

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté de Technologie

Département Ingénierie des Systèmes Electriques

Mémoire de Master

Présenté par

TAHMI Amina

Filière : Automatique

Spécialité : Automatique et Informatique Industrielle

**ETUDE ET REALISATION D'UNE CARTE
D'INTERFACE E/S COMMANDEE PAR PC**

Soutenu le 19 novembre 2020 devant le jury :

MESSAOUDI	Noureddine	MCA	UMBB	Président
NAFA	Fares	MCB	UMBB	Examineur
YAKHELEF	Yassine	MCA	UMBB	Encadreur

Année Universitaire : 2019/2020

« Le succès n'est pas la clé du bonheur. Le bonheur est la clé du succès. Si vous aimez ce que vous faites, vous réussirez »

Albert Schweitzer

REMERCIEMENTS

*Tout d'abord, Je tiens à remercier et glorifier Dieu le tout puissant et
miséricordieux,*

*D'avoir guidé mes pas avec clairvoyance pour accomplir ce travail et
de le Mener à terme.*

*J'aimerais dans ces quelques lignes remercier toutes les personnes qui
d'une manière ou d'une autre, ont contribué au bon déroulement de
mon travail, tout au niveau humain qu'au niveau scientifique.*

*Je tiens à remercier aussi mon promoteur,
Monsieur YAKHELEF Yassine D'avoir accepté de m'encadrer pour
l'exécution de ce Projet.*

*J'ajoute en particulière sa patience et ses encouragements, nous a
permis de travailler dans bonnes conditions.*

*J'exprime Mon Travail à l'ensemble des professeurs du département
qui ont contribué à ma formation, on les prie de bien vouloir croire à
la gratitude en espérant que cet humble travail de trois années fera
crédibilité de leurs efforts.*

*Mes remerciements s'adressent également à tous membres de Jury, qui
ont accepté de nous honorer de leur présence et de juger notre travail*

Merci.

*Et à toute personne ayant contribué de près ou de loin
à notre soutien moral.*

Amina

DEDICACE

Je dédie ce modeste travail :

A Mes Parents :

*Pour tous les sacrifices qu'ils ont faits pour moi, pour leur
Soutien continu durant mon travail. Que dieu vous protège.*

A mes frères et sœurs :

*Que Dieu vous garde, Je vous aime et je vous souhaite une vie
Pleine de succès et de réussite.*

A Mes amis :

*Qui m'ont soutenue durant toute cette période, dieu vous garde.
Vous guide dans le droit chemin.*

*Et enfin à tout ce ceux qui m'ont aidé de près ou de loin dans
Mon travail*

*Ainsi qu'à tous les membres de la promotion de master en
Automatique et Informatique Industrielle 2020.*

Amina

Résumé

Dans ce travail, on se propose de réaliser une carte d'interface et de control pour PC cette carte offre la possibilité de commander jusqu'à huit appareils et de lire autant de lignes digitales, ainsi que deux lignes analogiques, en utilisant le port sériel de n'importe quel PC.

La communication entre le PC et la carte, s'effectue par un protocole simple RS232, qui empêche les erreurs de communication.

Mots-clés : Entrée / Sortie, Ordinateur personnel (PC), RS232.

Abstract

In this work, we propose to realize an interface and control card for PC this card offers the possibility of controlling up to eight devices and reading as many digital lines, as well as two analogue lines, using the serial port. from any PC.

Communication between the PC and the card takes place via a simple RS232 protocol, which includes communication errors.

Keywords: Input / Output, Personal computer (PC), RS232.

ملخص

في هذا العمل ، نقترح تحقيق واجهة وبطاقة تحكم للكمبيوتر الشخصي ، توفر هذه البطاقة إمكانية التحكم في ما يصل إلى ثمانية أجهزة وقراءة أكبر عدد ممكن من الخطوط الرقمية ، بالإضافة إلى خطين تناظريين ، باستخدام المنفذ التسلسلي. من أي جهاز كمبيوتر.

يتم الاتصال بين الكمبيوتر والبطاقة عبر بروتوكول RS232 البسيط ، والذي لا يتضمن أخطاء الاتصال.

الكلمات الرئيسية: الإدخال / الإخراج ، الكمبيوتر الشخصي (PC) ، RS232

Sommaire

INTRODUCTION GENERALE.....	1
----------------------------	---

Chapitre I : Généralités sur la liaison série et la norme RS232

I.1 INTRODUCTION.....	2
I.2 PORT.....	2
I.3 PARALLELE OU SÉRIE.....	2
I.3.1 Liaison parallèle.....	2
I.3.2 Liaison série.....	3
I.3.2.1 Protocole synchrone.....	3
I.3.2.2 Protocoles asynchrones.....	4
I.3.3 La parité.....	5
I.4 BUS OU LIAISON.....	6
I.5 TYPES OU MODES DE LIAISON.....	7
I.5.1 Liaison unidirectionnelle « SIMPLEX ».....	7
I.5.2 Liaison bidirectionnelle.....	8
I.5.3 Liaison directionnelle simultanée.....	10
I.6 DESCRIPTIONS TECHNIQUES DE LA NORME RS232.....	11
I.6.1 Description électrique des signaux.....	11
I.6.2 La trame RS232.....	12
I.6.3 Fonction des signaux.....	16
I.6.4 Interconnexion des équipements.....	16
I.7 LES PROTOCOLES.....	20
I.7.1 Handshake.....	20
I.7.2 Xon-Xoff.....	21
I.7.3 ETX/ACK.....	22
I.8 CONCLUSION.....	22

Chapitre II : Microcontrôleur

II. 1. Introduction.....	23
II. 2. Les avantages des microcontrôleurs.....	23
II. 3. Les caractéristiques principales d'un microcontrôleur.....	24
II. 3. 1. Différentes Familles de microcontrôleurs.....	24
II. 3. 2. Définition d'un PIC.....	24
II. 3. 3. Les différentes familles des PIC.....	25
II. 3. 4. Le choix du microcontrôleur.....	25
II. 4. Le Microcontrôleur PIC16F876A.....	25
II. 4. 1. Présentation.....	25
II. 4. 2. Brochage.....	26
II. 4. 3. Les particularités électriques.....	27
II. 4. 3. 1. Organisation du 16F876A.....	27
II. 4. 3. 1. 1. La mémoire programme.....	27
II. 4. 3. 1. 2. La mémoire eeprom.....	27
II. 4. 3. 1. 3. La mémoire Ram et organisation.....	27
II. 4. 3. 1. 4. Watchdog.....	27
II. 4. 3. 1. 5. Le TIMER.....	28
II. 4. 3. 2. Les ports entrée/sortie.....	28
II. 4. 3. 2. 1. Particularité du port A.....	28
II. 4. 3. 2. 2. Particularités du port B.....	28

II. 4. 3. 2. 3. Particularités du port C.....	28
II. 4. 3. 3. L'oscillateur	29
II. 4. 3. 4. MCLR.....	29
II. 5. Conclusion.....	29

Chapitre III : Etude Théorique De La Carte

III.1. Introduction.....	30
III. 2. Schéma Synoptique Du Projet	30
III. 3. les éléments de la carte d'acquisition.....	31
III. 3. 1. Les Entrées Numériques	31
III. 3. 2. Les Entrées analogiques.....	31
III. 3. 3. LE MICROCONTROLEUR 16F876A	32
III. 3. 4. LA LIAISON SERIE RS232.....	32
III. 3. 4 .1 . LA COMMUNICATION SERIE.....	32
III. 3. 4. 2. LA PROCEDURE DE TRANSMISSION.....	34
III. 3. 5. Amplification Du Courant.....	36
III. 3. 5. 1. Les Sorties.....	36
III. 3. 5. 2. LE CIRCUIT INTÉGRÉ ULN2803	36
III. 3. 6. Les Relais	37
III.3. 7. Circuit d'alimentation	38
III. 4. INTRODUCTION.....	38

Chapitre IV : Réalisation Pratique

IV. 1. Introduction.....	39
IV.2. Partie Electronique (Hardware).....	39
IV. 2. 1. LE CHOIX DES COMPOSANTS.....	39
IV. 2. 1. 1. Régulateur 7805.....	39
IV. 2. 1. 2. Diode 1N4007	39
IV. 2. 1. 3. Les Diode Zener.....	40
IV. 2. 1. 4. Les Leds.....	40
IV. 2. 1. 5. Condensateur	41
IV. 2. 1. 6. Quartz.....	42
IV. 2. 2. Réalisation De la carte et simulation.....	42
IV. 2. 2. 1. Simulation.....	42
IV. 2. 2. 1. 1. Simulation De L'ISIS.....	43
IV. 2. 2. 1. 2. Carte D'acquisition.....	43
IV. 2. 2. 2. Réalisation pratique.....	43
IV. 2. 2. 2. 1. Présentation de l'ARES.....	43
IV. 3 . Partie Programmation (software).....	46
IV. 3. 1. Organigramme du fonctionnement du programme principal.....	47
IV. 3. 2. Langage de programmation Micro C.....	49
IV. 3. 3. Programme de Pic.....	49
IV. 3. 4. Interface (Supervision).....	49
IV. 3. 4. 1. Représentation LABVIEW.....	49
IV. 3. 4. 2. Réalisation de l'interface.....	49
IV. 4. Teste Résultat finale.....	52
IV. 5. Conclusion.....	53
CONCLUSION GENERALE.....	54

Bibliographie

Liste des figures

Chapitre I :

Fig.I.1. Exemple d'une liaison parallèle.....	3
Fig.I.2. Exemple d'une liaison série synchrone.....	4
Fig.I.3. Exemple d'une liaison série asynchrone.....	4
Fig.I.4. Transmission série asynchrone.....	5
Fig.I.5. Une liaison.....	6
Fig.I.6. Exemple d'un bus.....	7
Fig.I.7. Exemple de liaison unidirectionnelle.....	7
Fig.I.8. Chronogramme d'un exemple de liaison unidirectionnelle.....	8
Fig.I.9. Exemple de liaison half duplex.....	9
Fig.I.10. Chronogramme d'un exemple de liaison half duplex.....	10
Fig.I.11. Exemple de liaison full duplex.....	10
Fig.I.12. Chronogramme d'un exemple de liaison full duplex.....	11
Fig.I.13. Chronogramme d'une trame de données (Trame DATA).....	13
Fig.I.14. Chronogramme d'un exemple de trame de données/Valeur 55 h/85 d, pas de parité, un bit de stop.....	13
Fig.I.15. Chronogramme d'un exemple de trame de données/Valeur E1h/255 d, pas de parité, un bit de stop.....	14
Fig.I.16. Durée d'un bit à 2400 Bauds. Exemple de la transmission de l'octet : 55h /85d / 01010101b.....	15
Fig.I.17. Liaison complète entre un DTE et DCE.....	17
Fig.I.18. Liaison complète entre deux DTE (Null Modem.....	18
Fig.I.19. Liaison deux fils (DTE/DTE).....	19
Fig.I.20. Liaison trois fils (DTE/DTE).....	19
Fig.I.21. Liaison complète vert équipement trois fils (DTE/DTE).....	20
Fig.I.22. Chronogramme protocole matériel.....	20
Fig.I.23. Chronogramme protocole Xon-Xoff.....	21

Chapitre II :

FigII. 1: Brochage de PIC 16F876.....	26
---------------------------------------	----

Chpitre III :

Fig. III. 1. Schéma synoptique de fonctionnement du projet à réaliser.....	30
Fig. III. 2. Schéma d'implantation des entrées.....	31
Fig. III. 3. Brochage de la SUB-DB9.....	33
Fig. III. 4. schéma synoptique de la transmission série asynchrone.....	34
Fig. III. 5. Chronogramme de la transmission série asynchrone.....	35
Fig. III. 6. Brochage du MAX232.....	36
Fig. III. 7. Schéma interne ULN2803.....	37
Fig. III. 8. Symbole du relais.....	38
Fig. III. 9. Montage électrique de l'alimentation.....	38

Chapitre IV :

Fig. IV. 1. Régulateur 7805.....	40
Fig. IV. 2. Symbole de Diode.....	41
Fig. IV. 3. Symbole de Diode Zener.....	41
Fig. IV. 3. Les Leds.....	42
Fig. IV. 4. Plusieurs types de condensateurs.....	43
Fig. IV. 5. Un quartz de 4 MHz.....	43
Fig. IV. 6. Schéma de simulation de la carte.....	44
Fig IV. 7. La carte en 3 dimensionnement.....	45
Fig IV. 8. Schéma circuit imprimé de l'interface (face 1).....	46
Fig IV. 9. Schéma circuit imprimé de l'interface (face 2).....	46
Fig IV. 10. La carte électronique.....	47
Fig IV. 11 Organigramme du programme principal.....	48
Fig IV. 12 Organigramme du réception des données.....	49
Fig IV. 13. Fenêtre descriptive du micro c.....	50
Fig IV. 14. Programme principale.....	50

Fig IV. 15. Fenêtre de l'environnement de développement sur LabVIEW Face avant (à droite) et Diagramme (à gauche).....	51
Fig. IV. 16. Face avant (l'interface).....	52
Fig. IV. 17. Face arrière (programme).....	52
Fig. IV. 18. Interface avec simulation Proteus ISIS.....	53
Fig. IV. 19. Réalisation finale du Projet.....	53

Liste des tableaux

Tableau I.1 : Exemple de parité paire ou impaire.....	6
Tableau I.2 : Équivalence niveaux logiques/ tensions.....	11
Tableau I.3 : Tableau de correspondance entre la valeur de débit et le temps de maintien des données.....	15
Tableau I.4 : Description fonctionnelle des signaux.....	16
Tableau I.5 : Signification DTE/DCE.....	17
Tableau I.6 : Déroulement chronologique de l'HandShaking.....	21
Tableau II.1. Caractéristiques du PIC 16F876A.....	26

Liste des abréviations

ARES : Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS.

E/S : Entrée/Sortie.

IEEE : Electrical and Electronics Engineers.

IR : InfraRouge.

ISIS : Logiciel permet également de simuler les schémas.

LabVIEW : Laboratory Virtual Instrument Engineering Workbench.

LED : Signalisation d'état et de Défauts.

PC : Personal Computer.

PIC : Programmable Interface Controller.

RS232 : Recommended Standard 232.

TTL : Transistor-Transistor Logic.

USB : Universal Serial Bus.

VISA : Virtual Instrument Software Architecture.

INTRODUCTION GENERALE

Le développement à grande échelle des systèmes d'automatisation de toutes sortes, à fait apparaître dans les foyers, dans les lieux de travail et même dans les lieux publics, des télécommandes IR, des claviers en tous genres, de petites centrales destinées à la commande de toutes sortes d'appareils électriques, des commandes d'alarmes, d'éclairages, de chauffages, etc.

Pour plus de commodité, pour chacun des appareils, il y a un contrôle, tant est si bien, qu'il devient de plus en plus difficile de les gérer tous.

Confronté à ce problème, la meilleure solution est de confier le contrôle à un seul élément et qu'existe-t-il de mieux que l'ordinateur ?

En fait, dotant cet ordinateur d'un programme approprié, il devient la centrale de commande de tous les automatismes, sans pour autant être monopolisé par cette tâche.

Naturellement, l'ordinateur ne peut pas effectuer ces opérations tout seul. Par l'intermédiaire d'un des deux ports de communication, il peut analyser les données nécessaires et intervenir en conséquence, de façon automatique ou assistée par l'utilisateur.

Le projet que nous proposons dans ces lignes est une interface générique pour PC. Elle est équipée de 8 sorties sur relais et de 8 entrées TTL, actives aux 0 logiques. Par ailleurs, elle dispose de deux canaux analogiques, capables de lire des tensions continues d'une valeur comprise entre 0 et 5 volts. [1]

Dans le premier chapitre de notre travail on rappelle les notions générales sur la liaison série et la norme RS232.

Ensuite dans le deuxième chapitre on rappelle les notions sur les Microcontrôleur.

Après dans le troisième chapitre on va étudier théoriquement le système de contrôle et les éléments qui rassemblent notre carte.

Enfin dans le dernier chapitre on présente l'interface graphique qui affiche toutes les mesures sur un Ordinateur et les traitent à l'aide du logiciel Labview qui va nous permettre d'atteindre notre but de commande.

On termine ce mémoire par une conclusion générale sur le travail théorique et pratique qui sera fait avec l'espoir d'apporter au lecteur la possibilité d'avoir une idée plus claire sur le système réalisé.



Chapitre I

Généralités sur La Liaison série et la norme RS232

I.1. INTRODUCTION

Une liaison série est une ligne où les bits d'information (1 ou 0) arrivent successivement, soit à intervalles réguliers (transmission synchrone), soit à des intervalles aléatoires, en groupe (transmission asynchrone).

La liaison RS232 est une liaison série asynchrone.

L'octet à transmettre est envoyé bit par bit (poids faible en premier) par l'émetteur sur la ligne Tx, vers le récepteur (ligne Rx) qui le reconstitue.

La communication peut se faire dans les deux sens (duplex), soit émission d'abord, puis réception ensuite (half-duplex), soit émission et réception simultanées (full-duplex).

La transmission étant du type asynchrone (pas d'horloge commune entre l'émetteur et le récepteur), des bits supplémentaires sont indispensables au fonctionnement: bit de début de mot (start), bit(s) de fin de mot (stop).

D'autre part, l'utilisation éventuelle d'un bit de parité, permet la détection d'erreurs dans la transmission.

Dans ce chapitre nous avons donné une brève présentation sur la liaison série avec les deux types synchrone et asynchrone et la norme RS-232 avec descriptions techniques de la norme.

I.2. PORT

Un port est le terme permettant de spécifier une connexion normalisée. Ce terme désigne implicitement le type de liaison ou de bus et la connectique qui y est rattachée. En micro-informatique, on parle communément du port RS232 ou du port parallèle [2].

I.3. PARALLELE OU SÉRIE

I.3.1 Liaison parallèle

La méthode la plus répandue, consiste à relier les deux équipements par un bus parallèle. Ce type de connexion a l'avantage de satisfaire des débits très importants avec un nombre de bits variant suivant le matériel utilisé (4, 8, 16, 32, etc.). Elle reste limitée à de courtes distances ; un à deux mètres dans le meilleur des cas. De plus, on peut être confronté à des problèmes de

diaphonie dans les câbles ainsi que sur les cartes. L'encombrement peut être également un problème en termes de surface de pistes sur les cartes, d'autant plus que ce type de liaison impose des connecteurs de grandes tailles. Pour illustrer, on peut citer les liaisons ou bus parallèle les plus connus : le port Centronics, le bus SCSI, le bus IEEE488, etc. [2]. Le schéma simplifié d'une liaison parallèle est donné par la figure I.1.

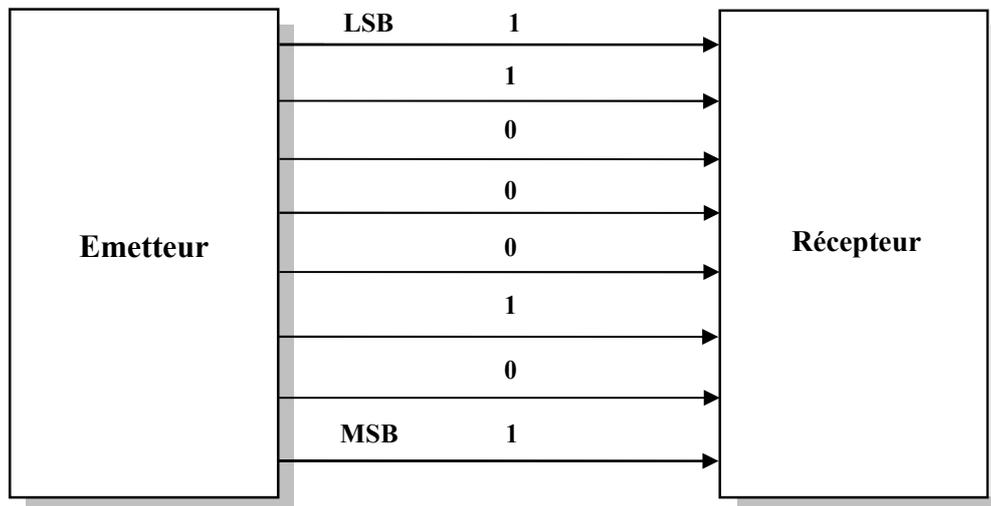


Fig. I.1. Exemple d'une liaison parallèle [2]

Une autre méthode consiste à transmettre les données les unes après les autres sur un seul fil. Cette méthode porte le nom de liaison série.

I.3.2. Liaison série

Pour interpréter un message binaire transmis en série, le récepteur et l'émetteur doivent pouvoir identifier le début et la fin d'un caractère ou bien d'un message grâce à une procédure appelée « protocole de communication série ». Il en existe plusieurs qu'on peut les regrouper en deux grandes classes : **synchrones** et **asynchrones**

I.3.2.1. Protocole synchrone

La caractéristique essentielle de transfert synchrone des données en série est l'asservissement exact des données au signal d'horloge.

Une fois que la vitesse de transmission (en bauds) a été fixée, le dispositif émetteur doit transmettre un bit à chaque impulsion d'horloge.

En plus de la vitesse de transmission, un protocole synchrone doit encore :

- Fixer la longueur d'un mot (ou caractère) de données.
- Permettre au dispositif récepteur de se synchroniser. Un message doit commencer par un ou deux caractères spéciaux, dits « caractère de synchronisation » (mot SYNC)

Ainsi en transmission synchrone, l'émission une fois commencée est continue, l'émetteur comble les trous éventuels d'un message par l'envoi d'un ou plusieurs caractères SYNC pour permettre au récepteur de rester synchronisé. Sur un seul support « fil », on a succession de 0 et 1 qui constituent la trame comme le montre la figure I.2.

Le temps qui sépare l'envoi de deux messages doit être un multiple de la durée du bit.

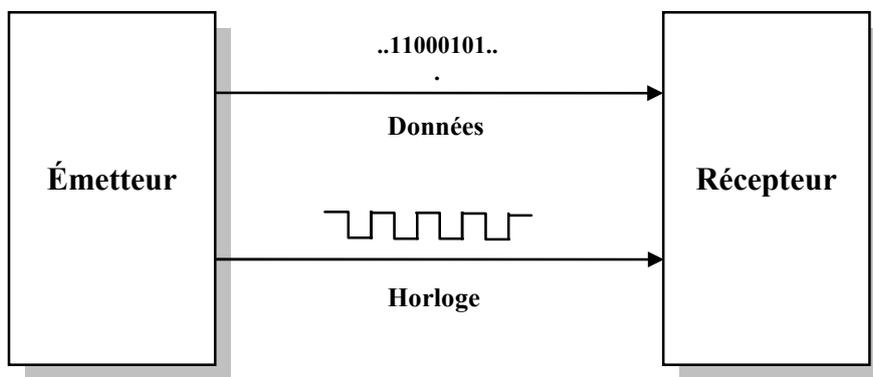


Fig. I.2 Exemple d'une liaison série synchrone

I.3.2.2. Protocoles asynchrones

Le principe de transmission est identique à la liaison série synchrone mais sans la ligne d'horloge. Les deux équipements possèdent leur propre horloge indépendante l'une de l'autre. L'horloge du récepteur se synchronise uniquement sur le premier bit de la trame, puis elle continue sur la lancée. Sur un seul fil, on transmet le caractère qui constitue l'information. Celle-ci est précédée du signal START et terminée par le « ou les » bit de STOP. Le schéma de la figure I.3 représente une liaison série asynchrone.

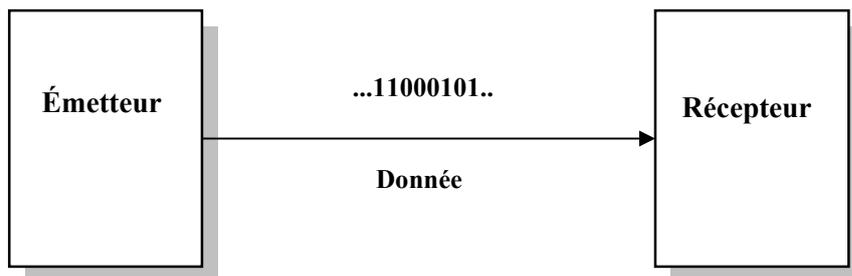


Fig. I.3. Exemple d'une liaison série asynchrone

La grande différence entre les protocoles synchrones et asynchrones est que, dans ce deuxième cas, le flot des données est DISCONTINU ; l'émission se fait caractère par caractère, au fur et à mesure que ceux-ci sont disponibles. Entre deux caractères, l'émetteur envoie un signal de rupture consistant en un niveau de tension haut. Ce niveau haut (appelé MARK signal en technique des transmissions), pour permettre de commencer l'envoi d'un caractère par un bit de départ à l'état bas (ou zéro).

Chaque mot « donnée » en transmission asynchrone, étant séparé, a besoin, pour être identifié par le récepteur, de contenir sa propre information de synchronisation. Ceci se fait en cadrant les bits d'information utile entre un bit dit de départ (START bit) et un ou deux bits d'arrêt (STOP bit).

Dans le domaine des micro-ordinateurs, le format d'une unité de données transmise est montré à la figure I.4

- Un bit START unique, de valeur 0 (norme universelle) ;
- Un mot d'information comportant 5,6 ou le plus souvent 7 bits (code ASCII) ;
- Un bit (facultatif) de parité, paire ou impaire ;
- Un ou deux bits STOP (exceptionnellement 1,5).

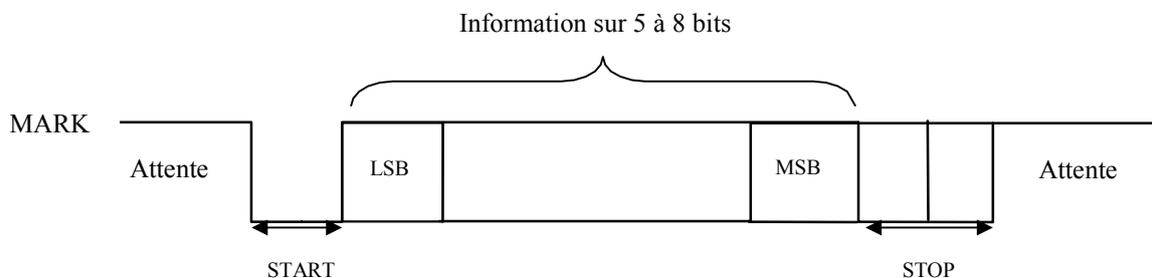


Fig.I.4. Transmission série asynchrone

I.3.3. La parité

Pour s'assurer que le caractère envoyé est bien celui qui a été reçu, on peut utiliser le contrôle de parité. Le principe est de rajouter un bit de parité directement après le caractère. Ce bit est mis à 0 ou à 1 selon que l'on veuille obtenir une parité paire (le nombre de 1 est pair) ou impaire (le nombre de 1 est impair). Si on souhaite transmettre un caractère ayant un nombre impair de bit à 1, et que l'on utilise un contrôle de parité paire, il faudra que le bit de parité soit positionné à 1 pour que le nombre total de bits à 1 soit pair. Ceci permet à l'organe de réception de vérifier si on a le même nombre de 1 dans le caractère à l'arrivée qu'il y en avait au départ.

Ce contrôle n'est pas infaillible, mais la probabilité d'avoir plusieurs inversions de bits dans un même caractère est faible.

Exemple :

Tableau I.1 : Exemple de parité paire ou impaire

Caractère transmise (7 bits)	Bit de parité (parité paire)	Bit de parité (parité impaire)
1001100	1	0
0000000	0	1
1010101	0	1

I.4. Bus ou Liaison [2]

De la même manière, pour chacune des deux natures d'interconnexions (parallèle ou série), il existe deux types d'architectures.

A. Le premier type est la liaison. C'est celle qui a été décrite ci-dessus. Il s'agit de l'interconnexion de deux équipements quelle que soit leur nature (voir la figure I.5) : émetteurs, récepteurs, ou émetteurs/récepteurs.

Cette catégorie ne présente pas de difficulté particulière, c'est également celle qui est utilisée pour la RS232.

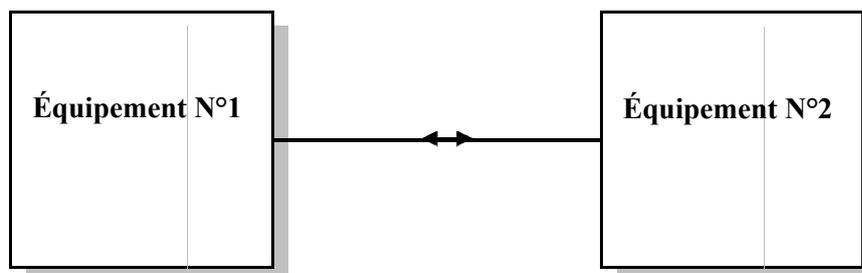


Fig. I.5. Une liaison

B. Le deuxième type d'architecture est le bus donné par la figure I.6. La structure matérielle permet ici d'interconnecter plusieurs équipements entre eux sur le même support physique. Cette architecture est beaucoup plus complexe que l'interface bus des équipements puisse supporter une interconnexion. De même, au niveau logiciel, un protocole est indispensable pour gérer les communications entre les différents équipements.

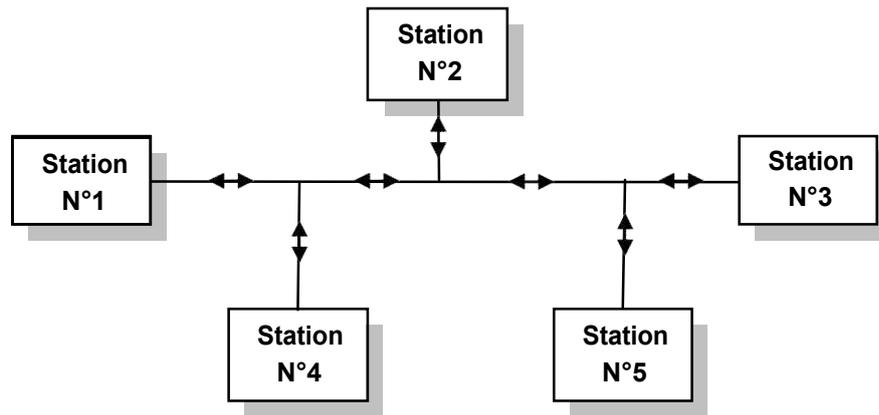


Fig. I.6. Exemple d'un bus

I.5. Types ou modes de liaison

Du point de vue de la capacité de transfert de la ligne de transmission, on distingue trois types de liaisons **unidirectionnelles** « **SIMPLEX** », **half duplex** et **full duplex**.

I.5.1. Liaison unidirectionnelle « SIMPLEX »

Il s'agit ici vraiment de la plus simple liaison que l'on puisse trouver. La communication simplex est un mode de communication unidirectionnel, dans lequel chaque appareil est soit toujours émetteur soit toujours récepteur. Elle ne permet que la transmission dans un seul sens [2]. Ainsi l'information est transmise de l'émetteur vers le récepteur, supportée par un seul fil auquel s'ajoute la ligne de masse « Ground » comme le montre la figure I.7.

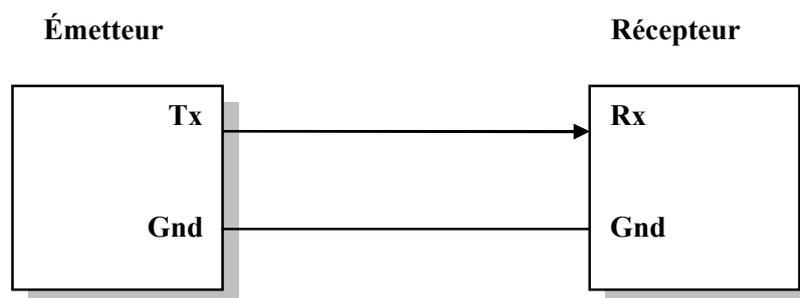


Fig. I.7. Exemple de liaison unidirectionnelle

Ce mode de communication est notamment utilisé quand il n'est pas nécessaire pour l'émetteur d'obtenir une réponse de la part du récepteur. Un circuit électronique comme un capteur qui envoie régulièrement et de manière autonome des données pourra utiliser une liaison simplex.

C'est aussi un mode de communication utilisé pour la diffusion, c'est à dire lorsqu'un même émetteur transmet simultanément à de nombreux récepteurs. Ainsi, la liaison entre un émetteur de télévision et les postes récepteurs est une liaison simplex.

Pour terminer sur un exemple encore plus simple, la télécommande de votre téléviseur communique avec ce dernier par une liaison simplex: quand vous pressez sur un bouton pour changer de chaîne, un train de signaux infrarouge est émis par la télécommande. Mais celle-ci est incapable de savoir si l'ordre a bien été reçu par le téléviseur ou pas. Cette solution est retenue car elle simplifie la conception du système et en réduit les coûts. Par ailleurs, aucun retour de type « accusé de réception » n'est nécessaire pour cette application: en effet, l'utilisateur est parfaitement en mesure de déterminer si la transmission s'est bien passée ou pas, et le cas échéant de ré-appuyer sur le bouton [3].

D'autres exemples de liaison simplex, on peut donner la liaison de :

- L'ordinateur vers l'imprimante.
- La souris vers l'ordinateur.

La figure I.8 donne le chronogramme de l'information sous la forme de trames envoyées à des temps aléatoires sur un seul sens.

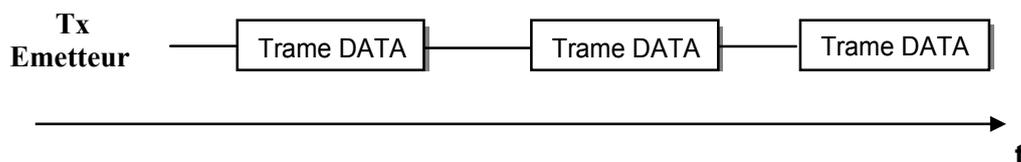


Fig. I.8. Chronogramme d'un exemple de liaison

I.5.2. Liaison bidirectionnelle (en Anglais: **Half-Duplex** ou **SEMI-DUPLEX**)

Ce type de liaison permet le dialogue dans les deux sens mais pas simultanément (donc à chacun son tour). Elle est également peu utilisée. En effet, seuls certains systèmes, lents ou ayant une unité de contrôle RS232 peu performante, fonctionnent encore uniquement dans ce mode comme le montre la figure I.9. [2]

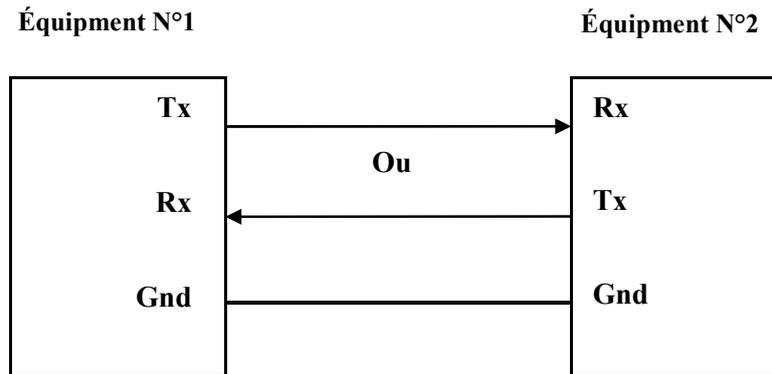


Fig. I.9. Exemple de liaison 'Half duplex'

L'avantage de ce système de communication par rapport au mode 'Full-duplex' est qu'il réduit par deux le nombre de canaux de communication nécessaires [3].

Par contre, il impose que les deux systèmes communicants soient en mesure de déterminer qui a le droit de parler. Dans le cas contraire, on risque d'avoir une collision (quand les deux systèmes tentent de parler simultanément) ou un blocage (quand les deux systèmes se mettent à l'écoute simultanément). De plus, un délai supplémentaire peut être induit lors du basculement du sens de communication d'une direction à l'autre [3].

Plusieurs stratégies sont possibles pour permettre aux deux systèmes de se coordonner. Par exemple, on peut envisager un multiplexage temporel dans lequel un timing précis indique à chacun le sens de la communication. Il peut également exister un canal de commande supplémentaire chargé d'indiquer à chaque périphérique s'il doit être en réception ou en transmission. Dans le même ordre d'esprit, un des deux systèmes peut être par défaut en réception et l'autre en émission, l'inversion ne se faisant qu'à la demande explicite du système en émission. Cette dernière solution s'approche des notions de maître esclave ou encore de jeton [3].

Un exemple de ce style de communication est le télégraphe Morse: celui-ci est constitué par une ligne électrique munie à chacune de ses extrémités d'un émetteur-récepteur. Quand un opérateur tape un message sur son manipulateur, les impulsions électriques sont transmises sur la ligne et produisent un son et/ou une transcription sur papier à l'autre extrémité. Comme c'est la même ligne qui achemine les signaux dans les deux sens, il s'agit bien d'un système 'Half-duplex' dans lequel les opérateurs doivent se coordonner pour ne pas transmettre simultanément [3].

La figure I.10 [2] donne le chronogramme de ce mode de liaison, il montre bien la façon

de fonctionnement de ce mode.

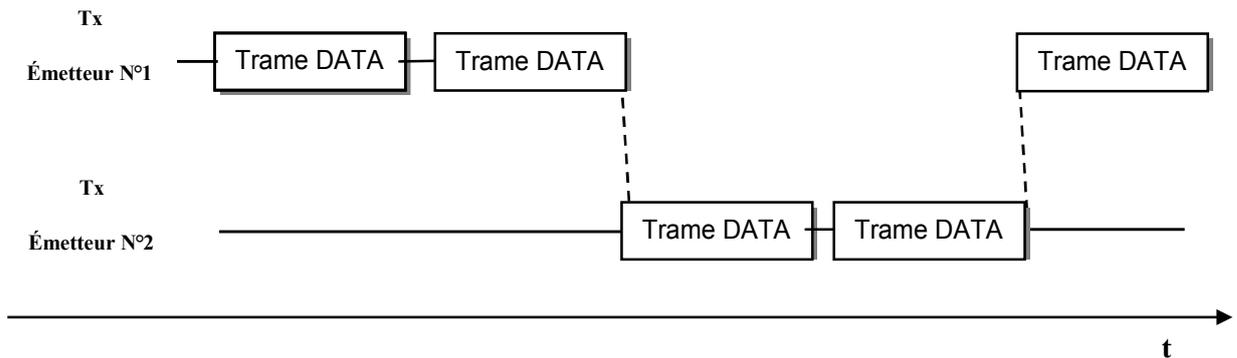


Fig. I.10. Chronogramme d'un exemple de liaison half duplex

I.5.3. Liaison directionnelle simultanée (Anglais: Full-Duplex) ou Plein-Duplex

Cette liaison dispose du même câblage que la liaison 'Half duplex' mais cette fois les deux interlocuteurs peuvent émettre et recevoir en même temps. Ce mode de transmission est asynchrone, les émissions et les réceptions n'ont pas à être synchronisées [2].

Dans la figure I.11, nous avons donné un exemple de liaison 'Full duplex' tel que l'équipement N°1 représente DTE et N°2 représente DCE.

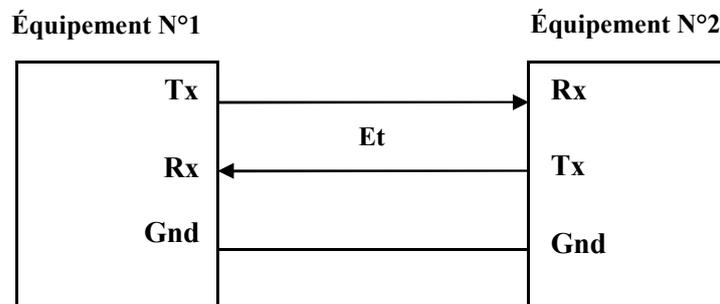


Fig. I.11. Exemple de liaison full duplex

Outre l'existence d'un canal de transmission dédié à chaque sens de communication, ce mode de communication exige aussi que chacun des deux systèmes soit capable de traiter à la fois des données entrantes et sortantes [3].

Un exemple simple est le téléphone: en effet, lors d'un appel, il est tout à fait possible aux deux correspondants de parler simultanément et de s'entendre l'un l'autre.

De la même manière, certains disques durs permettent de simultanément lire un fichier et en écrire un autre. Cette fonctionnalité requiert un bus de communication ‘full-duplex’ comme SAS (Serial Attached SCSI).

Dans ce cas l'émission de données indépendantes entre l'émetteur n°1 et le second est aléatoire comme le montre la figure I.12 [2].

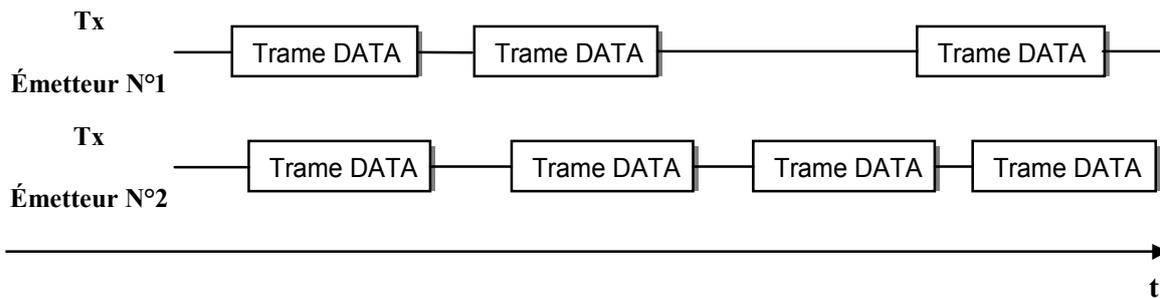


Fig. I.12. Chronogramme d'un exemple de liaison full duplex

I.6. DESCRIPTIONS TECHNIQUES DE LA NORME RS 232 [2]

I.6.1. Description électrique des signaux

Nous allons nous intéresser au niveau de tension présente sur la liaison, afin de mieux comprendre son fonctionnement [2], et nous le trouvons dans le tableau I.1.

Tableau I.2 : Équivalence niveaux logiques/ tensions

Niveau Logique	sur SUB-D du PC
« 0 » Low	+ 12 V (de + 5 à + 15 V)
« 1 » Hi	- 12 V (de - 5 à - 15 V)

On peut se rendre compte que les niveaux logiques et les tensions sont inversés. Le niveau haut est représenté par une tension négative et le niveau bas par une tension positive. On parle couramment des tensions + 12 V / - 12 V, mais la norme définit une fourchette allant de + 5 V / - 5 V à + 15 V / -15 V.

On constatera que sur les équipements, on utilise des tensions de plus en plus basses. Notamment sur les PC portables où désormais les tensions sont généralement de + 5 V / - 5 V.

Attention, certaines publications indiquent un minimum à + 3 V / - 3 V pour la RS232. Ces tensions correspondent aux seuils de comparaison à l'entrée des drivers. Par conséquent, si vous travaillez avec des tensions aussi basses, vous vous exposez à des aléas de fonctionnement causés par des parasites qui seront transformés en ligne par les drivers.

C'est pour descendre effectivement à + 3 V / - 3 V qu'a été créée la norme EIA232.

Normalement, chacune des sorties doit pouvoir fournir 20 mA, mais de nombreux constructeurs ont pris l'initiative d'abaisser la valeur du courant. Ne soyez donc pas surpris d'être limité à 10 mA sur votre PC, voire encore moins s'il s'agit d'un portable. De même, vous ne risquez pas la destruction de l'étage de sortie en cas de surcharge ou de court-circuit car le courant des signaux de sorties est limité.

Les tensions utilisées sont bipolaires, c'est-à-dire qu'elles sont référencées à la masse. Cette dernière est donc très importante pour garantir la qualité de la transmission. En général, il y a très peu de problèmes, car sur les PC, la masse est à la terre. Les choses sont un peu différentes lorsqu'il s'agit d'un PC portable sur batterie ou d'un PC n'étant pas connecté sur la même prise de terre. Il peut apparaître une différence de tension en mode commun entre les deux masses, ceci pouvant occasionner des erreurs de transmission et endommager les composants de communication.

La longueur du câble utilisé pour transmettre les données a son importance. À l'origine, la norme fixait une longueur maximale de câble de 8 mètres. Mais depuis peu, elle spécifie une impédance complexe pour ce câble. Le câble utilisé doit présenter une charge minimale de 3 k Ω et maximale de 7 k Ω , de même le câble ne doit pas posséder une charge capacitive supérieure à 2,5 nF. De par cette nouveauté, en choisissant bien le câble, on peut espérer transmettre sur une longueur d'environ 60 mètres.

De plus, la norme impose un 'slow rate' maximal de 30 V / μ s afin de limiter les effets électromagnétiques.

Le débit maximum est à présent défini à 20Kb/s (19 200 bauds), le futur débit maximal doit être de 64Kb/s (en préparation pour la norme Y.28).

Les débits que nous venons d'évoquer sont liés à la norme, le matériel quant à lui, est largement capable de dépasser le mégabit par seconde.

I.6.2. La trame RS232

Transmettre une donnée en mode série impose l'utilisation d'une structure de trame. Dans notre cas, elle est définie par la norme RS232 comme défini dans la figure I.13.

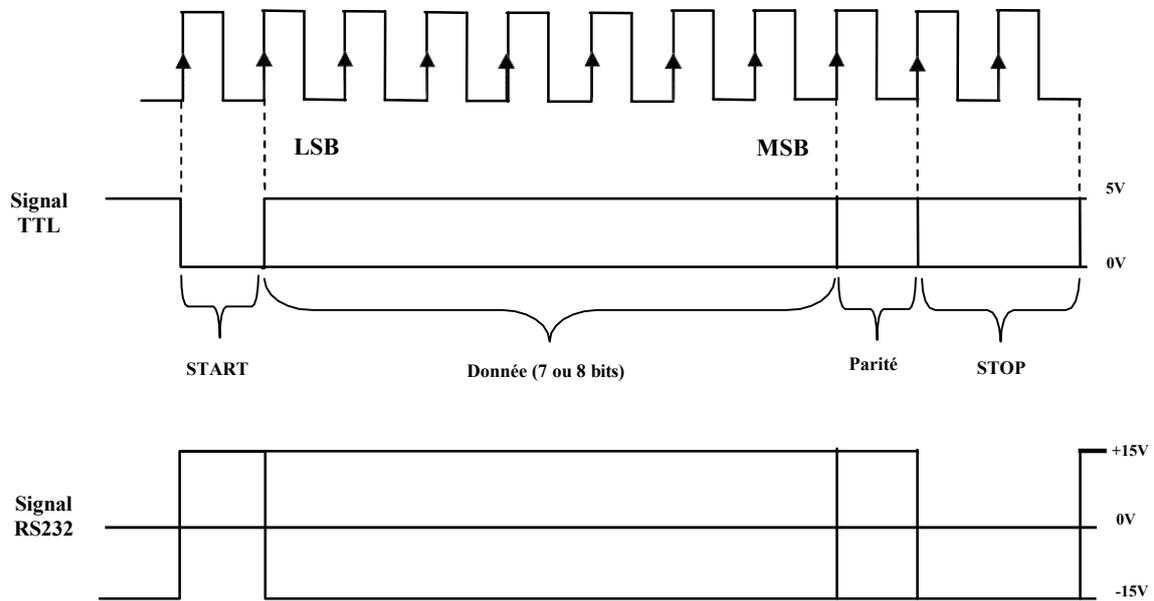


Fig. I.13. Chronogramme d'une trame de données (Trame DATA)

L'état initial vaut « 1 », c'est également l'état de repos, puis on émet le bit de START correspondant à un bit à «0».

Ensuite, on émet la donnée en commençant par le bit de poids le plus faible. Le nombre de bits peut aller, d'après la norme, de 5 à 8, mais usuellement, on utilise seulement des transmissions en 7 ou 8 bits, 7 bits pour ASCII et 8 bits pour les données. De plus, le bios du PC ne permet que ces des formats.

Nous donnons quelques exemples sur les trames de données dans la figure I.14 et la figure I.15.

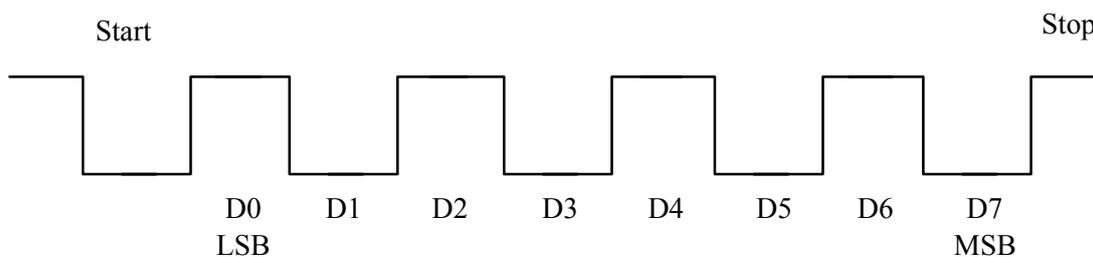


Fig. I.14. Chronogramme d'un exemple de trame de données/Valeur

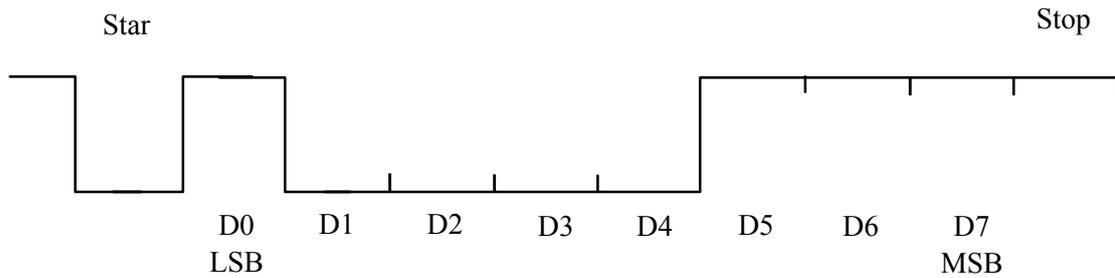


Fig. I.15. Chronogramme d'un exemple de trame de données/Valeur E1h/255 d, pas de parité, un bit de stop

La possibilité d'avoir ou non un bit de parité permet de vérifier la donnée. Son fonctionnement est simple. L'utilisateur choisit la parité paire ou la parité impaire. Ensuite, le bit de parité rend pair ou impair l'ensemble des bits (données + bit de parité).

Exemple N° 1 : 8 bits de données avec parité paire data = 55h/85d/01010101b donc un bit de parité qui vaut «0».

Exemple N° 2 : 8 bits de données avec parité impaire data = E1h/255d/11100001b donc un bit de parité qui vaut «1».

Enfin, on transmet 1 ou 1,5 ou 2 bits de stop. L'un des intérêts de faire varier la durée du bit de stop, est de laisser au récepteur d'avantage de temps pour être à nouveau prêt à recevoir une trame.

L'horloge n'est bien sûr pas la même des deux côtés de la liaison, car elle n'est pas transmise. Il s'agit en fait de deux horloges (une de chaque côté) qui sont synchronisées par le START de la trame et qui doivent avoir la même fréquence. Une tolérance de 5 % d'écart entre les deux fréquences est acceptée, au-delà, de nombreuses erreurs de réception sont à prévoir.

Cette horloge n'est pas définie en hertz par la norme, elle est indiquée en débit d'informations, c'est-à-dire en bauds.

L'utilisateur peut donc vérifier la vitesse de transmission à l'oscilloscope avec un octet facile à identifier.

La figure I.16 représenté durée d'un bit à 2400 Bauds par exemple de la transmission de l'octet: 55h / 85d / 01010101b.

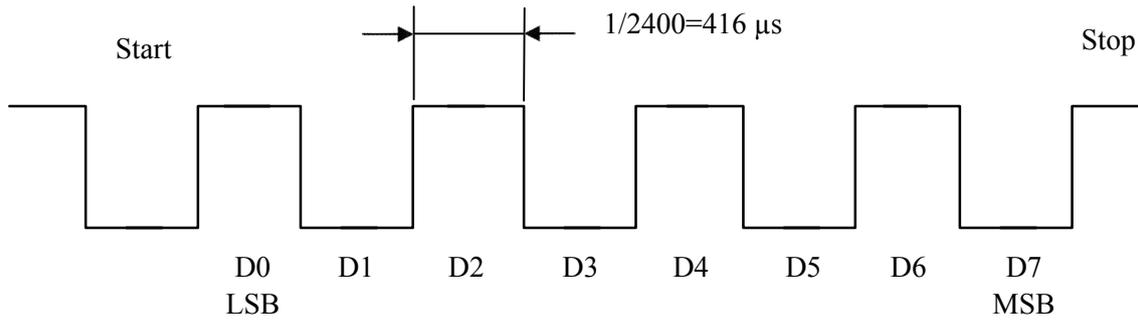


Fig. I.16. Durée d'un bit à 2400 Bauds. Exemple de la transmission de l'octet : 55h / 85d / 01010101b

Et le tableau I.2 représenté les différentes vitesses « Débits » en Baud et le temps de maintien d'un bit.

Tableau I.3 Tableau de correspondance entre la valeur de débit et le temps de maintien des données.

Débits en Bauds	Temps de maintien d'un bit
75	13,333 ms
150	6,666 ms
300	3,333 ms
600	1,666 ms
1 200	833 us
2400	416 us
4800	208 us
9600	104 us
19200	52 us

I.6.3. Fonction des signaux

On peut rassembler les signaux de la RS232 en deux groupes de fonctions. D'une part, les signaux de communication proprement dits (Tx, Rx), d'autre part, les signaux de 'Hand Shaking' (DTR, DSR, RTS, CTS, DCD et RI). Le premier groupe est indispensable à la communication. Le second est « optionnel » et seul, il ne peut servir à rien.

Dans le tableau ci-dessous (tableau I.3), on donne pour chaque signal: sa désignation définie par la norme, suivie d'une traduction interprétée, et enfin, entre parenthèse, la fonction remplie dans le protocole « Hand Shake ».

Tableau I.4 : Description fonctionnelle des signaux.

Nom	N° Broche SUB-D9	Sens pour un DTE	Désignation
Tx	3	Sortie	Transmit Data-Donnée émise
Rx	2	Entrée	Receive Data-Donnée reçue
DTR	4	Sortie	Data Terminal Ready-Terminal de données prêt (prêt à recevoir ?)
DSR	6	Entrée	Data Set Ready-Mise à disposition des données (donnée prête ?)
RTS	7	Sortie	Request To Send-Demande de transmettre (donnée à transmettre !)
CTS	8	Entrée	Clear To Send-Ouverture de transmission (prêt à recevoir !)
DCD	1	Entrée	Data Carrier Detect-Détection de message de données (liaison établie !)
RI	9	Entrée	Ring Indicator-Indicateur de communication (demande de comm. du DCE)

I.6.4. Interconnexion des équipements

A. Liaison complète DTE/DCE

À l'origine, la liaison RS232 a été décrite uniquement pour faire communiquer un ordinateur avec un périphérique. On a même prévu des appellations particulières pour l'ordinateur (DTE) et pour le périphérique (DCE). (Tableau I.5).

Un DTE désignait, à l'origine, uniquement les ordinateurs et les terminaux. À présent, on

y retrouve quasiment la totalité des équipements pourvus d'une liaison RS232.

Tableau I.5 : Signification DTE/DCE

Nom	Signification
DTE	Data Terminal Equipement
DCE	Data Communication Equipement

Un DCE désignait auparavant tous les périphériques comme les imprimantes, les traceurs, et les modems entre autres. Désormais, ce groupe est pratiquement délaissé, seuls certains modems sont encore en mode DCE.

Nous allons donc commencer par décrire le dialogue prévu par la norme à l'origine, et la figure I.17 montre la liaison complète entre un DTE et DCE.

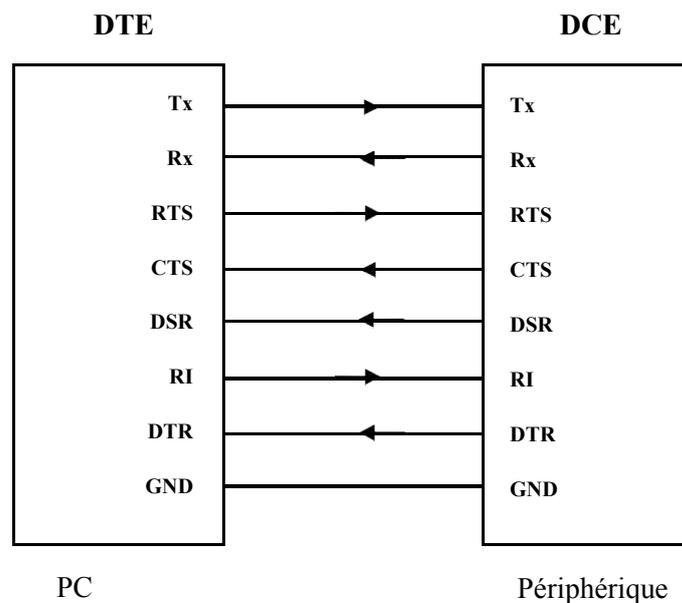


Fig. I.17. Liaison complète entre un DTE et DCE

Attention dans ce cas particulier, le sens des signaux d'un DCE est l'opposé de celui d'un DTE!

Cette manière de communiquer entre un DTE et un DCE peut être assimilée à un protocole qui est appelé « Hand Shake ».

Ce protocole permet de réguler le débit de transmission directement par le matériel et non par le logiciel. En effet, du point de vue du logiciel, ce protocole est transparent. Ce sont les UARTs qui gèrent le changement d'état des signaux.

B. Liaison complète DTE/DTE

De nos jours, les conditions d'utilisation ont beaucoup changé. Il ne s'agit plus de faire communiquer un DTE avec un DCE mais désormais, dans de nombreux cas, de faire communiquer deux DTE ensemble. (La figure I.18).

En effet, on est souvent amené à connecter deux PC ensemble ou par exemple à connecter un PC avec un appareil de mesure « intelligent » répondant aux spécifications d'un DTE.

Par conséquent, il faut adapter le câblage à la situation, car c'est la seule chose modifiable dans ce cas. C'est cette modification de câblage qui rend la communication toujours possible.

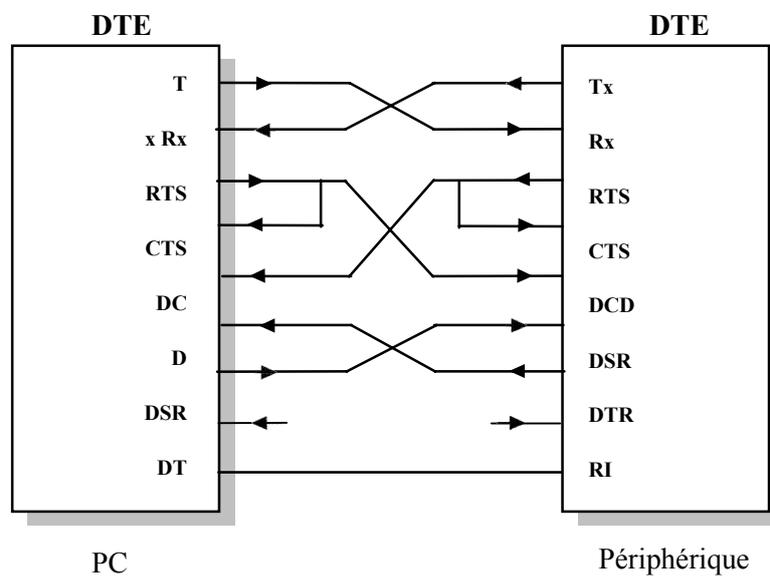


Fig. I.18. Liaison complète entre deux DTE

On remarque donc le croisement des différents signaux ainsi que le re-bouclage du RTS sur le CTS. Ce câblage est la seule manière de remplir le mode de fonctionnement de la liaison. De plus, le signal RI ne sera pas utilisé. Ce type de liaison porte le nom de Null Modem.

C. Liaison deux fils DTE/DTE

Il s'agit de la liaison la plus simple (voir la figure I.19). C'est une liaison unidirectionnelle appliquée à la RS232.

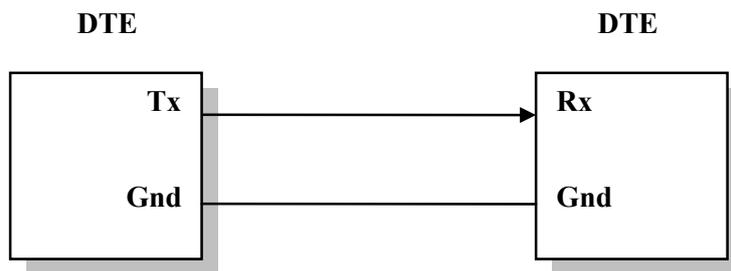


Fig. I.19. Liaison deux fils (DTE/DTE)

D. Liaison trois fils DTE/DTE

Ce type de câblage est sans doute le plus utilisé, car il répond au mieux à un besoin de facilité d'utilisation ainsi que de mise en œuvre. C'est également ce type de liaison qui est utilisé sur les microcontrôleurs. La figure I.20 explique cette liaison.

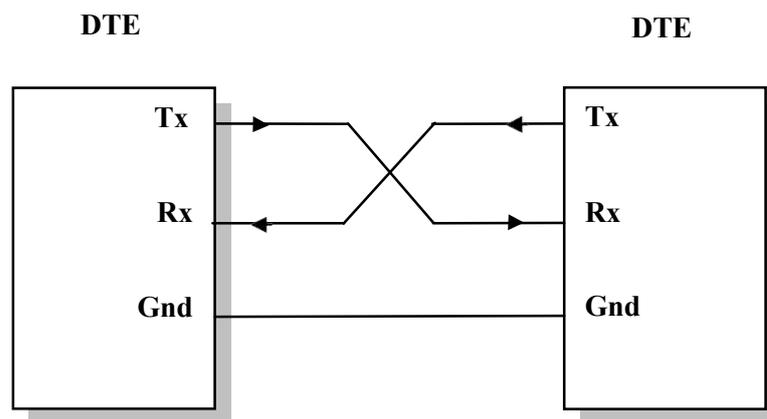


Fig. I.20. Liaison trois fils (DTE/DTE)

E. Liaison complète vers trois fils DTE/DTE

On peut être confronté au besoin d'interconnecter deux équipements différents. L'un est prévu pour fonctionner en mode liaison complète alors que l'autre ne peut communiquer qu'en trois fils.

La solution consiste à reboucler l'équipement « liaison complète » sur lui-même, la figure I.21 explique la liaison complète vers équipement trois fils et rebouclage les signaux.

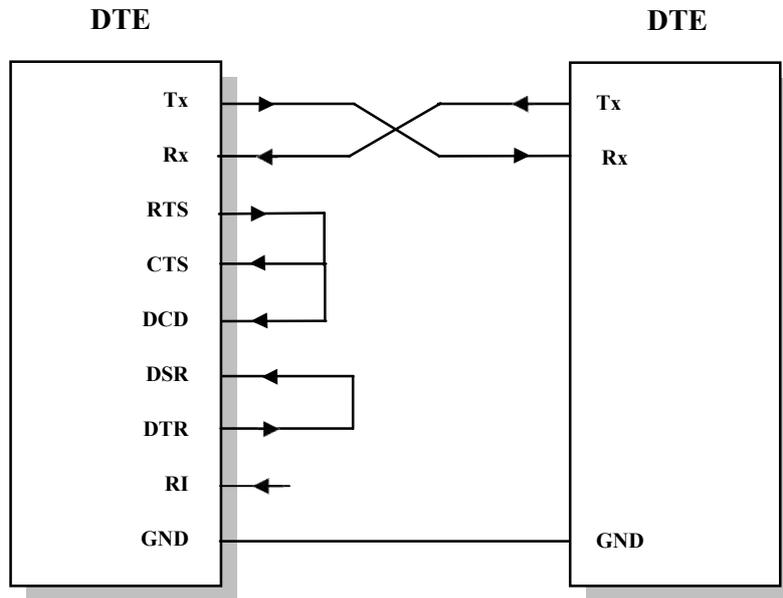


Fig. I.21. Liaison complète vert équipement trois fils (DTE/DTE)

I.7. Les protocoles

Lors de la mise en place d'une communication, nous sommes confrontés à un problème de régulation des données. Dans le premier cas de figure, la communication est parfaitement maîtrisée et aucun artifice n'est nécessaire. Dans un deuxième cas, le récepteur peut être occupé à d'autres tâches et doit pouvoir réguler lui-même le flux de données [2].

I.7.1. 'Hand shake' [1]

La méthode la plus courante, pour réguler le flux de données, est le 'Hand Shaking'. Elle utilise une liaison 'full duplex' complète. Ce protocole est géré directement par l'UART.

Vous trouverez sur le chronogramme suivant (Figure I.22), le mécanisme d'une communication.

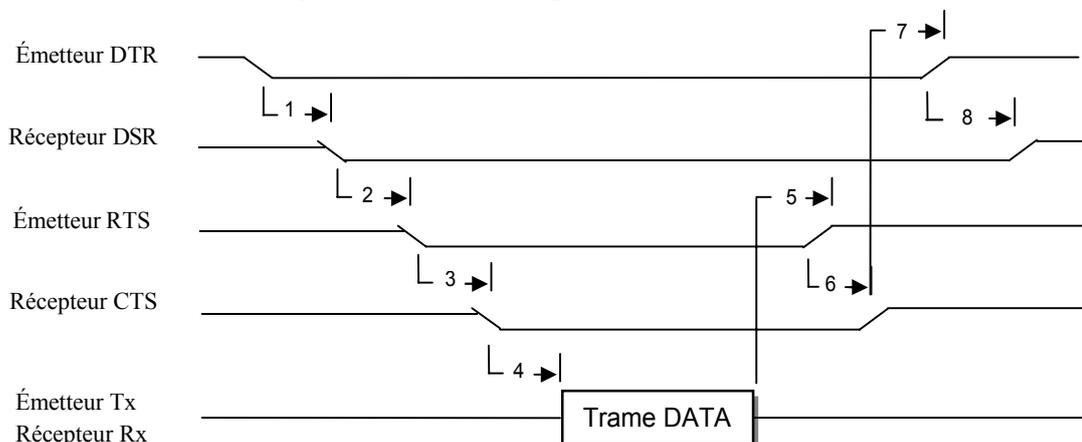


Fig. I.22. Chronogramme protocole matériel

Dans le tableau I.6, on explique le déroulement chronologique de ‘Hand Shaking’.

Tableau I.6 Déroulement chronologique de l'HandShaking.

1	L'émetteur demande si le récepteur peut recevoir des données (7)
2	Le récepteur répond, oui je peux recevoir des données (!)
3	L'émetteur demande s'il peut transmettre maintenant des données (7)
4	Le récepteur répond, oui je suis prêt maintenant (1)
Data	Transmission de la trame de données. Une fois la donnée transmise, on remet les signaux dans l'ordre initial tout en accusant réception des informations
5	Fin de transmission
6	OK fin de transmission
7	Fin de requête
8	OK Protocole terminé

Ce protocole n'a pas de contrainte de temps importante, car il s'agit d'une logique asynchrone détectant les changements d'état, non pas sur front mais sur niveau.

I.7.2. Xon-Xoff [1]

On peut également signaler, qu'une autre méthode que le ‘Hand Shaking’ est possible pour contrôler le flux des données.

Cette méthode est communément appelée Xon-Xoff. Elle utilise comme support physique une liaison ‘full duplex’ avec trois fils au minimum, et fonctionne de la manière suivante: l'émetteur transmet ses données au récepteur. Lorsque le récepteur est plein, il émet à l'émetteur un ‘Xoff’. L'émetteur par la réception de ce ‘Xoff’, arrête d'émettre les données et attend un ‘Xon’ de la part du récepteur pour reprendre l'émission, et la figure I.23 montre tout.

Dans la table de caractères, ASCII Xon est égal à 17 et Xoff à 19 en décimale.

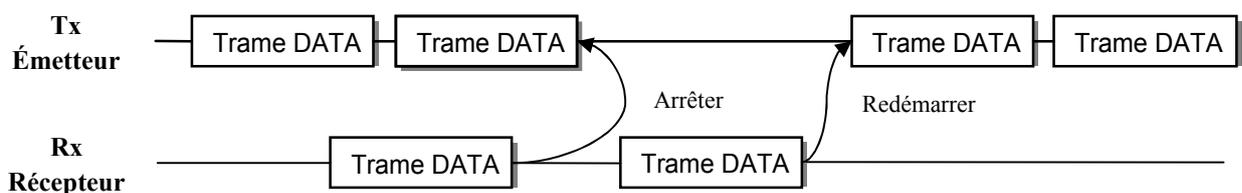


Fig.I.23. Chronogramme protocole Xon-Xoff

I.7.3. ETX/ACK [3]

Dans ce protocole, les données sont envoyées par blocs (trames). L'émetteur envoie un bloc de données auquel il ajoute le code ASCII 03h (appelé ETX pour End of Text) puis se met en attente. De son côté, le récepteur traite les données et renvoie à l'émetteur le code ASCII 06h (appelé ACK pour ACKnowledge) pour signaler qu'il a bien reçu le bloc de données précédent et qu'il est prêt à recevoir de nouveau des données. Le récepteur peut dans certains cas renvoyer le caractère ASCII 15h (appelé NACK pour Negative ACKnowledge) pour signaler qu'il a détecté une erreur dans la transmission.

I.8. CONCLUSION

La transmission des signaux numérique peut se faire en parallèle ou en série. Mais pour les longues distances c'est la transmission série qui prédomine. Parmi les deux types c'est la liaison asynchrone qui a un rapport direct avec notre réalisation



Chapitre II

Microcontrôleur

II.1. Introduction

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

II.2. Les avantages des microcontrôleurs

Les points forts des microcontrôleurs sont nombreux et bien réels. Il suffit, pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années.

Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui avant nécessitait une dizaine d'éléments séparés. 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'adresse et de données d'un composant à l'autre. [4]

Aussi le microcontrôleur permet :

- ✓ Diminution de l'encombrement du matériel et du circuit imprimé
- ✓ Simplification du tracé du circuit imprimé (plus besoin de tracer de bus !)
- ✓ Augmentation de la fiabilité du système
 - Nombre de composants diminués
 - Connexions composants, supports et composants circuit imprimé diminués
- ✓ Intégration en technologie MOS, CMOS, ou HCMOS Diminution de la consommation

- ✓ Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux:
 - Moins cher que les composants qu'il remplace
 - Diminution des coûts de main d'œuvre (conception et montage)
- ✓ Environnement de programmation et de simulation évolués

II. 3. Les caractéristiques principales d'un microcontrôleur

Les microcontrôleurs sont des composants qui permettent la gestion des cartes, ils sont caractérisés par :

- ✓ De nombreux périphériques d'E/S
- ✓ Une mémoire de programme
- ✓ Une mémoire vive (en général de type SRAM)
- ✓ Eventuellement une mémoire EEPROM destinée à la sauvegarde par programme de données à la coupure de l'alimentation.
- ✓ Un processeur 8 ou 16 bits
- ✓ Faible consommation électrique [4]

II. 3. 1. Différentes Familles de microcontrôleurs

On trouve plusieurs familles des microcontrôleurs citons :

- ☞ La famille Atmel AT91.
- ☞ La famille Atmel AVR.
- ☞ Le C167 de Siemens.
- ☞ La famille Hitachi H8 .
- ☞ La famille des PIC de Microchip.
- ☞ La famille des DSPIC de Microchip.
- ☞ La famille PSOC de Cyprées.
- ☞ La famille LPC21xx ARM7-TDMI de Philips.
- ☞ La famille V800 de NEC.
- ☞ La famille K0 de NEC.

II. 3. 2. Définition d'un PIC

Un PIC est un microcontrôleur, c'est une unité de traitement d'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes.

Les PICs sont des composants RISC (Reduced Instructions Set Computing) 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.

Alors, les microcontrôleurs 16F877 et 16F876A représentent le cerveau de notre projet, d'où nous allons les étudier [4].

II. 3. 3. Les différentes familles des PIC

La famille des PICs est subdivisée en trois grandes familles :

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

A cet effet, pour identifier un PIC, on utilise simplement son numéro :

16 : indique la catégorie du PIC, c'est un 'Mid-range'.

L : indique qu'il fonctionne avec une plage de tension beaucoup plus tolérante.

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

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

XX : représente la fréquence d'horloge maximale que le PIC peut recevoir.

Une dernière indication qu'on le trouve est le type de boîtier [5].

II. 3. 4. Le choix du microcontrôleur

Le choix d'un microcontrôleur est primordial car c'est de lui que dépendent en grande partie les performances, la taille, la facilité d'utilisation et le prix du montage. En fait ce choix est imposé dans le cahier de charge.

Le PIC 16F876A possède en plus des instructions très puissantes donc un programme à développer réduit, une programmation simple grâce au mode série. En fait la cause principale du choix du 16F876A est qu'il dispose de l'option du convertisseur A/D pour satisfaire le coté Acquisition de données.

II. 4. Le Microcontrôleur PIC16F876A

II. 4.1. Présentation

Le PIC 16F876A, utilisé est un circuit intégré de type CMOS. Son boîtier est un DIL (*Dual In Line*) de 2x14 pattes. Chacune de ces broches lui est associée a une ou plusieurs fonction.

En effet ces dernières peuvent jouer plusieurs rôles (entrée, sortie) tout dépend de leurs configuration qui s'effectue lors de la programmation du PIC [4].

Dans le tableau Tableau II.1 suivant, on montre les caractéristiques du PIC 16F876A.

Tableau II.1. Caractéristiques du PIC 16F876A

PIC	FLASH	RAM	EEPROM	I/O	A/D	PORT //	Port série
16F876A	8K mots de 14 bits	368 octets	256 octets	22	5	NON	USART/ MSSP

II.4.2. Brochage

Le 16F876A est un circuit intégré de 28 broches, que l'on peut trouver dans un boîtier PDIP.

28-Pin PDIP

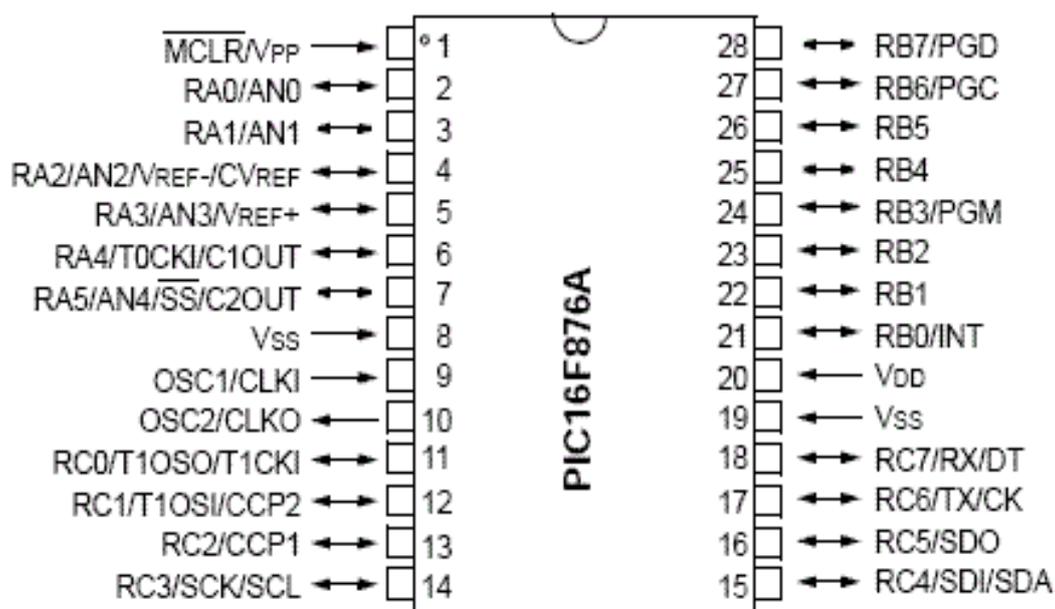


Figure II. 1: Brochage de PIC 16F876A

Le 16F876A possède jusqu'à 22 entrées/sorties :

- 6 dans le port A (RA0 à RA5)
- 8 dans le port B (RB0 à RB7)
- 8 dans le port C (RC0 à RC7)

On note qu'il y a deux broches de masse (broches 8 et 19) [6].

II. 4. 3. Les particularités électriques

On constate que sur le schéma concernant le 16F876A, on a 2 connexions « VSS » qui sont reliées à la masse. En fait, en interne, ces pins sont interconnectés. La présence de ces 2 pins s'explique pour une raison de dissipation thermique. Les courants véhiculés dans le PIC sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles.

Le constructeur a donc décidé de répartir les courants en plaçant 2 pins pour l'alimentation VSS, bien évidemment, pour les mêmes raisons (dissipation thermique), ces pins sont situées de part et d'autre du PIC, et en positions relativement centrales.

II. 4. 3. 1. Organisation du 16F876A

La mémoire du 16F876A est divisée en 3 parties. Pour mieux le connaître, nous regardons de près son schéma fonctionnel.

II. 4. 3. 1. 1. La mémoire programme

La mémoire programme est constituée de 8K mots de 14 bits. C'est dans cette zone que nous allons écrire notre programme.

II. 4. 3. 1. 2. La mémoire EEPROM

La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory), est constituée de 256 octets que nous pouvons lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi permanents.

II. 4. 3. 1. 3. La mémoire RAM et organisation

La mémoire RAM est celle que nous allons sans cesse utiliser. Toutes les données qui y sont stockées sont perdues lors d'une coupure de courant. La mémoire RAM disponible du 16F876 est de 368 octets. Elle est répartie de la manière suivante:

- ☑ 80 octets en banque 0, adresses 0x20 à 0x6F
- ☑ 80 octets en banque 1, adresses 0xA0 à 0XEF
- ☑ 96 octets en banque 2, adresses 0x110 à 0x16F
- ☑ 96 octets en banque 3, adresses 0x190 à 0x1EF
- ☑ 16 octets communs aux 4 banques, soit 0x70 à 0x7F = 0xF0 à 0xFF ; 0x17F=0x1F0 à 0x1FF.

II. 4. 3. 1. 4. ‘Watch dog’

Sous ce nom étrange nous allons découvrir une fonction capable de surveiller le bon fonctionnement du programme que le microcontrôleur exécute. Le rôle du ‘Watch dog’ (ou chien de garde) est de "reseter" le microcontrôleur si l'on ne remet pas à zéro périodiquement (à intervalle définissable) un registre interne grâce à l'instruction ‘clrwdt’ (clear watchdog), si le programme tourne par exemple dans une boucle sans fin (c'est un bug !) il ne peut remettre à 0 le chien de garde et ainsi le microcontrôleur se ‘reset’ afin de relancer le programme. Cette fonction est bien sûr désactivée au moment de la programmation du microcontrôleur, c'est la directive d'assemblage.

II. 4. 3. 1. 5. Le TIMER

Un timer est un registre interne au microcontrôleur, celui-ci s'incrémente au grès d'un horloge, ce registre peut servir par exemple pour réaliser des temporisations, ou bien encore pour faire du comptage (par l'intermédiaire d'une broche spécifique : RA4/TOKI). Le PIC 16F876 possède trois timers sur 8 bits (il compte jusqu'à 256) configurable par logiciel.

II. 4. 3. 2. Les ports entrée/sortie

On dispose de 22 broches d'entrées/sorties, chacune configurables soit en entrée soit en sortie (PORTA, PORTB, PORTC).

Un registre interne au PIC, nommé TRIS, permet de définir le sens de chaque broche d'un port d'entrées/sorties. En règle générale, un bit positionné à « 0 » dans le registre TRIS donnera une configuration en sortie pour la broche concernée ; si ce bit est positionné à « 1 », ce sera une broche d'entrée [7].

II. 4. 3. 2. 1. Particularité du port A

Le 16F876 dispose de 5 canaux d'entrée analogique. Nous pouvons donc échantillonner successivement jusque 5 signaux différents avec ce composant. Les pins utilisés sont les pins AN0 à AN4 (qui sont en fait les dénominations analogiques des pins RA0 à RA3 + RA5).

On peut noter également que les pins ANx sont des pins d'entrée. Il n'est donc pas question d'espérer leur faire sortir une tension analogique. Ceci nécessiterait un convertisseur numérique/analogique dont n'est pas pourvu notre PIC.

II. 4. 3. 2. Particularités du port B

Hors de sa fonction principale autant que ports d'entrées /sorties, on note le pin RB0 qui, en configuration d'entrée, est de type « trigger de Schmitt » quand elle est utilisée en mode interruption « INT » ; La lecture simple de RB0 se fait, elle, de façon tout à fait classique, en entrée de type TTL. Encore il y a (RB3-RB6-RB7) qui peut servir dans la programmation (en mode LVP) en cas d'absence de programmeur commercial.

II. 4. 3. 2. 3. Particularités du port C

C'est un port tout ce qu'il y a de plus classique, or qu'il a deux pins qu'on utilisera plus tard dans la communication série avec le PC à travers (TX et RX) (pin17 et 18).

II. 4. 3. 3. L'oscillateur

L'horloge système peut être réalisée soit avec un quartz (a), soit avec une horloge extérieure (b), soit avec un circuit RC(c). Dans ce dernier cas, la stabilité du montage est limitée.

La fréquence maximale d'utilisation va dépendre du microcontrôleur utilisé. Le suffixe indiqué sur le boîtier donne la nature de l'horloge à utiliser et sa fréquence maximale.

II. 4. 3. 4. MCLR

La broche MCLR permet de réaliser un Reset du circuit quand elle est placée à 0V

II. 5. Conclusion

Nous avons présenté dans ce chapitre les notions du microcontrôleur en évoquant leurs avantages ainsi que leur différente caractéristique.

Ensuite, nous avons pris le PIC16F876A comme titre d'exemple où nous avons présenté leurs différentes compositions.

Dans ce que suit nous allons évoquer l'étude théorique de la carte d'acquisition que nous souhaitons réaliser.

Chapitre III

Etude Théorique de la Carte

III.1. Introduction

D'après les objectifs qu'on a fixés, et d'un point de vue fonctionnel, le rôle de notre carte d'acquisition permettra de commander jusqu'à huit appareils et de lire autant de lignes digitales, ainsi que deux lignes analogiques, en utilisant le port série de n'importe quel ordinateur compatible PC.

On va étudier théoriquement dans ce chapitre les éléments qui rassemble notre carte, et qui on peut le voir dans le schéma synoptique suivant :

III. 2. Schéma Synoptique du Projet

Commençons d'abord par donner le schéma synoptique de notre carte D'acquisition.

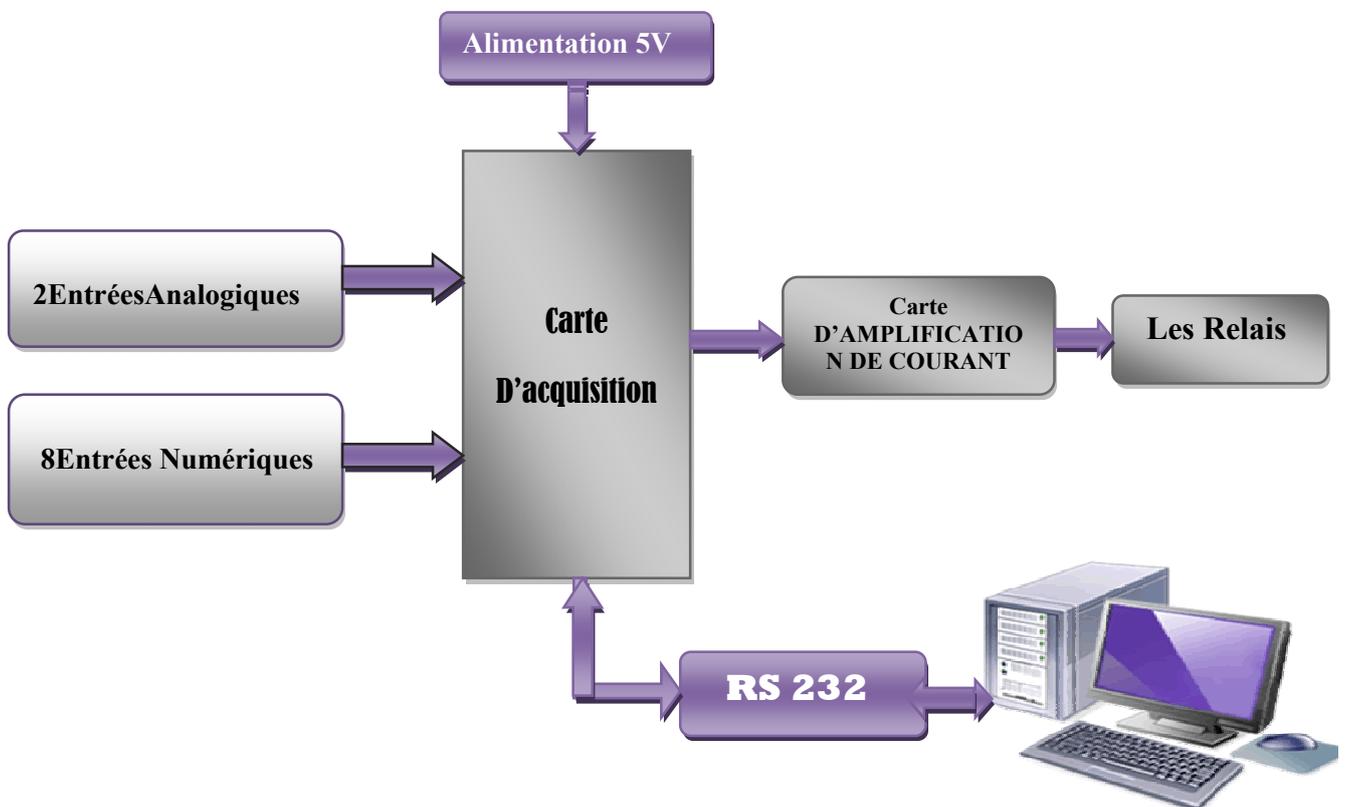


Fig. III. 1. Schéma synoptique de fonctionnement du projet à réaliser

III. 3. Éléments de la carte d'acquisition

Comme illustré sur le synoptique précédent, la carte d'acquisition doit rassembler :

- Les Entrées Numériques
- Les Entrées Analogiques

- un microcontrôleur PIC16F876A.
- Un port série pour la communication avec le PC (RS232).
- Amplification du Courant.
- Les Relais.
- Circuit d'alimentation 5V.

III. 3. 1. Les Entrées Numériques

Nous avons utilisés 08 entrées numériques. Les entrées sont protégées par une diode Zener, qui élimine la surtension et la tension négative. Bien que cette protection est basique mais elle suffit amplement pour notre système. Une protection plus perfectionnée peut être mise-en place en assurant l'isolation galvanique du circuit.

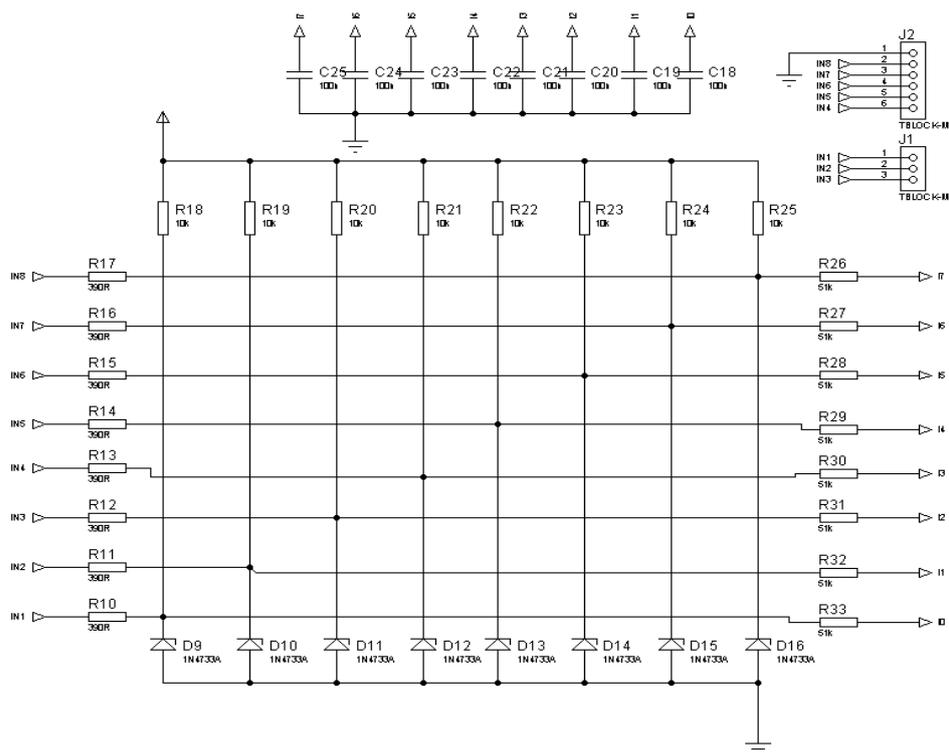


Fig. III. 2. Schéma d'implantation des entrées

Il est rappelé que le port C, en entrée, ne dispose pas de résistance ‘pull up’ comme le port B. Cependant on a choisit précisément ce port comme entrée car il dispose de trigger de Schmidt ce qui complète notre Bloc d’entrée.

III. 3. 2. Les Entrées analogiques

Les entrées analogiques, sont AD1 et AD2, et sont protégées par les résistances, qui limitent le courant consommé par les diodes de protection interne au microcontrôleur, au cas où une tension inverse lui serait appliquée.

Rappelons, que le système peut lire des tensions continues de 0 à 5 volts et que la conversion analogique/numérique (A/D) est effectuée avec une résolution de 8 bits, donc sur 256 niveaux.

C'est un degré de précision, qui nous semble plus que suffisant pour la majorité des applications.

III. 3. 3. LE MICROCONTROLEUR 16F876A

L'élément fondamental de la carte d'acquisition est un microcontrôleur PIC16F876A qui transforme les signaux analogiques en données numériques et qui indique les niveaux de liquide (surveillance du niveau par exemple) à l'aide d'un système d'affichage intégré. Le PIC 16F876A est un microcontrôleur à 14 bits. 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.

III. 3. 4. LA LIAISON SERIE RS232

Les liaisons série permettent la communication entre deux systèmes, numérique en limitant les nombres de fils de transmission. La liaison série aux normes RS232 est utilisée dans les domaines de l'information (ex : port de communication COM1 et COM2 des PC, permettant la communication avec des périphériques tels que modem et souris). La transmission série nécessite un minimum de 2 fils de connexion comportant les trames de données en émission (Tx) et en réception (Rx) [8].

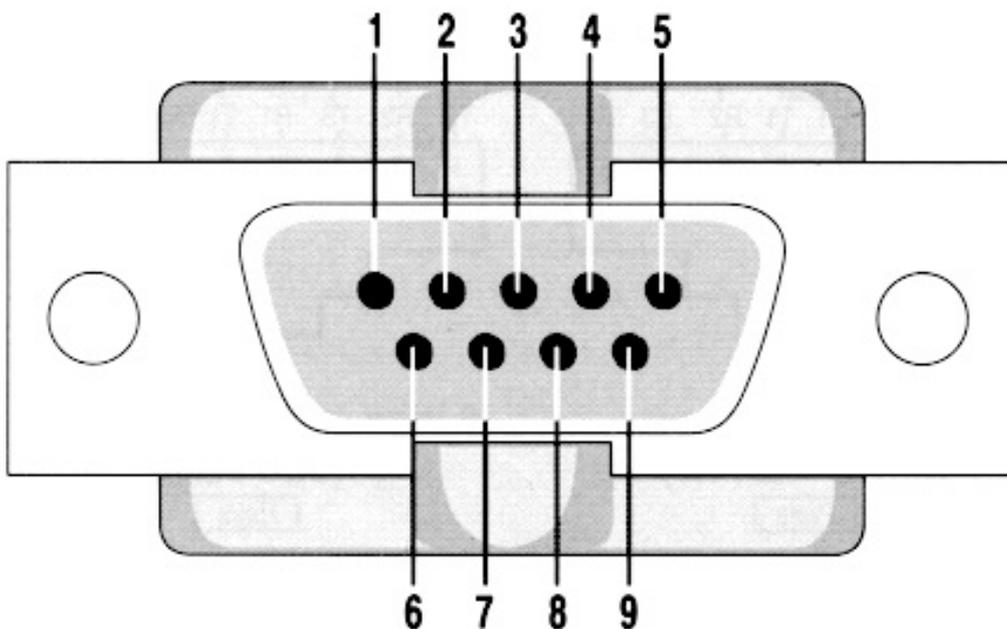
III. 3. 4. 1. LA COMMUNICATION SERIE

Dans une transmission série, les informations numériques sont transmises bit après bit, à une fréquence de transmission égale; la transmission série est donc plus lente que la transmission parallèle par contre nous allons voir clairement ce quelle nous rapporte de bien. La difficulté majeure de ce type de transmission est la synchronisation par l'horloge des échanges entre émetteur et récepteur. L'horloge délivre un signal périodique qui cadence la transmission des bits. On peut distinguer deux familles de transmission série :

➤ **La transmission série synchrone :**

Elle impose l'envoi des signaux de synchronisation par l'émetteur. Cependant pour que le récepteur effectue une bonne lecture, il doit se référer à chaque fois à ces signaux ; le problème c'est qu'elle nécessite le transfert de deux informations, non seulement la donnée elle-même mais aussi l'horloge de synchronisation. Pour cela ce type de transmission n'est pas répandu aux milieux industriels à cause de son taux élevé d'erreur ainsi que sa complexité [8].

La figure suivante illustre une configuration d'un port de communication série synchrone (SUB-DB9).



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request To Send
3	Transmitted Data	8	Clear To Send
4	Data Terminal Ready	9	Ring Indicateur
5	Signal Ground		

Fig. III. 3. Brochage de la SUB-DB9

➤ *La transmission série asynchrone :*

Ce type de communication série impose l'horloge du récepteur à la même fréquence que l'émetteur. (voir la figure III.5).

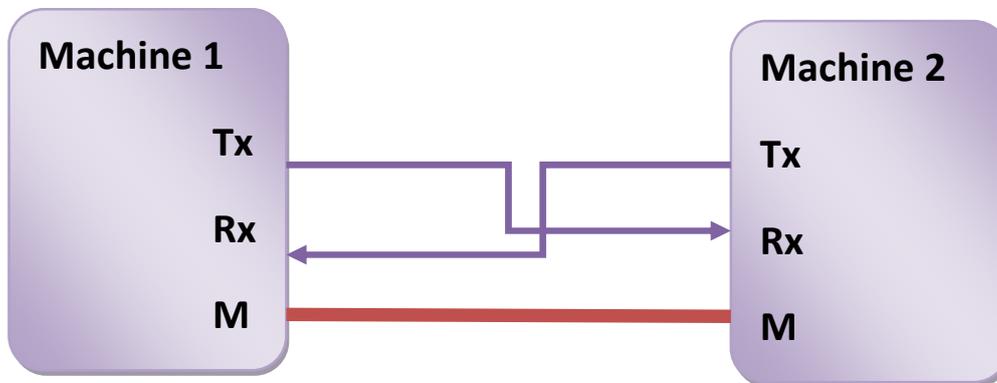


Fig. III. 4. schéma synoptique de la transmission série asynchrone

L'octet à transmettre est envoyé bit par bit (poids faible en premier) par l'émetteur sur la ligne Tx, vers le récepteur (ligne Rx) qui le reconstitue. La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses sont exprimées en BAUDS (1 baud=1 bit/seconde). Il existe différentes vitesses normalisées : 9600, 4800, 2400, 1200, ... bauds.

III. 3. 4. 2. LA PROCEDURE DE TRANSMISSION ASYNCHRONE

La procédure de transmission se fait suivant un ensemble de paramètres sur lesquels l'émetteur et le récepteur sont mis en accord. La transmission sur la liaison RS232 étant du type asynchrone (pas d'horloge commune entre l'émetteur et le récepteur), des bits supplémentaires sont indispensables au fonctionnement : bit de début de mot (Start) et les bits de fin de mot (c'est le 'Stop'). D'autre part, l'utilisation éventuelle d'un bit de parité permet la détection d'erreurs dans la transmission. Le chronogramme suivant représente les principaux constituants de la transmission série [8].

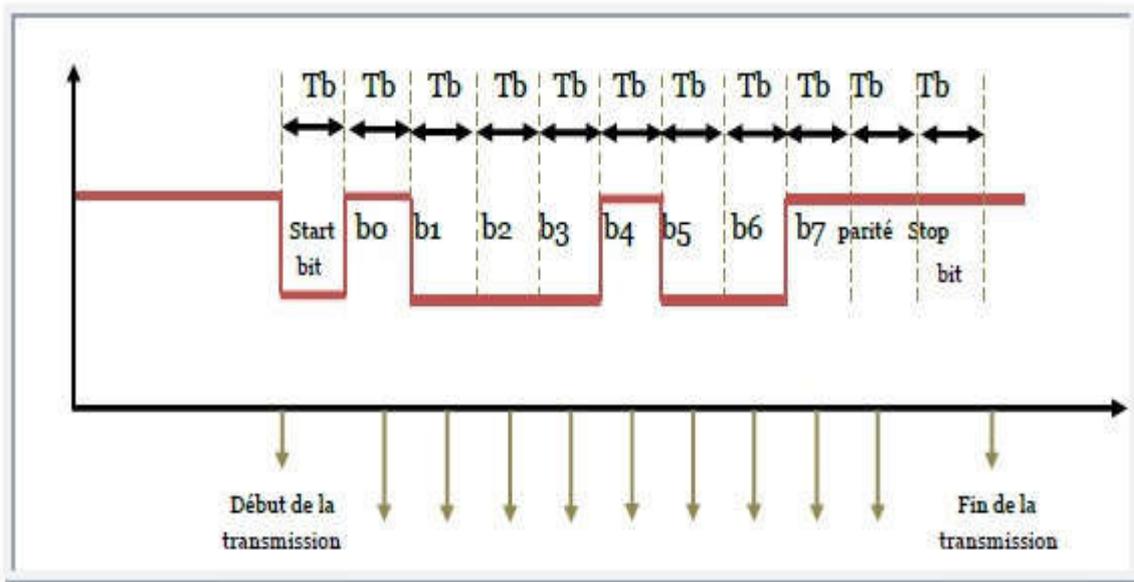


Fig. III. 5. Chronogramme de la transmission série asynchrone

Dans ce chronogramme, on a :

- **LE BIT 'START' :** Pour démarrer la transmission (la transition niveau haut-niveau bas de la ligne indique au récepteur qu'un caractère est émis, cela permet au récepteur de « caler » son horloge de réception. Le caractère est émis suivant un type de codage (ASCII :7 bits, IBM :8 bits, Téléx :5 bits...).
- **LE BIT DE PARITE (FACULTATIF):** Il est généré à l'émission et testé par le récepteur. Il permet de détecter les erreurs de transmission.
- **LE BIT 'STOP' :** La fin de l'émission d'une trame est indiquée par l'envoi de un ou deux bits de « STOP » qui sont toujours à l'état haut. Toutes ces données (bit Start, Stop, Parité, Vitesse ...) font partie du protocole de communication qui doit être naturellement le même pour l'émetteur et le récepteur, il faut noter qu'une fois que l'horloge du récepteur sera synchronisé par le bit Start, la détection d'un un (1) ou d'un zéro (0) se fera à chaque front montant de l'horloge.
- **LA PARITE :** La parité est une technique qui permet de vérifier que le contenu d'un mot n'a pas été changé accidentellement lors de sa transmission. L'émetteur compte le nombre de '1' dans le mot et met le bit de parité à '1' si le nombre trouvé est impaire, ce qui rend le total paire c'est la parité paire. On peut aussi utiliser la parité impaire. Ainsi, en parité paire, si à la réception il y a un nombre impaire de bits reçus, c'est qu'il y a eu une erreur de transmission (et l'émetteur devra réémettre la trame).

- **LES BITS DE DONNEES** : Ce sont les bits qui portent l'information, ils peuvent être de 4 jusqu'à 8, d'ailleurs c'est la valeur max que permet le bios.

III. 3. 4. 3. LE MAX232

C'est un circuit (driver de bus) chargé de convertir les niveaux des signaux entre PIC et PC.

❖ STRUCTURE EXTERNE DES MAX 232 :

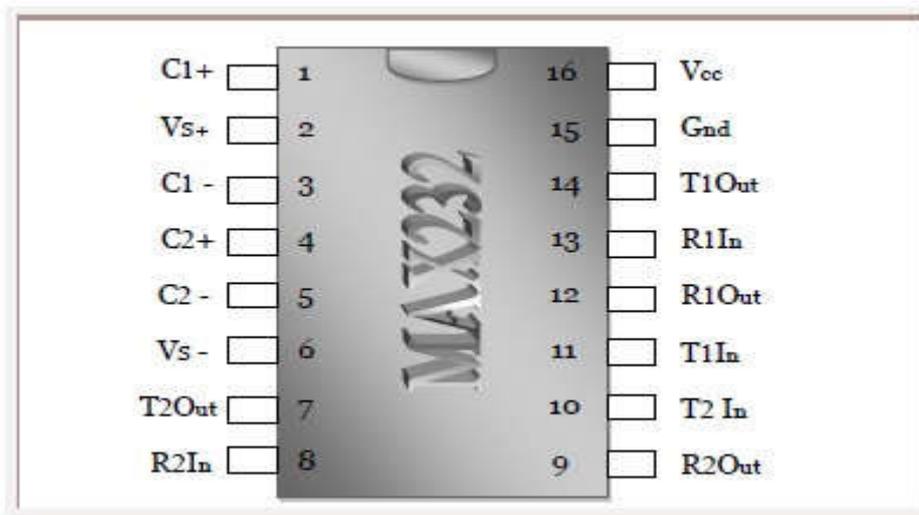


Fig. III. 6. Brochage du MAX232

Ce circuit contient un double convertisseur à double direction. Autrement dit, il dispose de :

- 2 blocs, dénommés T1 et T2, qui convertissent les niveaux entrés en 0V/5V en signaux sortis sous +12V/-12V. En réalité, on n'a pas tout à fait +12V et -12V, mais plutôt de l'ordre de +8,5V/-8,5V (en théorie +10V/-10V), ce qui reste dans la norme RS232.

Les entrées de ces blocs sont donc dirigés vers le PIC, les sorties sont connectées sur le port RS232.

- 2 blocs, dénommés R1 et R2, qui convertissent les niveaux entrés en +12V/-12V en signaux sortis sous 0V/5V. Les entrées de ces blocs sont donc connectées sur le port RS232, les sorties sur le PIC [10].

Il s'agit donc de lignes de sélection, que nous pourrions gérer, si le besoin s'en fait sentir, via des lignes I/O classiques de notre PIC. Le double convertisseur du MAX232 s'explique d'ailleurs.

III. 3. 5. Amplification du Courant

III. 3. 5. 1. Les Sorties

Les sorties sont constituées de 8 relais de 12V (réf. :) commandés par un composant de puissance 'ULN 2803', qui leur fournit le 0 V. Tous les relais sont connectés au +12V.

Chaque relais est mis en parallèle avec un indicateur constitué d'une LED et une résistance.

Le besoin de chaque relais ainsi que de son indicateur est au environ de 60 mA (voir FS Etude relais 01/02). Le PIC ne peut débiter un tel courant (voir data sheet), l'utilisation du 'ULN 2803' est justifiée, sachant que 'ULN 2803' peut débiter jusqu'à 500 mA par sortie.

Pour avoir le 0 V aux pins de sortie de 'ULN 2803' il faut appliquer une tension de 5V aux entrées correspondantes. Il faut rappeler que les entrées de 'ULN 2803' sont des entrées TTL (voir data sheet).

III. 3. 5. 2. LE CIRCUIT INTÉGRÉ ULN2803

En fait, l'ULN2803 est tout simplement un réseau de 8 transistors NPN Darlington intégrés dans un boîtier de 18 broches. Et ce petit boîtier tout intégré va nous faire gagner de la place sur notre circuit imprimé, du temps de réalisation (18 broches à souder contre 24 avec des transistors indépendants) [10].

C'est le compagnon idéal pour alimenter une série de LED haute intensité, des relais, un gros afficheur 7 segments, ... ou tout autres choses dans le même esprit. Il permet même l'utilisation d'une alimentation externe

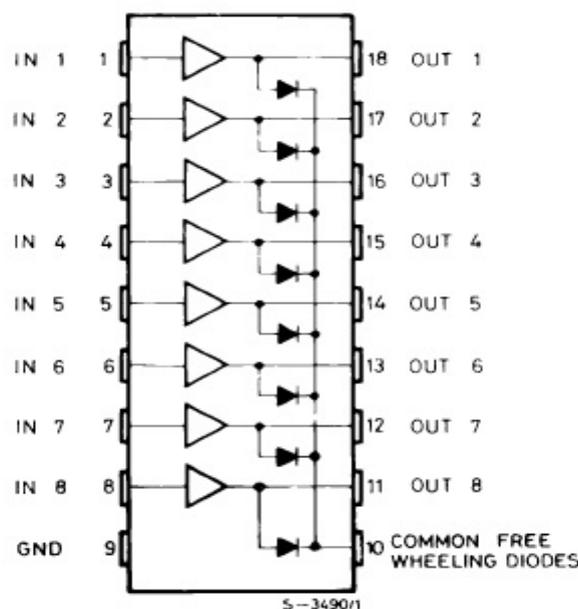


Fig. III. 7. Schéma interne ULN2803

III. 3. 6. Les Relais

Un relais électronique est un interrupteur qui se commande avec une tension continue de faible puissance. Ainsi, un relais permet l'ouverture et la fermeture d'un circuit électrique de puissance à partir d'une information logique. La partie interruptrice sert à piloter des charges secteur de forte puissance (jusqu'à 10 A couramment). [12].

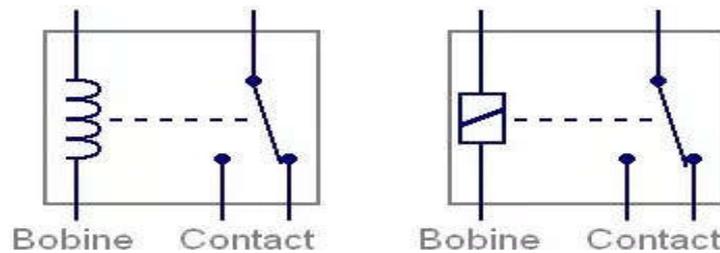


Fig. III. 8. Symbole du relais

Le symbole du relais doit inclure [12] :

- quelque chose qui représente la bobine
- quelque chose qui représente le ou les interrupteurs (contacts).

III. 3.7. Circuit d'alimentation

Tout montage électronique nécessite une alimentation pour fonctionner. Notre montage Nécessite une alimentation 5V pour alimenter le PIC et ces périphériques.

Autre contrainte présente dans ce circuit est qu'il peut être, dans certains cas, alimenté par une batterie dans des zones qui ne sont pas couvertes par le réseau électrique.

Pour ces raisons nous avons optés pour une alimentation externe, qui sera régulée et stabilisée vers les tensions nécessaires.

Le circuit d'alimentation reste un classique dans le genre, un régulateur de tension de type LM7805 nous fournit une tension positive de 5V avec un courant maximum de 1A.

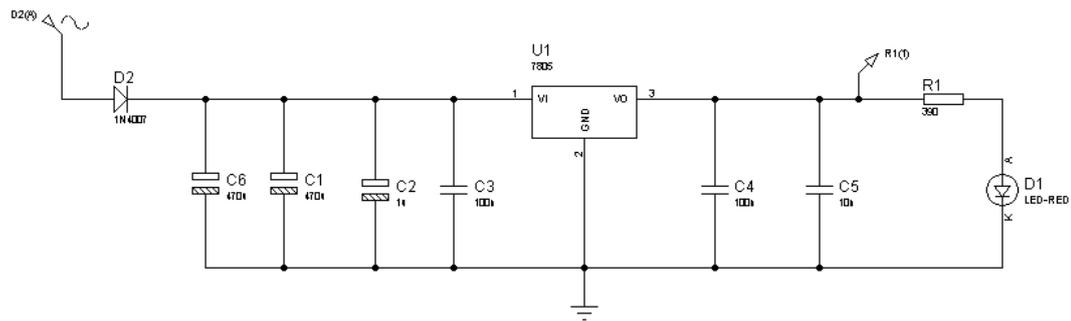


Fig. III. 9-. Montage électrique de l'alimentation

III. 4. Conclusion

Dans ce chapitre on a vu les différents éléments constituant la carte d'acquisition réalisée, leur fonctionnement ainsi que le principe de la transmission série.

Nous verrons dans le chapitre suivant la partie pratique et les essais obtenus ainsi que les résultats de notre réalisation.



Chapitre IV

Réalisation Pratique

IV. 1. Introduction

Après avoir décrit les différents composants de notre système qui permet la possibilité de commander jusqu'à huit appareils et de lire autant de lignes digitales, ainsi que deux lignes analogiques, nous passons maintenant à la phase de la réalisation pratique. Cette phase est composée d'une partie électronique (la carte électronique a base d'un PIC 16F876A), et d'une autre partie de programmation (les logiciels à base des langages de programmation).

IV.2. PARTIE ELECTRONIQUE (HARDWARE)

IV. 2. 1. Le Choix des Composants

Le choix des composants se fait en fonction de plusieurs critères et les principales caractéristiques qui vont influencer ce choix sont :

- l'importance de la gamme de circuit.
- la vitesse de fonctionnement.
- l'immunité aux bruits.

Les éléments constituant la carte sont:

IV. 2. 1. 1. Régulateur 7805

Régulateur de tension positive sous forme de circuit intégré à trois broches (Fig. IV. 1.) est un élément qui permet de stabiliser une tension à une valeur fixe, et est nécessaire pour les montages électroniques qui ont besoin d'une tension qui ne fluctue pas. Ce genre de circuit est très simple d'utilisation : il suffit de brancher une broche à la masse, une autre vers la tension d'alimentation, et on obtient en sortie une tension régulée. Il est toutefois généralement nécessaire d'ajouter quelques condensateurs à l'entrée et en sortie servant de réservoir d'énergie.

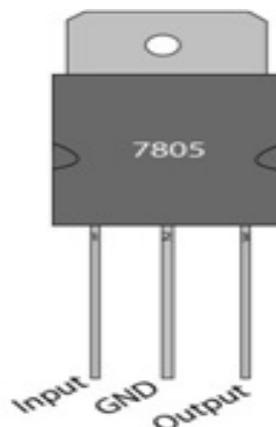


Fig. IV. 1. Régulateur 7805

Dans notre carte le régulateur reçoit environ 12 volts redressés et filtrés et génère du 5 volts stabilisé.

IV. 2. 1. 2. Diode 1N4007

La diode (Fig. IV. 2) est un composant électronique. C'est un dipôle non linéaire et polarisé (ou non-symétrique). Le sens de branchement d'une diode a donc une importance sur le fonctionnement du circuit électronique dans lequel elle est placée.

Sans précision ce mot désigne un dipôle qui ne laisse passer le courant électrique que dans un sens. Ce dipôle est appelé diode de redressement lorsqu'il est utilisé pour réaliser les redresseurs qui permettent de transformer le courant alternatif en courant unidirectionnel [13].

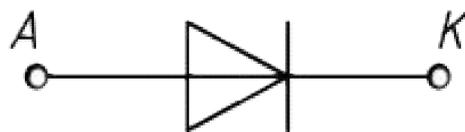


Fig. IV. 2. Symbole de Diode

IV. 2. 1. 3. Diode Zener :

Une diode Zener (Fig. IV. 3) est un assemblage de deux semi-conducteurs dont les propriétés électriques ont été découvertes par le physicien américain Clarence Zener. Contrairement à une diode conventionnelle qui ne laisse passer le courant électrique que dans un seul sens, le sens direct, les diodes Zener sont conçues de façon à laisser également passer le courant inverse, mais ceci uniquement si la tension à ses bornes est plus élevée que le seuil de l'effet d'avalanche. Ce seuil en tension inverse (tension Zener) est de valeur déterminée pouvant aller de 1,2 V à plusieurs centaines de volts. Certaines diodes Zener comportent une troisième broche qui permet de régler cet effet d'avalanche [14].

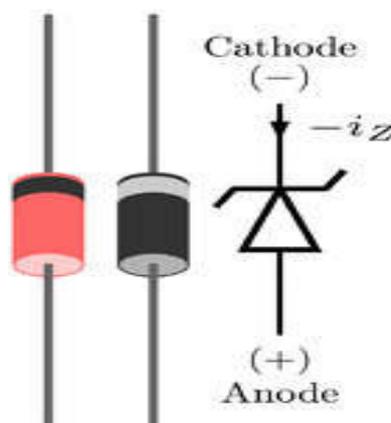


Fig. IV. 3. Symbole de Diode Zener

Dans notre cas les diodes Zener protègent le microcontrôleur contre d'éventuelles surtensions et garantissent une protection efficace, même dans le cas où la tension serait appliquée avec une polarité inversée.

IV. 2. 1. 4. LEDs (light Emitting Diodes)

Une diode électroluminescente (Fig. IV. 3) est un dispositif optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.

Une diode électroluminescente ne laisse passer le courant électrique que dans un seul sens et produit un rayonnement monochromatique ou poly chromatique non cohérent à partir de la conversion d'énergie électrique lorsqu'un courant la traverse.

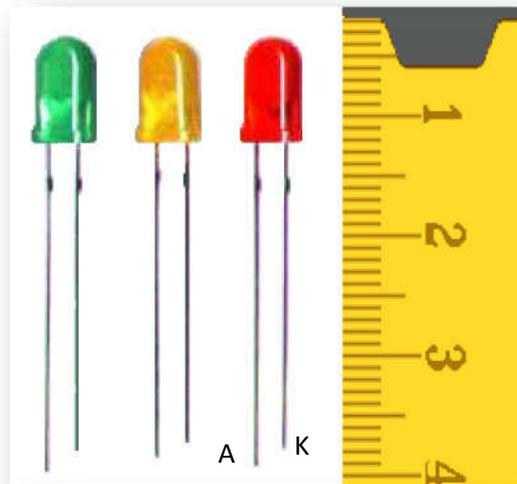


Fig. IV. 3. Les LEDs

IV. 2. 1. 5. Condensateur

Le Condensateur (Fig. IV.4) est un composant électronique élémentaire, constitué de deux armatures conductrices (appelées « électrodes ») en influence totale et séparées par un isolant polarisable (ou « diélectrique »). Sa propriété principale est de pouvoir stocker des charges électriques opposées sur ses armatures. La valeur absolue de ces charges est proportionnelle à la valeur absolue de la tension qui lui est appliquée. Le condensateur est caractérisé par le coefficient de proportionnalité entre charge et tension appelé capacité électrique et exprimée en Farads (F) [15].

Le condensateur est utilisé principalement pour :

- stabiliser une alimentation électrique (il se décharge lors des chutes de tension et se charge lors des pics de tension) ;
- traiter des signaux périodiques (filtrage...) ;
- séparer le courant alternatif du courant continu, ce dernier étant bloqué par le condensateur ;
- stocker de l'énergie, auquel cas on parle de super-condensateur.

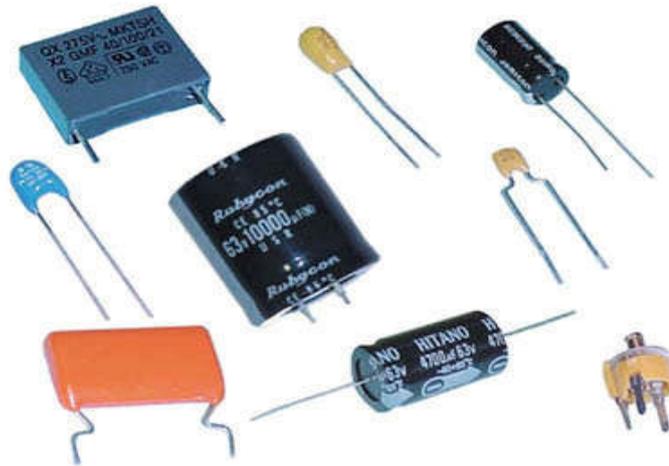


Fig. IV. 4. Plusieurs types de condensateurs

Les condensateurs électrolytiques utilisés dans notre carte, servent à filtrer les tensions requises pour le fonctionnement interne du MAX232

IV. 2. 1. 6. Quartz

Un Quartz (Fig. IV. 5) est un composant qui possède comme propriété utile d'osciller à une fréquence stable lorsqu'il est stimulé électriquement.

Les propriétés piézoélectriques remarquables du minéral de quartz permettent d'obtenir des fréquences d'oscillation très précises, ce qui en font un élément important en électronique numérique ainsi qu'en électronique analogique



Fig. IV. 5. Un quartz de 4 MHz

IV. 2. 2. Simulation et Réalisation de la carte

Après l'étude générale des différents éléments constituant la carte électronique, on passe maintenant à la réalisation physique de projet. Dans cette partie on touchera aux différents logiciels et outils utilisés pour création de la carte.

IV. 2. 2. 1. Simulation

IV. 2. 2. 1. 1. Simulation par Proteus ISIS

L'ISIS est un logiciel professionnel, utilisé dans l'électronique pour simuler des circuits et créer des typons. Il est également capable de simuler le fonctionnement du PIC avec tous les périphériques de la carte de commande.

L'utilisation du logiciel « ISIS » permet de mieux visualiser le bon déroulement du système ainsi que d'avoir une idée claire sur la partie matérielle et la conception des circuits imprimés [16].

IV. 2. 2. 1. 2. Carte d'Acquisition sur Proteus

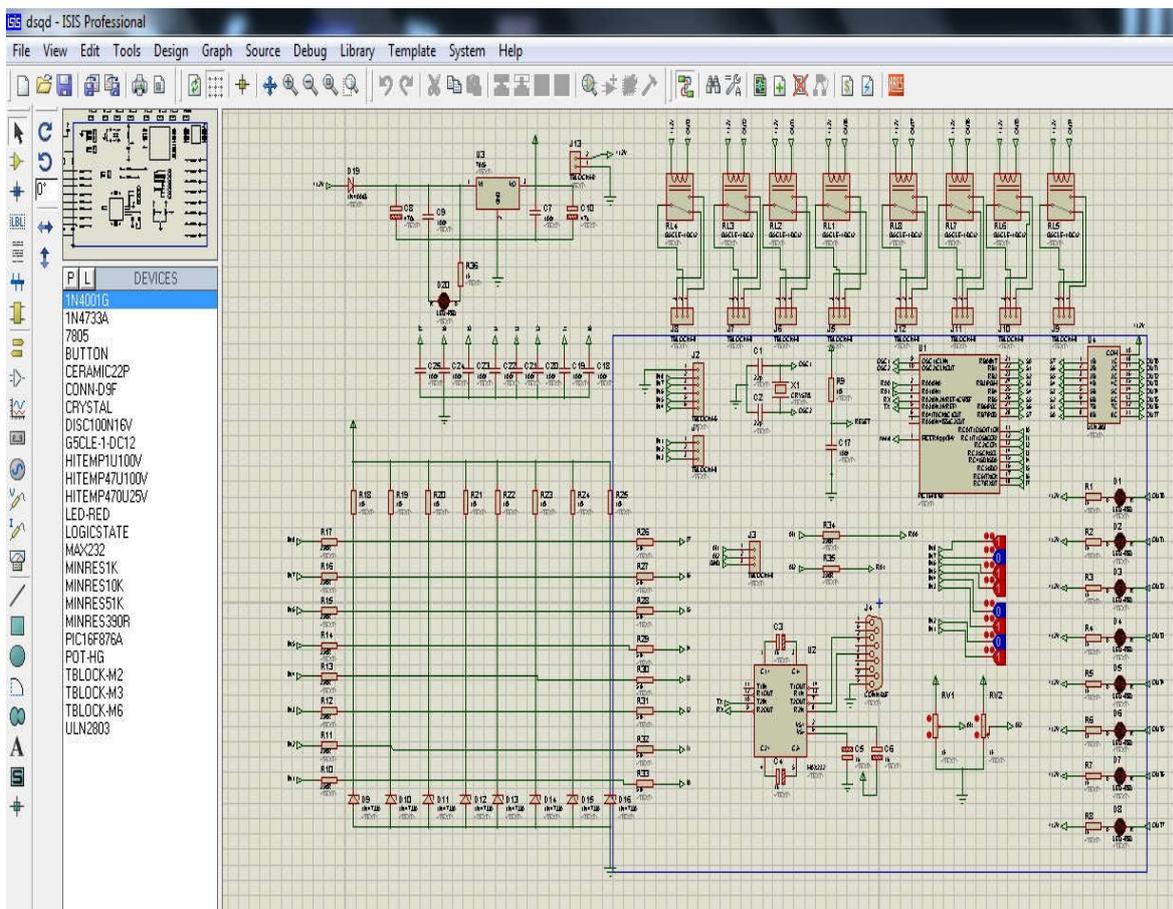


Fig. IV. 6. Schéma de simulation de la carte

IV. 2. 2. 2. Réalisation pratique

IV. 2. 2. 2. 1. Présentation de l'ARES

C'est un logiciel permettant le routage des cartes électroniques (constructions des pistes de connexion) en mode automatique ou manuel.

Cette fonctionnalité permet de réaliser des circuits de faible complexité en plaçant les composants et en traçant les pistes directement sur ARES. Une fois les connections établies, il est possible d'effectuer un routage automatique des pistes [16].

le logiciel permet présenter la carte électronique à réaliser en 3D (fig IV.7).

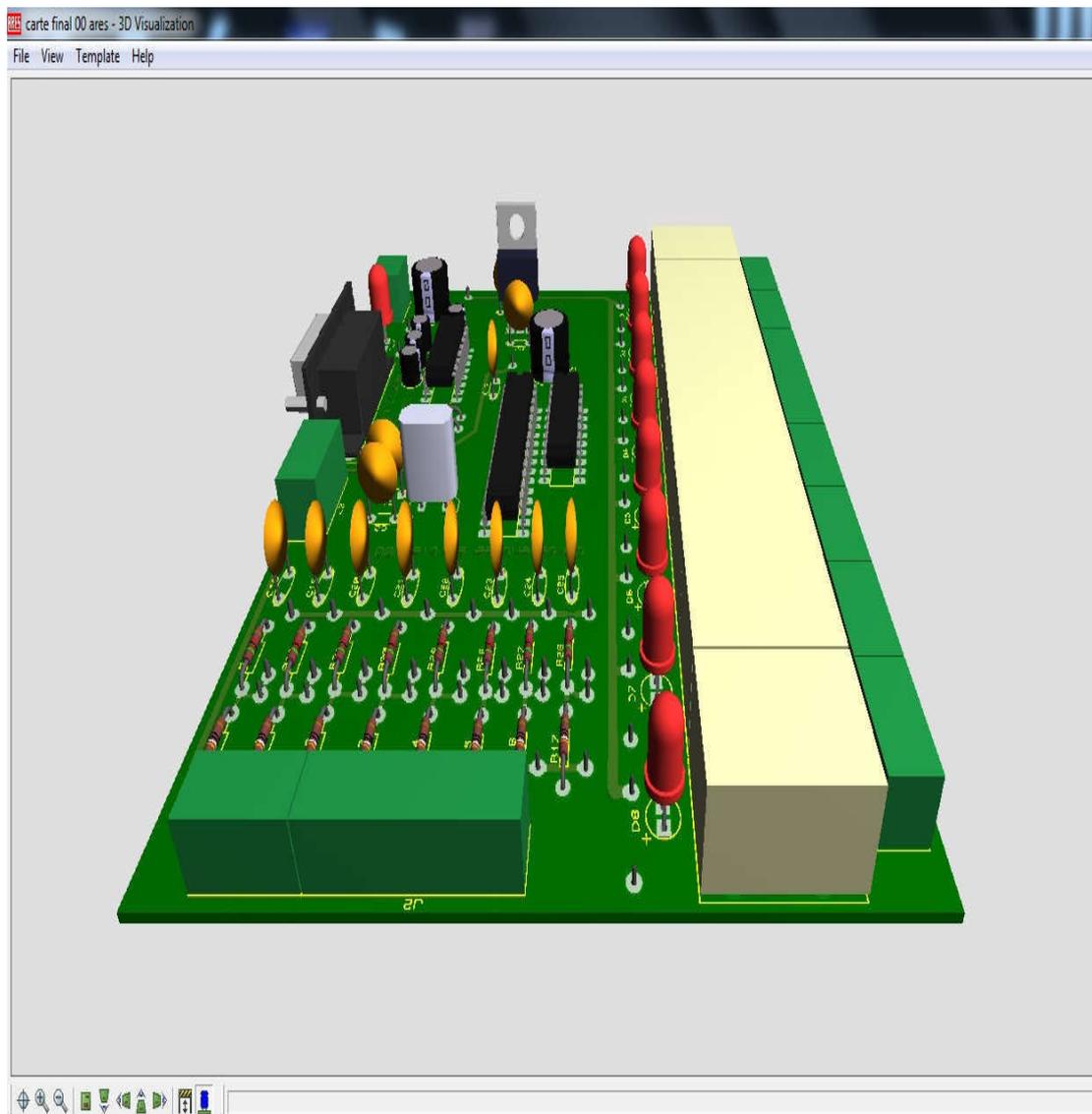


Fig. IV. 7. La carte en 3 dimensionnement

Dans notre réalisation nous avons opté d'utiliser un montage de type Multi faces pour créer un système plus compact et voici le schéma de circuit imprimé avec ses double face , sont illustrés dans les figures (IV.8) (IV.9)

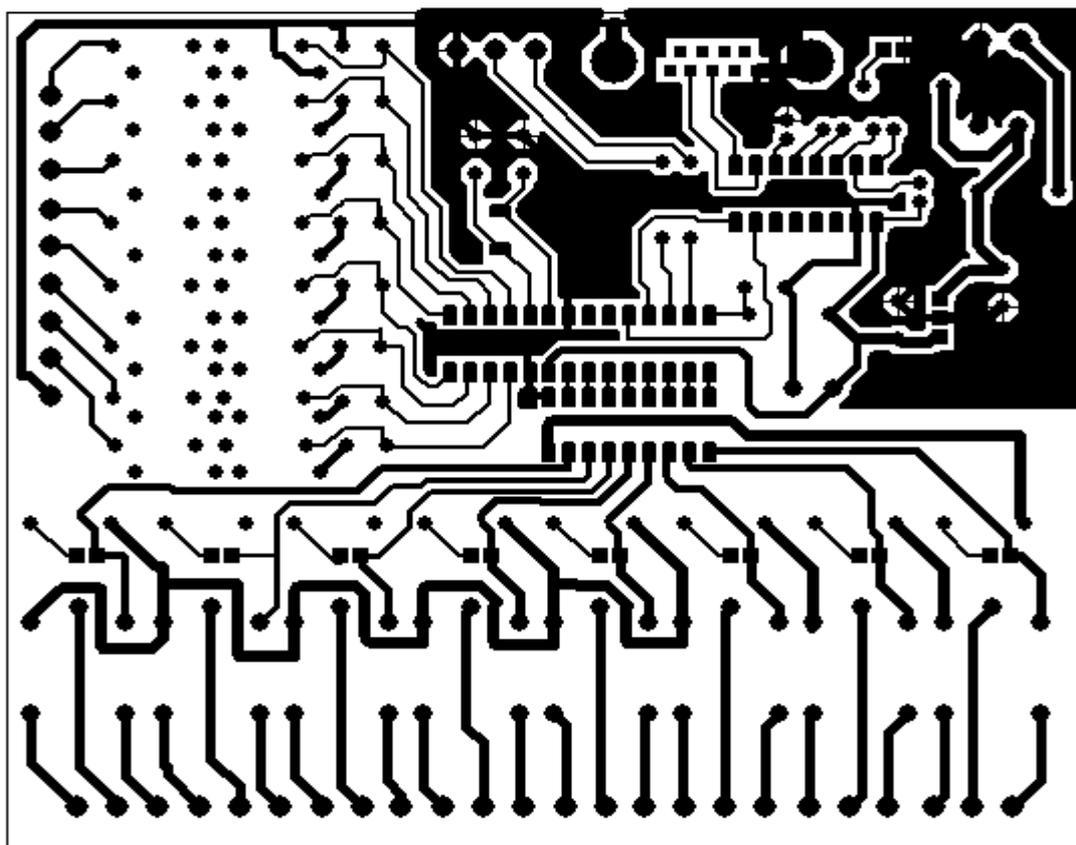


Fig. IV. 8. Schéma circuit imprimé de l'interface (face 1)

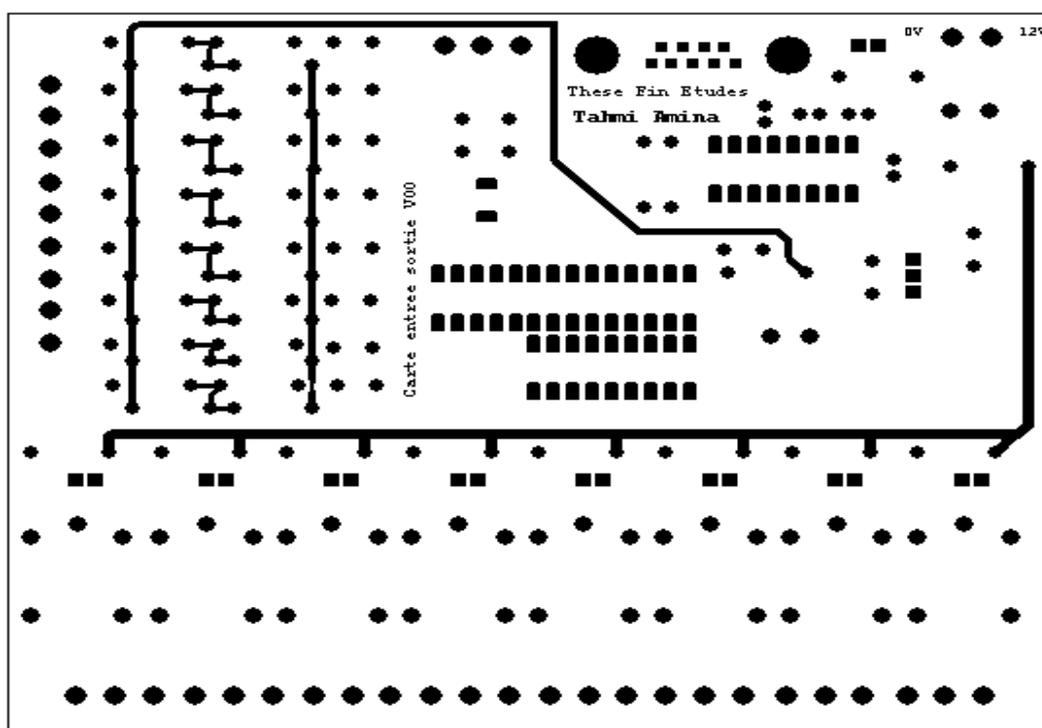


Fig. IV. 9. Schéma circuit imprimé de l'interface (face 2)

Après la réalisation des circuits imprimés il est impératif de vérifier la continuité de toutes les pistes, ainsi que l'absence de court-circuit entre pistes. Ensuite nous plaçons nos éléments commençant par les plus bas (résistances, diodes, supports des circuits intégrés) et en terminant par les plus hauts (condensateurs, en faisant attention à la polarité des électrolytiques, LED, que vous disposerez toutes à la même hauteur, en vous rappelant que la cathode est la patte la plus courte, etc.).

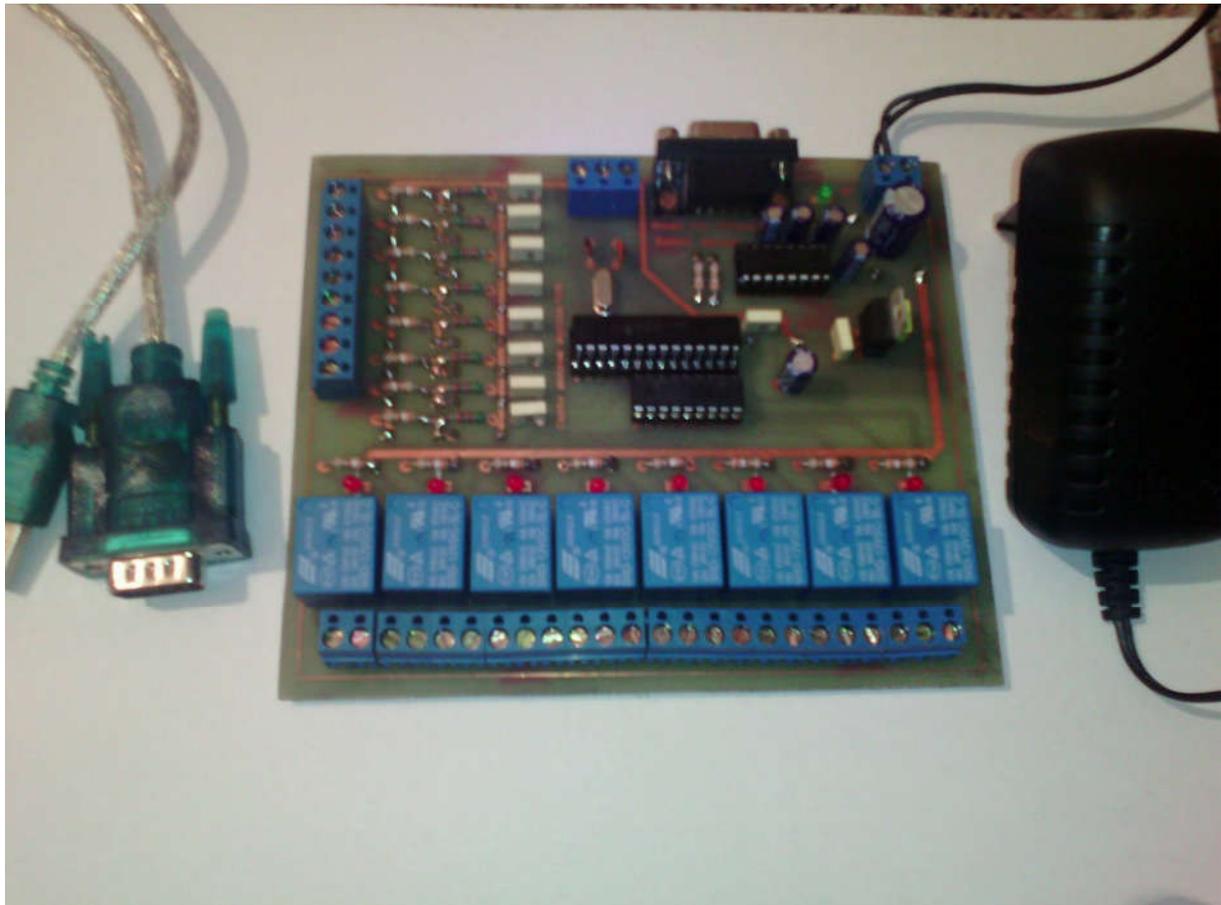


Fig. IV. 10. La carte électronique réalisée

IV. 3. Partie Programmation (software)

Dans cette partie le langage de programmation utilisé pour la carte d'Acquisition est le langage C et le logiciel d'interface 'LABview' a été utilisé pour la visualisation des données sous forme des graphes.

La programmation du PIC18F876A satisfait au fonctionnement de l'organigramme de la figure (IV.11) suivant qui facilite l'analyse, donc cet organigramme traduit les étapes d'exécution de notre programme.

IV. 3. 1. Organigramme du fonctionnement du programme principal

La figure IV.11 présente l'organigramme de programme principal du microcontrôleur suivi par un sous organigramme de la réception .

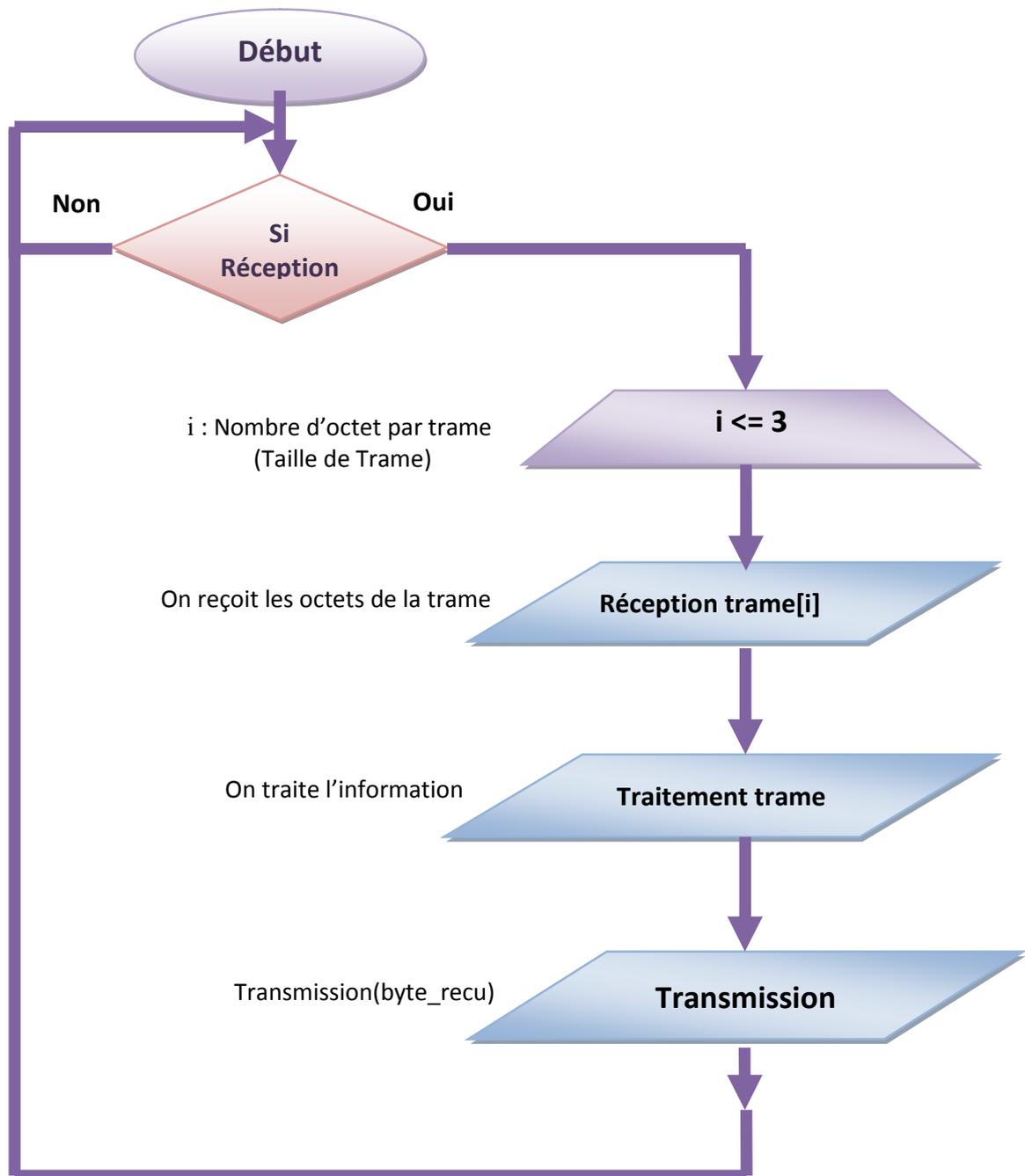


Fig. IV. 11 Organigramme du programme principal

Réception des données

Pour sous-organigramme de réception on utilise le fonctionnement suivant.

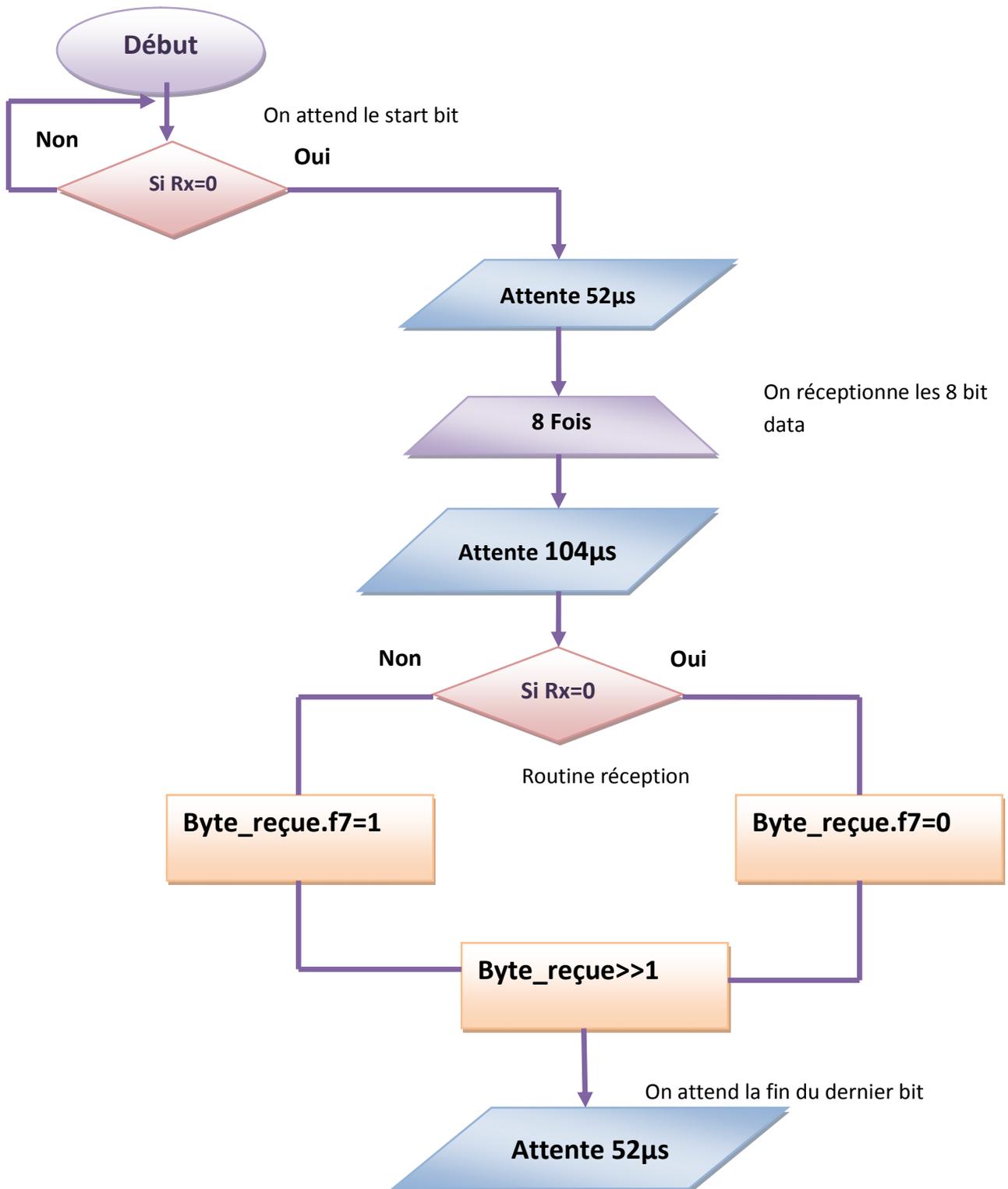


Fig. IV. 12 Organigramme de réception des données

IV. 3. 2. Langage de programmation Micro C

Dans le domaine de la programmation, on utilise aujourd'hui des langages de programmation de haut niveau (facilement compréhensibles par le programmeur) par opposition aux langages de bas niveau (de type assembleur) qui sont plus orientés vers le langage machine. Parmi les exemples de langages de haut niveau, on peut citer les langages C, C++, Java, cobol etc...

Le langage 'C'est un langage compilé (même s'il existe des interpréteurs plus ou moins expérimentaux). Ainsi, un programme 'C'est d'écrit par un fichier texte, appelé fichier source.

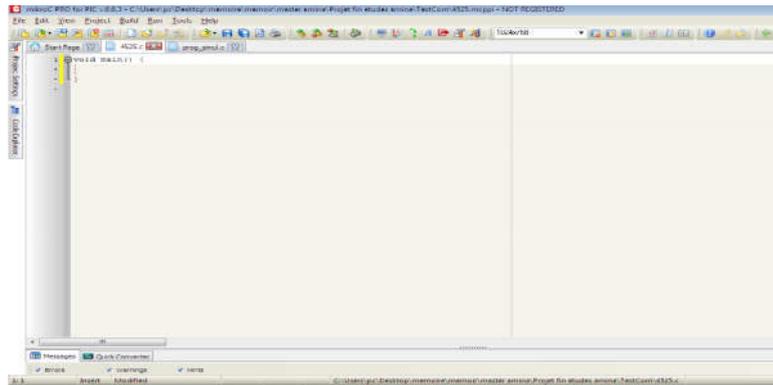


Fig. IV. 13. Fenêtre descriptive du micro c.

IV.3. 3. Programme de PIC

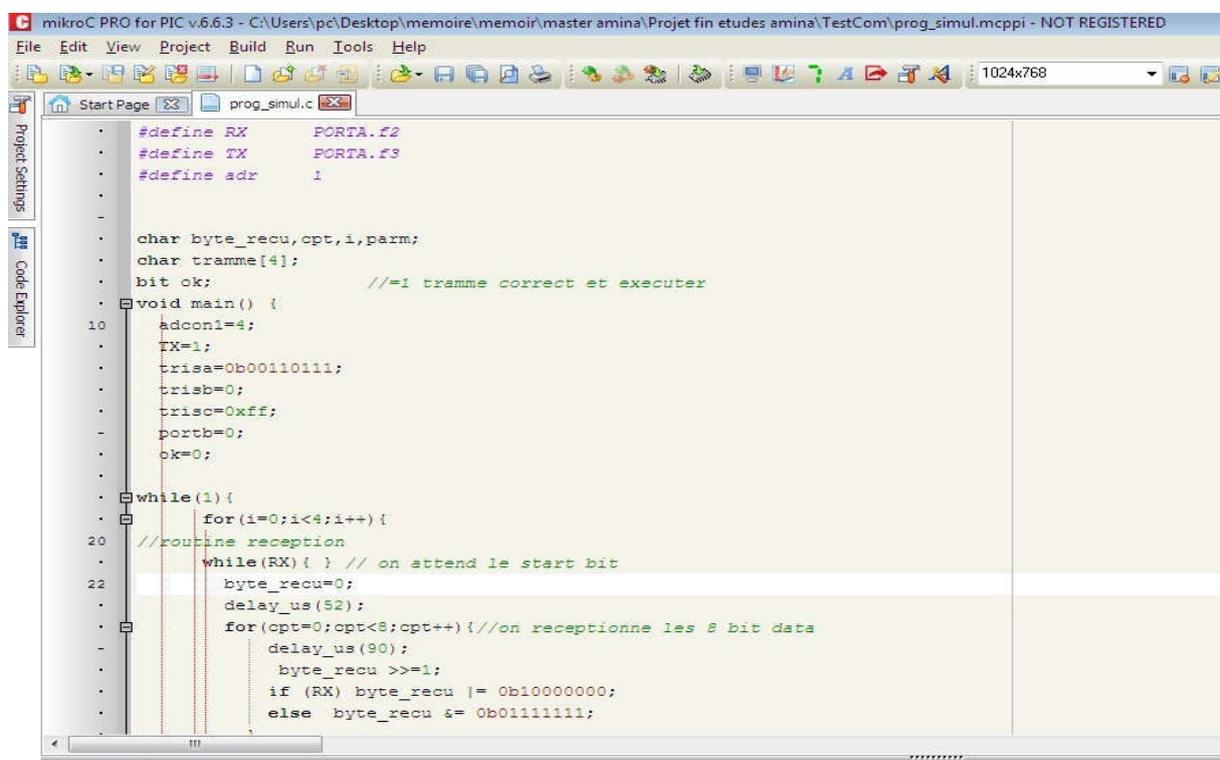


Fig. IV. 14. Programme principale

IV. 3. 4. Interface (Supervision)

Pour réaliser la supervision (l'interface) de ce système on utilise logiciel Labview.

IV. 4. 1. Représentation LABVIEW

Labview (Laboratory Virtual Instrument Engineering Workbench) est un langage de programmation graphique qui utilise des icones au lieu de lignes de texte pour créer des applications d'instrumentation plus particulièrement destiné a l'acquisition des données de mesure et leur traitement. Contrairement aux langages de programmation textuels où ce sont les instruments qui détermine l'ordre d'exécution du programme, Labview utilise la programmation par flux de données ; C'est le flux des données transitant par les nœuds sur le diagramme qui détermine l'ordre d'exécution des 'Vis' et des fonctions. Les 'Vis', ou instruments virtuels, sont des programmes LabView, vous construisez une interface utilisateur à l'aide d'un ensemble d'outils et d'objet, L'interface utiliser d'un VI est appelée la face-avant [17].

Ensuite nous créons le code en utilisant des représentations graphiques de fonctions pour commander les objets de face-avant. Ce code source graphique est aussi appelé code G ou code de diagramme. Le diagramme contient ce code. D'une certaine manière, le diagramme ressemble à un organigramme, aussi LabView permettra de développer des applications complexes en utilisant des fonctions mathématiques, de traitement de signal, d'analyse de données, et les moyens de communication actuels avec les instruments (RS232, Ethernet, USB ...).

Il se décompose en deux parties :

- ❖ La première partie (partie cachée ou interne) : elle contient l'algorithme du programme décrit sous la forme d'un diagramme flot de données en langage graphique.
- ❖ La seconde partie (partie visible) est constituée de l'interface utilisateur.

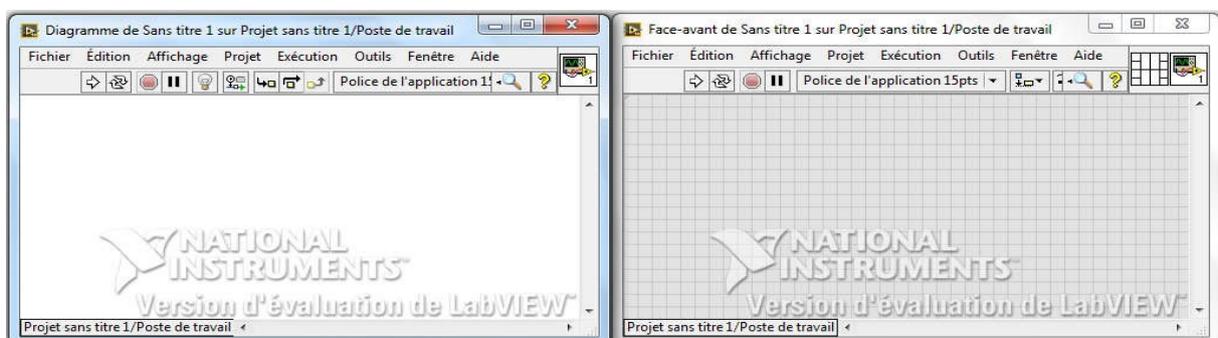


Fig. IV. 15. Fenêtre de l'environnement de développement sur LabView

Face avant (à droite) et Diagramme (à gauche).

IV. 3. 4. 2. Réalisation de l'interface :

Finalement le projet sera entièrement sur le programme de LabView comme suit :

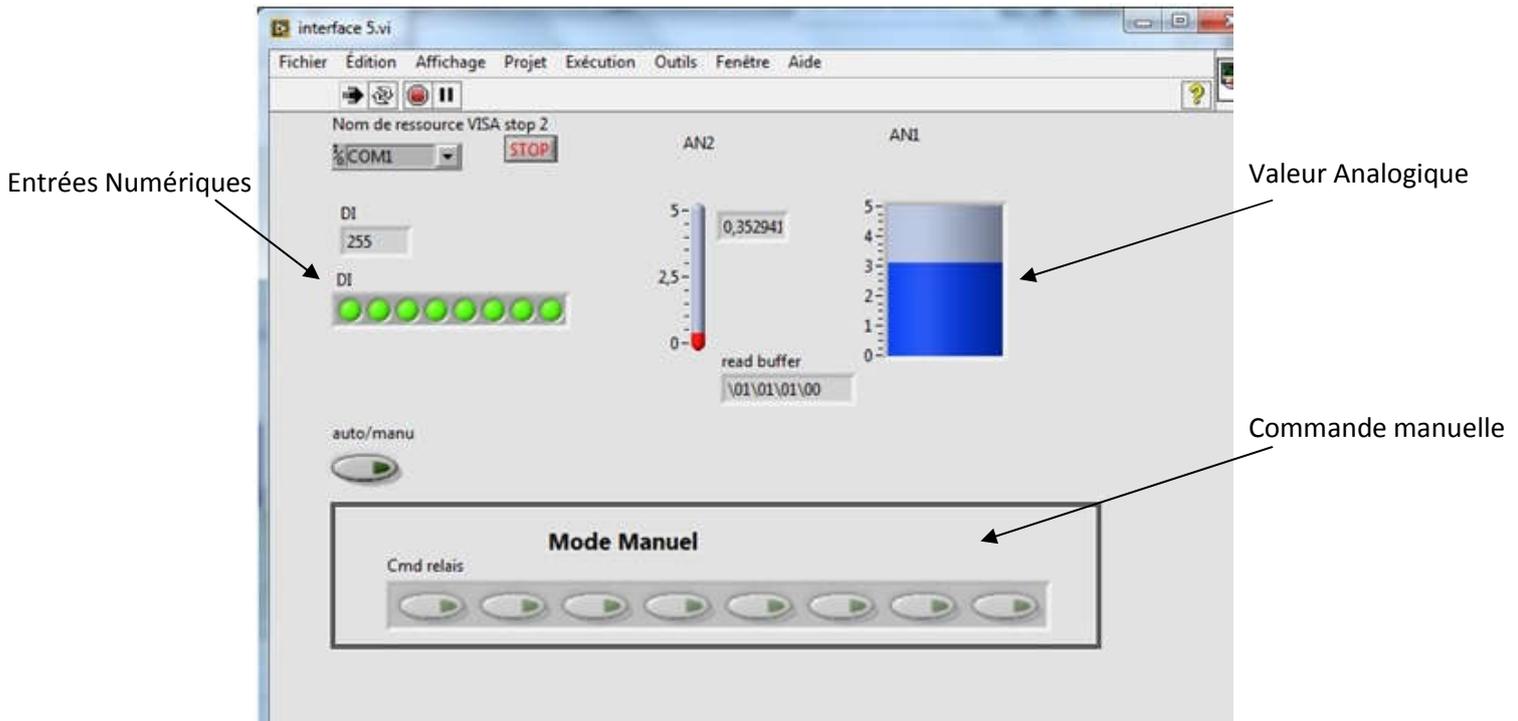


Fig. IV. 16. Face avant (l'interface)

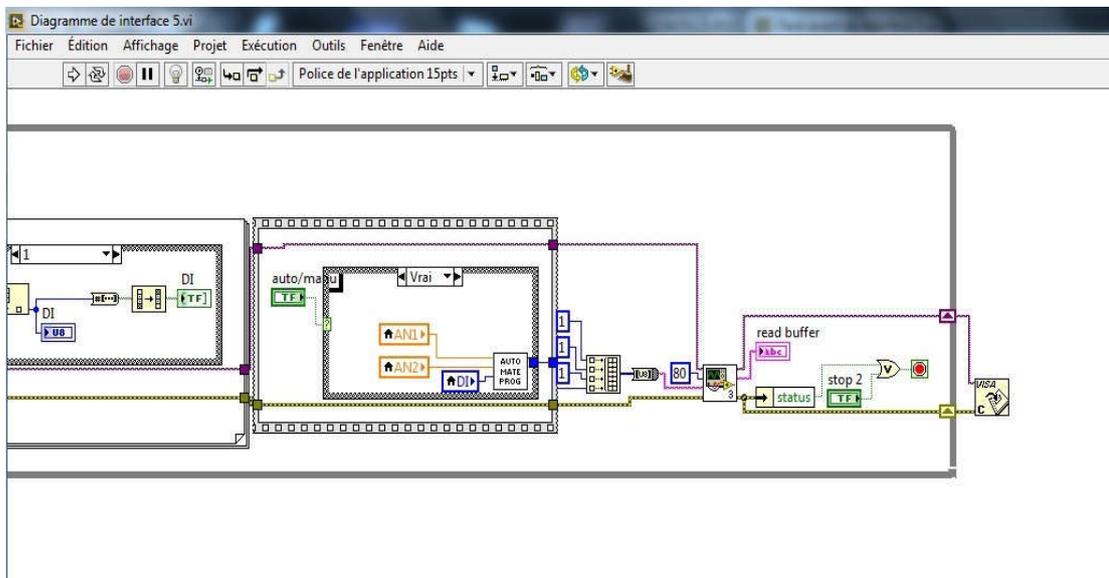


Fig. IV. 17. Face arrière (programme)

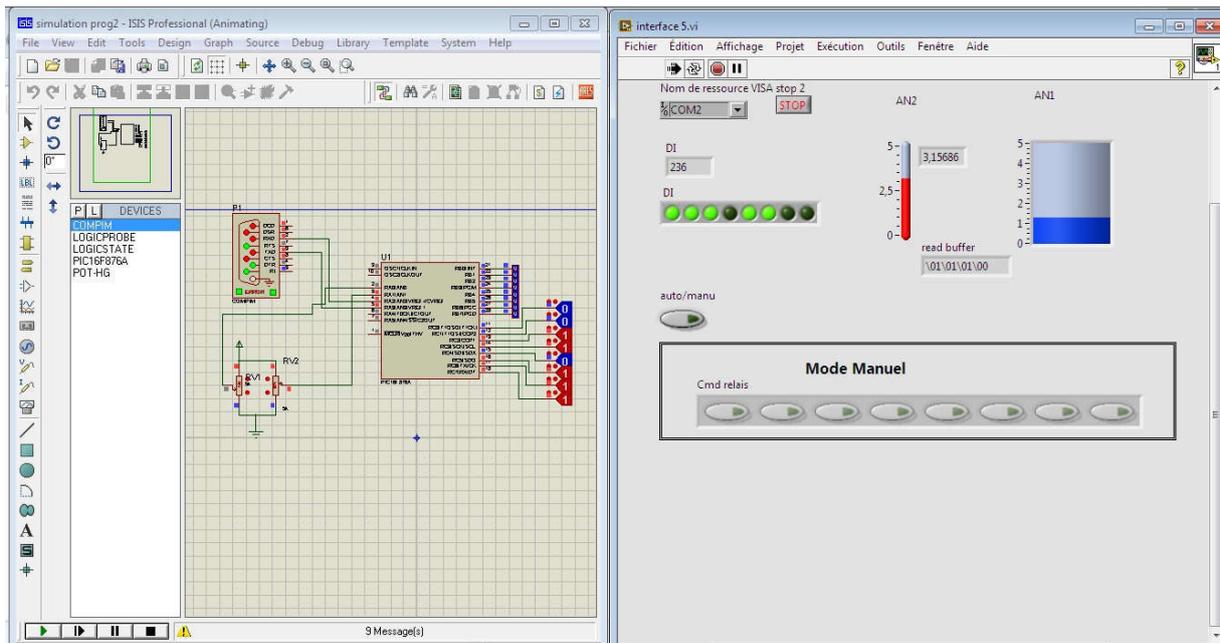


Fig. IV. 18. Interface avec simulation Proteus ISIS

IV. 4. Teste Résultat finale :

La figure suivante représente la réalisation de la carte électronique et le PC (l'interface) lié avec un câble de communication série USB/DB9.

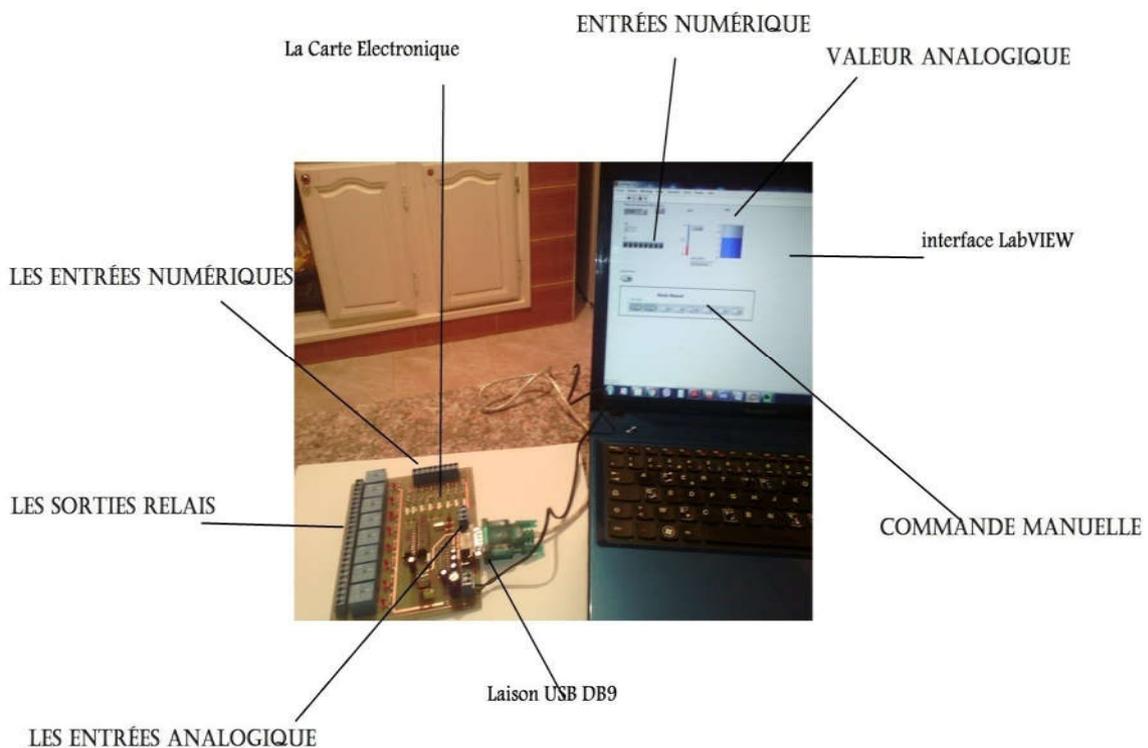


Fig. IV. 19. Réalisation finale du Projet

IV. 5. Conclusion

Dans ce chapitre nous avons présenté les différentes étapes de notre réalisation y compris la carte d'interface ainsi que les équipements et logiciels utilisés.

On a montré aussi des tests expérimentaux que nous avons à l'aide de l'interface graphique réalisée par le programme LabView. Les résultats obtenus ont montré que notre carte d'interface est totalement contrôlée par ordinateur.

CONCLUSION GENERALE

Objectif de notre travail était de réaliser une carte d'interface permettant de lire les entrées (numérique et analogique) et la commande des sorties Numérique.

La réalisation de ce projet nous a énormément appris, autant au niveau de l'électronique, de la programmation des microcontrôleurs. On a aussi appris des nouvelles connaissances au niveau de la Communication série.

Ce projet nous a permis de nous familiariser avec le domaine de la création et la manipulation électronique, par conséquent l'idée de construire divers montages est devenue simple.

Notre projet facilite la tâche à l'utilisateur afin de pouvoir contrôler à distance devant sont ordinateur jusqu'à 8 appareils grâce à une simple technique de transmission présenté par la liaison RS232.

Parmi Les difficultés de ce travail, nous avons eu un problème de lecture des données numériques à cause de parasite et phénomène d'anti-rebond. Pour éliminer ce problème, nous avons ajouté un filtre passe bas. De plus, le port C du PIC16876A ne dispose pas de résistance 'pull up' comme le port B. Cependant on a choisit précisément le port C comme entrée car il dispose de trigger de Schmidt ce qui complète notre étage d'entrée.

Comme perspective de ce projet nous proposant d'intégrer cette carte dans un réseau local LAN où le contrôle et la supervision sera réalisé par un ordinateur distant

Enfin on espère que ce travail sera utile et bénéfique pour le lecteur qui trouvera dans ce mémoire l'essentiel de ce qu'il faut savoir sur les cartes E/S.

Références bibliographie

- [1] J.PIERRAT, << Interface E/S commandée par PC >>, N°22 Electronique et Loisirs Magazine, JMJ éditions, Marseille, Mars 2001.
- [2] P.ANDRÉ, << La liaison RS232 Description technique et mise œuvre 2ème édition >>, Dunod, Paris, 2002.
- [3] http://www.chicoree.fr/w/Full-duplex,_half-duplex_et_simplex , Dernière modification le 22 mai 2011.
- [4] https://www.academia.edu/5163800/Chapitre3_microcontrolleur , Academia 2020.
- [5] V.TOURTCHINE, << Microcontrôleur de la famille PIC >>, Université M'hamed Bougara De Boumerdes , 2009.
- [6] PIC16F8X, document DS30430C, <https://www.microchip.com>
- [7] BIGONOFF , << La programmation des PIC par BIGONOFF >>,, Première partie révision 34, <https://www.abcelectronique.com/bigonoff/>
- [8] http://www.groupeisf.net/reseaux_informatiques/reseaux_et_telecommunications/reseaux/liasons_series/rs232.html , Dernière modification 2018
- [9] L. Rebuffie, << Communication par transmission série RS232 >>, académie de Rennes , Juin 2012
- [10] MAX232 datasheet , Texas INSTRUMENTS, <https://www.ti.com/>
- [11] ULN2803A datasheet, Texas INSTRUMENTS, <https://www.ti.com/>
- [12] <http://www.zpag.net/Electroniques/relais.htm> , Dernière modification 2019
- [13] W.Eccles ,<< Trésor de la langue française informatisé « diode » >>, Oxford English Dictionary ,2010.
- [14] <https://docplayer.fr/22637323-Diode-zener-presentation-de-la-diode-zener-composition-et-mode-de-fonctionnement.html> , Dernière modification 2016.
- [15] <http://www.epsic.ch/cours/electrotechnique/theorie/condos/090.html> , Dernière modification octobre 2020
- [16] M.LAKHDARI, << Introduction à la simulation et routage des circuits avec le logiciel PROTEUS V7 et V8 >>, Université des sciences et de la technologie d'Oran, 2017.
- [17] https://zone.ni.com/reference/fr-XX/help/371361R-0114/lvhowto/lv_getting_started/ , Dernière modification 2018.

