



**FACULTÉ DES SCIENCES APPLIQUÉES**  
**DÉPARTEMENT de GÉNIE ÉLECTRIQUE**



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

Spécialité : Génie Electrique

Option : ***Informatique Industrielle***

**Présenté par :**

Kessam Elyamine

Saidani Abdelkader

*Sujet du mémoire*

**Réalisation d'un Oscilloscope 4 voies par la liaison**

**RS232**

*Soutenu publiquement devant le jury composé de :*

Mr, B. SAHLI

Président

Mr, M. ABDICHE

Examineur

Mr, A. Tiffour

Encadreur

PROMOTION : 2015

**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE**  
**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE**  
**SCIENTIFIQUE**  
**UNIVERSITÉ IBN KHALDOUN DE TIARET.**

---



**FACULTÉ DES SCIENCES APPLIQUÉES**  
**DÉPARTEMENT de GÉNIE ÉLECTRIQUE**





# Dédicace

*Avant tous, je remercie dieu le tout puissant de m'avoir  
donner le courage et la patience pour réaliser ce travail  
malgré toutes les difficultés rencontrées.*

*Je dédie ce modeste travail :*

*A mes très chers parents, que dieu les garde et les protège  
pour leurs soutien moral et financier, pour leurs  
encouragements et les sacrifices qu'ils ont endurés.*

*A mon encadreur **abdelkader Tiffour***

*A mes frères*

*A mes sœurs*

*A l'esprit de mes grands pères*

*A l'esprit de ma grande mère*

*A mes chers amis*

*A tous les amis (es) d'études surtout ceux d'informatique  
industrielle promotion 2015.*



*Elyamine et  
abdelkader*

# Remerciements

*Remerciements à Dieu –le tout puissant– qui nous a aidé à réaliser ce modeste travail.*

*Nous tenons à remercier notre encadreur MS : **Abdelkader Tiffour** d'avoir accepté de nous encadrer et de nous suivre durant toute cette période.*

*Nos remerciements vont aussi au président du jury et aux membres du jury examinateurs qui nous fait l'honneur de participer au jury de ce travail.*

*enfin nous remercions l'ensemble, enseignants et collègues de notre promotion, qui nous ont aidé à réaliser ce modeste travail.*

Résumé

Liste des abréviations

Liste des figures et tableaux

Introduction

générale.....1

## **Chapitre I : Etude des signaux électriques et leur échantillonnage**

I.1 Introduction.....3

I.2 présentation d'oscilloscope.....3

I.3 Principes de base de l'oscilloscope .....4

I.4 Analogique vers numérique.....5

I.5 Mode d'emploi simplifié de l'oscilloscope numérique.....7

I.6 Le traitement de signal numérique .....9

I.7 Le problème de l'échantillonnage .....9

I.8 Les série de Fourier.....10

1.9 Echantillonnage .....12

I.10 Théorème de Shannon .....13

1.11. Représentation mathématique d'un signal Echantillonné.....13

I.12 Conclusion.....15

## **Chapitre II : Etude de la liaison RS232 et sa programmation**

II.1 Introduction ..... 17

II.2 Le microcontrôleur 16F877 ..... 17

II.2.1 Définition d'un PIC..... 17

II.2.2 Les différentes familles des PICS : ..... 17

II.2.3 Les caractéristiques du PIC 16F877A : ..... 17

II.2.4 Architecture externe du PIC 16F877A ..... 18

II.2.5 Architecture interne du PIC 16F877A ..... 18

II.2.6 Les mémoires : ..... 19

II.2.6.1 La mémoire programme ..... 19

II.2.6.2 La mémoire EEPROM : ..... 20

II.2.6.3 La mémoire RAM : ..... 20

II.2.6.4 Les ports d'entrées/sorties : ..... 20

II.2.7 Les différents registres du PIC 16F877A..... 21

# Sommaire

---

|  |    |
|--|----|
| II.2.7.1 Le registre W :   | 21 |
| II.2.7.2 Le registre STATUS ou registre d'états :                    | 21 |
| II.2.7.3 Le registre INTCON ou registre de contrôle d'interruptions: | 21 |
| II.2.7.4 Le registre OPTION-REG :                                    | 21 |
| II.2.7.5 Le registre T1CON :   | 21 |
| II.2.7.6 Le registre T2CON :   | 21 |
| II.2.7.7 Les registres CCP1CON et CCP2CON :                          | 22 |
| II.2.7.8 Les registres ADCON0 et ADCON1 :                            | 22 |
| II.2.8 Les interruptions   | 22 |
| II.2.9 Les circuits d'horloges                                       | 22 |
| II.2.9.1 Oscillateur à quartz :                                      | 22 |
| II.2.9.2 Oscillateur RC  | 23 |
| II.2.10 Les Timers   | 23 |
| II.2.10.1 Le Timer0 :  | 24 |
| II.2.10.2 Le Timer1 :  | 24 |
| II.2.10.3 Le Timer2 :  | 25 |
| II.2.11 Le convertisseur analogique numérique :                      | 25 |
| II.2.12 Les modules CCP1 et CCP2                                     | 26 |
| II.2.13 Domaines d'application des microcontrôleurs.....             | 26 |
| II.2.14 Les différents avantages des microcontrôleurs                | 27 |
| II.2.15 Les défauts des microcontrôleurs.....                        | 27 |
| II.3 Communication série RS-232.....                                 | 27 |
| II.3.1 Historique.....   | 28 |
| II.3.2 Description.....  | 28 |
| II.3.3 Utilisation.....  | 28 |
| II.3.4 Transmission des informations numériques.....                 | 29 |
| II.3.5 Bit de parité.....  | 29 |
| II.3.6 Les signaux électriques                                       | 30 |
| II.3.7 Caractéristique de la liaison V24 - RS232: la vitesse         | 30 |
| II.3.8 Raccordements   | 31 |
| II.3.9 Conclusion  | 31 |

# Sommaire

---

## **LISTE DES FIGURES**

Figure I.01 - principe de fonctionnement d'un oscilloscope

Figure I.02 - les différents types des signaux

Figure I.03 - forme sinusoïdale d'un signal

Figure I.04 - les Type de base d'un oscilloscope

Figure I.05 - les formes d'ondes

Figure I.06 - oscilloscope numérique

Figure I.07 - Mode d'emploi simplifié de l'oscilloscope numérique

Figure I.08 - schéma de principe de la conversion CAN

Figure I.09 - problème de l'échantillonnage d'un signal

Figure I.10- graph d'un signal échantillonné

Figure I.11 - comparaison entre fréquence sinusoïdale et plusieurs fréquences possibles

Figure I.12 - Reconstitution d'un signal carrée Fondamental et 7 harmoniques de rangs impaires d'un signal carrée

Figure I.13- spectre d'un signal analogique

Figure I.14- schéma signifie le théorème de Shannon

Figure I.15- Représentation mathématique d'un signal Echantillonné

Figure I.16- conversion analogique numérique

Figure I.17- échantillonneur bloqueur

Figure I.18- signal après l'échantillonneur bloqueur

Figure I.19- La chaîne complète de traitement de signal

Figure II.1- Microcontrôleur PIC 16F877A

Figure II.02- Boitier du PIC 16F877A

Figure II.03 - L'architecture interne du PIC 16F877A



## Liste des figures et des tableaux

---

Figure II.04 - Les différents ports du PIC16F877A

Figure II.05 - Oscillateur à quartz

Figure II.06 - Connexion d'une horloge externe

Figure II.07 - Oscillateur RC

Figure II.08 - Schéma du Timer0

Figure II.09 - Schéma descriptif du Timer1

Figure II.10 - Schéma descriptif du Timer2

Figure II.11- Schéma du module de conversion analogique numérique

Figure II.12- connecteur DB-9 femelle

Figure II.13- connecteur DB-9 male

Figure II.14- transmission des données

Figure II.15- schéma signifiant la transmission des signaux électrique via RS232

Figure II.16- raccordement DB9

Figure III.1- fenêtre au démarrage de Visual Basic

Figure III.2- fenêtre signifiant les caractéristiques ou propriétés des éléments qui composent l'interface

Figure III.3- les commandes disponibles pour un projet

Figure III.4 - interface oscilloscope

Figure IV.1-Schéma synoptique du système

Figure IV.2- PROTEUS 7

Figure IV.3- Synoptique d'un cycle de compilation

Figure IV.4-Organigramme de la programmation

Figure IV.5- Schéma synoptique du système

Figure IV.6- Schéma de sub circuit

## Liste des figures et des tableaux

---

Figure IV.11-Schéma fonctionnel de l'opto-coupleur

Figure IV.12-Vue d'ensemble du banc d'essai

Figure IV.13-MCC à aimants permanents

Figure IV.14-Carte d'entraînement et de développement 877-DATS.

Figure IV.15-Hacheur

Figure IV.16-Carte d'acquisition réalisée

Figure IV.17-Vue générale de la manipulation

Figure IV.18-Réponse en vitesse du moteur en boucle ouverte

Figure IV.19-Signal MLI pour consigne =120

Figure IV.20-Réponse en boucle fermé pour consigne = 360

Figure IV.21-Affichage pour consigne=360

Figure IV.22-Affichage pour consigne=474

Figure IV.23-Affichage pour consigne=784

Figure IV.24-Affichage pour consigne =784

Figure IV.25-Consigne et réponse en vitesse du moteur

Figure IV.26-Signal MLI avec pour consigne = 784

### **LISTE DES TABLEAUX**

Tableau II.01- Les interruptions

Tableau II.02- portes séries

Tableau III.1- barre de menu au visual basic

Tableau III.1- les commandes disponibles pour un projet

## Liste des figures et des tableaux

---

# LISTE DES ABREVIATIONS

---

## LISTE DES ABREVIATIONS

### Chapitre I

CAN : convertisseur analogique / numérique

Ne : représente une valeur numérique de  $v_e$  à un instant  $t$ .

Ns : représente une valeur numérique de  $v_e$  à un instant  $t + t_c$  (temps de calcul)

E : représente l'amplitude du fondamental

$T_e$  : période

$t + t_c$  : temps de calcul

$\omega$ : la pulsation

$\delta(t)$  : variation de temps

P : nombre naturel

$F_e$  : fréquence échantillonnage

$F_{max}$  : fréquence Maximale

$V_e$  : signal d'entrée

$V_i$  : signal initial

$V_b$  : signal d'échantillonneur bloqueur

MAC : multiplie and accumule

DSP : Digital Signal Processor

### Chapitre II

PIC : Programmable Interface Controller

RISC : Reduced Instructions Set Computer

RAM : Random Access Memory

ROM : Read Only Memory

MOS: Metal oxide Semiconductor

## LISTE DES ABREVIATIONS

---

CMOS : Complementary Metal Oxide Semiconductor

HCMOS : High Speed CMOS

PWM: Pulse width modulation

PSP: Parallèle Slave Port

CCP : modules de capture et de comparaison

MSSP : Master Synchronous Serial Port

SSP: Synchronous Serial Port

HS: High Speed

LP : Low Power

SCI : Serial Communication Interface

MCLR: Memory Clear

USART : Universal Synchronous Asynchronous Receiver Transmitter

WDT: Watch Dog Timer

EEPROM: Electric al Erasable Programmable Read Onlay Memory

RS: Recommended standard

TIA: Telecommunication Industry Association

MS-DOS: Microsoft-Disk operating System

IBM: International business Machines

RxD : Received Data

TxD : Transmitted Data

Bauds : nombre de bits de données transmis par seconde.

half-duplex: les données circulent dans les 2 sens mais pas simultanément

full-duplex: les données circulent de manière bidirectionnelle et simultanément

EIA 232 : Electronic Industry Associate

USB : Universal Serial Bus

# LISTE DES ABREVIATIONS

---

ISO : International Standardisation Organisation

## **Chapitre III**

VB : Visual Basic

BASIC: Beginners All-Purpose Symbolique Instruction Code

GUI: Graph cal User Interface

EDI : Environnement de développement intégré

IUG : Interface Utilisateur Graphique

## **Chapitre IV**

LCD: Liquid crystal display

PC: Personal computer

# Résumé

---

## **RESUME**

L'objectif de notre travail est de réaliser une interface permettant de visualiser sur un PC un signal électrique d'une amplitude comprise entre -10V et +10V.

La fréquence du signal ne doit pas dépasser les 15 KHz.

L'oscilloscope PC est un instrument de mesure constitué d'une interface matérielle et d'une interface soft développée en Visual Basic qui va nous servir à visualiser sur PC notre signal d'entrée.

Initialement, les oscilloscopes étaient des instruments autonomes sans capacités de traitement du signal ou de mesure et avec une capacité de stockage uniquement optionnelle.

Plus tard, les oscilloscopes ont commencé à utiliser de nouvelles technologies numériques pour introduire de nouvelles fonctions, mais ils restaient des instruments hautement spécialisés et onéreux.

Les oscilloscopes PC sont la récente étape de l'évolution des oscilloscopes qui associent la puissance de mesure de ces derniers à la performance croissante des PC.

## **ABSTRACT**

The aim of our work is to make a realization of an interface for viewing on a PC an electrical signal of a magnitude between -10V and + 10V with a maximum frequency of 15 KHz.

The PC Oscilloscope is a measuring instrument consists of a hardware interface and a software interface developed in Visual Basic that will serve us visualize our PC input signal.

Initially, oscilloscopes were self without instruments or measurement signal processing capabilities and with only optional storage capacity. Later, oscilloscopes began to use new digital technologies to introduce new functions, but they remained highly specialized and expensive instruments.

PC Oscilloscopes are the latest step in the evolution of oscilloscopes and combine the power measurement Oscilloscope for the convenience of your PC.

## *Introduction générale*



## **INTRODUCTION GENERALE**

Dans le domaine des sciences pour l'ingénieur, le système de mesure le plus simple et le plus répandu consiste à utiliser un micro ordinateur interfacé avec le système sur lequel on travaille.

Etablir un dialogue entre un micro ordinateur et un système extérieur au PC est possible par l'intermédiaire de cartes d'interface.

Ces cartes dites d'Entrées/Sorties sont directement implantées à l'intérieur du PC et connectées sur le bus PC via des Slots. Elles sont généralement équipées de compteurs (timers), de circuits convertisseurs offrant des entrées et des sorties analogiques et des circuits à base de portes logiques offrant des entrées et des sorties logiques à l'utilisateur.

Cet ensemble PC et carte d'interface est complété par un logiciel (généralement appelé driver) permettant la gestion de l'ensemble des fonctions de la carte Entrées/Sorties par l'intermédiaire d'un programme.

Le projet a pour but de réaliser une interface permettant de visualiser sur un PC un signal électrique d'une amplitude comprise entre -10V et +10V ayant une fréquence maximale de 15Khz.

Le travail est constitué de quatre principaux chapitres.

Dans le premier chapitre, nous avons parlé du principe de fonctionnement d'un oscilloscope et un rappel sur les signaux électriques et leur échantillonnage (conversion Analogique/ Numérique).

Nous commençons le deuxième chapitre par la description du microcontrôleur utilisé dans ce travail et sa propriété. Mais la majeure partie du chapitre est consacré à la liaison RS232 et de sa programmation en générale.

Dans le troisième chapitre, nous ferons l'étude de l'interface de transfert des données vers le PC et la visualisation du signal en temps réel.

Le quatrième chapitre sera consacré à la réalisation pratique et à l'expérimentation.

Nous présenterons brièvement les logiciels de simulation ainsi que les différents composants de notre expérimentation.

Enfin nous donnerons les principaux résultats expérimentaux obtenus ainsi que leurs interprétations.

[Eric Magarotto Licence EEA – IE Transmission & Acquisition de Données Informatique Industrielle 7Année 2003]

*Chapitre I : Etude des signaux électriques et  
leur échantillonnage (conversion Analogique/  
Numérique)*

## I.1 Introduction

Un oscilloscope permet avant tout de VISUALISER une tension dépendant du temps. il permet aussi d'effectuer des mesures mais de façon moins précise qu'un multimètre numérique. Avec des montages spécifiques, nous pouvons rendre les tensions proportionnelles à d'autres grandeurs physiques telles que l'intensité d'un courant ou la fréquence d'un signal. Nous pouvons ainsi visualiser des courbes expérimentales très diverses.

La conversion analogique-numérique transforme les signaux analogiques en nombres exploitables par l'ordinateur de manière numérique.

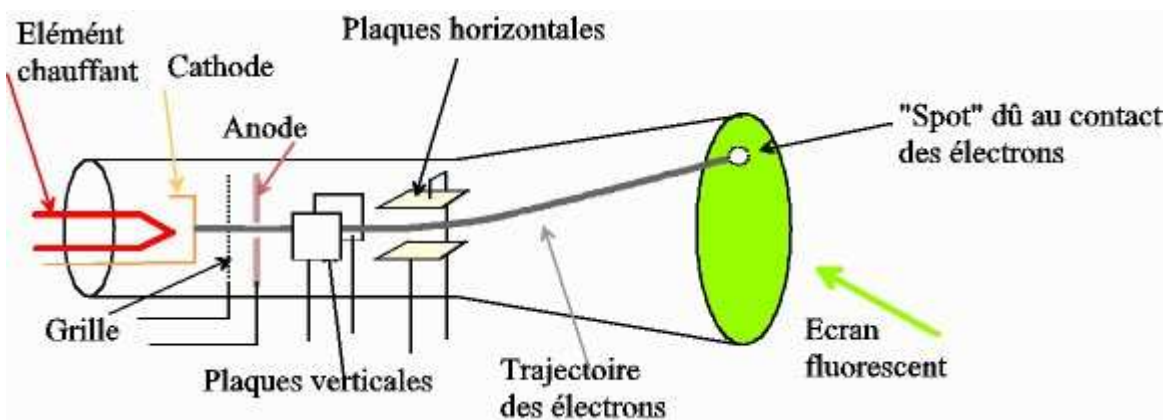
## I.2 L'oscilloscope

Bien qu'il permette de mesurer une différence de potentiel continue, l'oscilloscope est particulièrement adapté pour étudier les tensions alternatives dont il permet de mesurer non seulement l'amplitude mais aussi d'observer la forme de la variation dans le temps.

L'oscilloscope comporte un tube à rayons cathodiques ou canon à électrons, placé dans un tube en verre dans lequel il y a le vide (voir figure I.1). Les électrons sont émis par une cathode chauffée et accélérés par une forte tension appliquée à l'anode, percée d'un petit trou. Le faisceau d'électrons est envoyé sur un écran fluorescent où il laisse une trace visible ou spot. Avant d'atteindre l'écran, le faisceau d'électrons passe entre deux paires de plaques auxquelles on peut appliquer une différence de potentiel qui crée un champ électrique entre celles-ci.

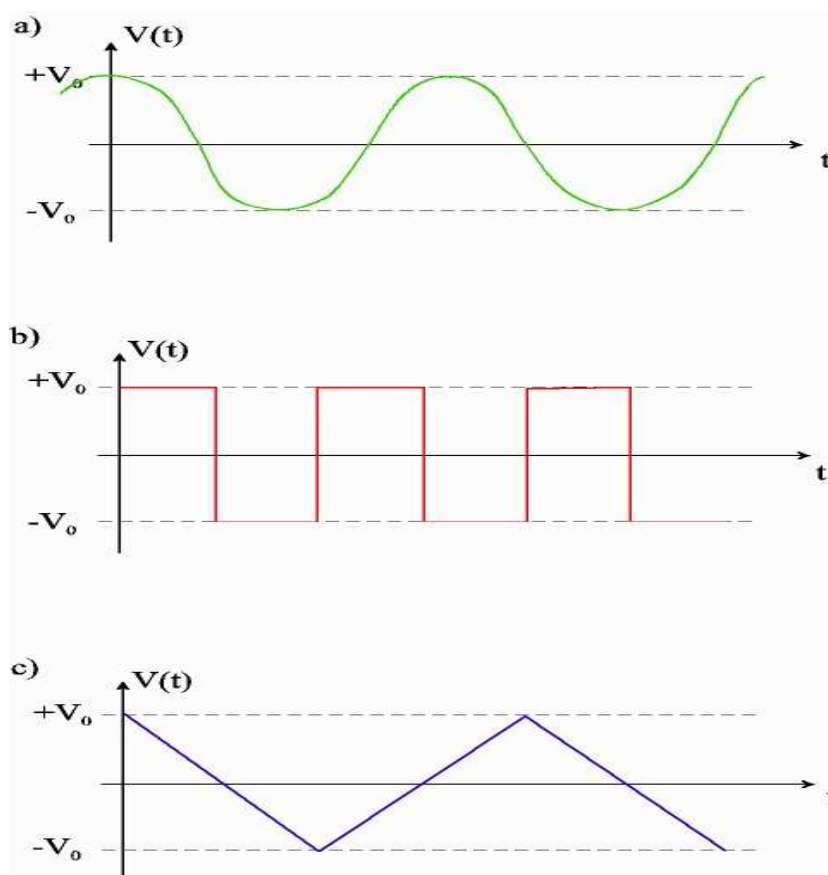
Par conséquent une force agit sur les électrons.

Une paire de plaques est verticale et permet de dévier le faisceau horizontalement, l'autre est horizontale et permet de dévier le faisceau verticalement. En variant les tensions des plaques, le spot laissé par les électrons sur l'écran se déplace sur celui-ci et dessine une trajectoire qui peut être observée.



*Figure I.1 principe de fonctionnement d'un oscilloscope*

Le mode le plus courant d'utilisation de l'oscilloscope consiste à appliquer une tension dite de balayage aux plaques verticales. Celle-ci fait dévier le spot de gauche à droite, à vitesse constante et le fait revenir rapidement à gauche lorsqu'il atteint l'extrémité droite de l'écran. La différence de potentiel à observer est placée entre les plaques horizontales et fait dévier le spot verticalement. La combinaison des deux déviations permet d'observer à l'écran la variation de la tension en fonction du temps.



*Figure I.2 les différents types des signaux*

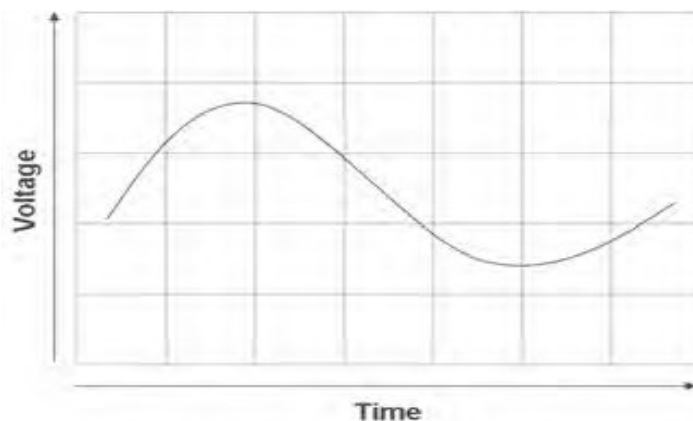
La figure I.2 montre quelques exemples de variation dans le temps d'une différence de potentiel ainsi qu'elle peut être observée à l'écran d'un oscilloscope : tension sinusoïdale (a), tension en créneaux (b) et tension en dents de scie (c). Un réticule calibré superposé à l'image du signal permet de faire des mesures d'amplitude et de période et donc de fréquence. [1]

### **I.3 Principes de base de l'oscilloscope**

Un voltmètre affiche le niveau de tension à un certain instant ou un niveau de tension moyen, tandis qu'un oscilloscope est capable de visualiser des niveaux de tension durant une période de temps. La tension est affichée verticalement (axe Y) par rapport au temps (axe X).[2]

Un oscilloscope est utilisé pour:

- ❖ voir la forme d'un signal
- ❖ mesurer l'amplitude et la fréquence d'un signal
- ❖ mesurer le temps entre deux repères de signal
- ❖ rechercher des anomalies comme le découpage, Parasites, distorsion, crêtes, ondulation, etc.



*Figure I.3 forme sinusoïdale d'un signal*

### **I.4 Analogique vs numérique:**

Il existe deux types de base d'oscilloscopes: analogiques et numériques.



**Oscilloscope analogiques**



**Oscilloscope numérique**

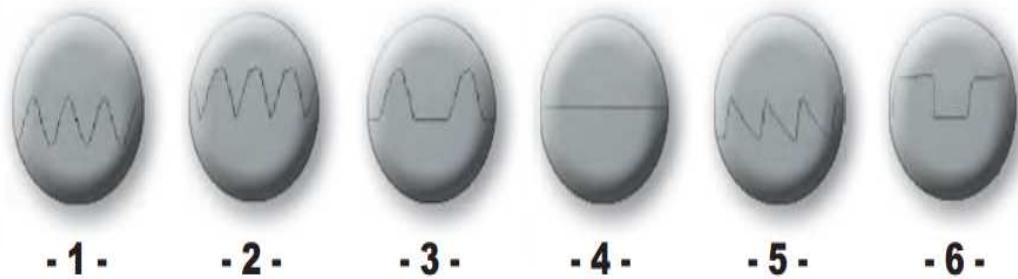
*Figure I.4 les Type de base d'un oscilloscope*

Chaque type se distingue par des applications typiques, avantages et désavantages. L'avantage des oscilloscopes numériques est qu'ils sont capables de capturer et de mémoriser l'affichage pour une étude plus approfondie. Ils sont également plus faciles à utiliser, puisqu'ils nécessitent moins de réglages pour la visualisation d'un signal. Pour nos expériences, nous utiliserons un oscilloscope

numérique. Formes d'ondes:

Un oscilloscope est généralement utilisé pour visualiser et mesurer des formes d'ondes. Une onde forme un patron qui se répète avec le temps, comme les vagues de mer. Un cycle ou une période d'une onde est la partie de l'onde qui se répète. Lors de l'affichage à l'écran d'un oscilloscope, ce phénomène est appelé forme d'onde.

Il existe de nombreuses formes d'ondes. Certaines d'entre elles seront utilisées dans nos expériences:



*Figure 1.5 les formes d'ondes*

- ❖ Onde sinusoïdale. Un exemple typique est la tension du réseau.
- ❖ Tension alternative redressée à pleine onde: la tension sortie d'un redresseur double alternance
- ❖ Tension alternative redressée à demi-onde: la tension de sortie d'une redresseuse simple alternance
- ❖ Tension CC. Oui, les oscilloscopes peuvent également mesurer la tension continue.
- ❖ Onde en dent de scie. Dans cet exemple: ondulation.
- ❖ Onde carrée. La carte dispose d'un simple oscillateur (à deux transistors) qui génère une onde carrée

## I.5 Mode d'emploi simplifié de l'oscilloscope numérique

Pour mesurer la tension  $u_{AB}$  sur la borne voie 1, il faut relier la borne A du circuit sur la borne rouge et la borne B sur la borne de masse (noire) de l'entrée 1 de l'oscilloscope. [3]



Figure I.6 oscilloscope numérique

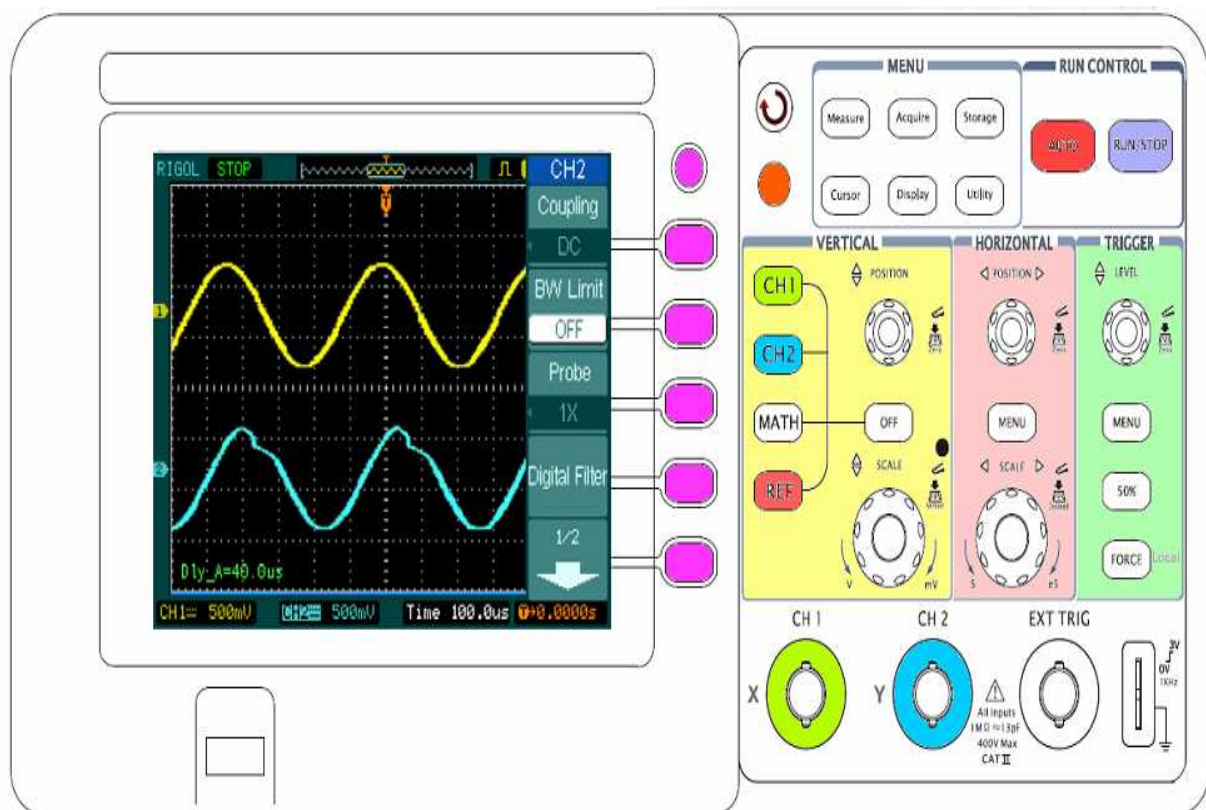


Figure I.7 Mode d'emploi simplifié de l'oscilloscope numérique

Les zones ainsi que les touches colorées sont utilisables au cours des TP de terminale.



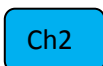
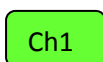
Réglages d'entrées et de sensibilités verticales



Sensibilité horizontale



Mode et seuil de déclenchement



} Activation de l'affichage des voies 1 ou 2  
(des connexions doivent être réalisées sur la prise  
Correspondante)



Mémorisation et rappel d'une courbe



Lancement d'une acquisition / Arrêt de l'enregistrement



Passage en mode tout automatique



Défilement dans les menus (validation en appuyant)

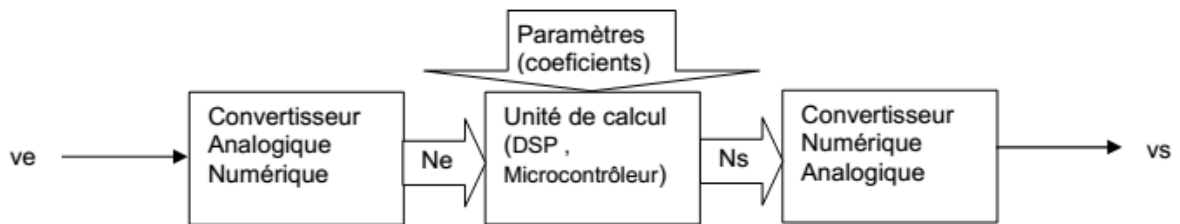


Sélection dans les menus contextuels



## I.6 Le traitement de signal numérique

Structure d'un système de traitement du signal numérique



*Figure I.8 schéma de principe de la conversion CAN [4]*

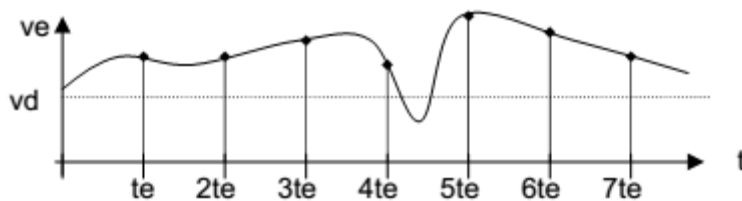
$Ne$  représente une valeur numérique de  $ve$  à un instant  $t$ .

$Ns$  représente une valeur numérique de  $ve$  à un instant  $t + t_c$  (temps de calcul)

La fonction de transfert  $Ns/Ne$  est une fonction mathématique (généralement une convolution)

## I.7 Le problème de l'échantillonnage

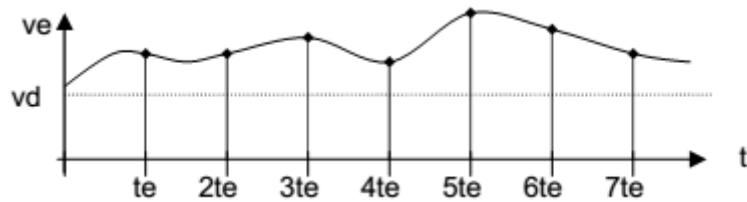
Le signal  $ve$  est mesuré (échantillonné) avec une période  $te$



*Figure I.9 problème de l'échantillonnage d'un signal*

Il n'est donc connu qu'aux instants  $nte$ .

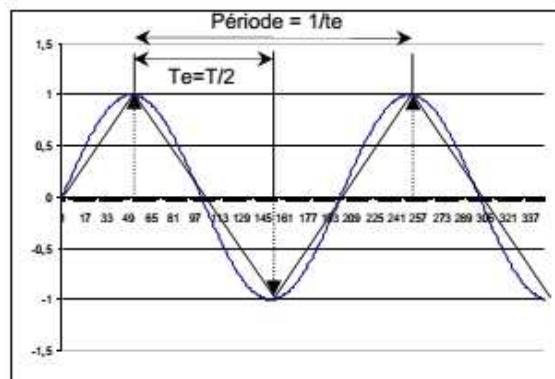
On voit ci dessus que le passage de  $v_e$  sous  $v_d$  n'est jamais détecté, tout se passe comme si l'on



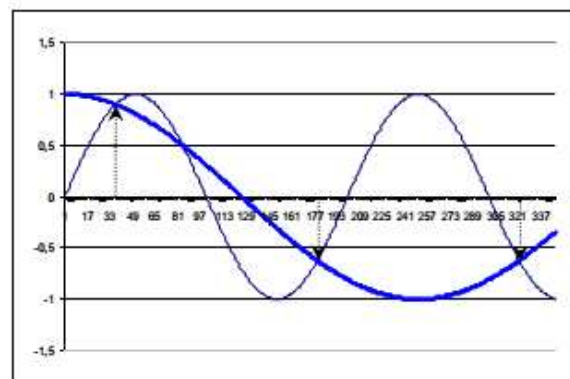
avait un signal  $v_e$  comme ceci :

**Figure I.10** graph d'un signal échantillonné

Il y a eu perte d'information. Le signal ci-dessus comporte moins d'harmoniques que le signal réel.



Ici le cas est limite  $Te=T/2$ , on peut trouver la fréquence de la sinusoïde par interpolation linéaire



Ici  $Te>T/2$ , il y a confusion entre plusieurs fréquences possibles

**Figure I.11** comparaison entre fréquence sinusoïdale et plusieurs fréquences possibles

Le critère de Nyquist établit que la fréquence d'échantillonnage ne doit pas être inférieure au double de la plus grande fréquence mesurée [4]

## I.8 Les séries de Fourier

Joseph Fourier a démontré que tout signal périodique pouvait se décomposer en une série de signaux sinusoïdaux. La première fréquence représente le fondamental (la fréquence propre du signal) et les autres représentent les harmoniques. Ce sont eux par exemple qui donnent le timbre à une note de musique. [4]

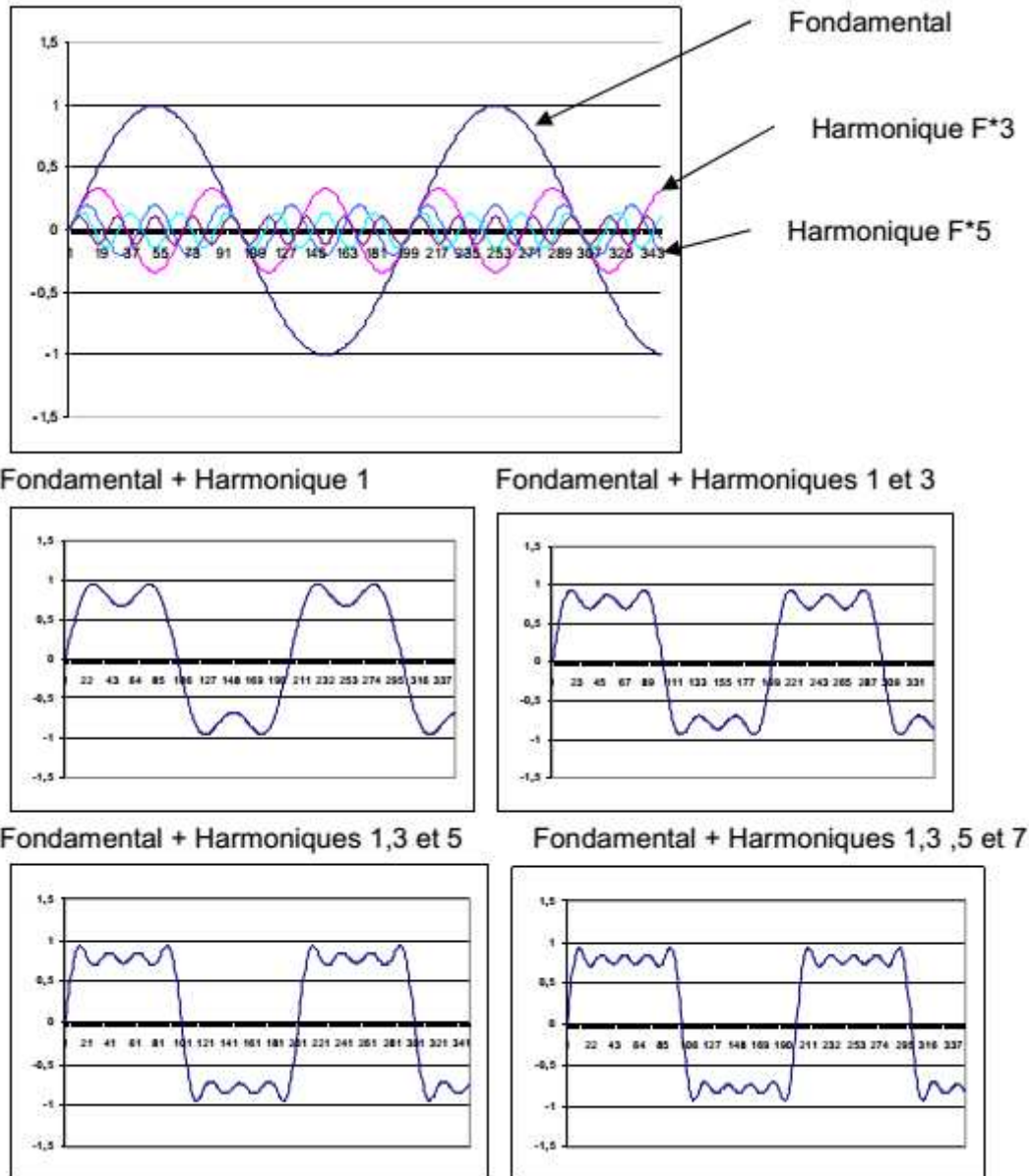
### **Exemple : reconstitution d'un signal carrée**

La décomposition d'un signal carrée en série de Fourier donne une somme de sinusoïdes multiples du fondamental et de rangs impaires Sa décomposition en série de Fourier s'écrit

$$u(t) = \frac{4E}{\pi} \left[ \sin \omega t + \frac{\sin 3\omega t}{3} + \frac{\sin 5\omega t}{5} + \frac{\sin(2p+1)\omega t}{2p+1} + \dots \right]$$

(E représente l'amplitude du fondamental et  $\omega$  sa pulsation)

Reconstitution d'un signal carré Fondamental et 7 harmoniques de rangs impaire d'un signal carré



**Figure I.12** Reconstitution d'un signal carré Fondamental et 7 harmoniques de rangs impaires d'un signal carré

On voit que plus le nombre d'harmoniques augmentent et plus l'on se rapproche d'un sigma carré. [4]

**Remarque :** Un signal carré possède une infinité d'harmoniques dont l'amplitude tend vers 0.

(Un signal carré n'existera jamais, on aura toujours affaire à une approximation représentée par les temps de montée et de descente du signal)

### 1.9 Echantillonnage

Si l'on respecte le critère de Nyquist  $F_e > F_{\max}/2$  on voit qu'il ne sera pas possible de traiter numériquement un signal carré.

Dans la réalité un signal carré n'existe pas, les fronts ont toujours un temps de montée non nul.

Le spectre d'un signal analogique est donc borné, il a généralement une forme de ce type

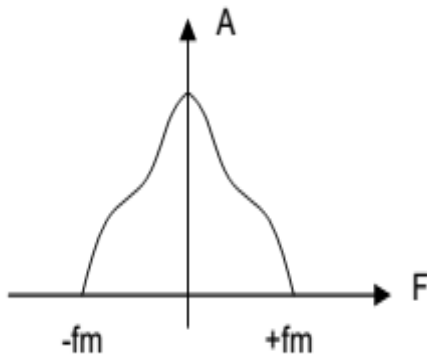
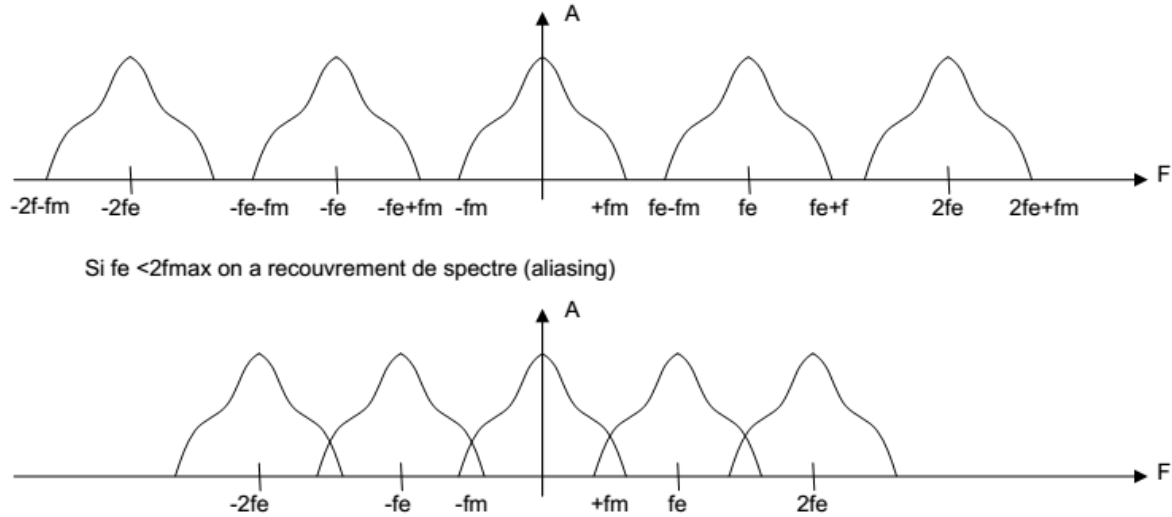


Figure I.13 spectre d'un signal analogique

**Remarque :** les fréquences négatives ne représentent qu'un concept mathématique de des séries de fourrier. [4]

## **I.10 Théorème de Shannon :**

L'échantillonnage a pour effet de multiplier le signal avec une infinité de sinusoides multiples de  $f_e$



*Figure I.14 schéma signifie le théorème de Shannon*

Le signal échantillonné n'est plus le signal d'origine

Le signal présent à l'entrée du CAN ne doit pas présenter de fréquences supérieures à  $f_e/2$ . On place toujours un filtre passe bas (appelé filtre anti-aliasing) avant le CAN. [4]

Plus la fréquence d'échantillonnage est grande :

Plus la détermination des différentes composantes d'un signal est facile.

Moins il a de contrainte sur les caractéristiques du filtre anti-repliement

Moins il a de bruit dû à l'échantillonnage.

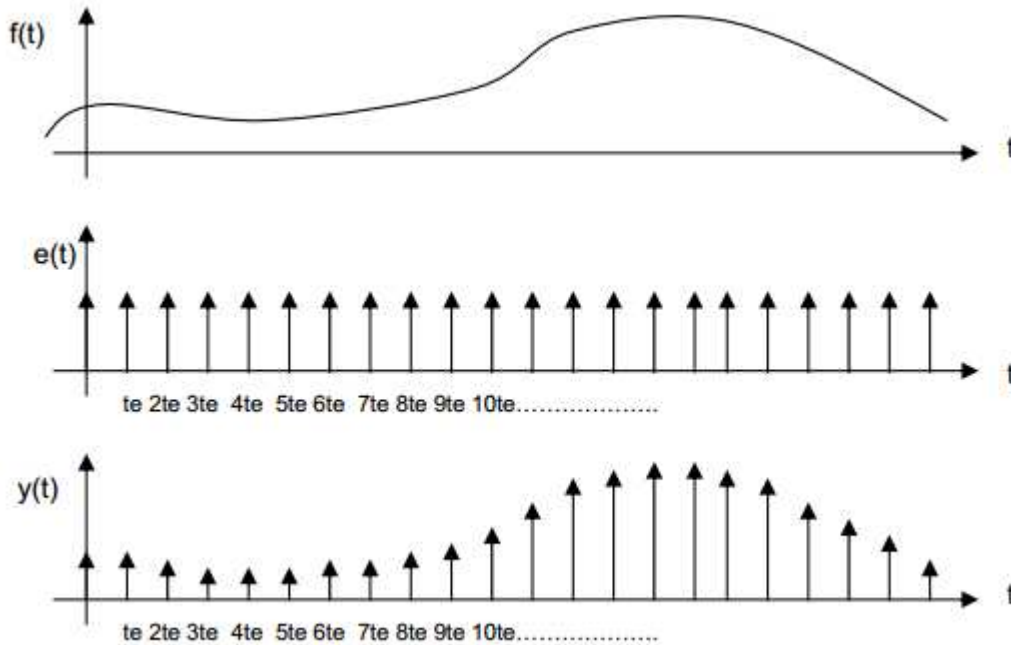
## **1.11. Représentation mathématique d'un signal Echantillonné**



*Figure I.15 Représentation mathématique d'un signal Echantillonné*

Une impulsion de Dirac est un concept mathématique, sa surface vaut  $\int \delta(t) = 1$

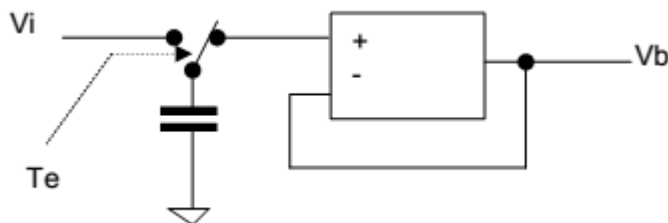
L'échantillonnage revient à multiplier le signal analogique par un peigne d'impulsions de Dirac



**Figure I.16** conversion analogique numérique

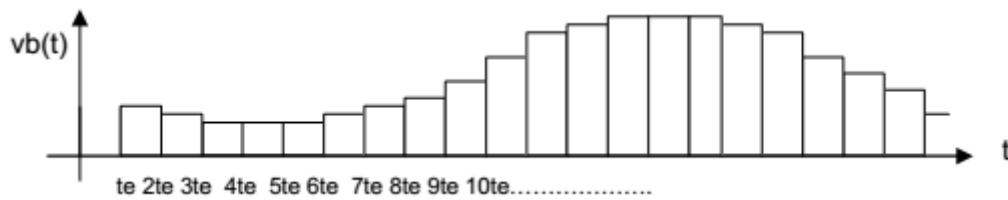
$$y(t) = \sum_{n=-\infty}^{+\infty} f(t) \times \delta(t - nte)$$

En pratique les convertisseurs analogiques numériques ont toujours un temps de conversion non nul. Il est indispensable que le signal mesuré n'évolue pas pendant la conversion. On place avant le CAN un échantillonneur bloqueur



**Figure I.17** échantillonneur bloqueur

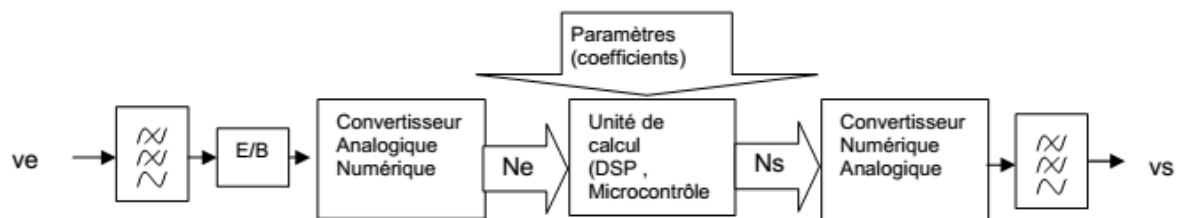
Le signal  $V_b$  a cette allure.



**Figure 1.18** signal après l'échantillonneur bloqueur

C'est un signal différent du signal à mesurer, il possède toutes les fréquences de celui ci plus des harmoniques héritées du peigne de Dirac.

La chaîne complète de traitement de signal est donc la suivante :



**Figure 1.19** La chaîne complète de traitement de signal

Après avoir saisi un échantillon, le calculateur définit la sortie  $N_s$ , le temps de calcul doit donc être inférieur à  $t_e$ . Les filtres numériques ne font appel qu'à des additions et à des multiplications, un microcontrôleur adapté au traitement de signal possédera une instruction MAC (multiplie and accumule) qui effectue une multiplication et additionne le résultat au résultat partiel.

Les DSP (Digital Signal Processor) possèdent toujours une instruction de ce type câblée, son temps d'exécution entre 10nS à 100nS. [4]

### **I.12 Conclusion**

Les avantages des systèmes numériques sont certains.

Cependant, notons que le passage dans le numérique s'accompagne d'une perte d'information puisque, du signal analogique ne sont conservés que des échantillons.

L'enjeu est donc de prendre suffisamment d'échantillons avec une cadence acceptable pour reconstruire au mieux le signal de départ tout en gardant un signal qui ne soit pas trop gourmand en espace.

*Chapitre II :*  
*Etude de la liaison RS232 et sa*  
*programmation*



## II.1 Introduction

Le développement des applications à base des microcontrôleurs PIC est devenu de plus en plus courant, ceci est dû à plusieurs causes : beaucoup de ressources internes (périphériques intégrés), mémoires embarquées de plus en plus grande, vitesse de calcul accrue... En effet, un microcontrôleur est un composant parfaitement adapté à des applications embarquées, il permet, en le programmant, d'effectuer et de contrôler une tâche tout en comparant son état à des conditions préfixées par l'utilisateur.

## II.2 Le microcontrôleur 16F877

Un microcontrôleur est un composant électronique ayant une unité de traitement de données, des mémoires, des interfaces de communication (entrées/sorties, ports séries..) et de multiples ressources internes. Les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications de grand public ou professionnelles, il existe plusieurs familles. La société américaine Micro-chip a mis au point un microcontrôleur CMOS, appelé PIC. Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

### II.2.1 Définition d'un PIC

Un PIC (Programmable Interface Controller) est un microcontrôleur de Micro-chip Technologie. Un PIC est un composant dit RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instruction réduit. Ces microcontrôleurs sont conçus sur une architecture dite HAVARD, elle est basée sur deux bus, un pour les données (bus de données) et l'autre pour les instructions (bus de programme).[5]

### II.2.2 Les différentes familles des PICS :

La famille des PICS est subdivisée à l'heure actuelle en trois grandes familles :

- Base-line: c'est la famille où l'unité centrale travaille sur 12-bits, exemple : 12CXXX...
- Mid-range : c'est la famille où l'unité centrale travaille sur 14-bits, exemple : 16F877A.
- High-end : c'est la famille où l'unité centrale travaille sur 16-bits, exemple : 18FXXX, 18CXXX ...

### II.2.3 Les caractéristiques du PIC 16F877A :

Le PIC 16F877A est un microcontrôleur de Micro-chip, qui fait partie intégrante de la famille Mid-rang (16) dont la mémoire programme est de type flash(F), ce µc est un circuit intégré à 40 broches dont les caractéristiques générales peuvent être résumé comme suit :



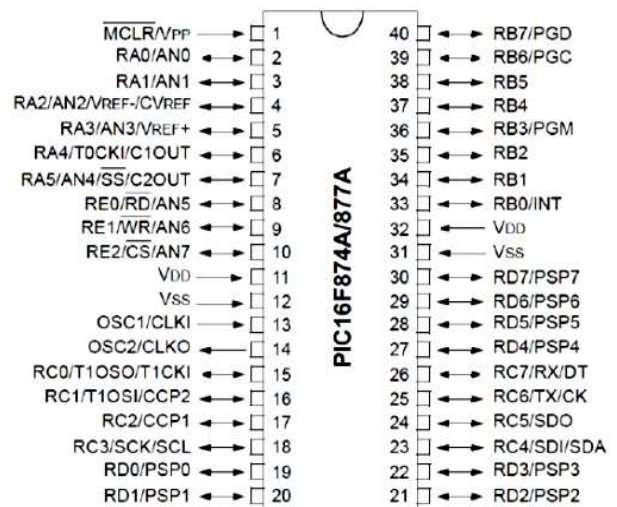
**Figure II.1** Microcontrôleur PIC 16F877A

- Une fréquence de fonctionnement allant jusqu'à 20MHz.
- Jeu d'instructions de 35 instructions
- Une mémoire de programme de type FLASH sur 8K mots (1 mot=14 bits).
- Une mémoire de données de type RAM sur 368 octets.
- Une mémoire de données de type EEPROM sur 256 octets.
- Une pile (Stack) à 8 niveaux.
- 14 sources d'interruptions.
- Un chien de garde WDT (Watch Dog Timer).
- Timer0 (compteur 8-bits avec pré-diviseur).
- Timer1 (compteur 16-bits avec pré-diviseur et possibilité d'utiliser une horloge externe).
- Timer2 (compteur 8-bits avec pré-diviseur et post-diviseur).
- Un convertisseur Analogique-Numérique 10-bits à 8 entrées multiplexées.
- Deux modules de capture/comparaison/PWM.
- Ports Entrées/Sorties : A, B, C, D et E.
- Une interface de communication série (MSSP, USART).
- Une interface de communication parallèle (PSP).
- Tension de fonctionnement est entre 2.0V et 5.5V. [5]

## II.2.4 Architecture externe du PIC 16F877A

Le boîtier ci-contre décrit l'architecture externe du 16F877A qui comprend 40 pins dont :

- 33 pins d'entrées/sorties multiplexées avec d'autres fonctions.
- 4 pins pour l'alimentation : VDD et VSS.
- 2 pins pour l'oscillateur : OSC0 et OSC1
- 1 pin pour le RESET : MCLR



**Figure II.2** Boîtier du PIC 16F877A [5]



Après compilation de votre code, le compilateur génère un fichier « \*.hex », une suite de codes hexadécimaux. Celui-ci est transféré ensuite dans la mémoire programme du PIC à l'aide du programmeur.[5]

Cette mémoire n'est pas reliée au bus de données (DATA Bus), sa vocation est déstocker le programme du PIC, mais pas les variables de votre programme. Le gros avantage de la mémoire FLASH c'est que vous pouvez la réécrire, donc implanter un nouveau programme dans le PIC.

### II.2.6.2 La mémoire EEPROM :

La mémoire EEPROM « Electrical Erasable Programmable Read Only Memory ». L'EEPROM est plutôt une mémoire de stockage de donnée à long terme, alors que la RAM est utilisée pour les variables du programme. Sur le PIC 16F877A, on a 256 octets d'EEPROM disponible. Les mémoires de type EEPROM sont limitées en nombre de cycles d'effacement / écriture. Ce nombre de cycle est tout de même de l'ordre du million pour le PIC. [5]

### II.2.6.3 La mémoire RAM :

La Mémoire RAM, qui fait partie de la zone d'adressage des données. Elle comprend tous les registres spéciaux permettant de contrôler le cœur du PIC (le microprocesseur) ainsi que ses périphériques. Elle contient également des cases mémoires à usage générique dans lesquelles pourront être stockées les variables de nos futurs programmes.

La mémoire RAM disponible du 16F877A est de 368 octets. [5]

### II.2.6.4 Les ports d'entrées/sorties :

Le PIC 16F877A dispose de 5 ports entrées/sorties (au total 33 pins) : port A, B, C, D et E (figure II.4). De plus il y a le PSP (Parallel Slave Port) : le Port Parallèle Esclave.

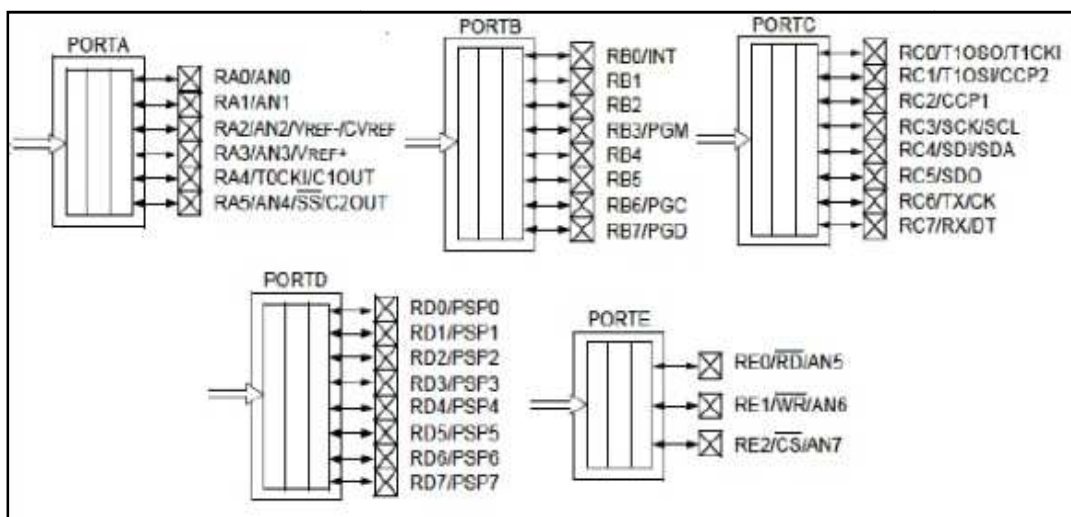


Figure II.4 Les différents ports du PIC16F877A

Un examen plus attentif du schéma ci-dessus nous montre cependant que les lignes d'entrées/sorties (les pattes du composant) correspondantes sont également utilisées par d'autres modules du PIC.

Ainsi, les pattes du PORTA servent également au convertisseur Analogique/Numérique, les pattes du PORTD au Port Parallèle Esclave, etc.

On voit également que les ports B, C et D ont 8 lignes d'entrée/sortie, alors que le port A n'en a que 6 et le port E que 3. Ils sont tous connectés au bus de donnée (DATA BUS), on pourra donc librement les adresser pour y lire ou écrire des données, et donc allumer des LED, commander des moteurs, des afficheurs LCD, lire les données envoyées par un clavier ou un bouton poussoir... [5]

On peut configurer les entrées/sorties de chaque port en entrée ou en sortie, grâce à un registre spécial dédié à chaque port (TRISX).

Bit k de TRISX = 0  $\Longrightarrow$  broche k de PORTX = sortie

Bit k de TRISX = 1  $\Longrightarrow$  broche k de PORTX = entrée

De plus, un des ports (le port B) possède des résistances de « pull-up » internes qui peuvent être validées ou non par logiciel.[15]

### II.2.7 Les différents registres du PIC 16F877A

Dans ce paragraphe, nous allons essayer de décrire seulement les registres utilisés dans notre projet.

#### II.2.7.1 Le registre W :

C'est le registre de travail (Works registre) possède 14 bits. Il est utilisé par les PICs pour réaliser toutes sortes de calculs. La destination du résultat peut en général être un emplacement dans la RAM ou le registre de travail (W). C'est donc un registre fondamental.

#### II.2.7.2 Le registre STATUS ou registre d'états :

Le registre d'état appelé STATUS est un registre 8 bits, son rôle est de donner diverses informations à l'utilisateur sur l'état de fonctionnement ou sur le résultat d'une opération.[5]

#### II.2.7.3 Le registre INTCON ou registre de contrôle d'interruptions:

Ce registre permet de configurer les différentes sources d'interruption.[5]

#### II.2.7.4 Le registre OPTION-REG :

Ce registre peut être lu et écrit, il contient les bits de contrôle qui permettent de configurer le pré-diviseur du Timer0, les interruptions externes et le choix du pull-up sur le port B.[5]

## II.2.7.5 Le registre T1CON :

Ce registre permet de contrôler le Timer1.[5]

## II.2.7.6 Le registre T2CON :

C'est un registre 8bits qui permet de paramétrer le pré diviseur et le post diviseur ainsi que d'autoriser ou non le fonctionnement du Timer2.[5]

## II.2.7.7 Les registres CCP1CON et CCP2CON :

Ces registres permettent de contrôler deux modules identiques CCP1 et CCP2.[5]

## II.2.7.8 Les registres ADCON0 et ADCON1 :

Sont des registres de configuration du convertisseur analogique numérique. [5]

## II.2.8 Les interruptions

Le PIC 116F877A possède 13 sources d'interruptions, les plus importantes à connaître, car ce sont les plus utilisées dans notre projet, sont :

*Tableau II.1 Les interruptions*

| Interruption | Source d'interruption   |
|--------------|---|
| TMR0I        | Dépassement du TIMER0 « débordement le passage de 255 à 0 »   |
| TMR1I        | Dépassement du TIMER1 « débordement le passage de 65535 à 0 » |
| TMR2I        | Débordement de TIMER2   |
| INT          | Une interruption externe, action sur la broche INT/RB0        |
| PSPI         | Lecture/Ecriture sur le port parallèle(PORTD)                 |
| ADI          | Fin de conversion analogique numérique                        |
| CCP1I        | Capture/comparaison/PWM avec module CCP1                      |
| CCP2I        | Capture/comparaison/PWM avec module CCP2                      |
| RCI          | Réception sur le port RS232                                   |
| TXI          | Emission sur le port RS232                                    |
| SSPI         | Caractère émis/reçu sur SSP (port série synchrone)            |
| EEL          | Ecriture dans EEPROM  |
| BCL          | Collision sur bus SSP en mode I <sup>2</sup> C                |

## II.2.9 Les circuits d'horloges

Le PIC 16F877A peut fonctionner en 4 modes d'oscillateur :

- **LP** : « Low Power » : version à quartz également, mais prévue spécialement pour les applications à très faible consommation. C'est donc une version dont la fréquence maximale de fonctionnement est limitée à 20KHz.
- **XT** : version à oscillateur à quartz «standard» fonctionnant jusqu'à une fréquence maximale de 4MHz.

- **HS**: « High Speed » : version à oscillateur à quartz capable de fonctionner jusqu'à une fréquence maximale de 20MHz.
- **RC**: « circuit RC » : capable de fonctionner jusqu'à 4MHz mais avec une stabilité de fréquence plus faible que les versions à quartz. [5]

## II.2.9.1 Oscillateur à quartz :

En mode LP, XT ou HS, un quartz est connecté aux pins OSC1/CLKIN et OSC2/CLKOUT équivalent à les pates 13 et 14 sur le boîtier du PIC 16F877A, pour établir l'oscillation. La figure II.5 montre comment ce quartz est connecté au PIC :

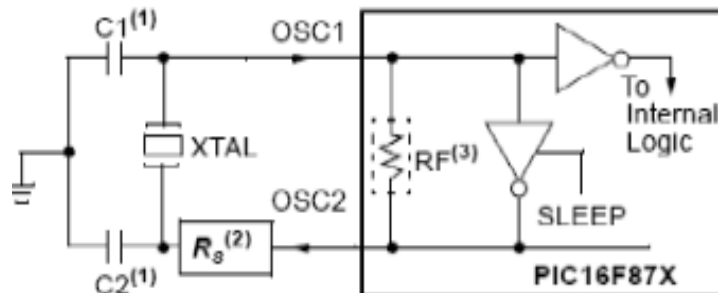


Figure II.5 Oscillateur à quartz

Toujours dans l'un de ces modes de fonctionnement, le microcontrôleur peut avoir une horloge externe connectée à la broche OSC1/CLKIN comme le montre la figure suivante :

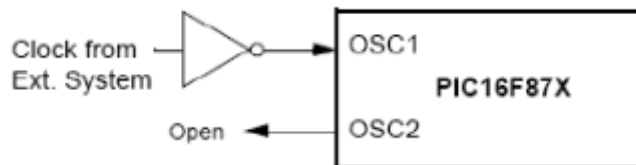


Figure II.6 Connexion d'une horloge externe [5]

## II.2.9.2 Oscillateur RC

La fréquence de l'oscillation dépend du voltage, des valeurs de R et C et de la température de fonctionnement. La figure (II.7) montre comment connecter un circuit RC au microcontrôleur :

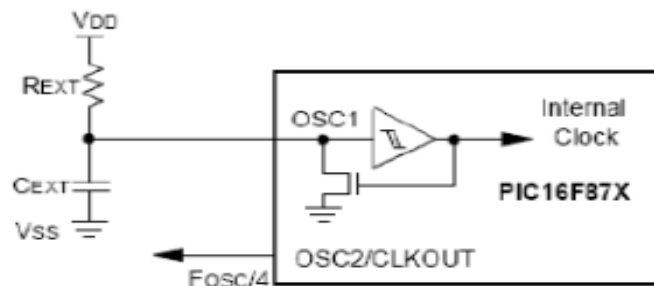


Figure II.7 Oscillateur RC

Pour générer un « cycle d'instruction », il faut 4 cycles d'horloge. Autrement dit, l'horloge système dite aussi horloge instruction est obtenue en divisant la fréquence par 4. Avec un quartz de 4 MHz, les instructions du programme s'exécutent à une cadence de 1 MHz. [5]

### II.2.10 Les Timers

Le PIC 16F877A dispose de 3 timers

#### II.2.10.1 Le Timer0 :

C'est un compteur 8 bits (0 à 255), Il peut être incrémenté par des impulsions extérieures via la broche « RA4/T0CKI » du port A (mode compteur) ou par l'horloge interne «Fosc/4 » (mode timer). [5]

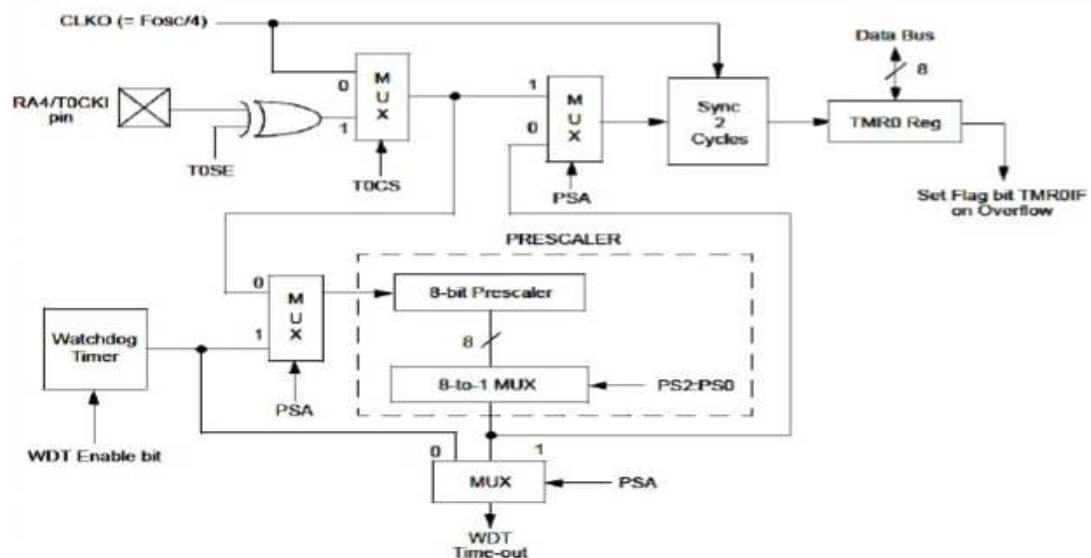
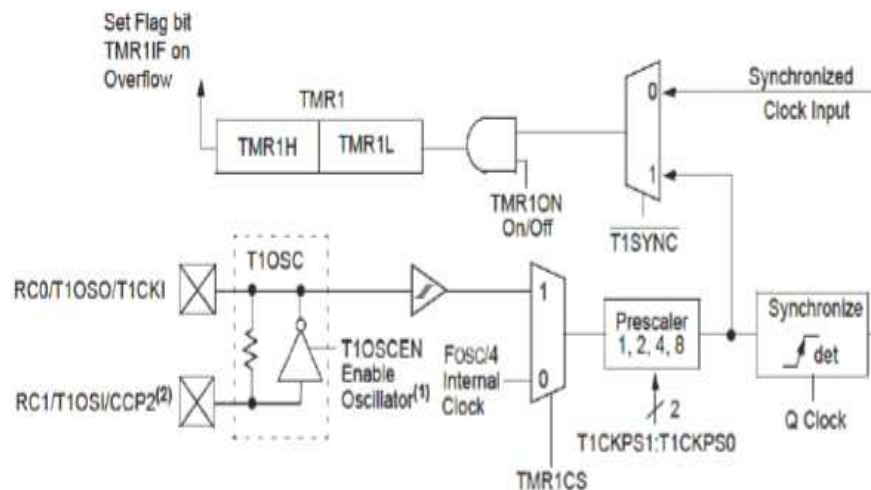


Figure II.8 Schéma du Timer0



## II.2.10.2 Le Timer1 :

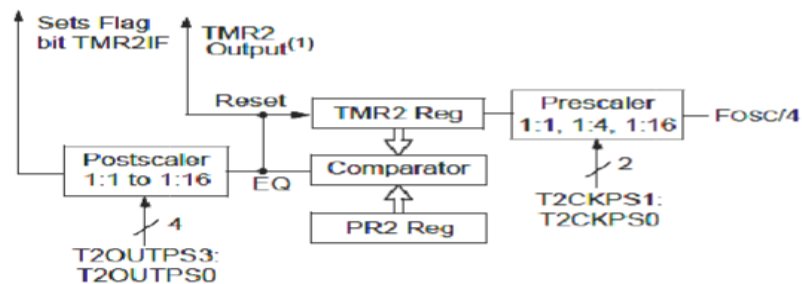
Le Timer1 fonctionne sur le même principe que le Timer0, mais avec un registre de comptage plus gros : 16 bits au lieu de 8, ce qui étend notablement ces capacités de comptage. De plus, il possède un mode de fonctionnement particulier : on peut l'utiliser en association avec un des modules CCP (modules de capture et de comparaison).[5]



*Figure II.9 Schéma descriptif du Timer1*

## II.2.10.3 Le Timer2 :

Le Timer2 a un fonctionnement différent des Timer0 et Timer1. C'est un compteur 8bits avec pré-diviseur et post-diviseur. On s'en sert pour générer des signaux carrés, ou, en association avec le module CCP, des signaux PWM. [5]



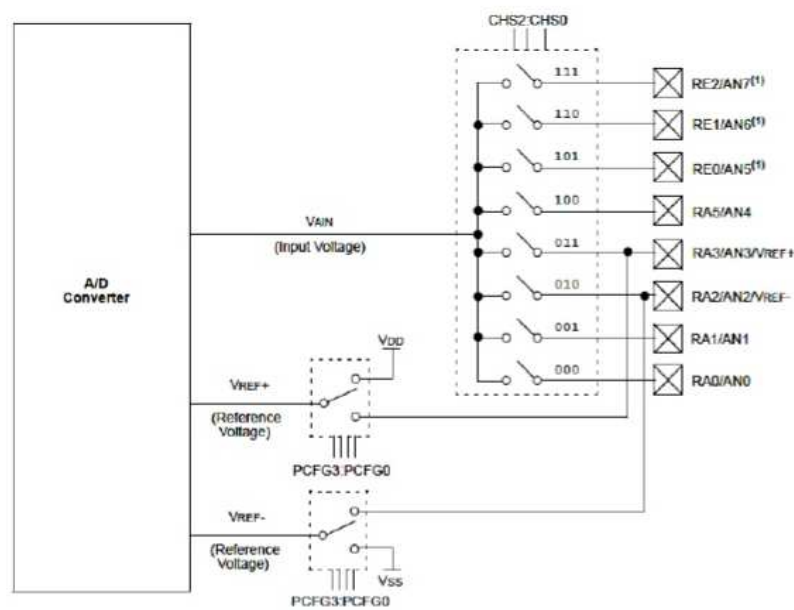
*Figure II.10 Schéma descriptif du Timer2*

## II.2.11 Le convertisseur analogique numérique :

Le convertisseur A/D convertit le signal analogique présent sur une de ses 8 entrées (figure II.11) en son équivalent numérique, notre 16F877A travail avec un convertisseur analogique numérique codé sur 10 bits.

Les pattes AN2 et AN3 peuvent être utilisées comme références de tension ou comme entrées analogiques standard, les références de tension étant dans ce dernier cas prises sur les tensions d'alimentations du PIC : VDD et VSS. (VDD pour le + et VSS pour le -).

On peut donc numériser jusqu'à 8 signaux analogiques. Pas tous en même temps bien sûr, étant donné qu'il n'y a qu'un seul module de conversion pour 8 signaux d'entrée multiplexés. [5]



*Figure II.11 Schéma du module de conversion analogique numérique*

## II.2.12 Les modules CCP1 et CCP2

Le PIC 16F877A possède deux modules CCP qui fonctionnent de la même façon. Chaque module contient un registre sur 16-bits qui peut fonctionner dans l'un des trois modes ci-dessous :

- Mode capture« CAPTURE »: ce mode permet en outre d'effectuer des mesures de temps.
- Mode comparaison« COMPARE »: ce mode permet en outre de générer des événements périodiques.
- Mode PWM signifie «Pulse Width Modulation», ce qu'on pourrait traduire par modulation de largeur d'impulsion, il s'agit d'un signal binaire de fréquence fixe dont le rapport cyclique peut être modulé par logiciel. Le rapport cyclique d'un signal binaire à fréquence fixe peut être défini comme étant le rapport entre le temps où il se trouve à l'état «1» par rapport au temps total d'un cycle.

Ces modules sont associés aux broches RC2/CCP1 et RC1/T1OSI/CCP2 suivant le mode choisi, les timer1 ou 2 vont être utilisés. Les modes capture et comparaison utilisent le timer1, tandis que le mode PWM utilise le timer2. [5]

### II.2.13 Domaines d'application des microcontrôleurs

Les domaines d'application sont extrêmement variés et touchent tous les secteurs : grand public, industriel, informatique, les télécommunications avec les téléphones portables, l'automobile pour toutes les fonctions autour du tableau de bord, de l'ouverture des portes, contrôle des sièges, feux, et autres alarmes. [5]

### II.2.14 Les différents avantages des microcontrôleurs

- ✓ Diminution de l'encombrement du matériel et du circuit imprimé
- ✓ Simplification du tracé du circuit imprimé
- ✓ Augmentation de la fiabilité du système
  - nombre de composants démunis
  - connexion composants / supports et composant circuit imprimé démunis
- ✓ 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 [5]

### II.2.15 Les défauts des microcontrôleurs

- Le microcontrôleur est souvent surdimensionné devant les besoins de l'application
- Investissement dans les outils de développement
- Écrire les programmes, les tester et tester leur mise en place sur le matériel qui entoure le microcontrôleur
- Incompatibilité possible des outils de développement pour des microcontrôleurs de même marque.
- Les microcontrôleurs les plus intégrés et les moins coûteux sont ceux disposant de ROM programmables par masque
- Le microcontrôleur présente l'avantage des microprocesseurs mais limités aux applications ne nécessitant pas trop de puissance de calcul (architecture courante 8bits)
- Il existe plusieurs architectures de microcontrôleurs de 4 à 128 bits pour les applications demandant une certaine puissance de calcul (injecteurs automobile)

## II.3 communication série RS-232

RS232 (parfois appelée EIA RS-232, EIA 232 ou TIA 232) est une norme standardisant un bus de communication de type série sur trois fils minimum (électrique, mécanique et protocole). Disponible sur presque tous les PC depuis 1962 jusqu'au milieu des années 2000, il est communément appelé le « port série ».

Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2, etc. Cela leur a valu le surnom de « ports COM », encore utilisé de nos jours. Cependant, il est de plus en plus remplacé par le port USB.

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

Les liaisons RS-232 sont fréquemment utilisées dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure, etc.). [6]

### II.3.1 Historique

Le protocole d'origine, RS-232, date de 1962. Des déclinaisons ont suivi, notamment les RS-232C en 1969 et RS-232D en 1986. Peu à peu tombé dans l'obsolescence, il finit par être remplacé par les ports USB et FireWire1 dans les années 2000. [6]

### II.3.2 Description

La connectique de cette liaison se présente fréquemment sous la forme du connecteur DE-9 ou DB-25, mais peut aussi être d'un autre type (RJ25). Seule la version DB-25 est vraiment standardisée, la DE-9 (très souvent appelée DB-9 dans le commerce) est une adaptation d'IBM lors de la création du PC AT. La transmission des éléments d'information (ou bit) s'effectue bit par bit, de manière séquentielle. [6]



*Figure II.12 connecteur DB-9 femelle*

### II.3.3 Utilisation

Bien que ce port de communication ait tendance à être remplacé par l'USB sur les PC, il reste encore très utilisé dans l'industrie, notamment grâce à sa robustesse et à sa simplicité. Ainsi, ce port est toujours d'actualité, en particulier dans les systèmes automatisés : le transfert de Gcode ou bien de lignes de programme pour machines-outils à commande numérique s'effectuent toujours par liaison RS-232.



*Figure II.13 connecteur DB-9 male*

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

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

En cas d'absence de port RS 232, il existe des adaptateurs USB/port série.[6]

### II.3.4 Transmission des informations numériques

- (a) Les informations à transmettre sont envoyés bit par bit (**poids faible en premier**) par l'émetteur sur la ligne Tx, vers le récepteur (ligne Rx) qui le reconstitue.
- b) 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 correspond à 1 bit / seconde, dans ce cas). Il existe différentes vitesses normalisées: 9600, 4800, 2400, 1200... bauds.
- c) Une communication peut se faire dans un sens (simplex), soit deux sens, émission d'abord, puis réception ensuite (half-duplex) soit émission et réception simultanées (full-duplex), ex : le téléphone. [7]

#### Remarque :

En général, une communication comprend 4 phases :

- ❖ Etablissement d'un circuit entre 2 correspondants
- ❖ Initialisation de la transmission
- ❖ Transmission proprement dite
- ❖ Libération de la transmission

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**).

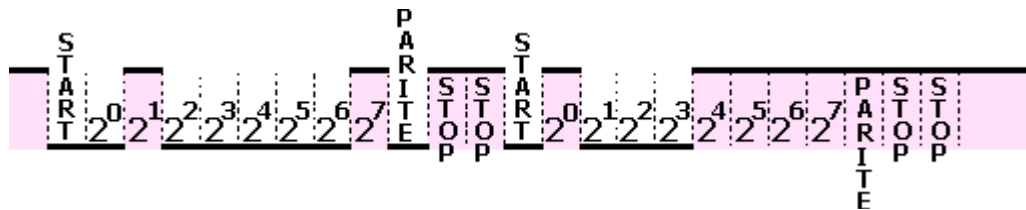
## II.3.5 Bit de parité

Pour détecter toute erreur dans la transmission des mots, on ajoute un bit de **parité**.

La parité est donc 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 impair, ce qui rend le total pair : c'est la parité paire.

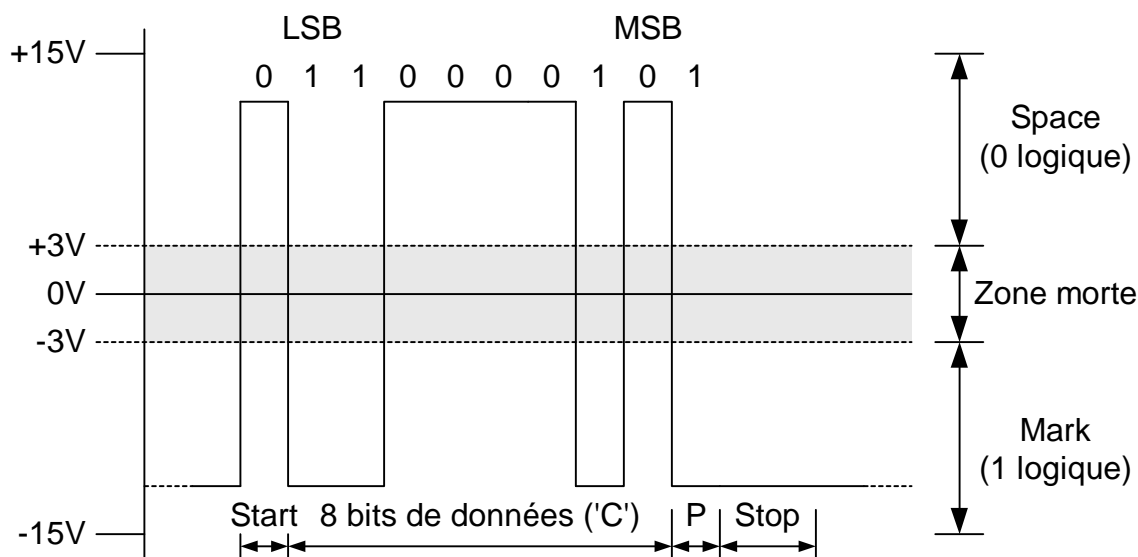
Idem pour la parité impaire : mise à 1 du bit de parité si le nombre de 1 dans le mot est pair. [7]



*Figure II.14 transmission des données.*

**Remarque :** Au repos, la ligne de transmission est au niveau logique 1.

## II.3.6 Les signaux électriques



*Figure II.15 schéma signifié la transmission des signaux électrique via RS232*

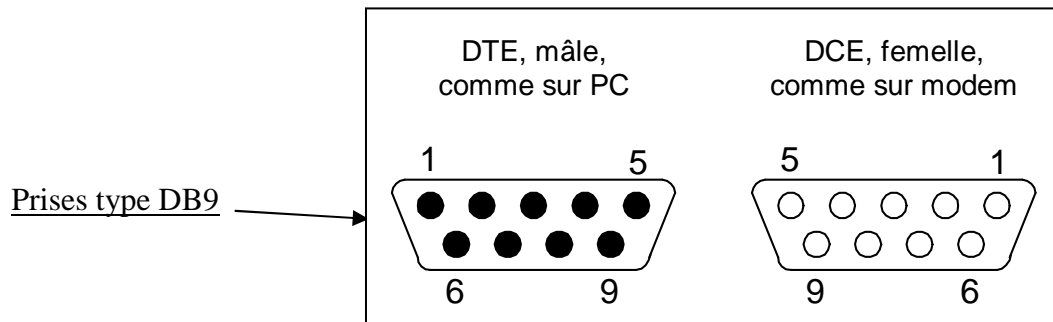
La « zone morte » est assez importante (6V), ce qui permet à cette liaison RS-232 de n'être que peu sensible aux perturbations, et donc de pouvoir être mise en place sur de longues distances (max : 50m) ; la vitesse de transmission des informations est limitée à 20kbit/s. [7]

## II3.7 Caractéristique de la liaison V24 - RS232: la vitesse

Il est possible de transmettre 960 mots d'1 octet, associé à un bit de Start, un bit de stop, sans bit de parité, à 9600 bit/s.

En effet, cela fait au total des trames (mots) de 10 bits, donc en 1 seconde,  $9600 / 10 = 960$  mots en 1 seconde. [7]

## II3.8 Raccordements



**Figure II.16** raccordement DB9 [7]

**Tableau II.2** portes séries

| DB 9 | DB 25 | Signal | Fonction   | Sens |
|------|-------|--------|--|------|
| 1    | 8     | DCD    | Détection de porteuse/tonalité (Data Carrier Detected) | ⇐ E1 |
| 2    | 3     | RxD    | Réception des données (Received Data)                  | ⇐ E  |
| 3    | 2     | TxD    | Transmission des données (Transmitted Data)            | ⇒ S  |
| 4    | 20    | DTR    | Terminal prêt (Data Terminal Ready)                    | ⇒ S1 |
| 5    | 7     | SG     | Masse du signal (Signal Ground)                        |      |
| 6    | 6     | DSR    | Données prêtes (Data Set Ready)                        | ⇐ E1 |
| 7    | 4     | RTS    | Demande d'émission (Request To Send)                   | ⇒ S1 |
| 8    | 5     | CTS    | Prêt pour l'émission (Clear To Send)                   | ⇐ E1 |
| 9    | 22    | RI     | Indicateur de sonnerie (Ring Indicator)                | ⇒ E1 |

## II3.9 Conclusion

Les anciennes entrées/sorties de nos micro-ordinateurs ont largement profité des développements rapides de ces dernières années. Les ports série et parallèle déjà présent sur les premiers PC commencent à être à bout de souffle et ne répondent plus vraiment aux besoins des utilisateurs.

*Chapitre III : Etude de l'interface de transfert  
des données vers le PC et la visualisation du  
signal en temps réel.*



# Etude de l'interface de transfert des données vers le PC et la visualisation du signal en temps réel.

---

## III.1 Introduction

Pour la visualisation d'un signal électrique en temps réel on a besoin d'une interface de transfert des données sur un PC, cette interface faire la réception des données et tracer au même temps dans une graph.

## III.2 définition

Visual Basic est l'outil le plus facile à utiliser pour créer des applications Microsoft Windows. Visual Basic se compose des deux mots Visual et Basic.

Le mot « Visual » fait référence à la méthode utilisée pour créer l'interface graphique utilisateur (GUI, Graphical User Interface). Au lieu de rédiger de multiples lignes de code pour décrire l'apparence et l'emplacement des éléments d'interface, il vous suffit de glisser déplacer des objets prédéfinis à l'endroit adéquat sur l'écran.

Le mot « Basic » fait référence au langage BASIC (Beginners All-Purpose Symbolique Instruction Code), langage le plus utilisé par les programmeurs depuis les débuts de l'informatique. Visual Basic constitue une évolution par rapport au langage BASIC initial et comporte aujourd'hui plusieurs centaines d'instructions de fonctions et de mots clés, dont un grand nombre font directement référence à l'interface graphique utilisateur (GUI) de Windows. [8]

## III.3 Environnement de développement intégré (EDI) :

L'environnement de développement intégré de Visual Basic permet de créer, exécuter et déboguer des programmes Windows dans une seule application (§ savoir Visual Basic). [8]

## III.4 Présentation de l'Environnement de Développement Intégré (EDI) :

Au démarrage de Visual Basic, la boîte de dialogue suivante, intitulée Nouveau projet (cf. barre de titre), S'affiche. Elle permet de choisir le type de projet que l'on souhaite créer. [8]

Etude de l'interface de transfert des données vers le PC et la visualisation du signal en temps réel.

---



*Figure III.1* fenêtre au démarrage de Visual Basic

## Etude de l'interface de transfert des données vers le PC et la visualisation du signal en temps réel.

### III.5 Développement d'un programme sous VB

– Trois étapes :

- Création de l'interface utilisateur à l'aide des contrôles VB
- Définition des caractéristiques ou propriétés des éléments qui composent l'interface
- Ecriture du code de programmation pour un ou plusieurs éléments de l'interface en fonction des besoins.

Après avoir chargé Visual Basic, l'écran présente :

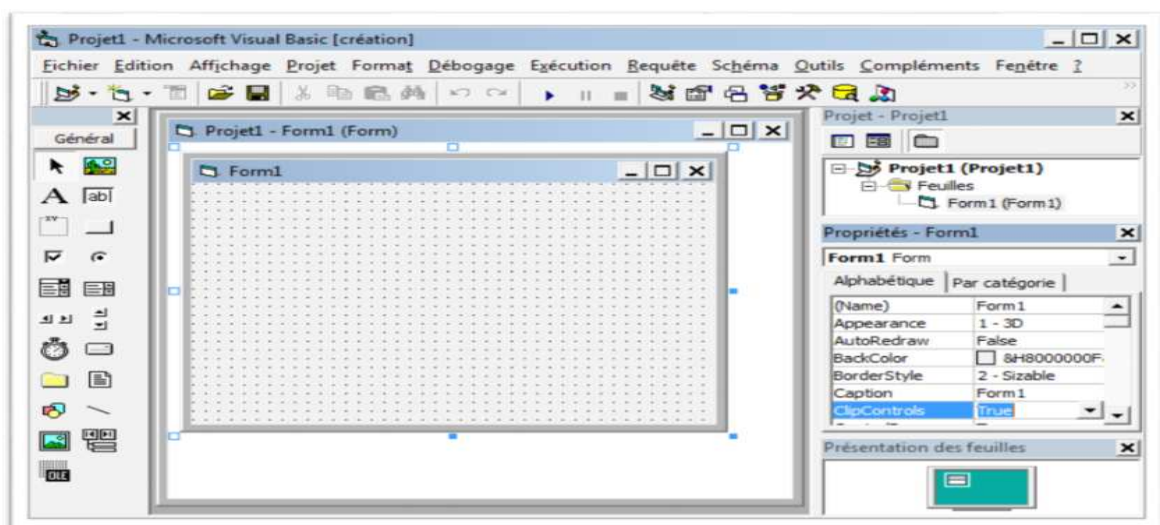
- ❖ la barre de menus qui contient le nom des menus à dérouler.
- ❖ La barre d'outils qui contient des raccourcis vers les éléments des menus fréquemment utilisés.
- ❖ La feuille qui est l'écran du programme à réaliser. Elle contiendra les contrôles.
- ❖ La boîte à outils qui propose les contrôles que l'on peut placer sur la feuille.

Pour placer un contrôle sur la feuille, il suffit de le sélectionner avec la souris puis de fixer sa position et sa taille sur la feuille.

- ❖ La fenêtre des propriétés qui donne accès aux propriétés de la feuille et des contrôles dessinés.

Pour modifier une propriété, il suffit de modifier sa valeur dans la colonne de droite.

Si l'une ou l'autre de ces parties de l'écran n'est pas affichée, le menu affichage permet de la faire apparaître. [9]



**Figure III.2** fenêtre signifie les caractéristiques ou propriétés des éléments qui composent l'interface

## Etude de l'interface de transfert des données vers le PC et la visualisation du signal en temps réel.

- Le tableau suivant récapitule les menus accessibles à partir de la barre de menu, située en haut de l'écran. En dessous sont situés, dans une barre d'outils, les raccourcis graphiques (icônes) représentant les commandes les plus courantes.

**Tableau III.1** barre de menu au visual basic

| Menu        | Description   |
|-------------|---|
| Fichier     | Contient des options pour ouvrir, fermer, écrire, ... des projets.  |
| Edition     | Contient des options telles que couper, copier, coller, etc.  |
| Affichage   | Contient des options concernant l'EDI.  |
| Projet      | Contient des options pour ajouter au projet des particularités telles que des feuilles.   |
| Format      | Contient des options pour aligner et verrouiller les contrôles d'une feuille.   |
| Débogage    | Contient des options pour déboguer le programme.  |
| Exécution   | Contient des options pour exécuter, stopper, ... un programme.  |
| Requête     | Contient des options pour manipuler des données récupérées d'une base de données.   |
| Schéma      | Contient des options pour éditer des bases de données.  |
| Outils      | Contient des options pour particulariser l'EDI.   |
| Compléments | Contient des options pour utiliser, installer et désinstaller des compléments (programmes visant à augmenter la performance de Visual Basic). |
| Fenêtres    | Contient des options pour disposer les fenêtres à l'écran.  |
| Aide        | Contient des options pour obtenir de l'aide.  |

La barre de contrôles, située dans la partie gauche de l'écran, contient des contrôles permettant de réaliser l'interface Utilisateur Graphique

(IUG) (i.e., la partie de la feuille visible par l'utilisateur).

Ces derniers permettent une programmation accélérée. La figure et le tableau suivants résument les contrôles contenus dans cette barre.



**Figure III.3** les commandes disponibles pour un projet [9]

## Etude de l'interface de transfert des données vers le PC et la visualisation du signal en temps réel.

**Tableau III.2** les commandes disponibles pour un projet [VB 6]

| Contrôle                     | Description  |
|------------------------------|--|
| Curseur                      | Permet d'interagir avec les contrôles de la feuille (en fait, ce n'est pas un contrôle).   |
| PictureBox (zone d'image)    | Permet l'affichage d'un fichier image (bmp, ico, wmf, ...).  |
| Label (étiquette)            | Permet l'affichage d'un texte non modifiable directement par l'utilisateur.  |
| TextBox (zone de saisie)     | Permet à l'utilisateur d'entrer du texte.  |
| Frame (cadre)                | Permet de regrouper d'autres contrôles tels que des cases à option.  |
| CommandButton (bouton)       | Ce contrôle est représenté par un bouton que l'utilisateur peut presser, ou cliquer, pour exécuter une action sous-jacente.                        |
| CheckBox (case à cocher)     | Permet de fournir un bouton de choix ( <i>checked</i> ou <i>unchecked</i> ).   |
| OptionButton (case à option) | Les cases à option sont utilisées en groupe (dans le même cadre), sachant que seulement une case peut être sélectionnée ( <i>True</i> ) à la fois. |
| ListBox (liste)              | Permet l'affichage de différents items (éléments).   |
| ComboBox (liste combinée)    | Permet de combiner les fonctionnalités des zones de saisie et des listes.  |
| HScrollBar                   | Une barre de défilement horizontale.   |
| VScrollBar                   | Une barre de défilement verticale.   |
| Timer (horloge)              | Permet la répétitivité de tâches (ce contrôle est non visible pour l'utilisateur).   |
| DriveListBox                 | Permet un accès aux différents disques du système.   |
| DirListBox                   | Permet un accès à des répertoires.   |
| FileListBox                  | Permet d'accéder aux fichiers d'un répertoire.   |
| Shape                        | Permet de dessiner des cercles, des rectangles, des carrés ou des ellipses.  |
| Line                         | Permet de dessiner des lignes.   |
| Image (dessin)               | Similaire au contrôle PictureBox avec des capacités moindres.  |
| Data                         | Permet la connexion à une base de données.   |
| OLE                          | Permet d'interagir avec d'autres applications Windows.   |

# Etude de l'interface de transfert des données vers le PC et la visualisation du signal en temps réel.

## III.6 Interface oscilloscope

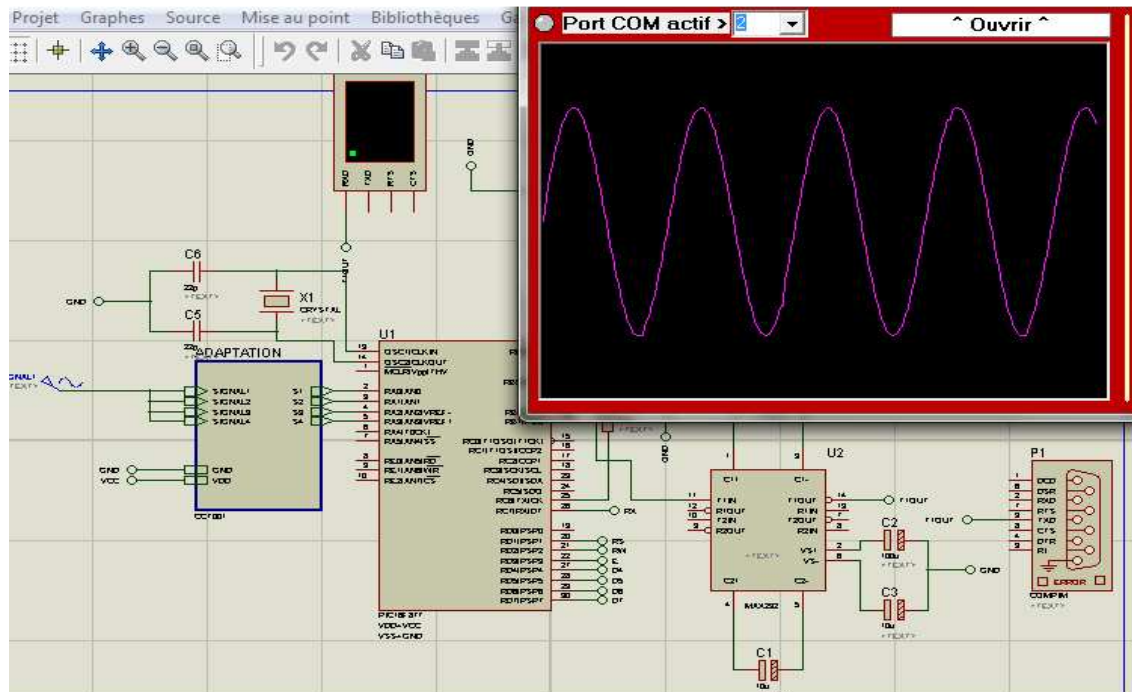


Figure III.4 interface oscilloscope

## III.7 Conclusion :

Bien qu'associé à une interface de contrôle intéressante, celle-ci se doit de communiquer en permanence ses données avec le PC.

Il a été implémenté pendant ce projet, une liaison série basée sur le protocole RS-232.

Cette technologie permet de disposer d'un débit suffisant pour l'application, mais nécessite également la connexion par fils entre le PC et la carte d'acquisition.

Cela restreint les déplacements possibles des données, en le limitant à la longueur des fils de transmissions.

De plus, cette longueur ne peut être augmentée indéfiniment, le standard RS-232 limitant la portée à quelques mètres.

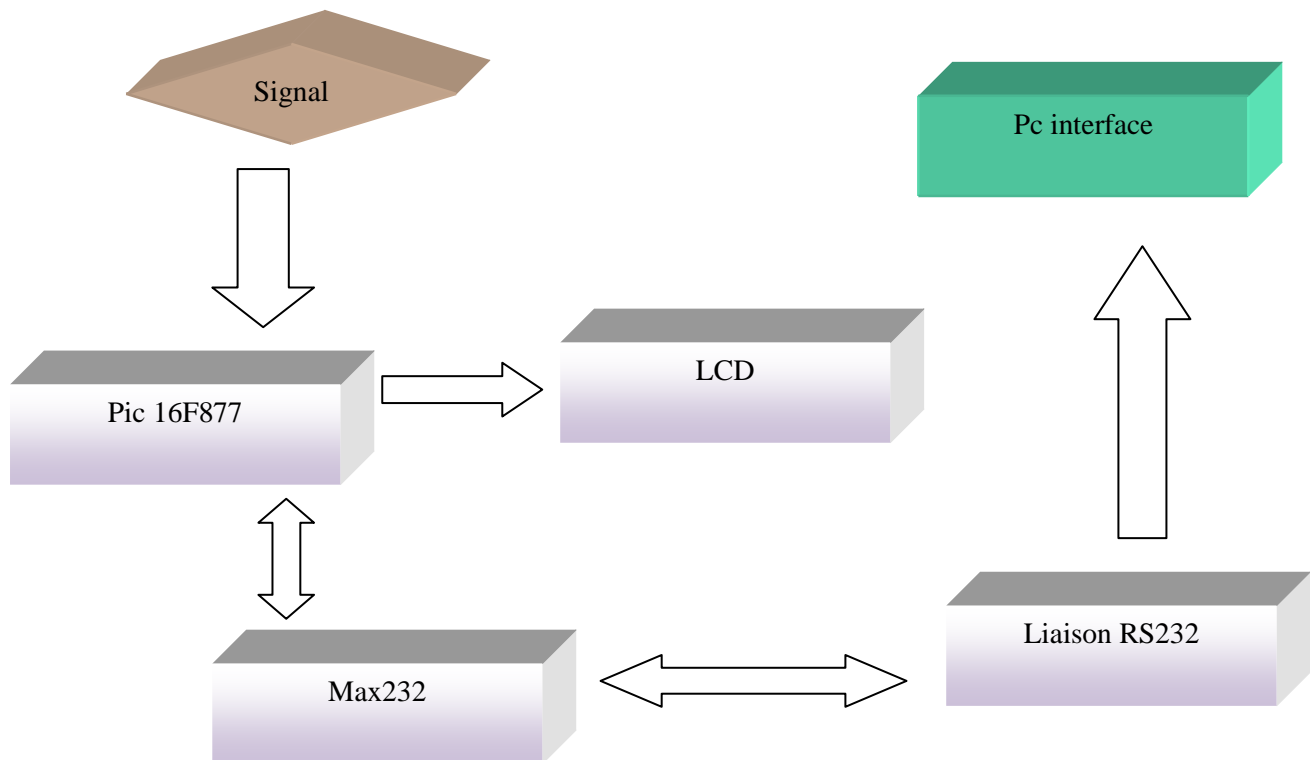
*Chapitre IV : Réalisation de la carte de transfert, son câblage au Pc et visualisation des signaux.*

## IV.I Introduction

Dans ce chapitre, nous allons tout d'abord présenter notre environnement de travail, les logiciels PROTEUS et l'outil de programmation PIC C Compiler utilisés pour la simulation et la réalisation pratique du projet, et le logiciel Visual basic pour l'interfaçage.

Ensuite, nous présenterons les résultats obtenus lors de l'expérimentation.

## IV.2 Vue d'ensemble du projet



*Figure IV.1 Schéma synoptique du projet*



### IV.3 Le logiciel de simulation Proteuse

Proteuse est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels inclus dans Proteuse permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES.

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages :

- Pack contenant des logiciels facile et rapide à comprendre et utiliser
- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet



*Figure IV.2 : PROTEUS 7 [10]*

#### ✓ ISIS :

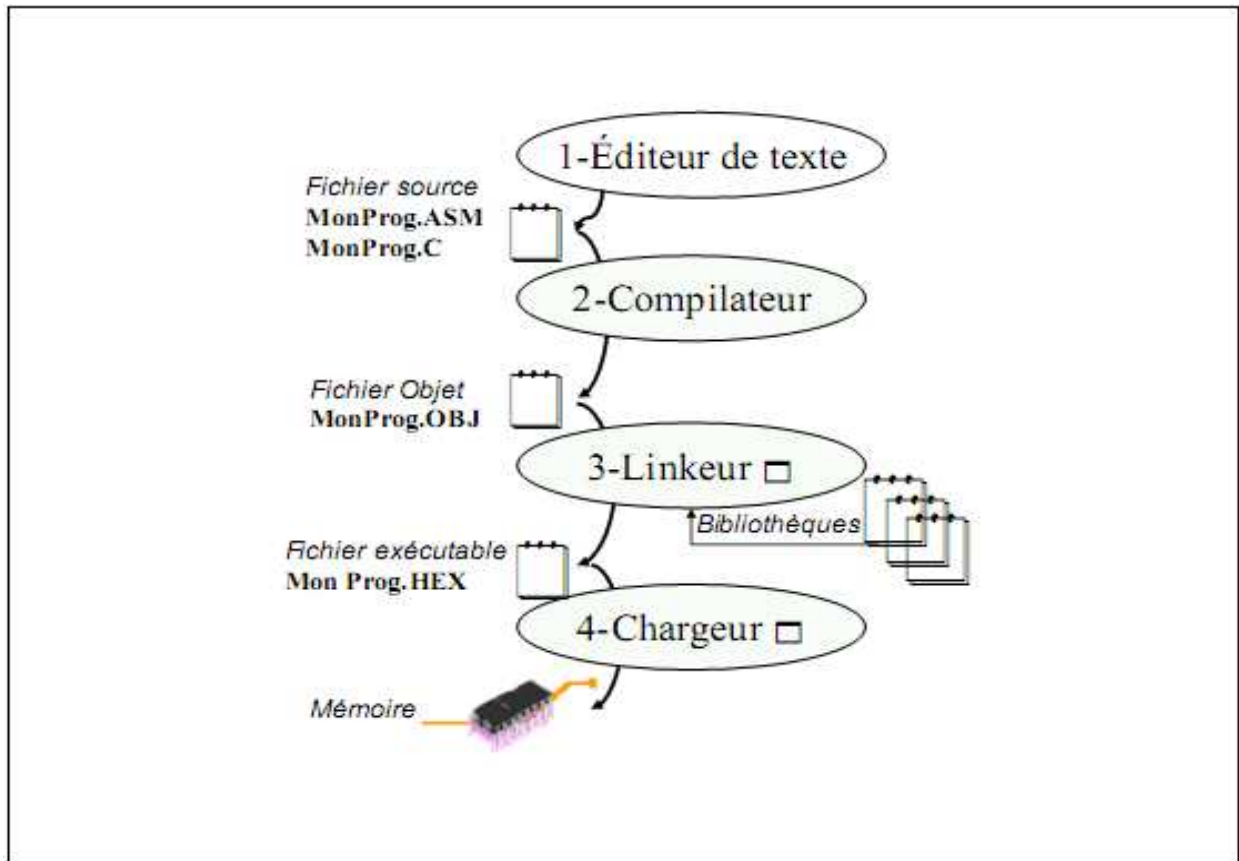
Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits. [10]

#### ✓ ARES :

Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement. [10]

### IV.4 Logiciel de programmation PIC C COMPILER DE CCS

Le PIC C Compiler est un compilateur pour langage que l'on utilise pour programmer les microcontrôleurs (16F877). Le cycle de création d'un programme exécutable par un microprocesseur est le suivant :

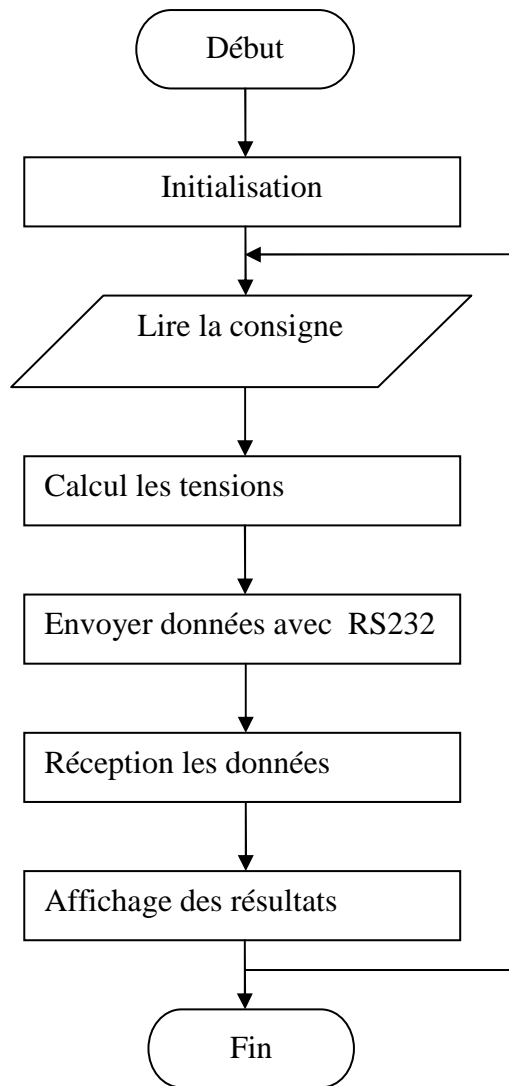


**Figure IV.3 :** Synoptique d'un cycle de compilation

- L'éditeur de texte met en forme le fichier source C à partir de directives(`#define`,`#include`)
- Le compilateur transforme le fichier C en un fichier objet (code machine), les fonctions pré-compilés sont déclarées (`*.h`)
- L'éditeur de lien LINKEUR lie (donne des adresses aux fonctions) tous les fichiers objets et crée un fichier exécutable (fichier pour debug, programme exécutable)
- Le chargeur charge le programme exécutable (`.hex`) sur le microcontrôleur. [10]

### IV.5 Organigramme principal

C'est le graphique qui représente la structure d'organisation du travail à effectuer notamment le programme principal.



**Figure IV.4 :** Organigramme principal

## IV.6 Schéma électronique du système sur PROTEUS

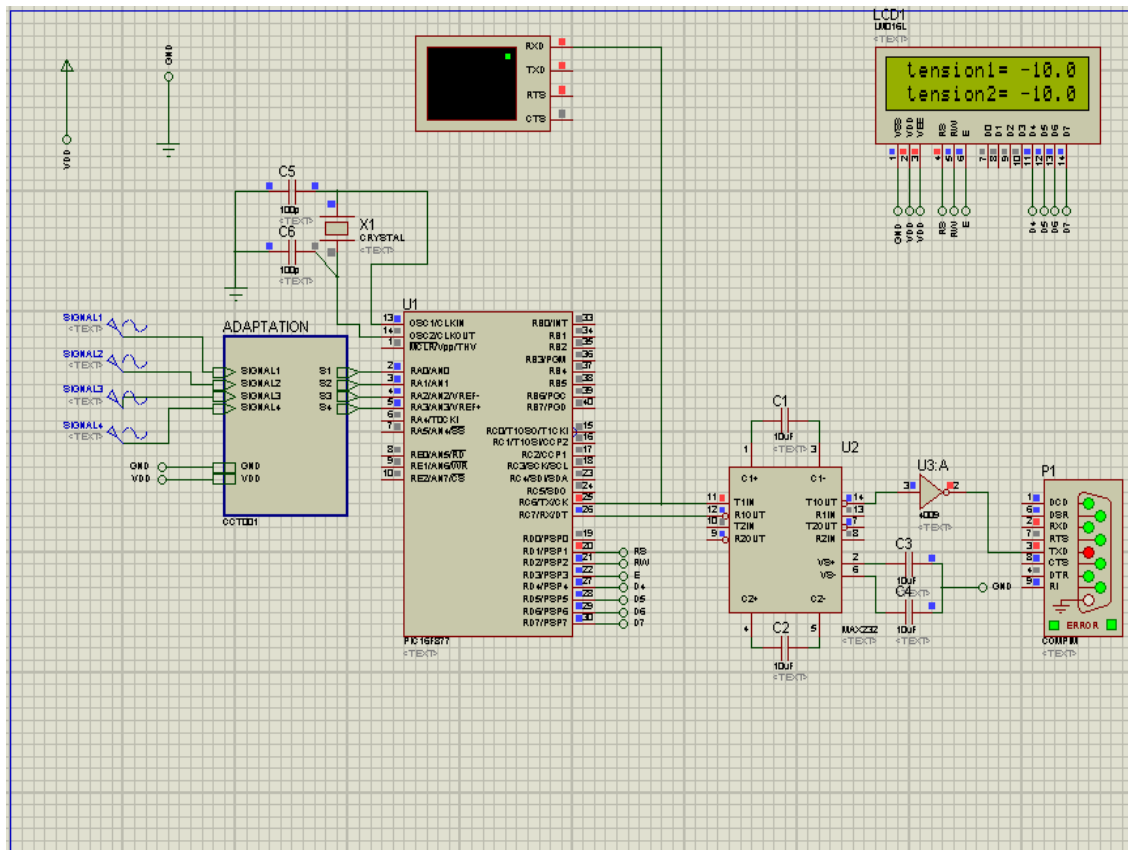


Figure IV.5 Schéma électronique du système

Pour effectuer la simulation, nous avons réalisé sur ISIS le circuit ci-dessus composé de :

- un PIC16F877 qui lit la valeur de la consigne et de la mesure et calcule la tension
- un écran LCD pour l'affichage : C'est un afficheur standard à 2 lignes de 16 (voir même plus puisque des modèles 4 lignes au même brochage existent également sur le marché) qui est raccordé comme l'indique la figure précédente. Comme vous pouvez le constater, deux restrictions sont imposées par ce mode de câblage.
- Tout d'abord l'afficheur doit fonctionner en mode 2 X 4 bits puisque seules les lignes D4 à D7 sont câblées. Cela ne présente pas vraiment d'inconvénient si ce n'est de compliquer un tout petit peu le logiciel de commande puisqu'il faut éventuellement l'adapter à ce mode de fonctionnement.

La seconde restriction est peut-être un peu plus gênante pour certaines applications mais elles restent tout de même fort peu nombreuses.

En effet, la ligne lecture et d'écriture de l'afficheur (R/W) est reliée en permanence à la masse et n'est donc pas pilotée par un port du PIC.

Il est donc possible d'écrire dans l'afficheur mais pas de relire sa mémoire de caractères ou d'affichage. Cette fonctionnalité reste cependant assez peu employée pour que cette limitation ne soit pas rédhibitoire en phase de test d'un programme.[10]

- max 232 Il sert d'interface entre une liaison série TTL (0-5V) et une liaison série RS232 (+12 - 12V) et ce avec une simple alimentation 5V.
- La liaison RS232 est un protocole de transfert de données asynchrones qui utilise pour se faire un fil de signal et un fil de masse, la norme RS232 précise les niveaux électriques des signaux chargés de véhiculer l'informations et ajoute à la donnée envoyée un certain nombre de signaux de contrôle.
- Un Sub circuit pour convertir la tension de (+10v, -10v) jusqu'à (0v, 5v) contient les 4 voies des tensions chaque voie contient 2 diviseur des tensions.

### IV.10 Conclusion

Dans ce chapitre, nous avons tout d'abord présenté notre environnement de travail à savoir les différents logiciels de simulation et le matériel utilisé pour la réalisation pratique.

Ensuite nous avons effectué une série de tests sur les circuits réalisés dans le laboratoire pour vérifier que les résultats de la simulation étaient bien compatibles avec la pratique.

Après la réalisation du montage et le câblage de différents éléments du banc d'essai, nous avons obtenue le même résultat dans la simulation.

## *Conclusion générale*

Du fait du développement et de la disponibilité de plus en plus des circuits numériques performants, le contrôle de vitesse ou de position d'une machine à courant continu est souvent réalisé en utilisant une commande numérique.

Les travaux présentés dans ce mémoire ont porté sur l'étude et la réalisation d'une commande d'un moteur à courant continu. Il s'agit de réaliser les ces circuits nécessaires pour la commande mais aussi pour la régulation de ce moteur et d'élaborer un programme en langage C permettant d'atteindre un fonctionnement performant qui sera implémenté dans un microcontrôleur pour assurer cette régulation.

Les résultats obtenus lors de nos tests nous montrent que le PIC 16F877 s'adapte bien à la commande en vitesse du moteur à courant continu.

Ce travail peut également être amélioré avec l'utilisation d'un moteur courant continu de grande puissance et l'augmentation de la tension d'alimentation du hacheur jusqu'à 220V.

Ils montrent également que le banc d'essai réalisé autour de la carte 877-DATS notamment le hacheur et la carte d'acquisition peut être utilisé à des fins pédagogiques pour effectuer des travaux pratiques avec les étudiants d'informatique industrielle.

[1] Les appareils de mesures électriques. L'oscilloscope.

<http://w3.iihe.ac.be/~cvdvelde/Info/Cours/ChapIX.pdf>

[2]- Velleman NV • Legen Heirweg 33 B-9890 Gavere • Belgium

Principes de base de l'oscilloscope

<http://www.selectronic.fr/media/pdf/410513.pdf>

[3]- Agilent Technologies, Inc. 2008 Mode d'emploi simplifié de l'oscilloscope numérique

Imprimé en Malaisie 05/08 <https://www.agilent.com/>

[4]- christian.dupaty@ac-aix-marseille.fr Le traitement de signal numérique sur  $\mu$ -contrôleur

Lycée Fourcade - Académie d'Aix-Marseille Traitement de signal.doc

[5]- BIGONOFF, La programmation des PICS, La gamme MID-RANGE par l'étude des 16F87X (16F876-16F877).

[6]- [https://euler.ac-versailles.fr/isn/Communication\\_RS232.pdf](https://euler.ac-versailles.fr/isn/Communication_RS232.pdf) communication série RS-232

[7]- Lycée professionnel Jacques Prévert combes-la-ville. Transmission des informations numériques. Section de BAC PRO Électronique industrielle Embarquée

<http://www.calameo.com/books/00003919361e9853fe399>

[8]-Jean-Louis Boimond. VISUAL BASIC 6

DESS IAIE, ISTIA, Université d'Angers 2003/2004, 30h

[9]- A. Belaïd Université de Nancy 2 Cours VB LSC 2003-2004

Développement d'un programme sous VB [http://tebraoui.canalblog.com/docs/cours\\_VB.pdf](http://tebraoui.canalblog.com/docs/cours_VB.pdf)

[10]- Christian Tavernier, Programmation en C des PIC. Edition DUNOD, 2005

[11] Pic 16F87X DATA SHEET, microchip Technologies Inc.2001.

[12] DATA SHEET MAX232, MAX232I DUAL EIA 232 DRIVERS/RECEIVERS

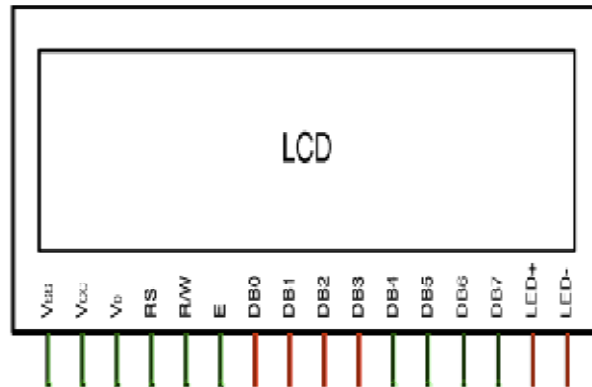
Texas Instruments, Post Office Box 655303, Dallas, [www.ti.com](http://www.ti.com)



# ANNEXES

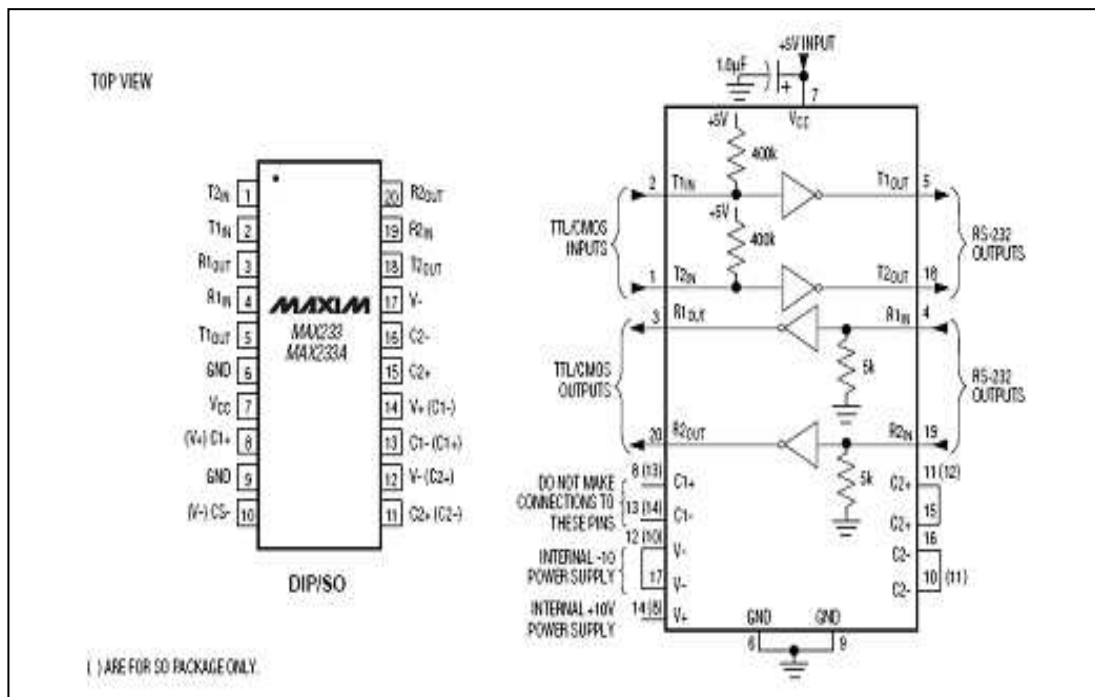
## ANNEXES

### 1. Brochage de LCD (LM016L) :



### 2. Max 232 :

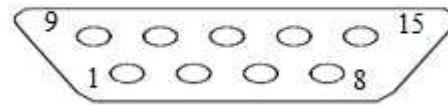
#### 2.1 Schéma simplifié de max232 : [12]



# ANNEXES

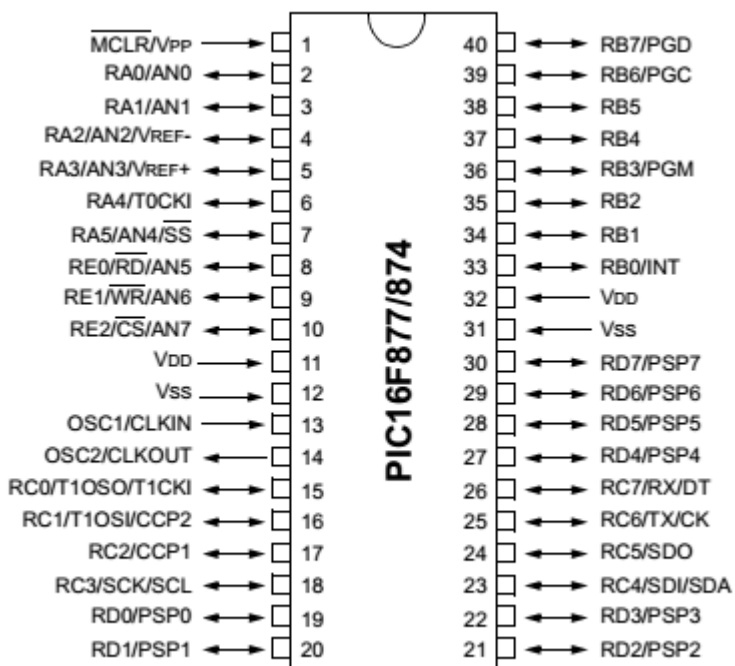
## 2.2 Connector P2 (serial I/O Pin out) [12]

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



## 3. Brochage de 16F877 (PLCC Pin out) :

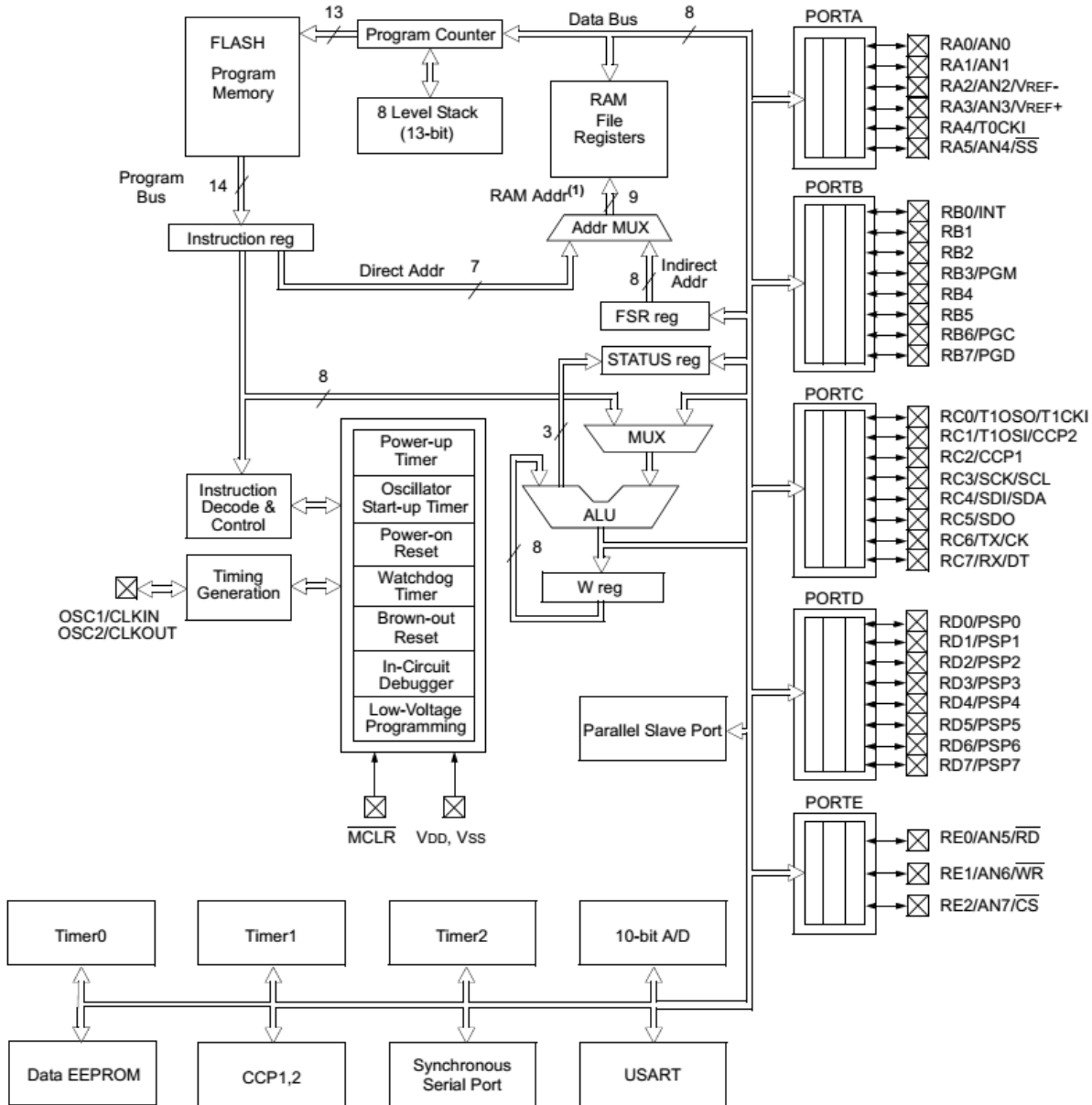
### Pin Diagram



# ANNEXES

FIGURE 1: PIC16F874 AND PIC16F877 BLOCK DIAGRAM [11]

| Device    | Program FLASH | Data Memory | Data EEPROM |
|-----------|---------------|-------------|-------------|
| PIC16F874 | 4K            | 192 Bytes   | 128 Bytes   |
| PIC16F877 | 8K            | 368 Bytes   | 256 Bytes   |



**Note 1:** Higher order bits are from the STATUS register.

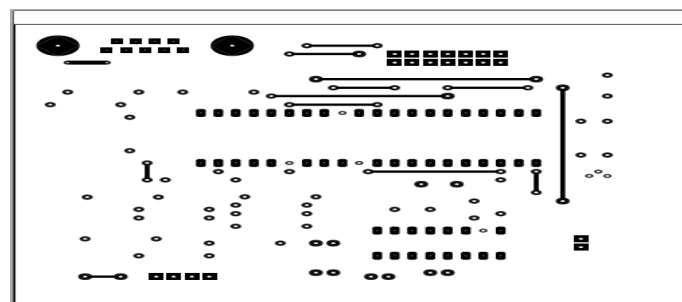
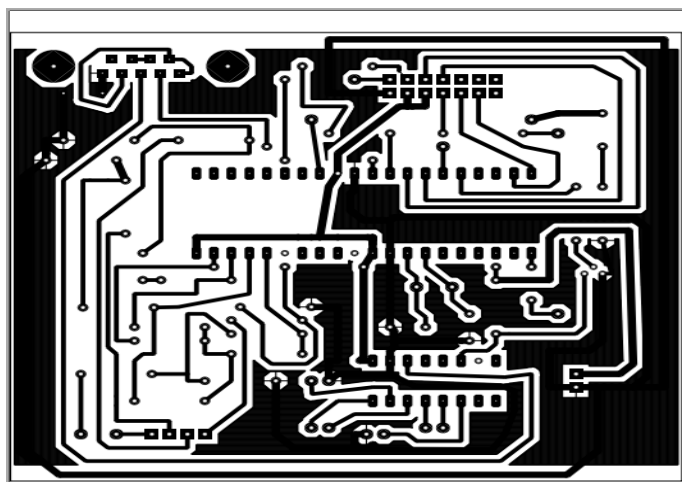
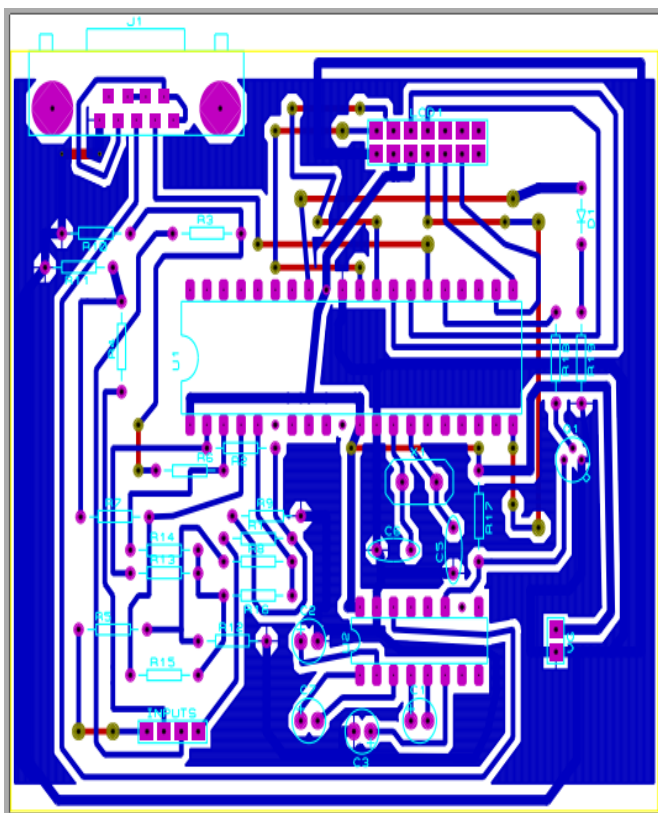
# ANNEXES

## PIC16F87X

| Key Features<br>PICmicro™ Mid-Range Reference<br>Manual (DS33023) | PIC16F873               | PIC16F874               | PIC16F876               | PIC16F877               |
|---|-------------------------|-------------------------|-------------------------|-------------------------|
| Operating Frequency   | DC - 20 MHz             | DC - 20 MHz             | DC - 20 MHz             | DC - 20 MHz             |
| RESETS (and Delays)   | POR, BOR<br>(PWRT, OST) | POR, BOR<br>(PWRT, OST) | POR, BOR<br>(PWRT, OST) | POR, BOR<br>(PWRT, OST) |
| FLASH Program Memory<br>(14-bit words)                            | 4K                      | 4K                      | 8K                      | 8K                      |
| Data Memory (bytes)   | 192                     | 192                     | 368                     | 368                     |
| EEPROM Data Memory  | 128                     | 128                     | 256                     | 256                     |
| Interrupts  | 13                      | 14                      | 13                      | 14                      |
| I/O Ports   | Ports A,B,C             | Ports A,B,C,D,E         | Ports A,B,C             | Ports A,B,C,D,E         |
| Timers  | 3                       | 3                       | 3                       | 3                       |
| Capture/Compare/PWM Modules                                       | 2                       | 2                       | 2                       | 2                       |
| Serial Communications   | MSSP, USART             | MSSP, USART             | MSSP, USART             | MSSP, USART             |
| Parallel Communications   | —                       | PSP                     | —                       | PSP                     |
| 10-bit Analog-to-Digital Module                                   | 5 input channels        | 8 input channels        | 5 input channels        | 8 input channels        |
| Instruction Set   | 35 instructions         | 35 instructions         | 35 instructions         | 35 instructions         |

### 3-La carte réalisée

- Circuit sous ARES



Circuit imprimé