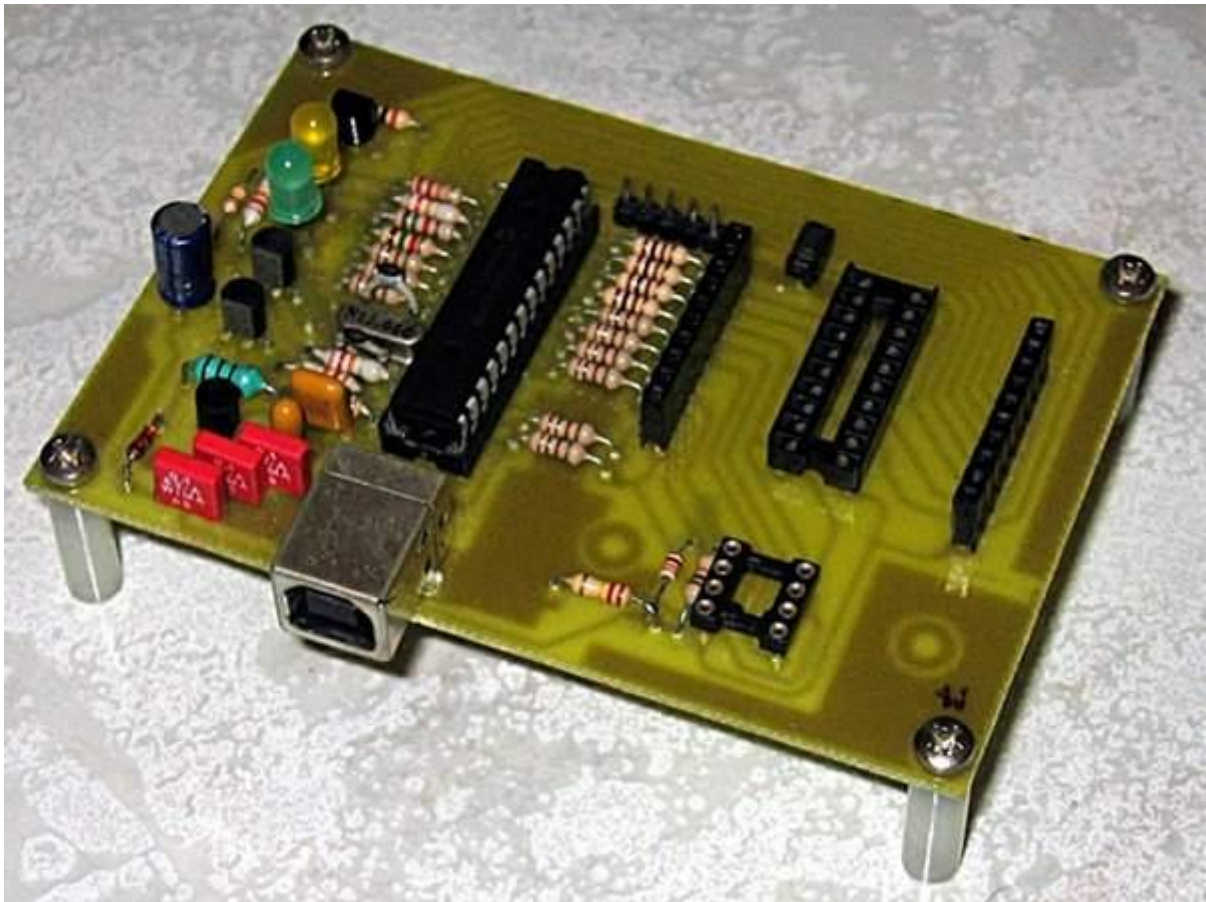
I/O Connectors

40 Way Connector P5

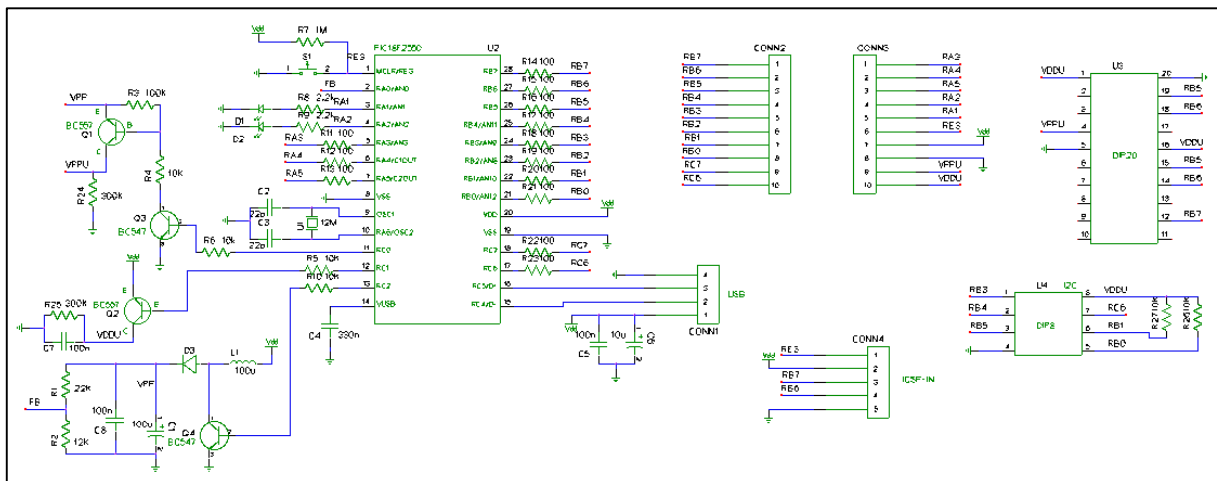


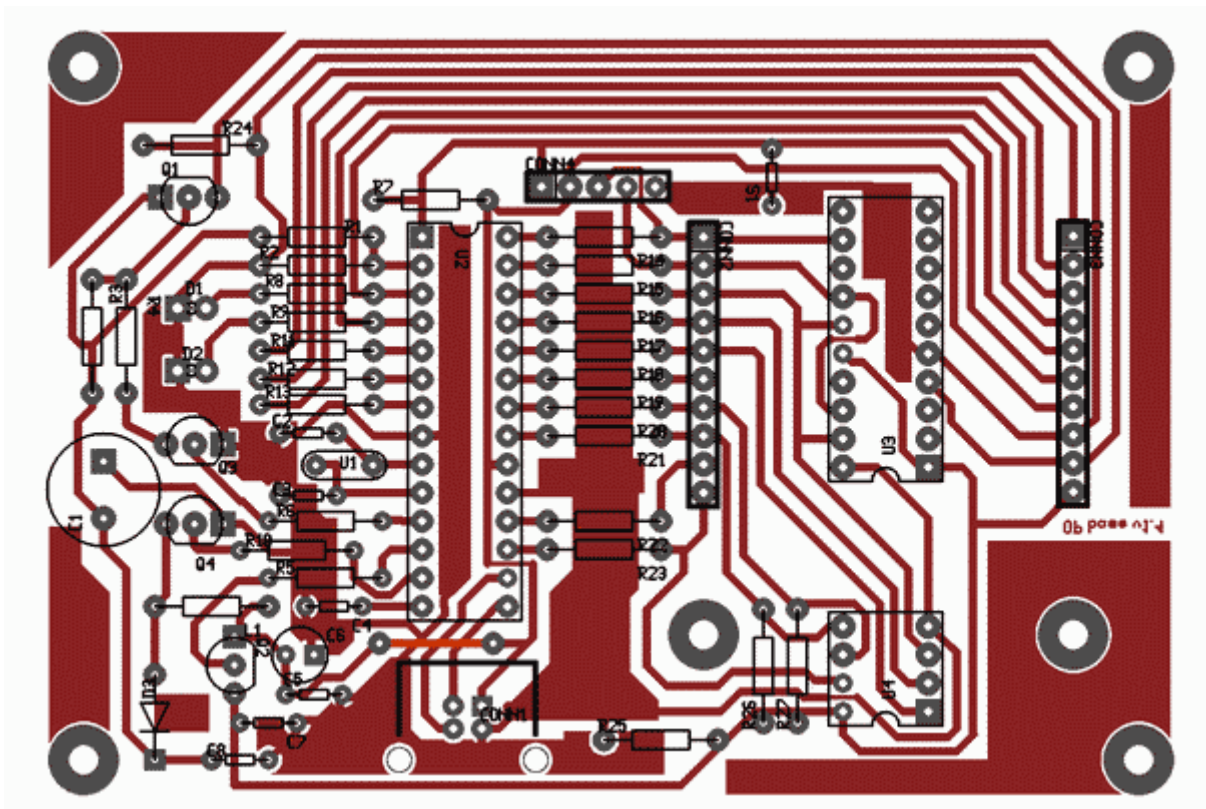
| Function | PIN No | Function | PIN No |
|----------|--------|----------|--------|
| N/C | 1 | N/C | 2 |
| N/C | 3 | N/C | 4 |
| N/C | 5 | N/C | 6 |
| N/C | 7 | N/C | 8 |
| N/C | 9 | N/C | 10 |
| N/C | 11 | N/C | 12 |
| +5 Volts | 13 | +5 Volts | 14 |
| RA0 | 15 | RE1 | 16 |
| RA1 | 17 | RE0 | 18 |
| RA2 | 19 | RA5 | 20 |
| RA3 | 21 | RA4 | 22 |
| RD0 | 23 | RD7 | 24 |
| RD1 | 25 | RD6 | 26 |
| RD2 | 27 | RD5 | 28 |
| RD3 | 29 | RD4 | 30 |
| RB0 | 31 | RB7 | 32 |
| RB1 | 33 | RB6 | 34 |
| RB2 | 35 | RB5 | 36 |
| RB3 | 37 | RB4 | 38 |
| 0 Volts | 39 | 0 Volts | 40 |

Programmateurs Open Programmer v0.9.x



Circuit





Components liste:

U1 12Mhz quartz (also 4, 8, 16, 20; reconfiguration of input divider options is required)

U2 18F2550 (also 2450,2455,2458,2553,4450,4455,4458,4550,4553)

U3 20p socket.

U4 8p socket.

Q1-2 BC557 (or any PNP, pay attention to polarity)

Q3-4 BC547 (or any NPN, pay attention to polarity)

D1-2 LED

D3 1N4148 (or any diode, better if Shottky)

L1 100uH resistor type or other

R1 22K

R2 12K

R3 100K

R4:6 10K

R7 1M

R8-9 2.2K

R10 10K

R11:23 100
R24-25 330K
R26-27 10K
C1 22-100uF 25V
C2-3 22pF
C4 >= 220nF
C5 100nF
C6 10uF
C7-8 100nF
CONN1 USB type B
CONN2-3 10 pin female stripline
CONN4 5 pin stripline

Code source

```
#include "18f2550.h"
#include <stdio.h>
#include <usart.h>
#include "usb.h"
#include "OPcontrol.h"

#pragma config PLLDIV=3, FOSC=HSPLL_HS //for 12 MHz crystal
#pragma config CPUDIV=OSC1_PLL2, USBDIV=2
#pragma config IESO=OFF, PWRT=OFF, BOR=ON_ACTIVE, VREGEN=ON, WDT=OFF
#pragma config MCLRE=OFF, PBADEN=OFF, STVREN=ON, LVP=OFF, XINST=OFF,
DEBUG=OFF

#if !defined(__18F2450)
#pragma config CCP2MX=OFF, BORV=3
#else
#pragma config BORV=21
#endif

#pragma code low_vector=0x8

void low_interrupt (void)
```

```

{
    _asm GOTO timer_isr _endasm
}

#pragma code

#if DEBUG_PRINT

static void InitializeUSART()
{
    TRISC &= 0xBF; // Set RC6 as an output
    TRISC |= 0x80; // Set RC7 as an input
    RCSTA = 0x90; // Enable serial port, enable receiver
    TXSTA = 0x24; // Asynch, TSR empty, BRGH=1

    // Baud rate formula for BRG16=1, BRGH=1: Baud Rate = Fosc/(4 (n + 1)),
    // or n = (Fosc / (4 * BaudRate)) - 1
    // At 48 MHz, for 115.2K Baud:
    // SPBRGH:SPBRG = n = Fosc / (4 * 115200) - 1 = 103.17
    BAUDCON = 0x18; // BRG16=1 txd inverted
    // SPBRGH = 0x00; // At 48MHz, SPBRGH=0, SPBRG=103 gives 115.2K Baud
    SPBRGH = 0x02; // 0x0270 gives 19200 Baud

#if CLK_48MHZ
    // SPBRG = 103; // For 48 MHz clock
    SPBRG = 0x70; // For 48 MHz clock
#else
    SPBRG = 52; // For 24 MHz clock
#endif

    printf("USB Test Startup\r\n");
}

#endif

// Entry point of the firmware
void main(void)
{

```

```
// Set all I/O pins to digital
ADCON1 |= 0x0F;
// Initialize USB
UCFG = 0x14; // Enable pullup resistors; full speed mode
deviceState = DETACHED;
remoteWakeup = 0x00;
currentConfiguration = 0x00;
// Call user initialization function
UserInit();
while(1)
{
// Ensure USB module is available
EnableUSBModule();
// As long as we aren't in test mode (UTEYE), process
// USB transactions.
if(UCFGbits.UTEYE != 1)
ProcessUSBTransactions();
// Application specific tasks
ProcessIO();
}
}
```

Résumé

Dans le domaine de l'industrie il y a beaucoup de communications entre les machines, pour la communication entre l'ordinateur et microcontrôleurs utilisé RS232 et USB, dans notre cas (plateforme ABITEC), nous utilisons RS232.

Le problème de plateforme ABITEC est les communications entre PIC16F877 (PIC programmable) et PC.

Pour de mettre à jour de la plateforme nous avons réalisé une carte pour communiqué PIC16F877 et PC, Pour cela, nous allons utiliser les interrupteurs automatique et manuel pour commande les deux pins de communication RX, TX.