

## SOMMAIRE

### REMERCIEMENTS

Liste des tableaux

Liste des figures

Liste des abréviations

INTRODUCTION.....	1
CHAPITRE I : LE RESISTIVIMETRE.....	2
1- La résistivité :.....	2
2- La conductibilité : .....	4
3- La loi d'Ohm :.....	4
4- Principe :.....	5
5- Dispositifs et acquisition : .....	7
CHAPITRE II : LE GENERATEUR DE COURANT.....	8
1. L'ALIMENTATION.....	8
2. REGULATION DU TENSION DE LA BATTERIE .....	10
3. LE GENERATEUR DE TENSION.....	10
II.4. LE GENERATEUR DE COURANT .....	16
CHAPITRE III : L'INTERFACE .....	22
III.1 STRUCTURE INTERNE D'UN MICROCONTROLEUR: .....	22
III.2 PRESENTATION DU PIC 16F877 :.....	24
III.3 STRUCTURE INTERNE DU 16F877.....	28
1)La mémoire du 16F877 .....	28
2) Les modules internes du 16F877 .....	32
3) Présentation de la liaison RS232 et du MAX 232 .....	45
III.4 PIC vers PC.....	47
III.5 LA PROGRAMMATION DU PIC :.....	51
CHAPITRE IV : RESULTATS ET DISCUSSIONS.....	62
RESULTAT :.....	63
DISCUSIONS :.....	66
CONCLUSION :.....	67

ANNEXES

BIBLIOGRAPHIE

**Liste des tableaux**

Tableau 1 : Porosité, perméabilité et résistivité de différentes roches .....	3
Tableau 2 : Valeurs des résistances Re et Rc max .....	18
Tableau 3 : Registre EECON1 .....	29
Tableau 4 : Les différentes pages de la mémoire .....	30
Tableau 5 : Registre ADCON0 .....	37
Tableau 6 : Vitesse de conversion .....	37
Tableau 7 : Temps de conversion TAD en fonction du Quartz et des bits du Clock select .....	38
Tableau 8 : Canal de sélection .....	38
Tableau 9 : Registre ADCON1 .....	39
Tableau 10 : Justification à gauche .....	39
Tableau 11: Justification à droite .....	39
Tableau 12 : Configuration des ports .....	40
Tableau 13: Registre TXSTA .....	42
Tableau 14 : Etat du bit BRGH .....	43
Tableau 15 : Facteur de division N avec un Quartz à 8MHz .....	44
Tableau 16: Registre RCSTA .....	61
Tableau 17 : Valeurs des résistances indiquées et mesurées .....	64
Tableau 18 : Modèle obtenu .....	65

## Liste des figures

Figure 1 : Résistivité d'un cylindre .....	3
Figure 2 : Equipotentiels et filets de courant pour une source unique.....	4
Figure 3 : Equipotentiels et filets de courant pour 2 sources.....	5
Figure 4 : Disposition des électrodes .....	6
Figure 5 : Montage avec LM338T .....	8
Figure 6 : Stabilisation avec ballast.....	10
Figure 7 : L'oscillateur du convertisseur .....	12
Figure 8 : Deux variantes envisageables pour commander les deux transistors.....	13
Figure 9 : Schéma du convertisseur .....	14
Figure 10 : Signaux obtenus par simulation de l'oscillateur .....	15
Figure 11 : Schéma de base du générateur de courant .....	16
Figure 12 : Zone de fonctionnement du générateur de courant .....	17
Figure 13 : Simulation du générateur de courant.....	19
Figure 14 : Typon du générateur de courant.....	20
Figure 15 : Carte du générateur de courant.....	21
Figure 16 : Les principaux blocs fonctionnels présents à l'intérieur d'un PIC.....	23
Figure 20 : Brochage du 16F877.....	25
Figure 21 : Brochage de l'oscillateur à quartz.....	26
Figure 22 : Les éléments constitutifs du 16F877 .....	28
Figure 23 : Algorithme du CAN à approximation successive .....	35
Figure 24 : Structure du CAN.....	37
Figure 25 : Processus de conversion du CAN.....	37
Figure 26 : Utilisation du MAX232 .....	45
Figure 27 : Le MAX 232 Brochage du DB9.....	47
Figure 28 : Le MAX232 .....	48
Figure 29: Brochage du MAX 232 avec le PIC et le DB9 .....	49
Figure 30 : Typon de l'interface.....	55
Figure 31: Carte de l'interface .....	56
Figure 32: Schéma et montage du programmeur. ....	56
Figure 33 : Interface de l'ICPROG .....	58
Figure 34 : Configuration du Programmeur.....	59

Figure 35 : Choix du composant .....	56
Figure 36: Transfère du fichier .....	58
Figure 37 : Schéma synoptique du résistivimètre .....	59
Figure 38 : Organigramme pour la mesure .....	60
Figure 39 : Synoptique de l'acquisition pour la mesure d'une résistance .....	65
Figure 40 : Interface logiciel – Mesure de la Résistance.....	65
Figure 41 : Interface logiciel – Mesure de la Résistivité .....	66

## Liste des abréviations

<b>ADCON0</b>	Analog to Digital Control <b>0</b>
<b>ADCON1</b>	Analog to Digital Control <b>1</b>
<b>ADRESL</b>	Analog to Digital Result Low
<b>ADRESH</b>	Analog to Digital Result High
<b>C.A.N</b>	Convertisseur Analogique Numérique
<b>CHS</b>	Channel Select Bit
<b>C.P.U</b>	Computer Personal Unit
<b>CISC</b>	Complex Instruction Set Computer
<b>D.I.L</b>	Dual In Line
<b>E/S</b>	Entrée/Sortie
<b>EEPROM</b>	Electrical Erasable Programmable Read Only Memory
<b>EIA</b>	Electronic Industry Associate
<b>EPROM</b>	Erasable Programmable Read Only Memory
<b>Fosc</b>	fréquence de l'oscillateur
<b>HS</b>	High speed crystal/Resonator
<b>I.C.S.P</b>	In Circuit Serial Programming
<b>I.O.G.A</b>	Institut et Observatoire de Géophysique d'Antananarivo
<b>I/O</b>	Input/ Output
<b>IC-Prog</b>	In-Circuit Programming Guide
<b>ICSP</b>	In Circuit Serial Programming
<b>L.S.B</b>	Lost Significant Bit
<b>LP</b>	Low power
<b>M.I.P.S</b>	Million d'Instructions Par Seconde
<b>M.S.B</b>	Most Significant Bit
<b>MHz</b>	Mega hertz
<b>MOS</b>	Metal Oxyd Semiconductor
<b>MUX</b>	Multiplexeur
<b>mV</b>	Millivolt
<b>P.C</b>	Personal Computer
<b>P.I.C</b>	Programmable Interface Controller
<b>P.W.M</b>	Pulse Width Modulation
<b>PROM</b>	Programmable Read Only Memory
<b>RAM</b>	Random Access Memory
<b>RC</b>	Resistor/Capacitor
<b>RISC</b>	Reduced Instruction Set Computer
<b>SAR</b>	Successive Approximation Register
<b>SCI</b>	Serial Communication Interface
<b>TTL</b>	Transistor Transistor Logique
<b>USART</b>	Universal Synchronous Asynchronous Receiver Transmitter
<b>V</b>	Volt
<b>WDT</b>	Watch Dog
<b>XT</b>	Crystal/Resonator
<b>Ω</b>	Ohm
<b>Ω m</b>	Ohm mètre

## INTRODUCTION

Le Laboratoire d'Instrumentation Géophysique de l'Institut et Observatoire Géophysique d'Antananarivo est opérationnel depuis Janvier 2003. Sa mission est de comprendre et d'analyser les blocs fonctionnels des matériels utilisés en prospection géophysique. Dès lors, nous avons conçu un Résistivimètre commandé par un ordinateur en utilisant une communication série et la nouvelle technologie « Microcontrôleur » qui constitue le cerveau du résistivimètre.

Le résistivimètre introduit un courant continu dans le sol par l'intermédiaire de deux électrodes « d'injection » A et B et mesure la différence de potentiel entre deux autres électrodes « de potentiel » M et N. Un générateur de courant est utilisé pour produire un courant, (0.2mA à 500mA), avec une tension stabilisée 300V à 50Hz à partir d'une batterie 12V.

Le microcontrôleur PIC 16F877 gère en même temps l'injection du courant, la mesure de la tension, la conversion et le transfert des données vers le PC grâce à ses différentes modules internes : mémoire programme, mémoire des données, convertisseur analogique numérique, et module USART pour le transfert. Toutes les commandes : mesure de la Polarisation Spontanée, Injection de courant, seront faites à partir d'un ordinateur sur lequel a été installée une interface logicielle réalisée avec Borland C++. Ce rapport décrit une partie de base de cette réalisation, à noter que ce travail est réalisé en étroite collaboration avec ANDRIAMBELOSON Joely, qui se chargera la partie simulation et modélisation.

Le premier chapitre relate le principe de fonctionnement d'un résistivimètre électrique. Le second chapitre sera consacré au générateur de courant et à sa réalisation. Le troisième chapitre décrira le rôle du microcontrôleur dans l'interfaçage de notre appareil. Enfin, au dernier chapitre, nous présenterons les résultats obtenus et les discussions.

## CHAPITRE I : LE RESISTIVIMETRE

Le courant électrique circule plus ou moins facilement dans les sols et des roches en fonction de :

- la nature lithologique du terrain (l'argile par exemple facilite le passage du courant électrique),
- la quantité d'eau qu'il renferme
- la salinité de cette eau (une roche contenant une grande quantité d'eau salée est très conductrice de courant).

Une investigation géophysique électrique consiste à injecter dans le sol un courant électrique d'intensité connue et de mesurer la différence de potentiel induite entre deux points donnés.

### 1- LA RESISTIVITE :

La résistivité  $\rho$  d'un milieu est la propriété physique qui détermine la capacité de ce milieu à laisser passer le courant électrique. La résistivité d'un terrain s'exprime en  $\Omega \cdot m$ . Elle correspond à la résistance théorique en Ohm d'un cylindre de terrain de  $1 \text{ m}^2$  de section et de 1 m de longueur.

$$R = \rho \frac{L}{S}$$

Avec :  $R$  = résistance ( $\Omega$ )

$\rho$  = résistivité ( $\Omega \cdot m$ )

$L$  = longueur (m)

$S$  = section ( $m^2$ )

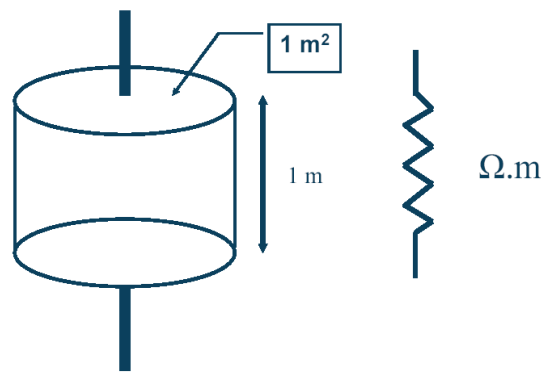


Figure 1 : Résistivité d'un cylindre

La résistivité est donc un élément très variable suivant les formations sous-jacentes. Le tableau suivant donne des valeurs typiques en fonction de la nature du terrain. Mais ces valeurs sont très indicatives et seule une mesure sur site permet de déterminer avec précision la valeur de la résistivité.

Roches	Porosité totale (%)	Perméabilité (cm/s)	Résistivité (ohms*m)
Argilite	35	$10^{-8} - 10^{-9}$	70 à 200
Craie	35	$10^{-5}$	30 à 300
Tuf volcanique	32	$10^{-5}$	20 à 300
Marnes	27	$10^{-7} - 10^{-9}$	20 à 100
Grès	3 à 35	$10^{-3} - 10^{-6}$	30 à 800
Dolomite	1 à 12	$10^{-5} - 10^{-7}$	200 à 10'000
Calcaires	3	$10^{-10} - 10^{-12}$	200 à 10'000
Métaschistes	2.5	$10^{-4} - 10^{-9}$	300 à 800
Gneiss	1.5	$10^{-8}$	1'000 à 20'000
Quartzite	<1	$10^{-10}$	1'000 à 10'000
Granite	1	$10^{-9} - 10^{-10}$	1'000 à 15'000
Gabbro	1 à 3	$10^{-4} - 10^{-9}$	6'000 à 10'000
Basalte	1.5	$10^{-6} - 10^{-8}$	800 à 15'000

Tableau 1 : Porosité, perméabilité et résistivité de différentes roches



## 2- LA CONDUCTIBILITE :

La conductibilité  $\sigma$  est l'inverse de la résistivité et s'exprime en  $\Omega/m$ .

$$\sigma = \frac{1}{\rho}$$

En hydrogéologie on emploie le plus souvent le  $\mu\Omega/cm$  ou  $\mu\text{siemens/cm}$ .  $\rho$  ( $\Omega.m$ ) = **10.000 /**  
 $\sigma$  ( $\mu\text{siemens/cm}$ ) = **1000 /**  $\sigma$  (**millisiemens/m**)

## 3- LA LOID'OHM :

Pour un terrain homogène et isotrope de résistivité  $\rho$  limité par une surface plane du côté de l'air, l'écoulement du courant  $I$  se fera par filets rectilignes rayonnant autour de l'électrode A et produira des variations de potentiel dans le sol à cause de la résistance ohmique de celui-ci. La répartition du potentiel peut être représentée par des demi-sphères centrées sur A

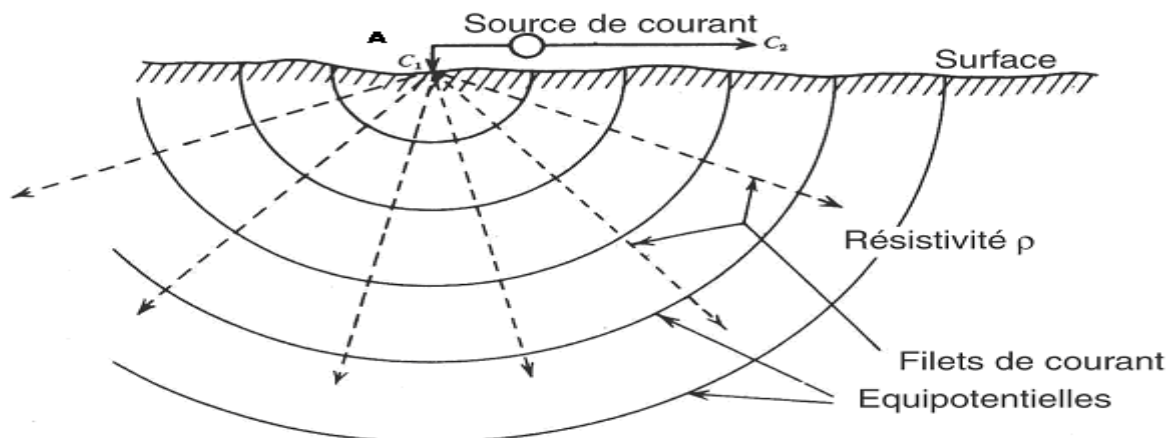


Figure 2 : Equipotentielles et filets de courant pour une source unique  
 (Terrain homogène et isotrope)

La figure 2 montre que dans un sol homogène et isotrope, les surfaces équipotentielles seront des demi-sphères centrées sur ce pôle. Pour deux pôles (figure 3), leur forme sera plus compliquée mais elles resteront sensiblement sphériques au voisinage des prises A et B. De même, les lignes de courant qui seraient des droites issues du pôle d'envoi si celui-ci est unique s'incurvent progressivement pour rejoindre la seconde prise.

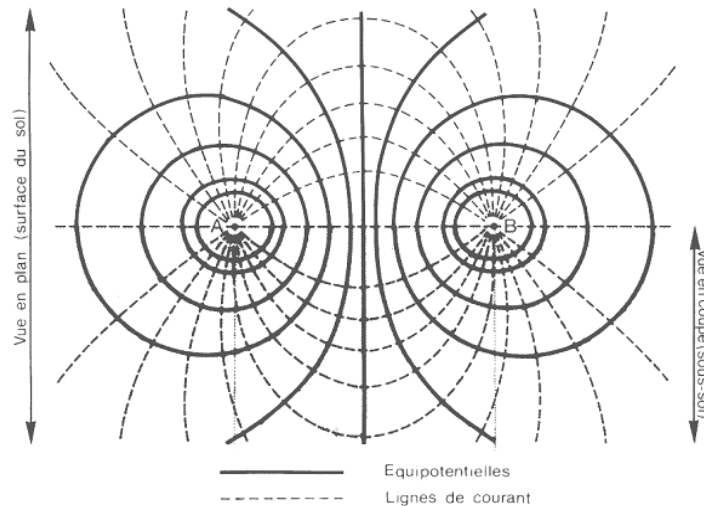


Figure 3 : Equipotentiels et filets de courant pour 2 sources

#### **4- PRINCIPE :**

La mesure de la résistivité d'une structures s'effectue en injectant un courant électrique dans celle-ci au moyen d'électrodes d'injection (souvent nommées A et B) et en mesurant la différence de potentiel créée par le passage du courant au moyen d'électrodes de mesure du potentiel (souvent nommées M et N). Le nombre d'électrodes, ainsi que leur agencement définit le dispositif électrique utilisé. La configuration du dispositif est choisie selon la problématique de l'étude. La différence de potentiel dépend de l'intensité du courant injecté, de la disposition des électrodes et de la résistivité électrique du matériau constituant l'objet étudié. (figure 4)

Il est alors possible de déduire une répartition de la résistivité vraie du matériau, ce qui n'est pas le cas pour des milieux hétérogènes. La notion d'homogénéité dépend toutefois de l'échelle à laquelle le milieu est observé. Nous faisons appel, dans le cas d'un milieu hétérogène, au concept de résistivité apparente. Le terme de résistivité apparente est relativement impropre. Le fait que cette grandeur ait pour dimension celle de la résistivité provient du choix de l'unité pour la résistivité du milieu homogène, choix parfaitement conventionnel.

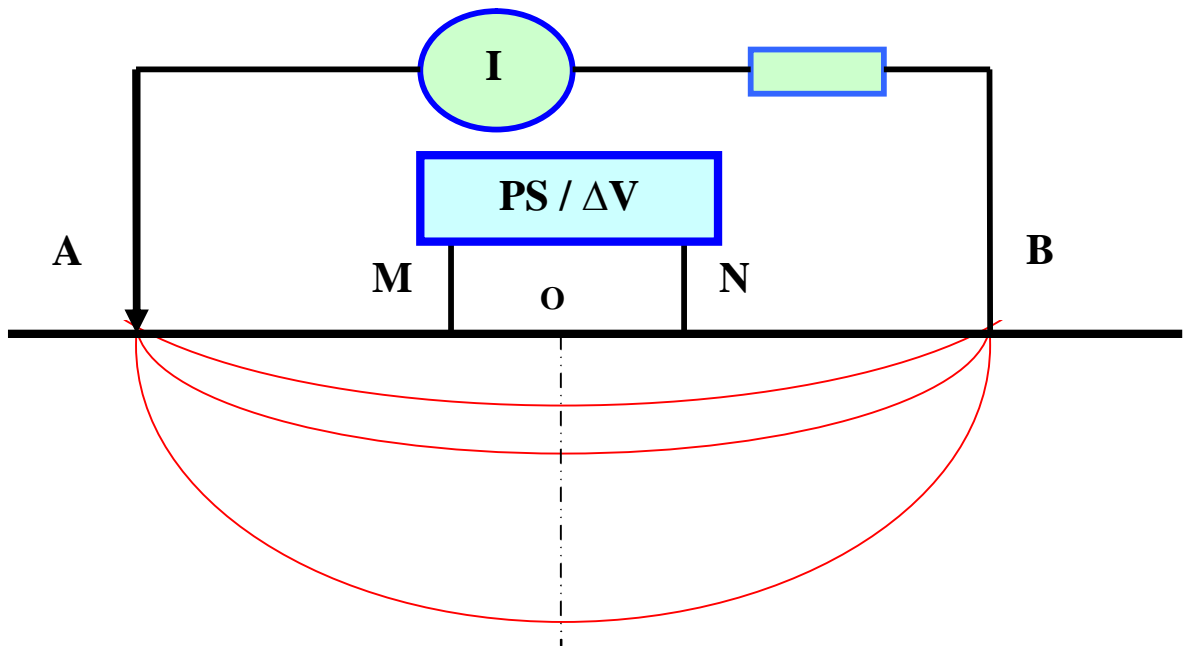


Figure 4 : Disposition des électrodes

$$\rho_{app} = k \frac{\Delta V}{I}$$

$\rho_{app}$  = résistivité apparente [ $\Omega \cdot m$ ]

$k$  = facteur géométrique [m]

$I$  = intensité du courant [mA]

$\Delta V$  = différence de potentielle [mV]

Le paramètre k est appelé le facteur géométrique; il dépend uniquement de la position des électrodes sur le terrain et s'exprime par la relation :

$$K = \frac{2\pi}{\frac{1}{AM} - \frac{1}{AN} - \frac{1}{BM} + \frac{1}{BN}}$$

Afin de sonder une gamme de profondeurs, les mesures sont faites en changeant l'espacement entre les électrodes afin que le courant électrique puisse circuler dans les couches géologiques plus ou moins profondes. Le courant électrique injecté est typiquement de 100 mA et les potentiels mesurés varient de quelques Volts à quelques mV selon que les électrodes M et N sont proches ou éloignées des électrodes d'injection A et B et également selon la conductivité électrique des roches.

## **5- DISPOSITIFS ET ACQUISITION :**

Avant chaque injection du courant, il est important de mesurer la Polarisation Spontanée (PS) du terrain due à l'activité électrochimique ou mécanique; sa valeur est de quelques millivolts ; elle sera prise en compte lors de la mesure de la différence de potentiel ( $\Delta V$ ). En surface, les mesures peuvent être effectuées en gardant le centre du dispositif fixe et en écartant les électrodes d'injection (sondage électrique) ou en déplaçant un dispositif à écartement constant (traîné électrique). Avec la première méthode nous obtenons la variation de la résistivité en 1D (une dimension) sous le dispositif et avec la seconde méthode nous étudions les variations latérales de ce paramètre. En combinant ces deux techniques, il est possible de réaliser des panneaux électriques, sensibles aux variations tant verticales qu'horizontales de la sub-surface.

Ces mesures sont habituellement représentées sous la forme de panneaux électriques (ou pseudo-sections en résistivités apparentes). Les mesures sont particulièrement sensibles aux variations de la résistivité sous le dispositif. Les points de mesure sont reportés à l'aplomb du centre du dispositif et à une ordonnée proportionnelle à la distance séparant les électrodes (AM/2 ou AB/2 par exemple) définissant des niveaux d'acquisition. Les valeurs sont ensuite interpolées pour tracer les lignes d'isorésistivité.

Les électrodes sont de simples broches en acier inoxydable d'environ 30 cm de long et 1 cm de diamètre. Des tensions de plusieurs centaines de Volts et des courants pouvant atteindre 500 mA sont utilisés, ce qui nous amène à réaliser un convertisseur continu - continu 12V / 300V et capable de fournir un courant constant, indépendamment de la résistivité du sous sol.

## CHAPITRE II : LE GENERATEUR DE COURANT

Pour générer le courant dans le sol, un convertisseur alimenté par une batterie externe 12V est nécessaire. Ce convertisseur transformera le 12V de la batterie à une valeur plus importante, de l'ordre de 300V. La tension de la batterie ne doit pas être en dessous de 10V. Les courants disponibles sont : 0.2mA, 1mA, 10mA, 100mA, 200mA et 500mA.

### II.1. L'ALIMENTATION

Deux fonctions principales sont toujours demandées à un circuit d'alimentation :

- Délivrer une ou plusieurs tensions (ou courants) bien précises et souvent stables.
- Fournir une énergie électrique avec un minimum de pertes.

Il peut être demandé en plus des fonctions de régulation ou de protection par exemple. Il existe 2 types d'alimentation:

- courant constant : courant stable quelque soit la charge
- tension constante : tension stable quelque soit la charge

- **Régulation avec un circuit spécialisé, le LM338T**

L'utilisation d'un régulateur intégré est nécessaire pour obtenir la tension voulue ; le LM338T est un régulateur intégré dont la sortie est ajustable entre 1,25 volts et 34 volts avec un courant maximal de 5 ampères. Il peut supporter une tension d'entrée maximum 37V. Le régulateur LM338T est protégé en interne contre les surcharges de courant et la surchauffe. Un refroidisseur est obligatoire pour notre application. *La figure 5* nous montre une application de base de ce régulateur.

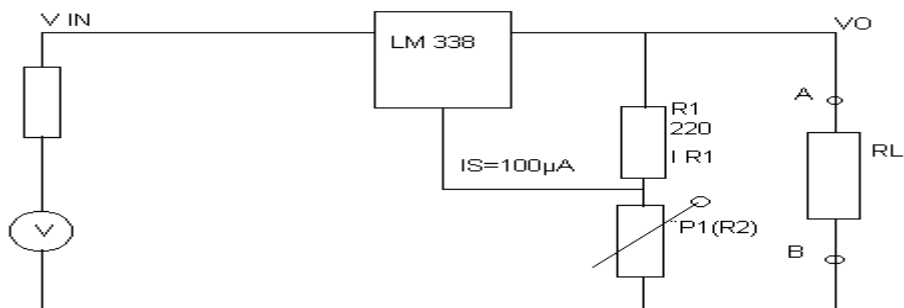


Figure 5 : montage avec LM338T

Les valeurs de R1 et R2 ne sont pas critiques mais R1 ne devra pas excéder 300 ohms. Pour une tension de sortie fixe, R2 sera sélectionnée pour donner avec R1 une valeur fixe. Notre montage a besoin de deux tensions précises et stables (quelques soit la charge) : 12V et 5V.

*Remarque :* Le régulateur assure une tension précise et stable tant que la différence de tension entre la tension d'entrée et la tension de sortie vaut 3V. Donc pour avoir une tension de 12V, il nous faut une tension d'entrée minimum à 15V ; ceci est résolu en mettant en série une autre batterie de 12V avec la batterie externe qu'on va utiliser sur le terrain.

La formule générale permettant de calculer la tension de sortie est donnée ci-après:

$$V_0 = 1.25(1 + \frac{R_2}{R_1})$$

$V_0$  est la tension de sortie. Elle ne dépend pas de la tension d'entrée. Le calcul de R2 pour  $V_0 = 5 \text{ V}$  et  $R_1 = 220 \Omega$  donne :

$$R_2 = [(5/1.25)-1]*220 = 660 \Omega$$

Pour  $V_0=12\text{V}$  :

$$R_2 = [(12/1.25)-1]*220 = 1892 \Omega$$

Remplaçons R2 par un potentiomètre noté P1 sur la *figure 5*, de valeur équivalente. Nous obtenons une tension de sortie ajustable, avec les restrictions suivantes :

Tension minimum en sortie = 1,25 V.

Tension maximum en sortie, suivant la formule, et au maximum  $V_{in} - 3 \text{ V}$

( $V_{in}$  est la tension d'entrée)

## II.2. REGULATION DU TENSION DE LA BATTERIE :

Au fur et à mesure de l'utilisation de la batterie externe, sa tension va diminuer, donc il est nécessaire de stabiliser la tension délivrée par la batterie en faisant attention au courant de sortie. La puissance est assurée par un transistor bipolaire NPN (2N6258) qui est monté sur un dissipateur en aluminium de façon à évacuer la chaleur. Ce transistor est piloté par l'unité de stabilisation ou régulateur de tension dont la théorie est expliquée auparavant. Ce régulateur est incapable de réguler des courants de quelques ampères voire 20 ou 25 ampères, d'où l'utilisation du transistor ballast. La figure 6 suivante nous montre le schéma de ce montage.

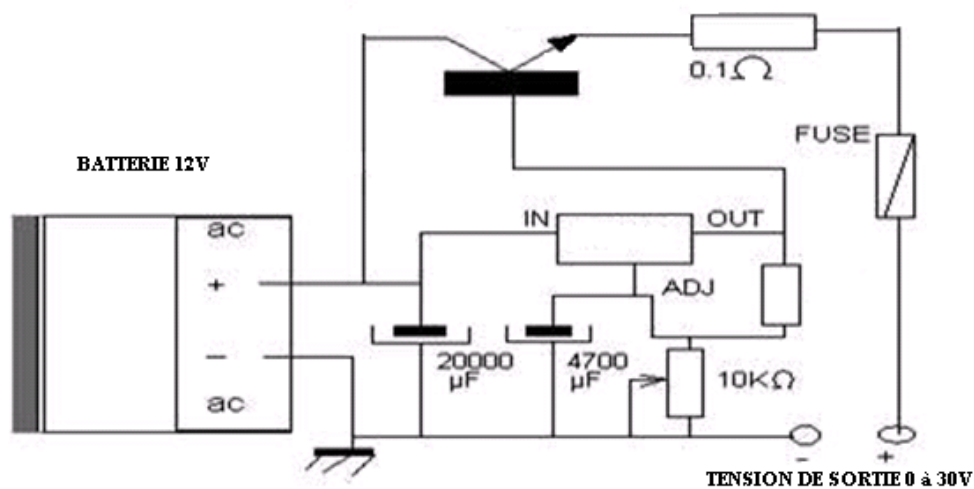


Figure 6 : stabilisation avec ballast

## II.3. LE GENERATEUR DE TENSION :

Pour obtenir une tension de 300V précise et stable à partir d'une tension de la batterie externe, nous avons réalisé un convertisseur piloté par un quartz qui est capable de nous fournir la tension voulue et peut débiter les courants nécessaires à fréquence fixe.

### **1. Les avantages du convertisseur pilote :**

Dans un convertisseur dit « piloté », les transistors de puissance sont commandés par une logique fonctionnant à une fréquence quasiment stable, pilotée par quartz si nécessaire. Un convertisseur piloté fonctionnera toujours à la même fréquence, que sa charge soit résistive, inductive, ou capacitive, même si une charge trop réactive compromet sérieusement son rendement. Ce type de convertisseur peut être construit à partir de transformateurs classiques à deux enroulements ou à point milieu. Enfin, on peut modeler la forme d'onde dans de large proportions, quitte à s'éloigner assez sensiblement du simple signal rectangulaire : tout entravaillant exclusivement en commutation, on peut arriver à approcher d'assez près cet idéal que représente la sinusoïde, notamment en ce qui concerne la tension crête.

### **2. Principe et réalisation :**

Le principal problème qui se pose lors de l'étude d'un convertisseur piloté d'une certaine puissance, est l'amplification en courant des signaux de l'oscillateur pilote : de quelques milliampères (logique CMOS ou TTL) ou au mieux de quelques centaines de milliampères (pour le CI 555), il faut passer à plusieurs dizaines d'ampères tout en commandant deux transistors en opposition de phase, puisqu'il faut absolument appliquer au transformateur un courant primaire de valeur moyenne nulle.

### **3. Choix des transistors de puissance :**

Avec des gains pas toujours bien glorieux à courant élevé pour les transistors courants, il faut recourir à des darlington à plusieurs niveaux : l'addition de tous les courants de base mène souvent à un chiffre non négligeable, qui pèse sur le rendement de l'appareil. Une solution intéressante consiste à faire appel à des transistors MOS de puissance, qui se commandent en tension (autrement dit dont l'électrode de commande ne consomme pas de courant en régime statique).

Cette technique permet de commuter directement un courant de plusieurs dizaines d'ampères à partir de niveaux logiques CMOS. Il faut cependant s'accommoder de la résistance à l'état passant de ces transistors, qui introduit une chute de tension purement ohmique, et donc proportionnelle au courant débité. Par comparaison, un transistor bipolaire saturé n'introduit qu'une chute de tension de l'ordre du volt, pratiquement indépendante du courant débité.



Cette chute liée au courant consommé va dans le sens d'une protection du convertisseur contre les surcharges, d'autant plus qu'elle augmente avec la température (exactement l'inverse d'un « emballement thermique »).

Les MOS de puissance se montent fort bien en parallèle, tandis qu'il en existe dont la résistance à l'état passant est très faible : selon la puissance souhaitée (en pratique celle du transformateur), on choisira donc le type et le nombre des transistors nécessaires en fonction des disponibilités ou des opportunités, sans que cela ne remette en cause si peu que ce soit le reste du montage.

**4. L'oscillateur :**

Le montage utilise un oscillateur à quartz qui garantit l'exactitude de la fréquence de 50Hz sans le moindre réglage. Cette valeur sera obtenue avec un cristal de 3,2768 MHz, valeur standard, tandis que le modèle encore plus courant de 3,58MHz mènerait à une fréquence supérieure d'environ 10%, 54.5MHz.

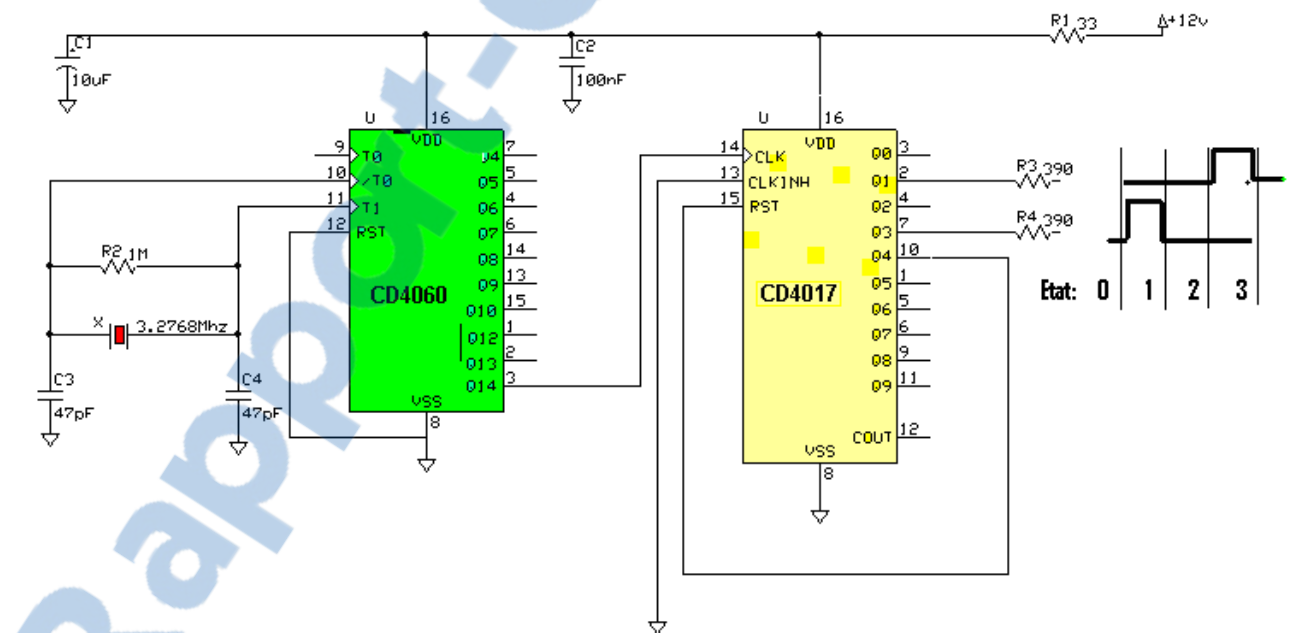
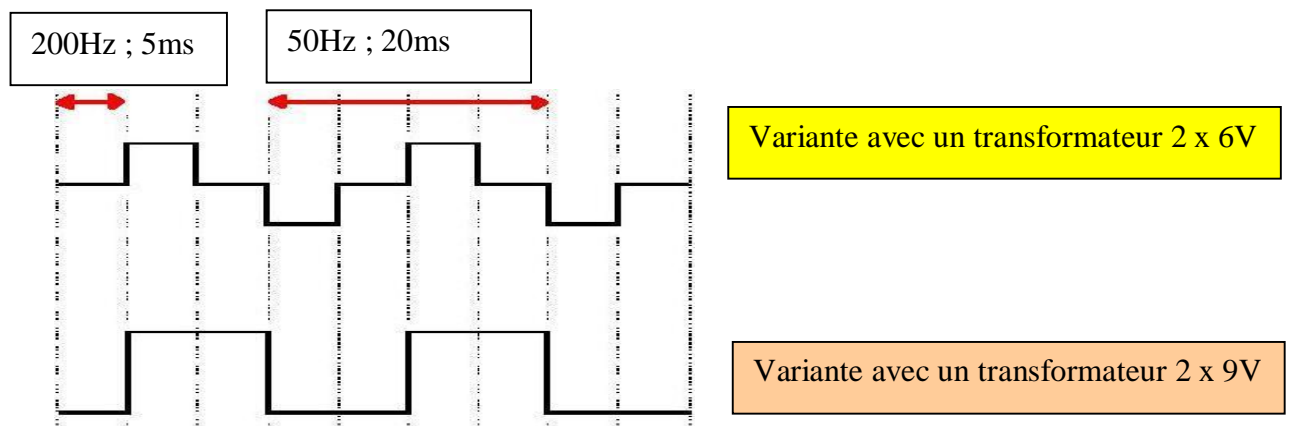


Figure 7 : l'oscillateur du convertisseur

Le circuit CMOS 4060 combine l'oscillateur proprement dit en un diviseur fournissant une fréquence de 200Hz. Appliquée à un compteur décodeur 4017, dont la remise à zéro est effectuée sur le cinquième état, cette fréquence lui fait prendre quatre états successifs durant 5ms chacun : 0,1,2 et 3. Il en résulte une période globale de 20ms, qui correspond bien à une fréquence de 50Hz. La *figure 8* montre que deux variantes sont envisageables pour commander les deux transistors qui alimentent le primaire de transformateur :



*Figure 8 : variantes envisageables pour commander les deux transistors*

- La première consiste à faire conduire l'un des transistors pendant l'état n° 1 du compteur et le second pendant l'état n° 3, les deux transistors restant bloqués ensemble pendant les états n° 0 et 2.

Moyennant un transformateur muni d'enroulements 6V, on obtiendra en sortie une onde pseudo sinusoïdale dont la tension crête sera de 300V pour une valeur efficace de 212V.

- La seconde utilise une conduction des transistors pendant les états n° 0 et 3, et respectivement 1 et 2 : il en résulte une onde strictement rectangulaire de 212V crête, dont l'unique avantage est de permettre l'utilisation d'un transformateur à deux enroulements de 9V, si une telle pièce est déjà disponible.

Nous avons opté pour la première variante à cause de la disponibilité d'un transformateur récupéré dans un onduleur, et aussi d'en titrer une tension crête de 300V lors du redressement.

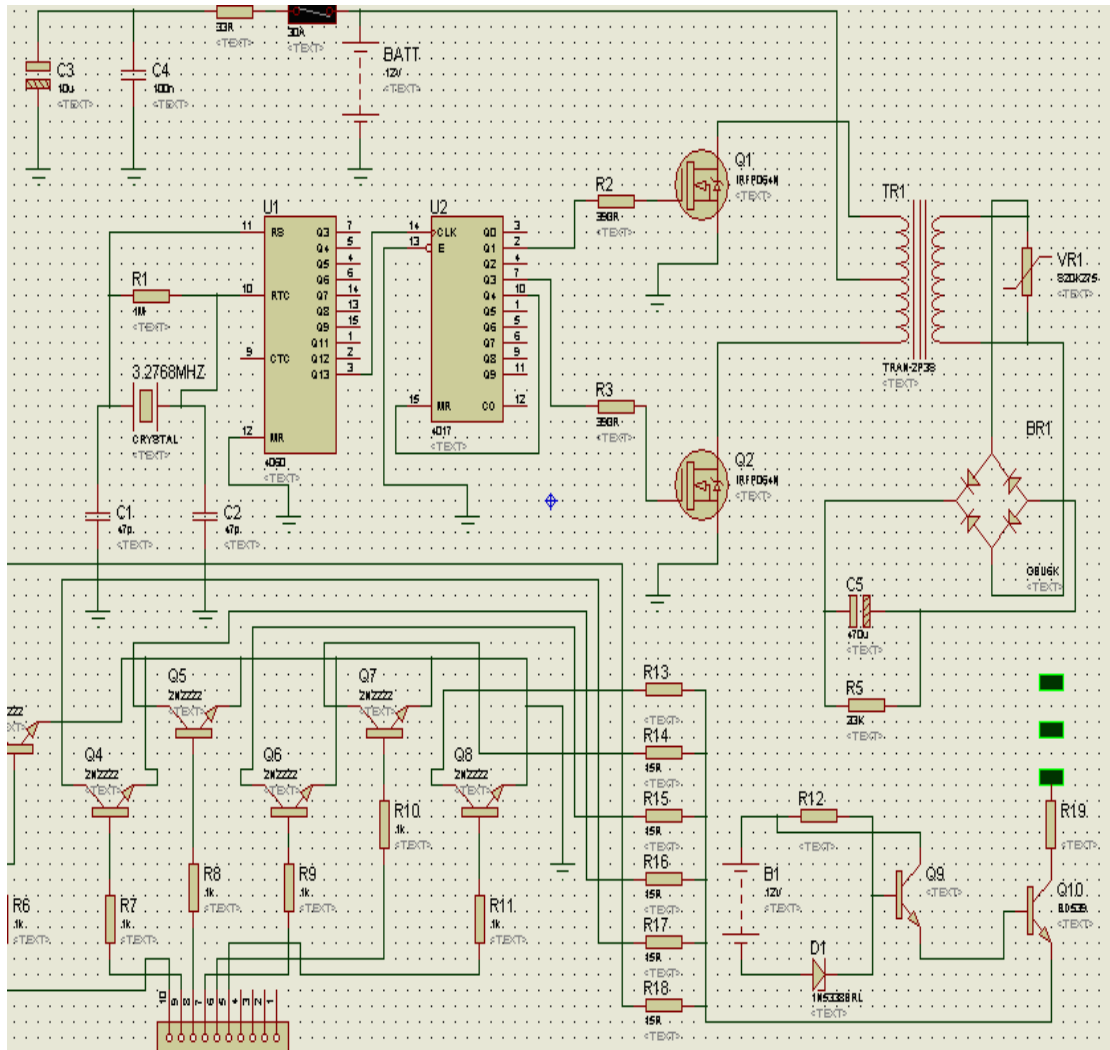
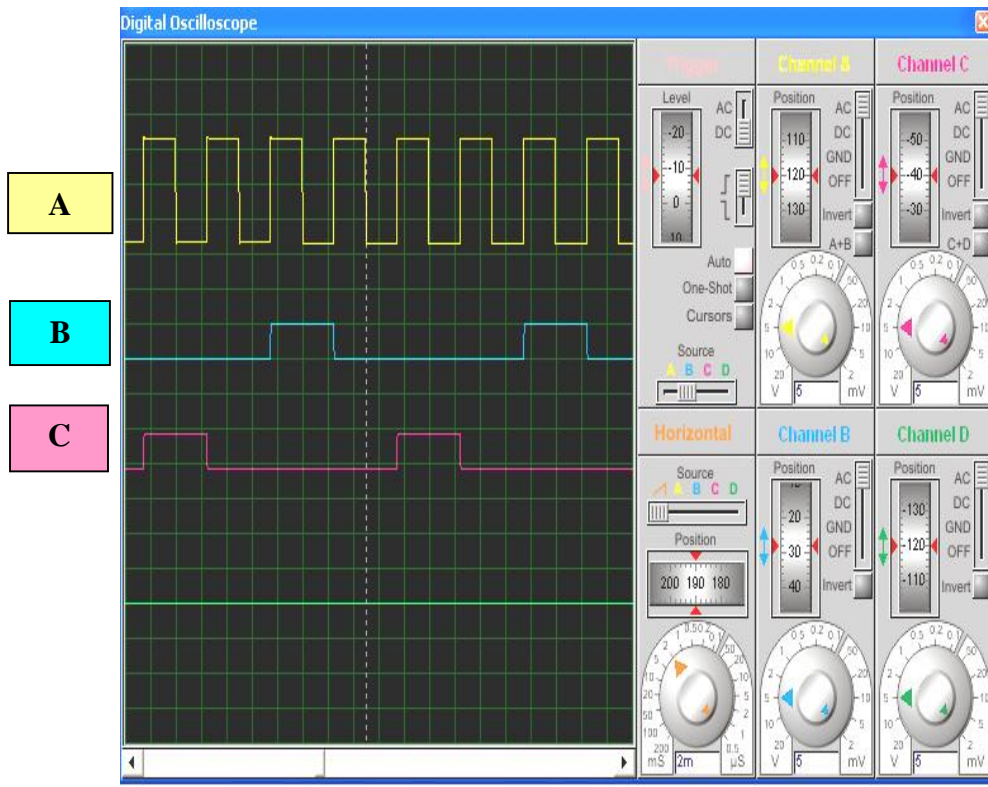


Figure 9 : schéma du convertisseur



**LEGENDES**

**Fréquence 200Hz à la sortie du CD4060**

**Fréquence 50Hz à la sortie du CD4017 pin**

**Fréquence 50Hz à la sortie du CD4017 pin**

*Figure 10 : Signaux obtenus par simulation de l'oscillateur*

*(Logiciel Proteus / version 7.1, Février 2007)*

### 5. Protections :

Chaque transistors est protégé par un transil de 30V, tandis q'une varistance de 275V élimine les pointes de tension pouvant subsister en sortie. Enfin, un fusible de 30A pour une puissance de 400VA est évidemment prévu dans l'alimentation batterie.

### 6. Redressement et filtrage :

A la sortie du transformateur, la tension sera filtrée à l'aide d'un module, pour éliminer les pics de tension, puis redressée à l'aide d'un pont de diode. Le montage délivre une tension redressée mais non continue. Pour obtenir une tension (quasi) continue, il suffit de mettre un gros condensateur en parallèle avec la charge. Un condensateur  $470\mu\text{F}$  avec une tension de service 450V fera bien l'affaire.

## II.4. LE GENERATEUR DE COURANT

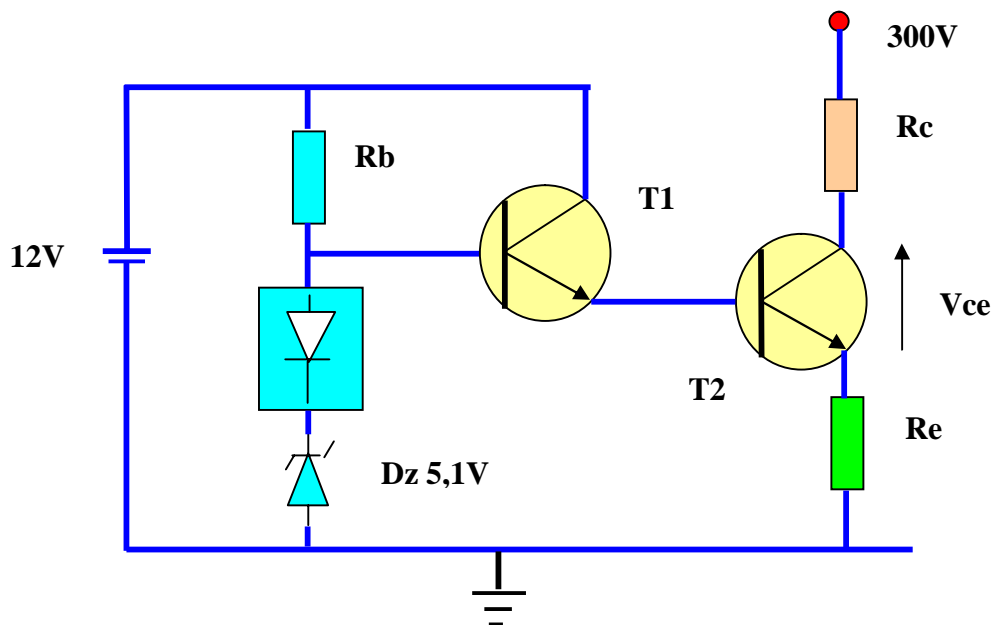


Figure 11 : schéma de base du générateur de courant

Le montage présenté à la *figure 11* (source : RAMBOLAMANANA Mamiharijao, rapport DEA) permet la réalisation d'un générateur de courant **constant** simple et efficace. Ce générateur fournit un courant constant même avec une variation de charge (**Rc**). La diode Zener de tension **Vz = 5,1V** polarisée correctement par la résistance **Rb** impose le potentiel de la base du transistor **T1**. La tension au borne du **Re = 4,7V**, cette valeur est fixée par le zener et ne varie pas avec la charge.

### Equation de la droite de charge :

$$V_{cc} = V_{ce} + R_c I_c + R_e I_e$$

(  $\beta$  très grande,  $I_c = I_e$  )

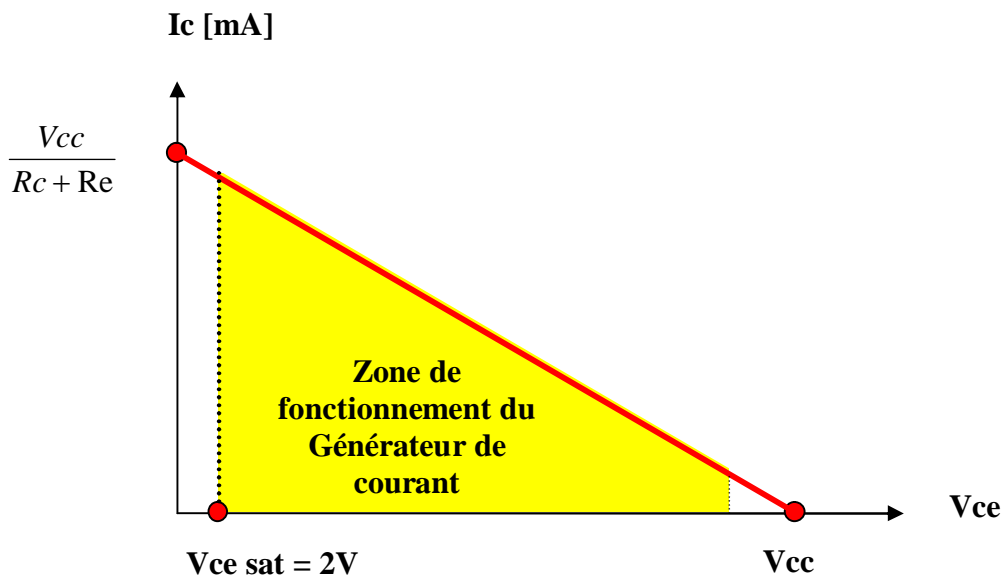


Figure 12 : zone de fonctionnement du générateur de courant

Le transistor T2 est dans son régime linéaire si et seulement si :

$$V_{ce\ sat} < V_{ce} < V_{cc} - [(R_c + R_e) I_{cb0}].$$

On peut donc calculer la valeur maximale de la charge **Rc max** en fonction du courant fourni par le générateur. **Vce sat** correspond à la tension de coude de la caractéristique, on l'appelle encore tension de déchet. Il s'agit de la valeur minimale que prend la tension **Vce** pour un **Ic** imposé. En fonction du courant et du type de transistor, cette valeur évolue entre **0,15 V** et **3V** environ.

- **Calcul de Rc max (charge maximale)**

$$V_{ce\ sat} < V_{ce} < V_{cc} - V_{ce\ sat}$$

$$V_{ce\ sat} < V_{cc} - (R_c + R_e) I_c < V_{cc} - V_{ce\ sat}$$

Dans le datasheet fournit par le constructeur, le TIP 162 prend une valeur **Vce sat = 2.8V** pour  $I_c = 6.5A$ . Pour notre réalisation,  $I_c$  max ne dépassera pas la valeur de 0.5A, ce qui nous permet de prendre une valeur  $V_{ce\ sat} = 2V$ . Le gain  $\beta$  du TIP 162 est égal à 200, on peut dire qu' $I_c \approx I_e$ .

Nous avons, donc :

$$\frac{V_{cc} - V_{cesat}}{I_c} - R_e > R_c$$

Le tableau suivant indique les valeurs des courants imposés  $I_c$  max, et les valeurs  $R_e$  correspondant, ainsi que les charges maximales  $R_c$  max.  $P_{max}$  est la puissance arrondie de la résistance  $R_e$ .

<b><math>I_c</math> max [mA]</b>	<b><math>R_e</math> [<math>\Omega</math>]</b>	<b><math>P_{max}</math> [W]</b>	<b><math>R_c</math> max [<math>\Omega</math>]</b>
0, 2	23 500	0.25	1 466 500
1	4 700	0.25	293 300
10	470	0.5	29 300
100	47	1	2 933
200	23,5	1	1466,5
500	9	3	586,6

Tableau 2 : valeurs des résistances  $R_e$  et  $R_c$  max

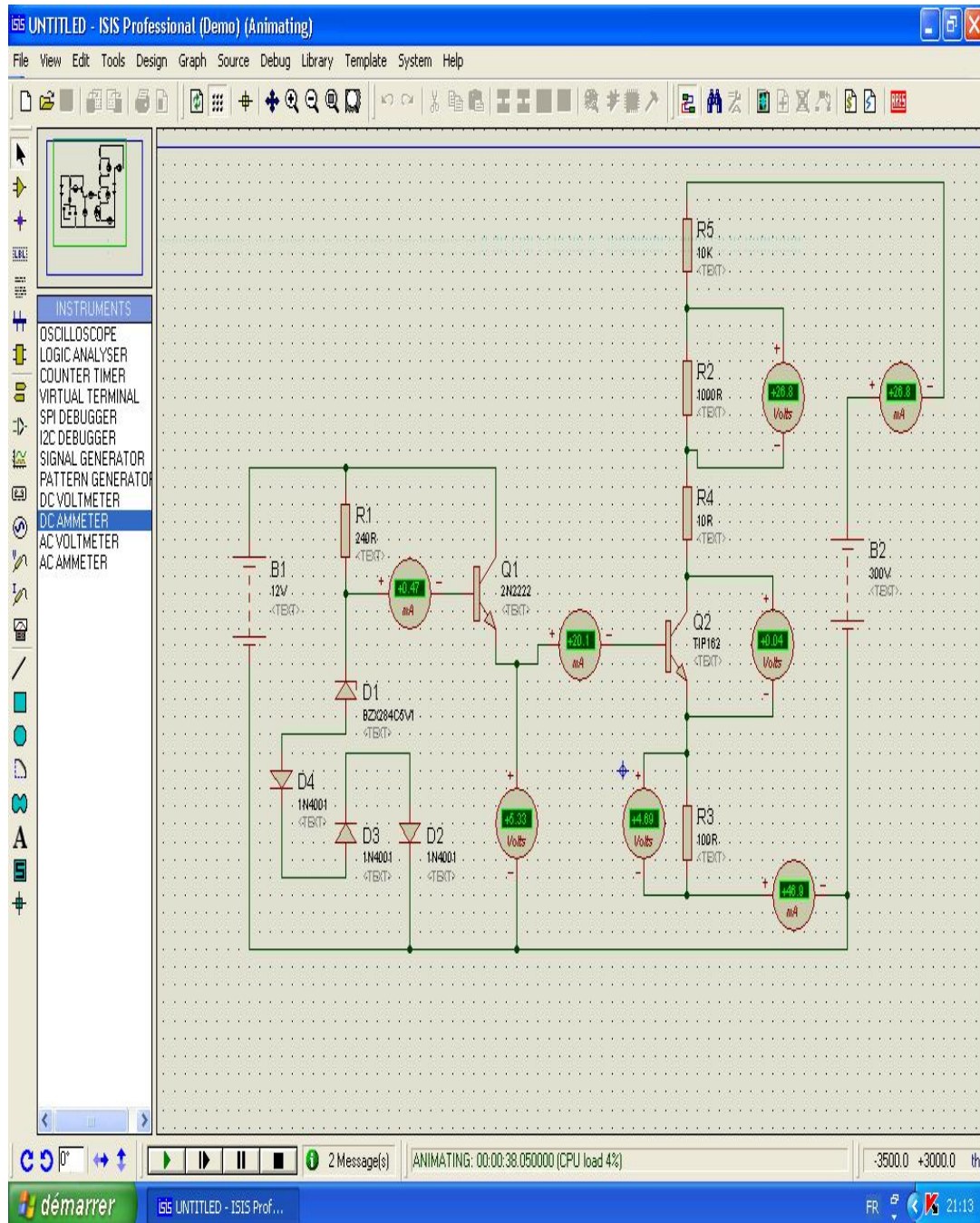


Figure 13 : Simulation du générateur de courant  
(Logiciel Proteus / version 7.1, Février 2007)





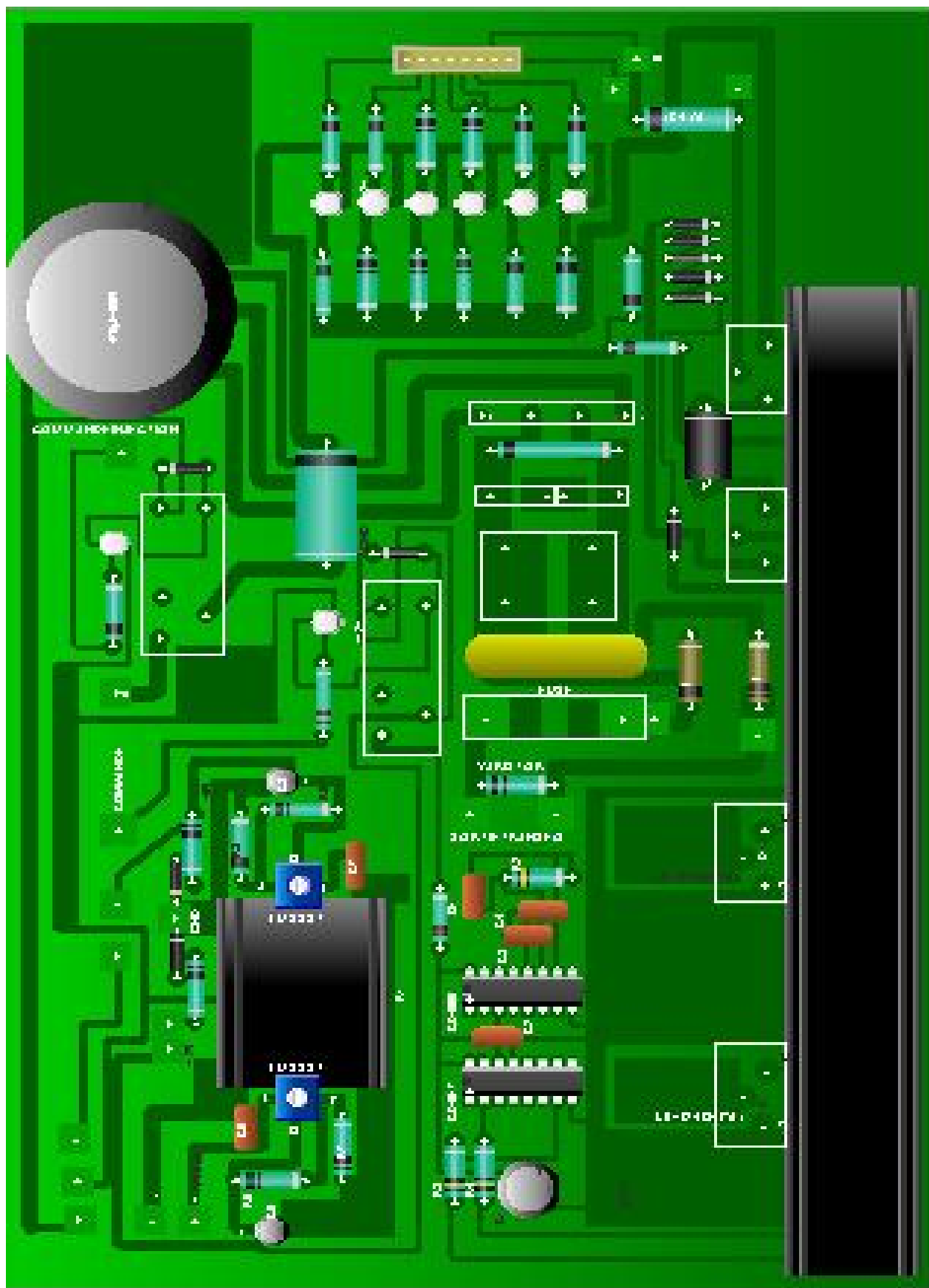


Figure 15 : Carte du Générateur de courant

### **CHAPITRE III : L'INTERFACE**

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 d'une carte d'interface. La carte est directement connectée sur le PC via le port série. Elle est équipée d'un microcontrôleur PIC 16F877 offrant la possibilité de gérer et commander notre résistivimètre. Cet ensemble PC et d'interface est complété par une interface logicielle permettant la gestion de l'ensemble de fonctions de la carte par l'intermédiaire d'un programme réalisé avec BORLAND C++. Le microcontrôleur regroupe dans un seul boîtier compact un processeur, de la mémoire, des périphériques.

#### **III.1 STRUCTURE INTERNE D'UN MICROCONTROLEUR :**

Le PIC 16F877 se présente sous la forme d'un circuit intégré réunissant tous les éléments d'une structure à base de microprocesseur. Il renferme (Figure 16):

- Un microprocesseur (C.P.U.),
- De la mémoire de donnée (RAM et EEPROM),
- De la mémoire programme (ROM, OTPROM, UVPROM ou EEPROM),
- Des interfaces parallèles pour la connexion des entrées / sorties,
- Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,
- Des timers pour générer ou mesurer des signaux avec une grande précision temporelle,
- Des convertisseurs analogique / numérique pour le traitement de signaux analogiques.

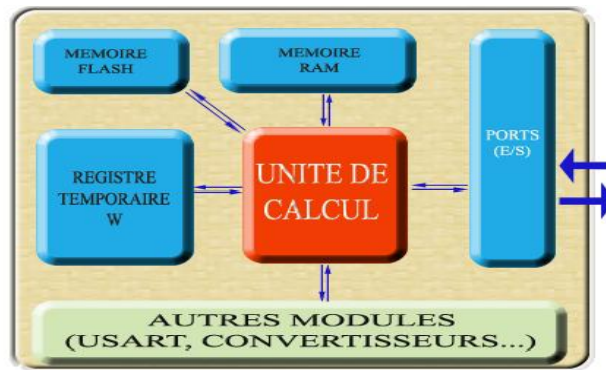


Figure 16 : Les principaux blocs fonctionnels présents à l'intérieur d'un PIC

L'Unité de Calcul est le coeur du microcontrôleur. Ici se déroulent toutes les opérations à une vitesse définie par la fréquence d'horloge (fréquence d'oscillation divisée par 4).

$$F_{horloge} = \frac{F_{oscillateur}}{4}$$

Un microprocesseur exécute séquentiellement les instructions stockées dans la mémoire programme. Il est capable d'opérer sur des mots binaires dont la taille, en bits, est celle du bus des données (parfois le double pour certains microcontrôleurs). Il est généralement constitué des éléments suivants :

- Un ou plusieurs registres **accumulateurs** contenant temporairement les opérandes ainsi que les résultats des opérations,
- Des **registres auxiliaires** permettant de relayer les accumulateurs,
- Des **registres d'index** pour le mode d'adressage indirect,
- Un **compteur programme** pointant l'adresse de la prochaine instruction à exécuter, sa taille est celle du bus des adresses,
- Une unité arithmétique et logique (**ALU**) permettant d'effectuer des opérations entre l'accumulateur et un opérande,
- Un **registre code condition** indiquant certaines particularités en ce qui concerne le résultat de la dernière opération (retenu, zéro, interruption...).

**Remarque :**

Il existe 2 catégories de microprocesseur : les CISC et les RISC.

**CISC** (Complex Instruction Set Computer) : Ce microprocesseur possède un nombre important d'instructions. Chacune d'elles s'exécute en plusieurs périodes d'horloges.

**RISC** (Reduced Instruction Set Computer) : Ce microprocesseur possède un nombre réduit d'instructions. Chacune d'elles s'exécute en une période d'horloge. (Cas du PIC).

**III.2. PRESENTATION DU PIC 16F877 :**

Les PICs sont des microcontrôleurs à architecture RISC (Reduce Instructions Construction Set), ou encore composant à jeu d'instructions réduit. L'avantage est que plus on réduit le nombre d'instructions, plus leur décodage sera rapide ce qui augmente la vitesse de fonctionnement du microcontrôleur.

La famille des PICs est subdivisée en 3 grandes familles :

- La famille **Base-Line**, qui utilise des mots d'instructions de 12 bits,
- La famille **Mid-Range**, qui utilise des mots de 14 bits dont fait partie la 16F877
- La famille **High-End**, qui utilise des mots de 16 bits.

Le PIC 16F877 est un microcontrôleur à mémoire FLASH. PIC ou « *Programmable Interface Controler* » ou Contrôleur d'Interface Programmable est un microcontrôleur fabriqué par la société américaine *Arizona MICROCHIP Technology*.

Mais avant toute chose faisons les présentations : le 16F877 est un circuit intégré contenu dans un boîtier dit « DIL 40 ». Il présente donc 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40, la broche 1 étant celle qui se trouve dans le coin situé à gauche de l'encoche de repérage.

Comme pour tout circuit intégré, chacune de ses broches a une ou plusieurs fonctions qui sont résumées par un sigle mnémotechnique. La description complète présentée ci-dessous se trouve dans la datasheet de Microchip. Nous présentons également un schéma simplifié du 16F877 avec les fonctions principales de ses 40 broches.

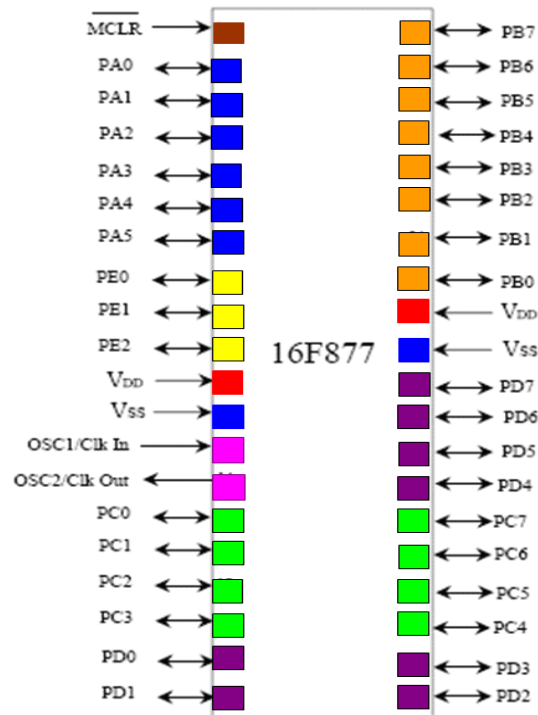


Figure 20 : Brochage du 16F877

Nous allons décrire les différentes broches du PIC 16F877 :

### **1) Les broches de « fonctionnement » :**

Les broches de « fonctionnement » sont les broches qui permettent au microprocesseur de fonctionner. Ces broches doivent obligatoirement être connectées pour que le 16F877 fonctionne d'où leur nom.

#### ➤ **Les broches d'alimentation :**

Tout d'abord, comme tout CI, le 16F877 a des broches d'alimentation : 2 pour le 0V (Bleu) et 2 pour le +5V (Rouge). Il suffit de connecter une de chaque à l'alimentation pour que le CI fonctionne. Cf figure 20

#### ➤ **Les broches du Quartz (en rose) :**

Ensuite, comme pour tout microprocesseur, le 16F877 va avoir besoin d'un « chef d'orchestre » pour « battre la mesure ». Ce « chef d'orchestre » va fixer la vitesse d'exécution des instructions. On utilise pour ce faire un quartz dont le rôle est de créer une impulsion de fréquences élevées. Dans le cas du 16F877, le quartz utilisé est typiquement un quartz de 4Mhz, c'est-à-dire qu'il va fournir 4 millions d'impulsions par secondes. Le temps qui s'écoule entre 2 impulsions s'appelle un cycle d'horloge. Un cycle d'horloge dure donc 250 nanosecondes (c'est-à-dire 250 milliardièmes de seconde).

Le microprocesseur va se baser sur cette fréquence pour son fonctionnement interne. Sachons que le 16F877 va exécuter une instruction élémentaire de langage assembleur en 4 cycles d'horloge. Il va donc pouvoir exécuter 1 million d'opérations élémentaires par seconde, soit le temps pour exécuter une instruction de  $1\mu\text{s}$

Le quartz de 4Mhz est connecté sur les 2 broches OSC1 et OSC2. En plus du quartz on ajoute une résistance de 1 M. et deux condensateurs de 22 pf reliés à la masse.

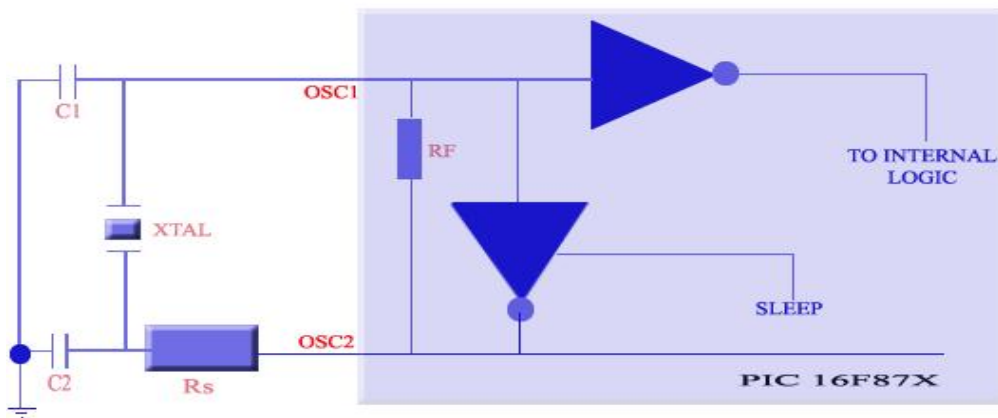


Figure 21 : brochage de l'oscillateur à quartz

Le filtre pass bas ( $R_s$ ,  $C_1$ ,  $C_2$ ) limite les harmoniques dus à l'écrêtage et réduit l'amplitude de l'oscillation, il n'est pas obligatoire.

➤ **La broche de réinitialisation (en marron) : cf figure 20**

Une broche particulière du microprocesseur est la broche  $\overline{\text{MCLR}}$ . Cette broche a pour effet de provoquer la réinitialisation du microprocesseur lorsqu'elle est connectée à 0. Ceci provoque l'arrêt du programme qui va recommencer à la première instruction.

Pour que le microprocesseur fonctionne, il faut donc que cette broche soit connectée au +5V en permanence.

## **2) Les Ports du 16F877 :**

Toutes les autres broches du 16F877 sont des broches de port. Elles peuvent aussi avoir d'autres rôles si les modules internes du 16F877 sont activés, comme nous allons le voir ultérieurement. **Toutes les autres broches peuvent donc être utilisées comme des broches de Port.**

Le 16F877 possède 5 ports différents :

- le port A (6 broches) en bleu
- le port B (8 broches) en orange
- le port C (8 broches) en vert
- le port D (8 broches) en violet
- le port E (3 broches) en jaune

Les broches de chaque port sont numérotées de 0 à n.

Les broches de ports sont désignées par un P suivi de la lettre du port et du numéro de broche.

Ainsi, la broche 0 du port A est désignée par PA0, la broche 4 du port B par PB4, etc...

**Remarque :**

Tout comme pour la numérotation des bits d'un registre, la 5e broche du port B porte le numéro 4 (et non pas 5).

Toutes les broches de ces ports sont des broches E/S, c'est-à-dire que l'on va pouvoir les configurer en entrée ou en sortie. Avec ses broches, on va donc pouvoir soit « sortir » des données du 16F877 sous forme de 1 et de 0 correspondant respectivement à une mise des broches au niveau HAUT ou BAS. On va également pouvoir « lire » des données sous forme de 1 et de 0 correspondant respectivement à un niveau HAUT ou BAS présent sur les broches. La structure de chaque port, à quelques variations près, est grosso modo la même ; chaque port possède assez logiquement un registre de données associé. Pour chaque broche configurée en sortie, chaque bit du registre de donnée détermine le niveau BAS ou HAUT présent sur cette broche. Pour chaque broche configurée en entrée, à chaque bit du registre de donnée correspond le niveau BAS ou HAUT présent sur cette broche

**EN RESUME :**

- le sens E/S des broches est configuré dans un registre appelé **TRIS**.
- un registre de données est attaché à chaque port.
- on peut brancher une LED via une résistance directement sur les broches de port en sortie.



### III.3 STRUCTURE INTERNE DU 16F877 :

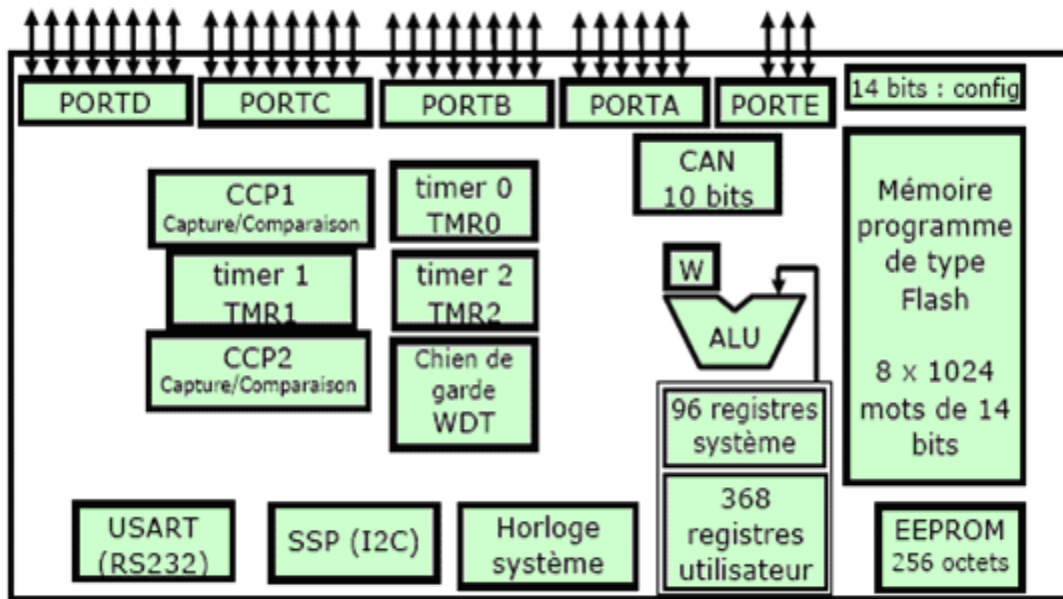


Figure 22 : Les éléments constitutifs du 16F877

#### 1) LA MEMOIRE DU 16F877 :

Il y a 2 zones mémoire EEPROM accessible par l'utilisateur. La zone DATA et la mémoire Flash de Programme.

- **La zone DATA** est organisée en mot de 8 bits. On peut programmer et lire les 256 octets. Elle est positionnée dans le chip entre les adresses h'2100' et h'21FF'.  
On l'adressera (de h'00' à h'FF') par le registre spécifique: EEADR en h'10D' (page 2).  
Les données transiteront par le registre: EEDATA en h'10C' (page 2).
- **La zone EEPROM** de programme qui est organisée en mots de 14 bits, est également accessible par l'utilisateur. Pour adresser les 8 K mots il faut 13 bits d'adresse. Un registre supplémentaire est disponible: EEADRH en h'10F' (page 2). Les 8 bits LSB de l'adresse seront passés par EEADR et les 5 bits MSB par EEADRH.  
Les données sur 14 bits seront échangées à travers le registre EEDATA pour les 8 bits LSB et grâce à un registre supplémentaire : EEDATH en h'10E' (page 2) pour les 6 bits MSB. Le tableau suivant nous montre l'emplacement de ces différents registres.

**MAPPING de la MEMOIRE DATA**

PAGE 0		PAGE 1		PAGE 2		PAGE 3	
00	INDF	80	INDF	100	INDF	180	INDF
01	TMR0	81	OPTION	101	TMR0	181	OPTION
02	PCL	82	PCL	102	PCL	182	PCL
03	STATUS	83	STATUS	103	STATUS	183	STATUS
04	FSR	84	FSR	104	FSR	184	FSR
05	PORTA	85	TRISA	105		185	
06	PORTB	86	TRISB	106	PORTB	186	TRISB
07	PORTC	87	TRISC	107		187	
08	PORTD(16F877)	88	TRISD(16F877)	108		188	
09	PORTE(16F877)	89	TRISE(16F877)	109		189	
0A	PCLATH	8A	PCLATH	10A	PCLATH	18A	PCLATH
0B	INTCON	8B	INTCON	10B	INTCON	18B	INTCON
0C	PIR1	8C	PIE1	10C	EEDATA	18C	EECON1
0D	PIR2	8D	PIE2	10D	EEADR	18D	EECON2
0E	TMR1L	8E	PCON	10E	EEDATH	18E	
0F	TMR1H	8F		10F	EEADRH	18F	
10	TICON	90		110	RAM 16 octets	RAM 16 octets	190
11	TMR2	91	SSPCON2				
12	T2CON	92	PR2				
13	SSBUF	93	SSPADD				
14	SSPCON	94	SSPSTAT				
15	CCPR1L	95					
16	CCPR1H	96					
17	CCP1CON	97					
18	RCSTA	98	TXSTA				
19	TXREG	99	SPBRG				
1A	RCREG	9A					
1B	CCPR2L	9B					
1C	CCPR2H	9C					
1D	CCP2CONL	9D					
1E	ADRESH	9E	ADRESL				
1F	ADCON0	9F	ADCON1				
20	RAM 96 octets	A0	RAM 80 octets	11F	RAM 80 octets	19F	RAM 80 octets
						120	
EF		EF		16F		1EF	
F0		F0		170		1F0	
7F		FF		17F		1FF	

Tableau 3 : les différentes pages de la mémoire

**Les registres utilisés par l'EEPROM sont :****EEDATA** : en h'10C' (page 2).**EEADR** : en h'10D' (page 2).**EEDATH** : en h'10E' (page 2).**EEADRH** : en h'10F' (page 2).**EECON1** : en h'18C' (page 3).**EECON2** : en h'18D' (page 3).**EECON1: (h'18C': page 3).**

Bit 7							Bit 0
<b>EEPGD</b>				<b>WRERR</b>	<b>WREN</b>	<b>WR</b>	<b>RD</b>

Tableau 4 : registre *EECON1*

Au reset : EECON1 = X000X000

**Bit 7 : EEPGD** =Program/Data Eeprom select bit : choix de la zone EEPROM.

1= Access à la zone EEPROM de programme.

0= Accès à la zone EEPROM de DATA.

Ce bit ne doit pas être changé pendant une opération de lecture ou d'écriture.

**Bit 3: WRERR** =EEPROM error flag.

1= Opération d'écriture prématurément terminée (causée par reset).

0= Opération d'écriture achevée normalement.

**Bit 2 : WREN** =EEPROM write enable bit.

1= Autorise cycle d'écriture en EEPROM.

0= Interdit l'écriture en EEPROM.

**Bit 1: WR** = Write Control bit.

1= Lance le cycle écriture de l'EEPROM. Ce bit est remis à "0" par hard.

0= Le cycle d'écriture est terminé.

**Bit 0: RD** = Read Control bit.

1= Lance un cycle de lecture de l'EEPROM. Ce bit est remis à "0" par hard.

0= Pas de cycle lecture.

**EECON2 : (h'18D' : page 3).**

Ce registre n'est pas un registre de configuration physique. Il n'est utilisé que pendant les séquences d'écriture en EEPROM.

Pour pouvoir écrire en EEPROM programme, il faut configurer correctement certains bits du REGISTRE de CONFIG en h'2007'.

Le bit WRT (Flash Program Memory enable) doit être forcé à "1". Les bits CP0 et CP1 doivent être tous les deux à "1" (Code Protect OFF), ou bien ne protéger qu'une zone particulière dans laquelle on ne voudra pas écrire.

**Lecture en zone EEPROM programme :**

- 1 - Mettre le MSB de l'adresse dans EEADRH.
- 2 - Mettre le LSB de l'adresse dans EEADR
- 3 - Mettre le bit EEPGD à "1" pour pointer la zone programme.
- 4 - Mettre le bit RD à "1" pour lancer le cycle de lecture.
- 5 - Attendre 2 cycles par deux NOP.
- 6 - Lire le MSB dans EEDATH et le LSB dans EEDATA.

**Ecriture en zone EEPROM programme :**

- 1 - Mettre le MSB de l'adresse dans EEADRH.
- 2 - Mettre le LSB de l'adresse dans EEADR.
- 3 - Mettre le MSB de la DATA dans EEDATH.
- 4 - Mettre le LSB de la DATA dans EEDATA.
- 5 - Mettre le bit EEPGD à "1" pour pointer la zone programme.
- 6 - Mettre le bit WREN à "1" pour autoriser l'écriture.
- 7 - Inhiber les interruptions par mise à "0" du bit GIE de INTCON.
- 8 - Lancer la séquence spécifique suivante :
  - Ecrire h'55' dans EECON2.
  - Ecrire h'AA' dans EECON2.
  - Lancer le cycle d'écriture par WR=1.
  - On attend 2 cycles par deux NOP.
  - Le PIC ignore maintenant toutes les instructions.
  - A la fin de l'écriture, le PIC exécute l'instruction suivante.

- 9 - On autorise les interruptions par GIE=1.
- 10 - On interdit l'écriture en EEPROM par WREN = 0.

## 2) LES MODULES INTERNES DU 16F877 :

Nous allons présenter ici une vue d'ensemble des différents modules utilisables qui sont présents dans la structure interne du 16F877.

Le 16F877 comprend donc :

- Trois timers / compteurs :
  - le timer 0
  - le timer 1
  - le timer 2
- Un convertisseur analogique numérique (CAN) 10 bits
- Un module de génération d'impulsions à période réglable (PWM)
- Un module de communication série synchrone
- Une USART
- Un module de communication en « port parallèle »
- Un « chien de garde »

### ➤ Trois timers / compteurs :

Le gros intérêt des modules de comptage, c'est qu'ils permettent de tenir compte d'évènements qui surviennent de façon répétée sans que le microprocesseur soit monopolisé par cette tâche. Dans la plupart des cas, une interruption n'a lieu que lorsque le compteur déborde (overflow).

#### ○ *Le timer 0*

Le timer 0 est un compteur 8 bits qui peut compter (de 0 à 255) :

- soit les impulsions de l'horloge via un pré diviseur
- soit des impulsions externes, via la broche PA4

Le débordement (overflow) du compteur, qui a lieu lorsque le compteur passe de 255 à 0, provoque une interruption.

- **Le timer 1**

Le timer 1 est un compteur 16 bits qui peut compter (de 0 à 65535) :

- soit les impulsions de l'horloge
- soit les impulsions externes, et en particulier les impulsions d'un quartz externe.

Là aussi, le débordement provoque une interruption.

- **Le timer 2**

Le timer 2 est un timer couplé au module dit CCP. Il peut compter de 0 à 65535 comme le timer1. C'est ce timer que nous utilisons pour compter les durées entre l'émission et la réception des signaux venant de l'émetteur.

➤ **Le module de génération d'impulsion à période réglable (PWM) :**

Ce module est très puissant pour créer des impulsions à des fréquences élevées, par exemple à 38 KHz, dans le cadre d'une communication infrarouge.

➤ **Un module de communication série synchrone :**

Le 16F877 dispose d'un module de communication série synchrone.

➤ **Un module de communication en « port parallèle » :**

Le 16F877 dispose d'un module de communication parallèle intégré, utilisant les 3 broches de contrôle habituelles CS, RS et WR.

➤ **Un « chien de garde » :**

Système anti-plantage du microcontrôleur, il s'assure qu'il n'y ait pas d'exécution prolongée d'une même suite d'instruction. Un compteur pré chargeable se décrémente régulièrement au rythme de la fréquence d'horloge. Si aucun pré chargement n'est effectué avant qu'il n'atteigne la valeur "0" un Reset est généré relançant ainsi le microcontrôleur. Il faut donc penser à pré charger régulièrement ce chien de garde par programme lorsqu'il est activé. Autrement dit, le chien de garde est un oscillateur capable de réveiller le microprocesseur mis à l'arrêt par le programme.

➤ **Convertisseur A/D :**

Le PIC 16F877 dispose d'un convertisseur analogique digital. Il permet de convertir une valeur Analogique 0-5 V en une valeur numérique pouvant aller jusqu'à 10 bits. Plus nous augmenterons la précision de la mesure et plus le temps de mesure sera élevé. Il faudra donc parfois faire un compromis vitesse/précision. Sa résolution de 10bits donne une précision en 5V de 5mV environ.

Le principe de conversion est simple, la tension à numériser est appliquée à un condensateur interne et, une fois le condensateur chargé, on applique le principe de l'approximation successive dont l'algorithme est indiqué par la *figure 23*.

**Nbit**= 10Bits : résolution du CAN

**Tension analogique U** : tension obtenue aux bornes de l'électrode MN : PS ou  $\Delta V$

**Refmax** : tension de référence positive = 5V

**Refmin** : tension de référence négative = 0V

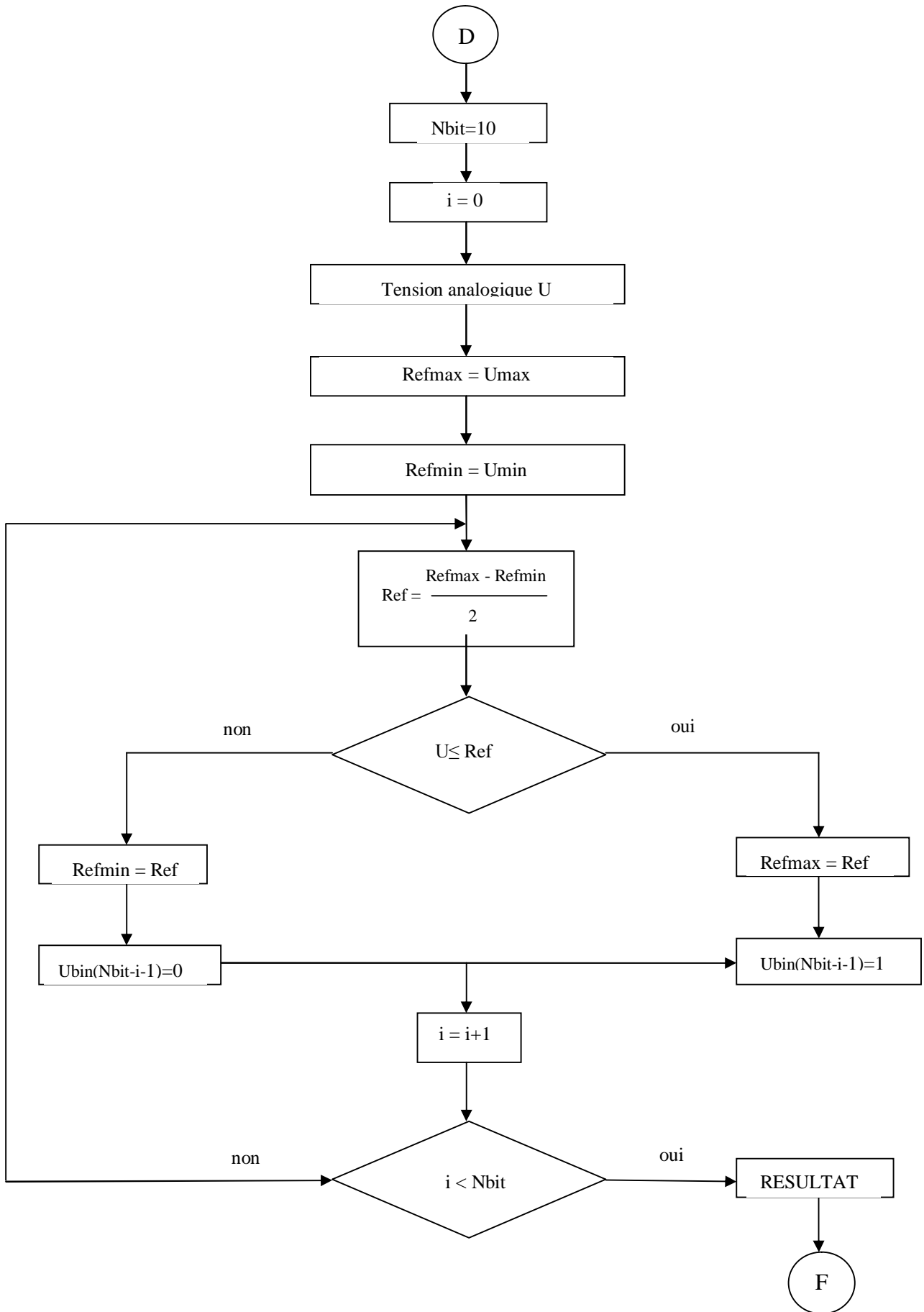


Figure23 : Algorithme du CAN à approximation successive



Le PIC 16F877 possède 8 entrées analogiques, (RA0..RA5 et RE0..RE2) multiplexées vers un C.A.N. à approximation successive (SAR = Successive approximation register). Ce dernier est précédé d'un échantillonneur / bloqueur permettant une stabilité de la tension d'entrée pendant toute la durée de la conversion. La figure suivante montre un schéma synoptique du CAN.

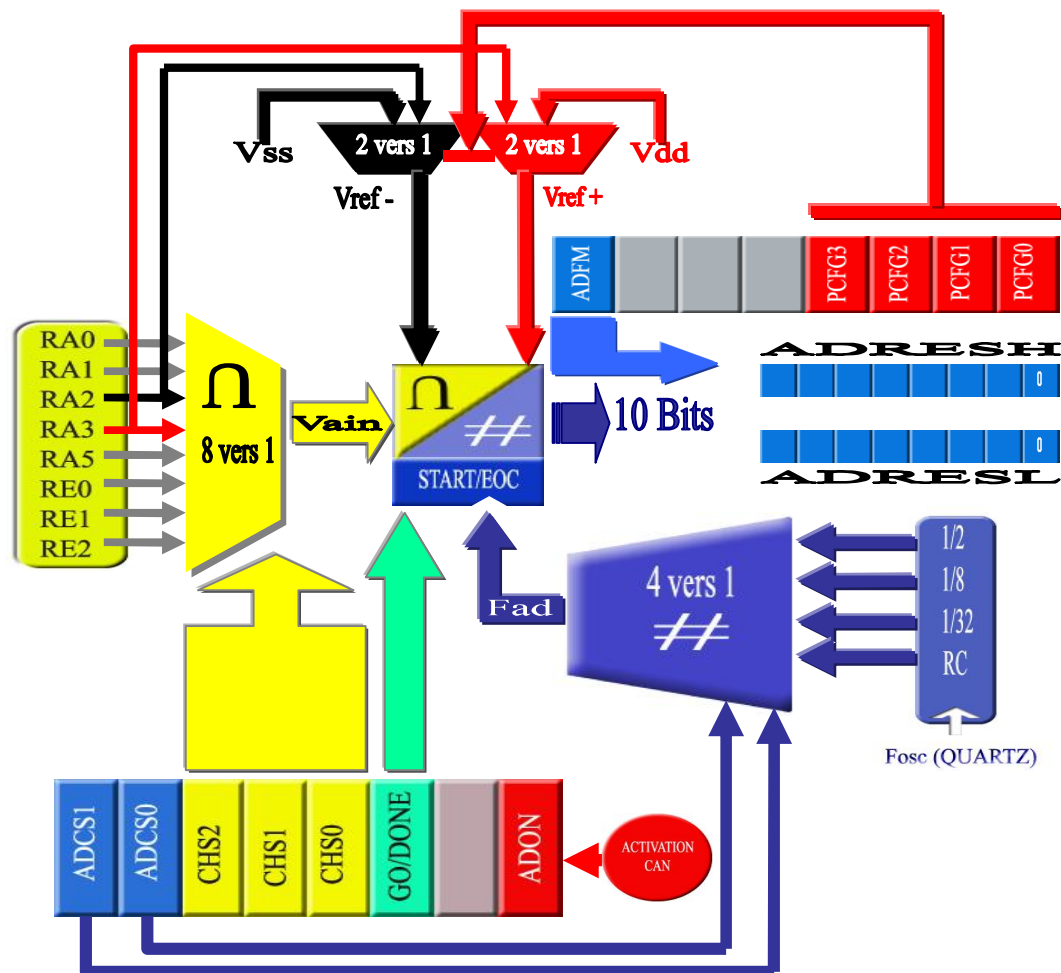


Figure 24 : Structure du CAN

La gestion de la conversion se fait grâce à 4 registres 8 bits :

- **ADRESH** et **ADRESL** (Analog to Digital result High and Low), contiennent le résultat de la conversion sur 10 bits. Rem: un octet ne contiendra que 2 bits.
- **ADCON0** et **ADCON1** (Analog to Digital Control 0, 1), permettent de configurer la conversion.

- La tension de référence peut être interne (**GND ou 5v**), ou externe (**RA2 et RA3**) selon la configuration. En aucun cas elle ne doit dépasser la tension d'alimentation.
- Le temps de conversion minimum est d'environ **20 µs**.

❖ *Configuration de la conversion :*

**REGISTRE ADCON0 : (h1F : page 0)**

<b>BIT 7</b>	<b>BIT 6</b>	<b>BIT 5</b>	<b>BIT 4</b>	<b>BIT 3</b>	<b>BIT 2</b>	<b>BIT 1</b>	<b>BIT 0</b>
<b>ADCS1</b>	<b>ADCS0</b>	<b>CHS2</b>	<b>CHS1</b>	<b>CHS0</b>	<b>GO/DONE</b>	Inutilisé	<b>ADON</b>
Sélection Fréquence Horloge		Sélection entrée de conversion			Lancement/Fin de conversion		Activation du C.A.N

Tableau 5 : Registre ADCON0

❖ *Sélection fréquence d'horloge du CAN :*

**ADCS1 et ADCS0 : Clock Select bits**

Ces bits permettent de choisir la vitesse de conversion

<b>ADCS1</b>	<b>ADCS0</b>	<b>Fréquence F<sub>AD</sub></b>	<b>Fréquence max. du Quartz</b>
<b>0</b>	<b>0</b>	<b>1/2 F<sub>osc</sub></b>	<b>1.25 MHz</b>
<b>0</b>	<b>1</b>	<b>1/8 F<sub>osc</sub></b>	<b>5 MHz</b>
<b>1</b>	<b>0</b>	<b>1/32 F<sub>osc</sub></b>	<b>20 MHz</b>
<b>1</b>	<b>1</b>	<b>Circuit RC interne <sup>(1)</sup></b>	

Tableau 6 : Vitesse de conversion

<sup>(1)</sup> La fréquence F<sub>AD</sub> engendrée par le RC est d'environ 250 KHz

Le temps de conversion d'un bit est T<sub>AD</sub>.

Pour une conversion totale des 10 bits il faut 12 x T<sub>AD</sub>

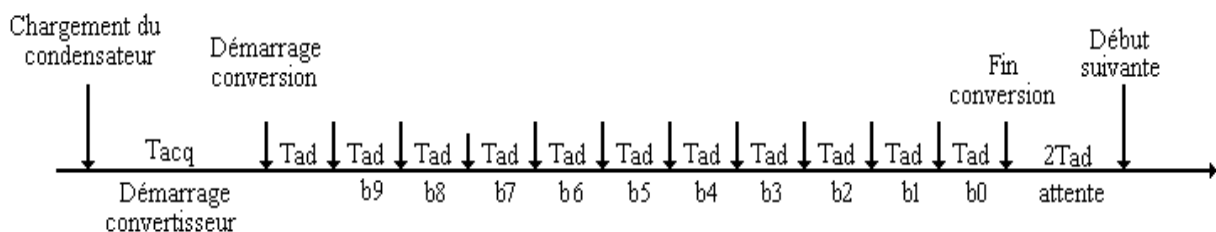


Figure 25 : Processus de conversion du CAN

Pour que la conversion soit correcte, il faut que  $T_{AD}$  soit au minimum de **1,6 $\mu$ s**, cf tableau 7.

QUARTZ	CLOCK	$T_{AD}$	12. $T_{AD}$	Ne convient pas si $T_{AD} < 1,6\mu s$
4 MHz	$F_{osc}/2 = 2 \text{ MHz}$	0,5 $\mu$ s	6 $\mu$ s	Ne convient pas
	$F_{osc}/8 = 500 \text{ KHz}$	2 $\mu$ s	24 $\mu$ s	OK
	$F_{osc}/32 = 125 \text{ KHz}$	8 $\mu$ s	96 $\mu$ s	OK
8 MHz	$F_{osc}/2 = 4 \text{ MHz}$	0,25 $\mu$ s	3 $\mu$ s	Ne convient pas
	$F_{osc}/8 = 1 \text{ MHz}$	1 $\mu$ s	12 $\mu$ s	Ne convient pas
	$F_{osc}/32 = 250 \text{ KHz}$	4 $\mu$ s	48 $\mu$ s	OK
12 MHz	$F_{osc}/2 = 6 \text{ MHz}$	0,16 $\mu$ s	1,92 $\mu$ s	Ne convient pas
	$F_{osc}/8 = 1,5 \text{ MHz}$	0,66 $\mu$ s	8 $\mu$ s	Ne convient pas
	$F_{osc}/32 = 375 \text{ KHz}$	2,6 $\mu$ s	32 $\mu$ s	OK
16 MHz	$F_{osc}/2 = 8 \text{ MHz}$	0,125 $\mu$ s	1,5 $\mu$ s	Ne convient pas
	$F_{osc}/8 = 2 \text{ MHz}$	0,5 $\mu$ s	6 $\mu$ s	Ne convient pas
	$F_{osc}/32 = 500 \text{ KHz}$	2 $\mu$ s	24 $\mu$ s	OK

Tableau 7 : Temps de conversion  $T_{AD}$  en fonction du Quartz et des bits du Clock select (Référence datasheet Microship)

❖ **Sélection de conversion :**

**CHS2 CHS1 CHS0: Channel Select bits**

Ces 3 bits permettent de choisir l'entrée qui va être convertie

CHS2	CHS1	CHS0	Canal Sélectionné
0	0	0	AN0 (RA0)
0	0	1	AN1 (RA1)
0	1	0	AN2 (RA2)
0	1	1	AN3 (RA3)
1	0	0	AN4 (RA5)
1	0	1	AN5 (RE0)
1	1	0	AN6 (RE1)
1	1	1	AN7 (RE2)

Tableau 8 : canal de sélection

❖ **Lancement et fin de conversion :**

Le bit **GO/DONE** lance la conversion lorsqu'il est positionné à "1". Il passe à "0" lorsque la conversion est terminée. Il est nommé **ADGO** dans le langage C HI-TECH.

❖ **Activation du C.A.N. :**

Le positionnement à "1" du bit ADON active le convertisseur A/N. Dans le cas contraire le C.A.N est inopérant.

**REGISTRE ADCON1**

<b>BIT 7</b>	<b>BIT 6</b>	<b>BIT 5</b>	<b>BIT 4</b>	<b>BIT 3</b>	<b>BIT 2</b>	<b>BIT 1</b>	<b>BIT 0</b>
<b>ADFM</b>	Inutilisé	Inutilisé	Inutilisé	PCFG3	PCFG2	PCFG1	<b>PCFG0</b>
<b>Mode de présentation du résultat</b>	<b>Inutilisés (sont lus à 0)</b>			<b>Sélection de la tension de référence et du nombre d'entrée analogique</b>			

Tableau 9 : Registre ADCON1

ADFM= Sélection du Format du résultat de la conversion :

Le résultat de la conversion sur 10 bits peut se présenter de 2 façons :

- **ADFM** = 0 justification à gauche (dans ce cas ADRESH donne un résultat sur 8 bits, correspondant à la partie la plus significative = poids fort). Ce mode est surtout utilisé lorsque l'on se contente d'un résultat sur 8 bits (variable de type "unsigned char").

7 ADRESH 0								7 ADRESL 0							
<b>D9</b>	<b>D8</b>	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	0	0	0	0	0	0

Tableau 10 : Justification à gauche

- **ADFM** = 1 justification à droite. Le résultat correspond directement à un entier de type "unsigned int" en langage C. Ce mode est donc tout indiqué lorsque l'on souhaite utiliser le résultat sur 10 bits.

7 ADRESH 0								7 ADRESL 0							
0	0	0	0	0	0	<b>D9</b>	<b>D8</b>	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	

Tableau 11: Justification à droite

## Sélection de la tension de référence et du nombre d'entrées analogiques :

## PCFG3 PCFG2 PCFG1 PCFG0 : Bits de contrôle de la configuration des ports.

Ces bits permettent de choisir le partage entre entrées analogiques et digitales sur les PORTS A et E ; ils permettent également de choisir pour VRéf + entre Vdd et PA3 et pour VRéf – entre Vss et PA2

**A** : Entrée Analogique

**D** : I/O Digitale

PCFG3 : PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	NOMBRE Ain
0000	A	A	A	A	A	A	A	A	VDD	VSS	8
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7
0010	D	D	D	A	A	A	A	A	VDD	VSS	5
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4
0100	D	D	D	D	A	D	A	A	VDD	VSS	3
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2
0110	D	D	D	D	D	D	D	D	VDD	VSS	0
0111	D	D	D	D	D	D	D	D	VDD	VSS	0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6
1001	D	D	A	A	A	A	A	A	VDD	VSS	6
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1

Tableau 12 : Configuration des ports

**SÉQUENCES À RESPECTER POUR LA CONVERSION :**○ **Séquence d'initialisation :**

1. Sélection de la tension de référence et du nombre d'entrées analogiques : **"PCFG 3:0"**.
2. Sélection du mode de présentation du résultat (justification droite ou gauche) : **"ADFM"**.
3. Sélection de fréquence d'horloge du convertisseur **"ADCS 1:0"**.
4. Activation du C.A.N. : **"ADON = 1"**.

○ **Séquence de conversion :**

1. Sélection de l'entrée analogique à convertir **"CHS 2:0"**.
2. Lancement de la conversion : **"GO/DONE = 1"**. (ADGO = 1;)
3. Attente de fin de conversion : **"GO/DONE = 0 ?"**. (While (ADGO == 1); ou While (ADGO);).
4. Lecture du résultat dans **ADRESH** et **ADRESL**.

**INTERRUPTION**

La fin de conversion A/N peut générer une routine d'interruption. Pour cela il faut positionner à "1" le bit d'activation général des interruptions **"GIE"** (Bit 7 du registre **INTCON**), le bit d'activation des interruptions périphériques **"PEIE"** (Bit 6 du registre **INTCON**), ainsi que le bit d'activation de l'interruption CAN **"ADIE"** (Bit 6 du registre **PIE1**). Lorsqu'une requête d'interruption est lancée par le CAN (A la fin d'un cycle de conversion), le drapeau (Flag) **"ADIF"** (Bit 6 du registre **PIR1**) passe à "1". Ce bit doit être repositionné à "0" avant toute nouvelle conversion et lors de l'initialisation.

La relation entre les grandeurs analogiques d'entrées et les grandeurs numériques s'écrit :

$$V_{num} = \left( \frac{V_{in} - V_{ref-}}{V_{ref+} - V_{ref-}} \right) \times (2^n - 1)$$

On définit par  $V_{ref+}$  la tension de référence positive et  $V_{ref-}$  la tension de référence négative ;

$V_{in}$  : tension d'entrée analogique et

$V_{num}$  : la tension de sortie numérique.

Comme nous utilisons une tension de référence négative de 0V, c'est-à-dire que la référence de tension négative est en réalité  $V_{ss}$ , alors :

$$V_{num} = \left( \frac{V_{in}}{V_{ref+}} \right) \times (2^n - 1)$$

$$\text{D'où } V_{in} = \left( \frac{V_{num}}{2^n - 1} \right) \times V_{ref+}$$

Après avoir convertir les données ; le PIC utilise le module USART pour les transmettre. Ceci permet aussi de recevoir ou d'envoyer des données en mode série, de façon synchrone ou asynchrone.

### ➤ L'USART :

**USART** signifie « Universal Synchronous Asynchronous Receiver Transmitter ». C'est un module qui permet d'envoyer et de recevoir des données en mode série, soit de façon synchrone, soit asynchrone. Nous nous intéresserons dans le mode série asynchrone.

#### **Le mode série asynchrone**

Une liaison série asynchrone, comme son nom l'indique, fonctionne, en émettant les bits les uns à la suite des autres, mais sans fournir le signal d'horloge qui a permis de les générer.

La liaison utilisera les 2 mêmes pins que pour la liaison synchrone, à savoir **PC6/TX/CK** et **PC7/RX/DT**. Les dénominations qui conviendront dans ce cas seront bien entendu **TX** pour l'émission et **RX** pour la réception. Ces pins devront être configurées en entrée via **TRISC** pour fonctionner en mode **USART**.

Les 5 registres utilisés sont :

Registre Emission : **TXREG** en h'19' page 0.

Registre Réception: **RCREG** en h'1A' page 0.

Registre d'état Emission : **TXSTA** en h'98' page 1.

Registre d'état Réception : **RCSTA** en h'18' page 0.

Registre du choix de la vitesse : **SPBRG** en h'99 page 1.

#### **REGISTRE TXSTA: (h'98': page 1).**

Bit 7						Bit 0	
CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D

Tableau 13: Registre TXSTA

Au reset : **TXSTA = 00000010**

**Bit 7 : CSRC** = Clock Source en synchrone. Sans importance en asynchrone.

**Bit 6 : TX9** = Autorisation d'émission sur 9 bits.

1 = Autorisé.

0 = Non autorisé.

**Bit 5 : TXEN** = Autorisation d'émission.

1 = Autorisé.

0 = Non autorisé.

**Bit 4 : SYNC** = Sélection mode Synchrone / Asynchrone.

1 = Mode synchrone.

0 = Mode asynchrone.

**Bit 3 : Non implémenté**

**Bit 2 : BRGH** = Sélection vitesse rapide en mode asynchrone.

1 = Vitesse haute sélectionnée.

0 = Vitesse basse sélectionnée.

**Bit 1 : TRMT** = bit d'état du registre à décalage Emission.

1 = Registre vide, donc émission terminée.

0 = Registre plein, donc émission en cours.

**Bit 0 : TX9D** = 9eme bit de Data transmise. Ce bit peut être le bit de la parité.

### REGISTRE SPBRG: (h'99' : page 1).

Le Baud Rate Generator est un registre 8 bits qui contient le facteur de division (N) de l'horloge interne qui permet d'obtenir la vitesse commune d'émission et de réception.

En mode Asynchrone (bit SYNC = 0) suivant l'état du bit BRGH on aura le choix entre 2 vitesses : haute pour BRGH=1 et basse pour BRGH=0.

BRGH=0 VITESSES BASSES	BRGH=1 VITESSES HAUTES
$VITESSE = \frac{F_{oscill}}{64(N+1)}$	$VITESSE = \frac{F_{oscill}}{16(N+1)}$
$N = \frac{F_{oscill}}{64.Vitesse} - 1$	$N = \frac{F_{oscill}}{16.Vitesse} - 1$

Tableau 14 : Etat du bit BRGH



Le nombre N est le nombre entier, arrondi de la valeur trouvée par les équations ci dessus.  
Il est recommandé d'utiliser si possible les vitesses hautes (BRGH=1), même pour des vitesses faibles, car dans ce cas on minimise l'erreur, en obtenant un nombre N plus grand.

**Valeurs de N pour diverses vitesses avec un Quartz de 8 MHz :**

VITESSES en Bits/sec ou BAUDS	VITESSES BASSES BRGH = 0	VITESSES HAUTES BRGH = 1
119200	×	4
57600	×	8
38400	×	12
19200	5	25
9600	12	51
4800	25	103
2400	51	207
1200	103	×

Tableau 15 : Facteur de division N avec un Quartz à 8MHz

**REGISTRE RCSTA: (h'18': page 0).**

Bit 7							Bit 0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

Tableau 16: Registre RCSTA

Au reset : **RCSTA = 0000000X**

**Bit 7 : SPEN**= Serial Port Enable. PC7 et PC6 configurés pour le port série.

1 = Port série en service.

0 = Port série désactivé.

**Bit 6 : RX9** = Autorisation de réception sur 9 bits.

1 = Autorisé.

0 = Non autorisé.

**Bit 5 : SREN** = Single Receive Enable. Réservé pour mode Synchrone. Non utilisé en mode Asynchrone.

**Bit 4: CREN** = Continous Receive Enable.

**1** = Autorise la réception en continu.

**0** = Désactive la réception en continu.

**Bit 3: ADDEN** = Adress Detect Enable. En mode Asynchrone 9 bits :

**1** = Autorise la détection d'adresse, et charge la Data dans le registre de réception RCREG quand le 9eme bit du registre de dé sérialisation vaut "1".

**0** = Désélectionne la détection d'adresse. Tous les octets sont reçus et le 9eme bit peut servir de bit de parité.

**Bit 2: FERR** = Framing Error .

**1** = Une erreur de Framing est survenue.

**0** = Pas d'erreur de Framing.

**Bit 1: OERR** = Overrun Error.

Un octet est reçu alors que le registre de réception n'a pas été vidé par lecture.

**1** = Erreur Overrun.

**0** = Pas d'erreur Overrun.

**Bit 0: RX9D** = 9eme bit de Data reçue. Ce bit peut être le bit de la parité.

### 3) PRESENTATION DE LA LIAISON RS232 ET DU MAX 232 :

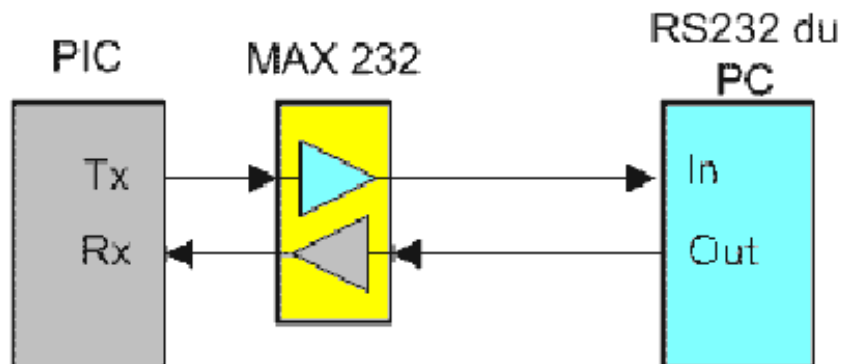


Figure 26 : Utilisation du MAX232

Le MAX232 convertit les niveaux 0v/5V du PIC en niveaux +12V/-12V à destination de la RS232 du PC, et réciproquement. La norme RS232 est également une norme qui précise un mode de fonctionnement full-duplex.

➤ . **Rôle**

Une liaison série est une ligne où les bits d'informations (1 ou 0) arrivent successivement, soit à intervalles réguliers (transmission synchrone), soit à des intervalles aléatoires (transmission asynchrone). La liaison RS 232 est une liaison asynchrone. Les intérêts de ce type de communication, c'est tout d'abord un nombre de fils réduits. En effet, la communication la plus simple peut être faite sur 3 fils (Tx, Rx et masse). Ensuite cela permet une communication sur de grandes distances à travers le réseau téléphonique par exemple.

➤ . **Principe**

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

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).

➤ . **Protocole de transmission**

Afin que les éléments communicants puissent se comprendre, il est nécessaire d'établir un protocole de transmission. Ce protocole devra être le même pour les deux éléments afin que la transmission fonctionne correctement.

- longueur des mots : 7, 8 bits.
- Vitesse de transmission : 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 bauds.
- Parité :

**Paire** : Le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit pair sur l'ensemble donné plus bit de parité.

**Impaire** : Le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit impair sur l'ensemble donné plus bit de parité.

- bit de START : la ligne au repos est à l'état logique 1 pour indiquer qu'un mot va être transmis, la ligne passe à l'état bas avant de commencer le transfert. Ce bit permet de synchroniser l'horloge du récepteur.
- bit de stop : après la transmission, la ligne est positionnée au repos pendant 1, 2 ou 1,5 période d'horloge selon le nombre de bits de stop.

➤ . **Format des trames.**

Le bit de START apparaît en premier dans la trame puis les données (poids faible en premier), la parité éventuelle et le bit de stop.

➤ . **Port série d'un PC.**

Les niveaux logiques à la sortie du port série sont les suivant :

- 1 :→ + 12 V

- 0 :→ - 12 V

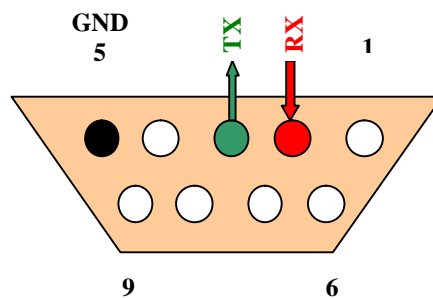


Figure 27 : Brochage du DB9

Les données arrivent du port série via la patte 2 et sont transmises via la patte 3. Il faut ajouter une masse à ce dispositif.

### III.4 PIC VERS PC :

Il importe en premier lieu d'analyser comment connecter physiquement notre PIC à notre port RS232. Le PIC utilise les niveaux 0V et 5V pour définir respectivement des signaux « 0 » et « 1 ». La norme RS232 définit des niveaux de +12V et -12V pour établir ces mêmes niveaux. Nous aurons donc besoin d'un circuit (driver de bus) chargé de convertir les niveaux des signaux entre PIC et PC. Le pin TX du PIC émettra en 0V/5V et sera convertie en +12V/-12V vers notre PC. La ligne RX du PIC recevra les signaux en provenance du PC, signaux qui seront converti du +12V/-12V en 0V/5V par notre circuit de pilotage du bus.

Nous utiliserons le circuit MAX232 pour effectuer cette adaptation de niveaux. Le MAX232 est un composant créé par MAXIM que l'on trouve sous d'autres références chez d'autres fabricants. Il sert d'interface entre une liaison série TTL (0-5V) et une liaison série RS232 (+12 -12V) et ce avec une simple alimentation 5V. En fait, il amplifie et met en forme deux entrées et deux sorties TTL/MOS vers deux entrées et deux sorties RS232.

Les broches 10 (T2in) et 11(T1in) correspondent aux 2 entrées TTL, les sorties RS232 correspondantes étant les broches 14(T1out) et 7(T2out).

Les broches 13(R1in) et 8(R2in) correspondent aux entrées RS 232, les sorties TTL correspondantes étant les broches 12(R1out) et 9(R2out).

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. 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.

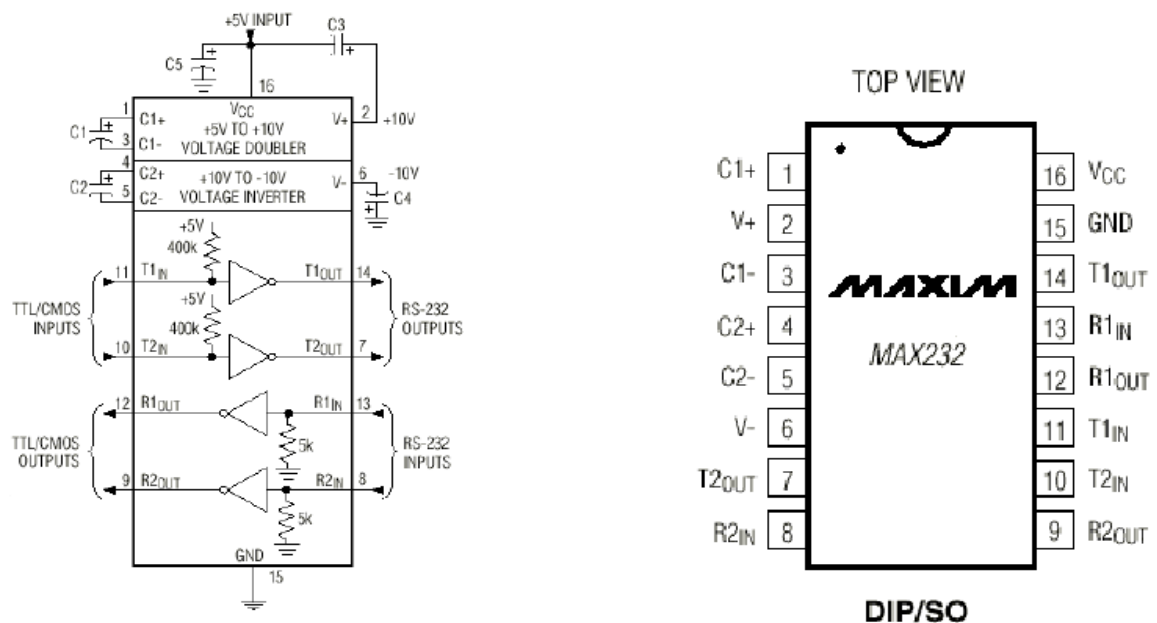


Figure 28 : le MAX 232

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 par la nécessité de convertir, outre RX et TX, les signaux RS232 supplémentaires.

Maintenant, nous allons avoir besoin de 2 logiciels, un du côté du PIC, et qui nous intéresse, et un du côté du PC, afin de recevoir et d'envoyer des informations.

Il nous faudra interconnecter les pins concernés en suivant les spécificités des « null-modem ». Autrement dit, dans la fiche DB9 de notre câble, du côté du PC, nous connecterons ensemble les pins 4,6, et 1. Un second pontage entre les pins 7 et 8 complétera l'intervention.

Le pin 3 de la fiche DB9 de notre PC est l'émission (vu par le PC), donc la réception vu du côté du PIC. Ce signal est dénommé TD (Transmitt Data).

Le pin 2 est le pin de réception de notre PC. Ce signal est dénommé RD (Receive Data).

Le pin 5 est la masse de notre RS232, qu'il ne faudra pas oublier de connecter à la masse de notre PIC.

Voyons le schéma obtenu :

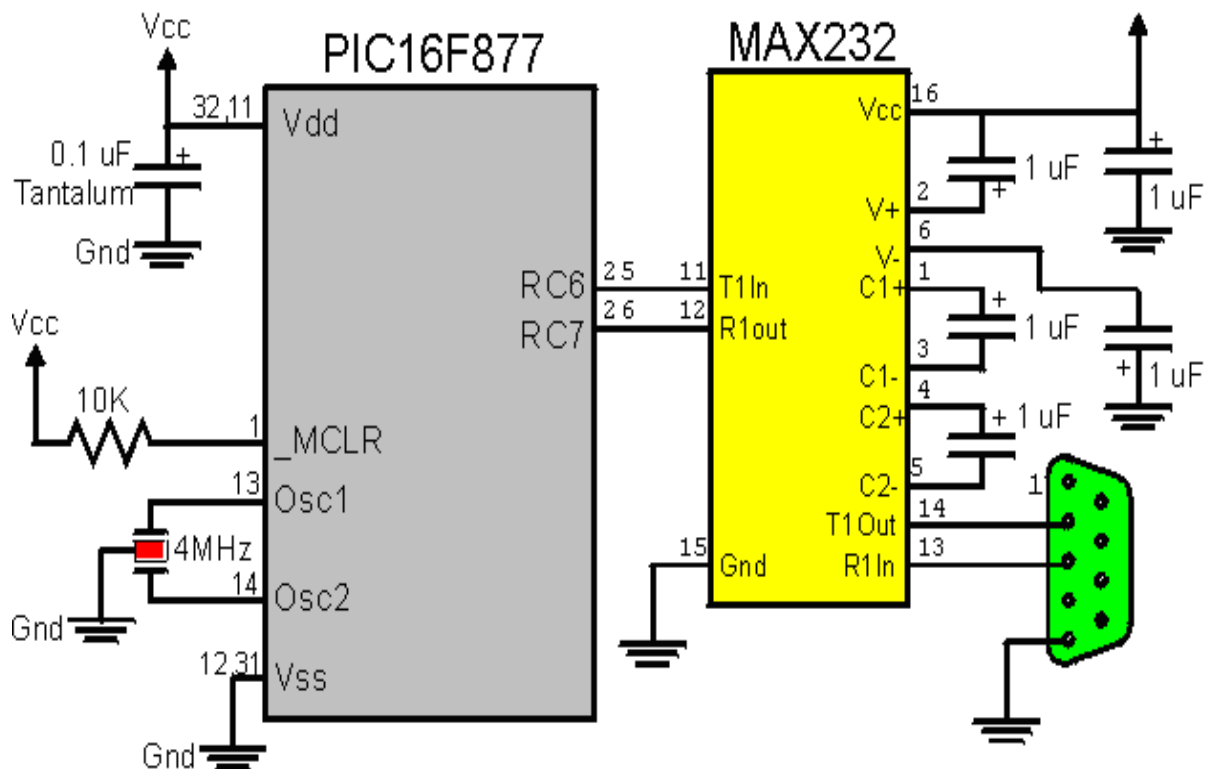


Figure 29 : Brochage du MAX232 Avec le PIC et le DB9

Concernant les condensateurs, leur valeur n'est pas critique (de 1 à 22  $\mu\text{F}$ ), mais ils sont polarisés. Les numéros des pins sont ceux correspondants à la fiche DB9 du PC

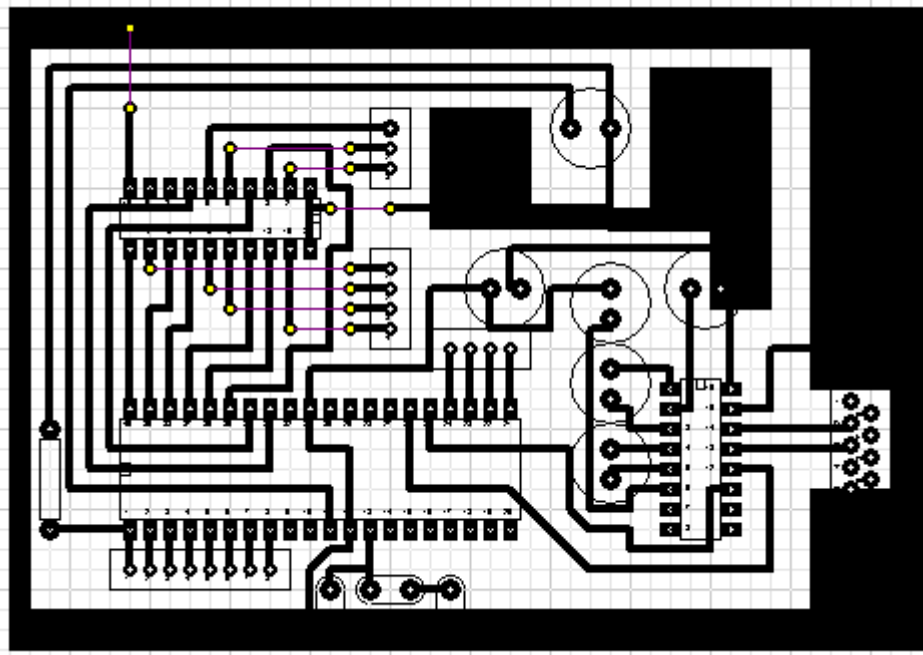


Figure 30 : Typon de l'interface (face composant)

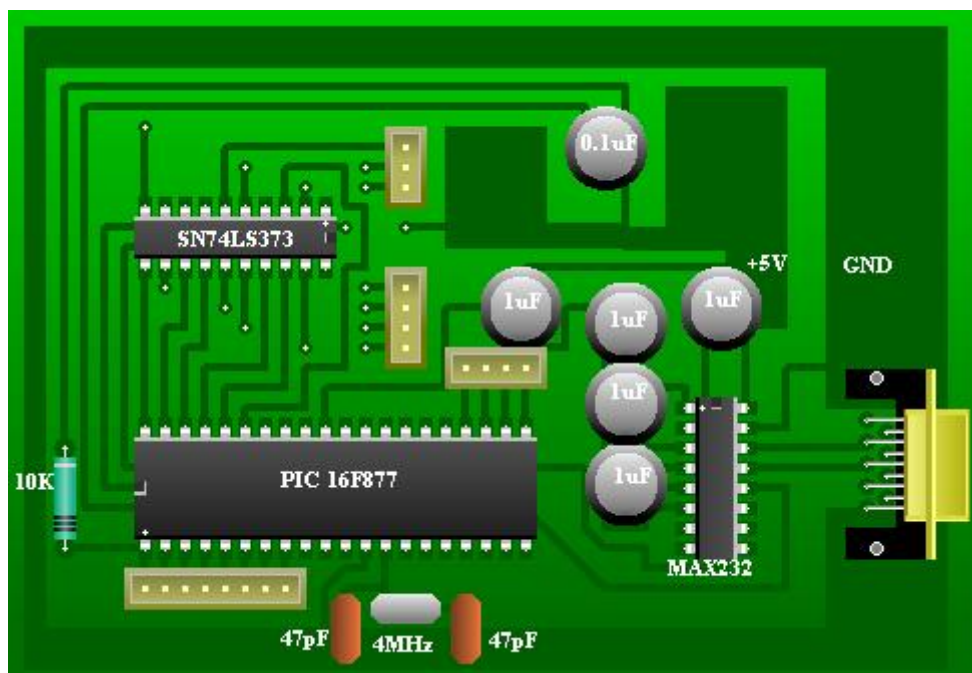


Figure 31: Carte de l'interface

### **III. 5 LA PROGRAMMATION DU PIC :**

La programmation du PIC consiste à entrer le programme dans sa mémoire flash, et des données dans sa mémoire EEPROM. Pour programmer un PIC il est important de connaître sa structure interne : registres, mémoires, ports d'entrées sorties, et tous les modules (CAN, USART, TIMER..).

Pour écrire un programme, il nous faut :

- un éditeur de texte pour rédiger les programmes
- un logiciel d'assemblage pour le traduire en langage machine
- un simulateur pour mettre le programme au point.

MPLAB est un outil de développement spécifique aux microcontrôleurs PIC.

Il permet :

- la rédaction du fichier source en langage assembleur (fichier.ASM)
- sa transformation en fichier objet (fichier.HEX) prêt à être chargé dans la mémoire programme du microcontrôleur
- l'ensemble des fichiers nécessaires à ces opérations est regroupé dans un espace « projet » (fichier.PJT)

*Remarque :* Tous les fichiers d'un même projet doivent porter exactement le même nom que le projet lui-même, seules diffèrent les extensions.

#### **1) CREATION D'UN NOUVEAU PROJET :**

Pour créer un nouveau projet, MPLAB demande d'indiquer le répertoire où on devra sauvegarder le projet; le nom du projet avec l'extension .PJT. Ensuite, le nom du fichier objet qui sera créé avec l'extension .HEX.

#### **2) ECRITURE DU PROGRAMME SOURCE :**

On écrit le programme en langage assembleur. Le PIC 16F877 possède 35 instructions. Elles sont toutes décrites dans la datasheet dans la section « instruction set summary ».

Pour la plupart des instructions, la structure peut s'apparenter à : opération – source – destination, avec la notation suivante pour les opérandes :

- F : signifie que l'on travaille avec des valeurs stockées dans la mémoire.
- W : signifie que l'on travaille par le registre de travail W.



- L : signifie que l'on utilise une valeur littérale, c'est-à-dire écrite dans le programme. Les valeurs littérales peuvent être rentrées sous la forme : décimale, hexadécimale ou binaire.
- B : signifie que le type de l'opération est « Bit à bit » ce qui signifie que l'on peut intervenir que sur certains bits d'un octet.

Une fois écrit, on sauvegarde le fichier dans le même répertoire que celui du projet (même nom mais extension .ASM).

### 3) CREATION DU PROGRAMME OBJET :

Il s'agit de transformer ce fichier .ASM en fichier.HEX directement exploitable par le programmeur de PIC. Pour ce faire on utilise le « *Make project* » du MPLAB.

### 4) CORRECTION DES ERREURS :

MPLAB détecte automatiquement les erreurs. Il affiche à la fin du transformation du fichier.ASM en fichier.HEX le message suivant : « *build failed* ». Après avoir corrigé les erreurs, on recompile en cliquant sur « *Make project* » et recommence jusqu'à obtenir le message « *build completed successfully* ».

### 5) SIMULATION :

MPSAM fournit une possibilité de simulation du programme aider la correction. La simulation permet de suivre le cheminement du programme, instruction par instruction. On passe en mode simulation par *Options – Development Mode – MPLAB – SIM Simulator*. Dans la simulation, on peut:

- voir le contenu des mémoires et registres après chaque exécution d'instruction,
- voir la valeur des variables
- modifier le contenu d'une mémoire

## 6) **FICHIERS CREES :**

MPLAB crée 5 fichiers différents :

- Le fichier Source (.ASM) : contient le programme tel qu'il a été tapé sur un traitement de texte.
- Le fichier liste (.LST) : fabriqué par l'assembleur et contient le source et le code machine correspondant.
- Le fichier objet (.COD) : contient le travail d'assemblage.
- Le fichier hexadécimal (.HEX) : contient le travail d'assemblage sous forme qui sera gravée dans le microcontrôleur.
- Le fichier des erreurs (.ERR) : contient, éventuellement les erreurs repérées dans le source par l'assembleur.

## 7) **LE PROGRAMMATEUR :**

Le laboratoire d'Instrumentation de l'IOGA a décidé de réaliser un programmeur de microcontrôleur à cause des travaux de recherche en cours qui utilisent essentiellement le PIC. Le schéma de montage ainsi que le typon sont téléchargeable gratuitement sur le site Internet. C'est un programmeur de PIC et d'EEPROM par port série RS232. Il est capable de supporter plusieurs types de microcontrôleur tels que la famille 16fxx, 12fxx, 12C5xx, 24LCxx, etc..., et permet en outre la programmation en ICSP. Pour mettre en œuvre le circuit, il suffit de la réaliser dont le schéma de montage est dans *la figure 24*.

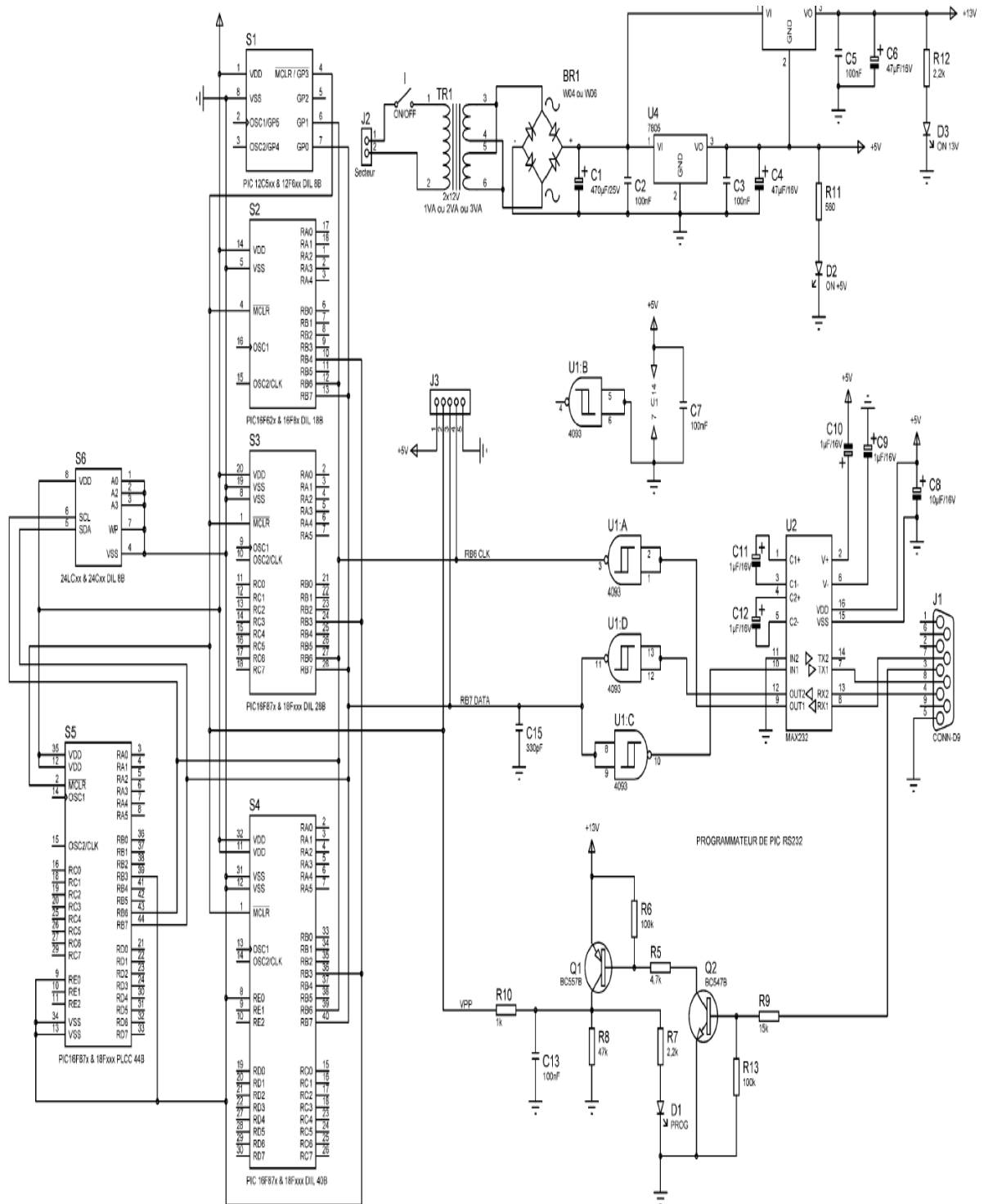


Figure 32: schéma de montage du programmeur

### Branchements :

La programmation se fait en série .Le PIC étant alimenté sous 5V, et la procédure est la suivante :

- RB6 et RB7 sont mises à 0
- La tension sur MCLR est portée à 13V
- RB6 devient alors une entrée horloge de synchronisation
- RB7 devient une entrée data.

### 8) ICPROG :

C'est un programme qui nous permet de transférer le fichier.HEX vers le PIC. Il est compatible avec la plupart des systèmes d'exploitation. Disponible gratuitement sur le site <http://www.ic-prog.com>. Il est nécessaire de configurer l'ICPROG selon le type du programmeur. La figure ci-dessous montre un aspect général de l'ICPROG.

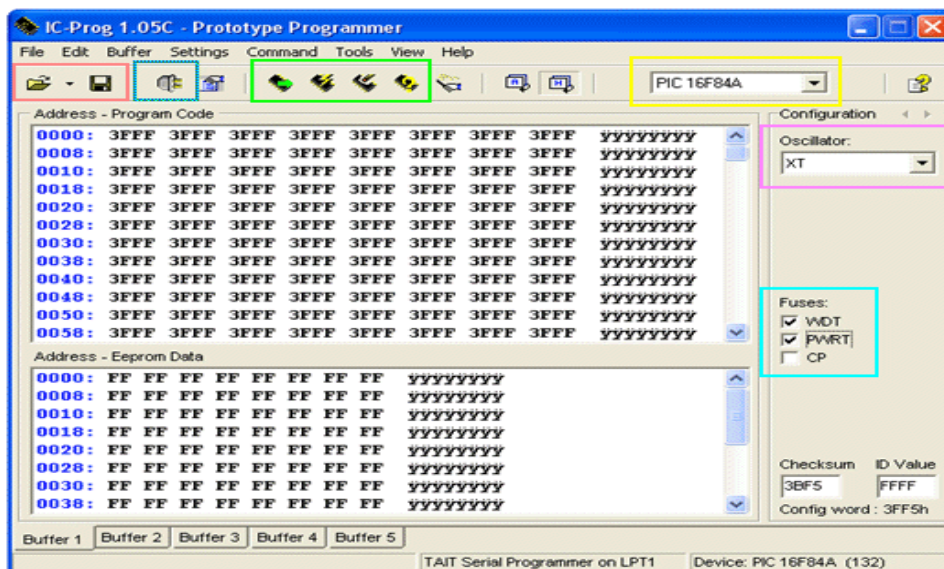


Figure 33 : Interface de l'ICPROG

Encadré en rouge : Changement ou sauvegarde d'un fichier au format HEX ou BIN

Encadré en bleu-vert : la configuration du hardware

Encadré en vert clair : les principales fonctions du logiciel : Lecture, programmation, effacement, vérification du contenu mémoire

Encadré en jaune : choix du microcontrôleur ou mémoire EEPROM

Encadré en violet : choix du type d'oscillateur (XT, HS, RC, LP)

Encadré en bleu clair : Positionnement des fusibles

**Configuration de l'ICPROG :**

Depuis le menu « settings » puis « hardware », on sélectionne selon le programmeur une des options ci-dessous :

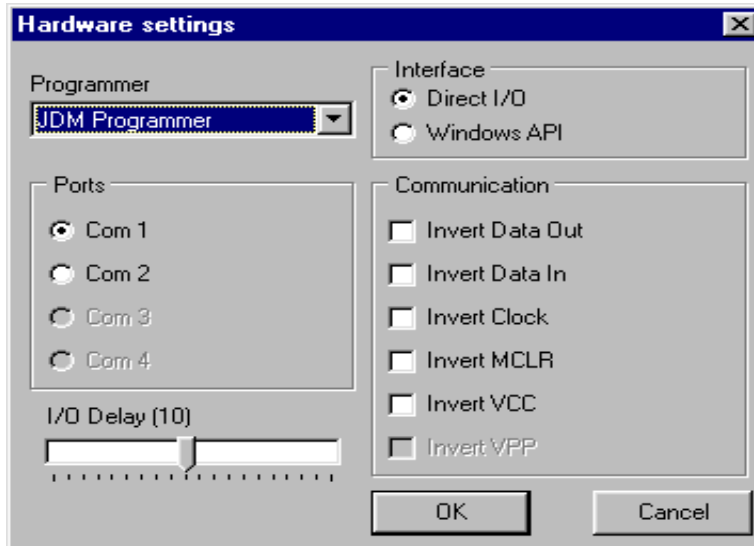


Figure 34: Configuration du Programmeur

Ensuite, on choisie le composant qui nous intéresse (dans notre cas le PIC16f877).

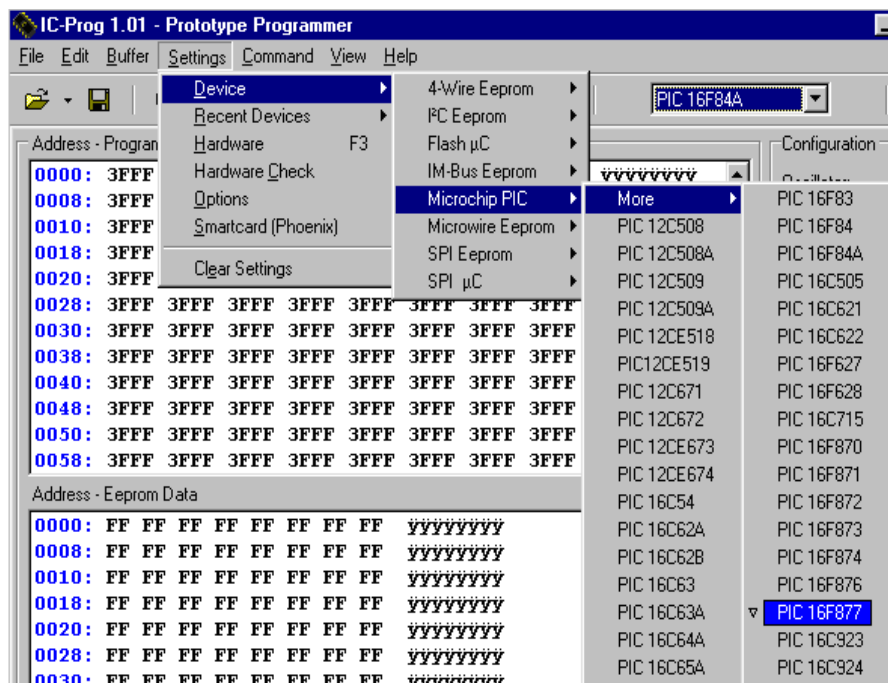


Figure 35: choix du composant.

Une fois le composant sélectionné, il ne nous reste plus qu'à ouvrir notre fichier et à le transférer en cliquant dans le menu « file » puis sélectionner le fichier compilé (\*.hex) avec « Open File ». A ce stade, on peut encore modifier certaines options qui nous sont proposées avec les cases à cocher :

- **CP (code protect)** : si cette option est cochée, alors il ne nous sera plus possible de relire le programme de notre PIC. Il est possible bien sûr de le programmer avec un nouveau programme et nous aurons alors la possibilité de retirer le CP.
- **WDT (watchdog timer)** : le watchdog vérifie que notre programme ne s'est pas « perdu » dans une boucle sans fin. L'activation de cette option nous incite à mettre l'instruction de raz du watchdog, c'est-à-dire mettre l'instruction clrwdt dans notre programme sinon il y aura un « reboot » à l'adresse 0 dès que la période watchdog sera atteinte, en clair notre programme va sans cesse repartir de 0.
- **PWRT (Power Up Timer)** : Si elle est cochée, un timer interne provoquera un arrêt typique du PIC pendant 72ms après la mise sous tension (entre 1,2V et 1,7V), cette option permet entre autre un démarrage stable de l'horloge principale du PIC.

**L'oscillateur** : il y a 4 modes dans cette option : LP (low power crystal), XT, HS (High speed), RC (Resistor/capacitor). Nous avons choisi dans cette option le mode XT car elle nous permet d'utiliser un quartz de 4MHz.

Le transfère du fichier peut commencer maintenant à l'aide du commande « command » puis « programm all »

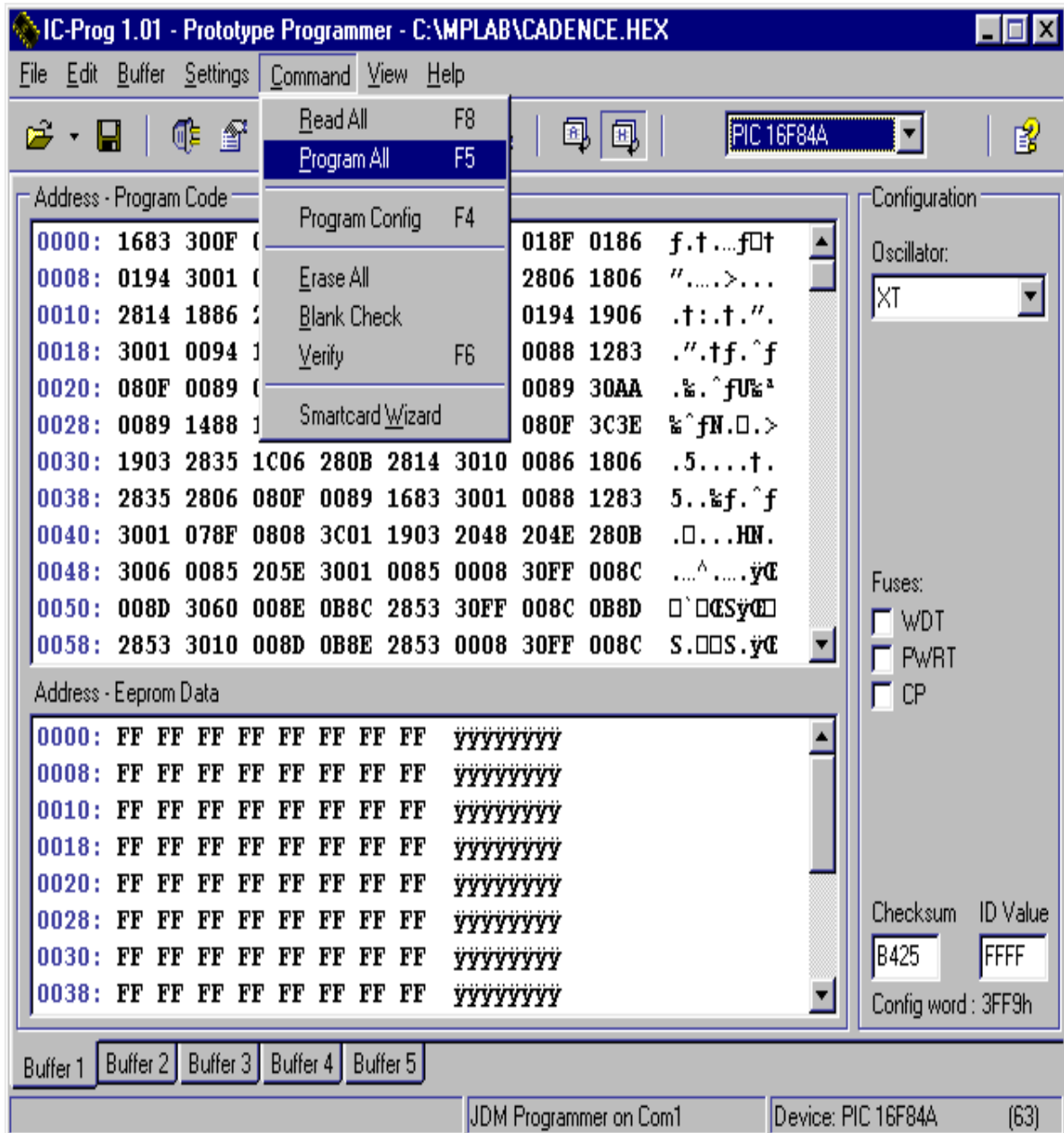


Figure 36 : transfère du fichier

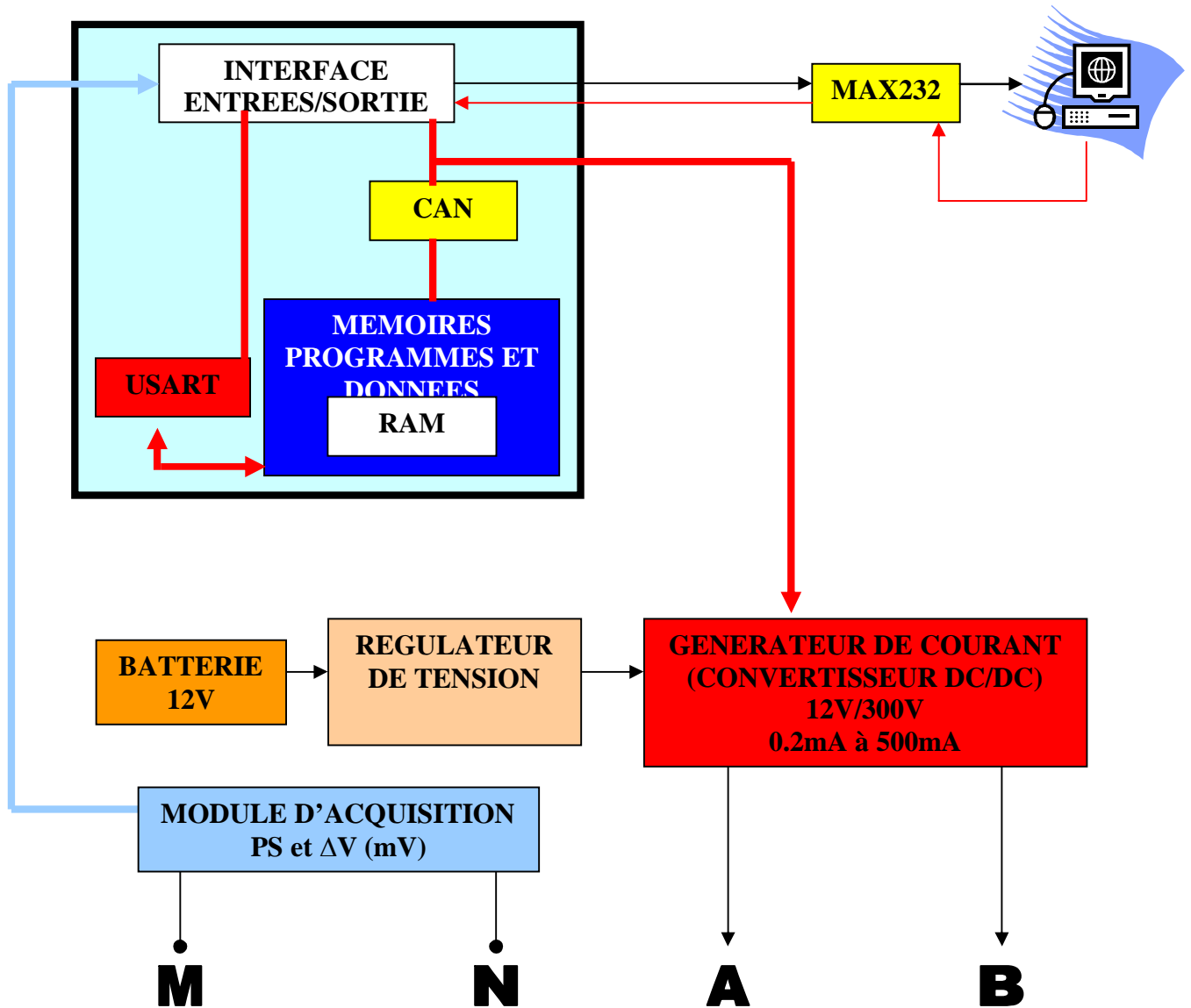


Figure 37 : Schéma synoptique du résistivimètre



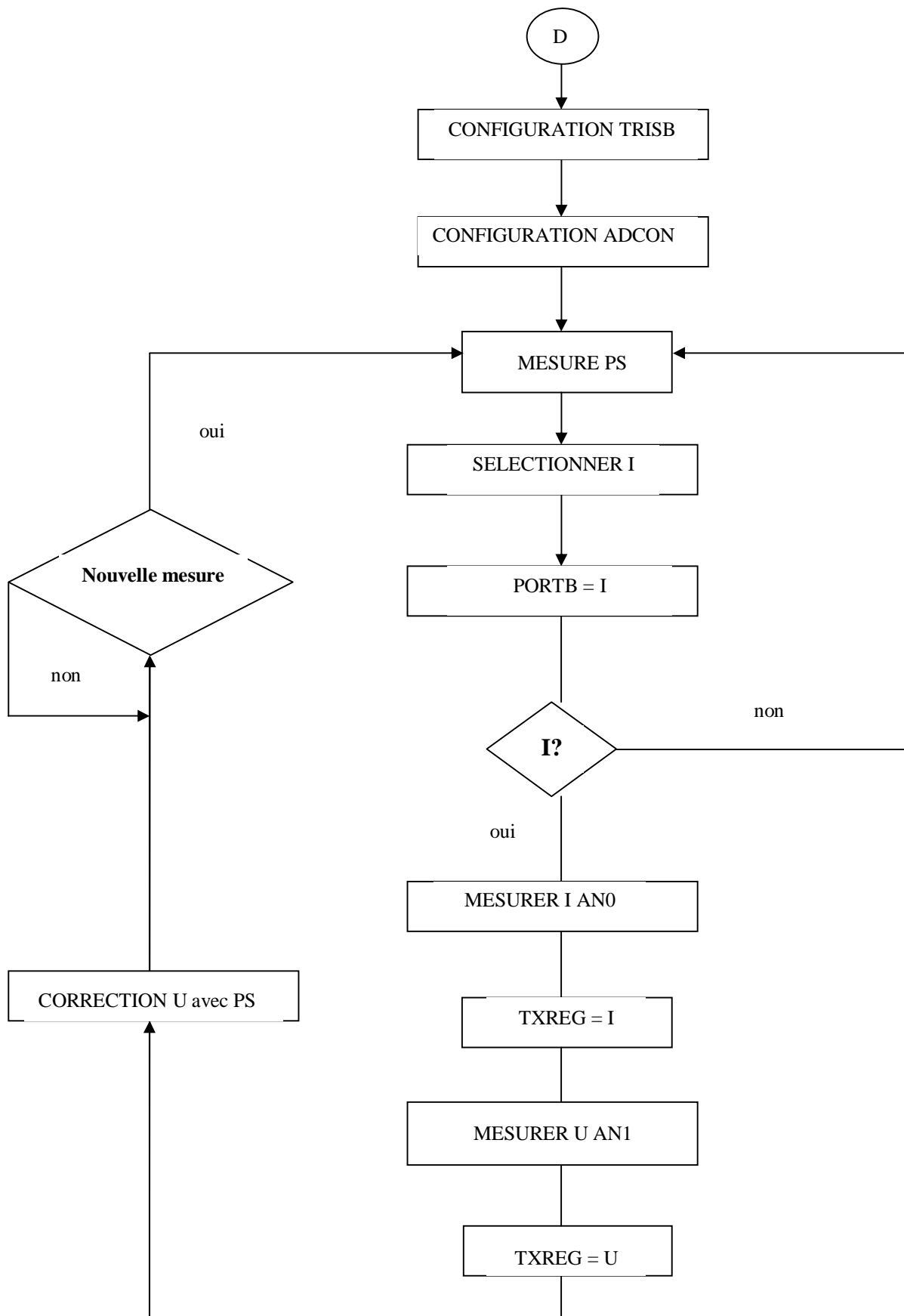


Figure 38 : Organigramme pour la mesure

### Configuration du port :

Les ports B du PIC seront configurés comme ports de sélection de courant, une impulsion venant de l'un du port B sera envoyée au LATCH (SN74LS373) qui actionnera le générateur de courant avec l'intensité de courant choisie :

COURANT à INJECTER	PORTB
0.2mA	b'10000000'
1mA	b'01000000'
10mA	b'00100000'
100mA	b'00010000'
200mA	b'00001000'
500mA	b'00000100'

Tableau 17 : Configuration du port B pour les choix du courant

Les ports AN0 et AN1 sont configurés comme ports d'entrées des tensions analogiques :

AN0 : pour l'intensité de courant I

AN1 : pour la PS et le ddp entre MN

Ces tensions analogiques seront converties par le CAN avant d'être transférées au PC par l'intermédiaire du module USART. Voici une partie du programme d'envoi.

```
ENVOI_data    macro
                BANKI                ; passer en page 1
                bsf TXSTA, TXEN      ; pour lancer la transmission
                BANK0                ; passer en page0
                movf DATA_H, 0      ; copier contenu DATA_H dans w
                movwf TXREG          ; charger DATA_H dans TXREG
                BANK1                ; passer dans la page1
at1_1         btfs, TXSTA, TRMT      ; s'assurer que TRMT = 1 avant de transmettre
                goto at1_1          ; attendre si non
                BANK0                ; passer en page 0
                movf DATA_B, 0      ; copier contenu DATA_B dans w
                movwf TXREG          ; charger DATA_H dans TXREG
                BANK1                ; passer dans la page1
At1_2         btfs, TXSTA, TRMT      ; s'assurer que TXIF = 1 avant de transmettre
                goto at1_2          ; attendre si non
                bcf TXSTA, TXEN      ; arrêter la transmission
                endm
```

## CHAPITRE IV : RESULTATS ET DISCUSSIONS

### RESULTATS:

Deux sortes d'essai ont été réalisées ; le premier consiste à mesurer quelques résistances avec des valeurs connues ; pour ce faire, nous avons utilisé deux multimètres différents pour connaître la valeur de ces résistances ; ensuite, le résistivimètre mesure ces mêmes résistances.

La figure suivante montre le schéma synoptique pour la mesure de la résistance à trouver ( $R_3$  lors de notre essai); la résistance  $R_1$  sert à connaître la valeur du courant qui est injecté, courant qui traverse la résistance  $R_3$ . Sa valeur est de  $10\Omega$ . Connaissant la différence de potentiel au borne de la résistance  $R_3$  (M et N) dû à l'injection du courant fournit par le générateur de courant et l'intensité du courant qui le traverse, la loi d'Ohm nous donne la valeur exacte de la résistance  $R_3$ .

$$R_3 = \frac{\Delta V}{I}$$

$R_3$  : valeur de la résistance à tester [ $\Omega$ ]

$\Delta V$  : différence de potentiel aux bornes de  $R_3$  [mV]

$I$  : intensité de courant traversant  $R_1$ ,  $R_3$  [mA]

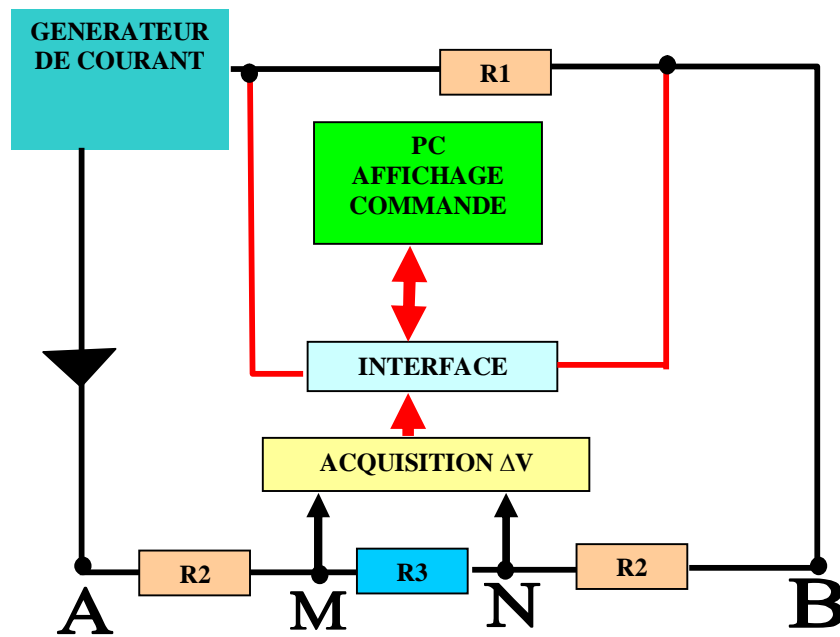


Figure 39 : Synoptique de l'acquisition pour la mesure d'une résistance

Les résultats obtenus sont assignés dans ce tableau 17:

Valeur de la résistance indiquée par son constructeur	Valeur de la résistance indiquée par le multimètre 1	Valeur de la résistance indiquée par le multimètre 2	Valeur de la résistance indiquée par le résistivimètre
1 MΩ	1.072 MΩ	1.073 MΩ	1.071 MΩ
500 KΩ	502 KΩ	502 KΩ	502 KΩ
300 KΩ	317 KΩ	316 KΩ	315 KΩ
20 KΩ	20.2 KΩ	20.1 KΩ	20.1KΩ
10Ω	10.7 Ω	10.6 Ω	10.9Ω

Tableau 18 : valeurs des résistances indiquées et mesurées

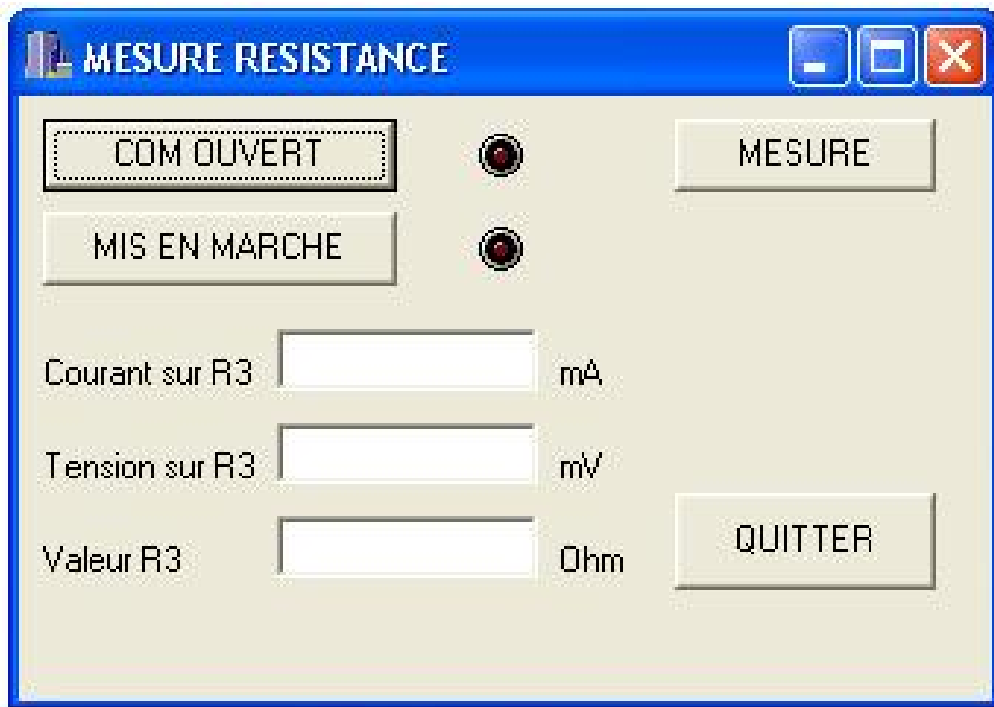


Figure 40 : Interface logiciel – Mesure de la Résistance

Une interface logicielle, réalisée avec C++ Builder, nous a permis d'essayer la communication de notre montage avec le PC, et de voir les résultats obtenus. Cf. figure 33

Le deuxième essai a été réalisé sur le terrain d'Ankatso ; pour évaluer la résistivité du terrain, nous avons utilisé le SYSCAL R2 (résistivimètre actuellement utilisé à l'IOGA). Le logiciel ISEMS1 interprète le sondage électrique ainsi mesuré. On obtient les modèles de terrains suivants :

	SYSCAL R2	<i>résistivimètre</i>
1 <sup>er</sup> Terrain / résistivité	2000Ωm	1800 Ωm
Epaisseur	3.5m	2.9m
2 <sup>ème</sup> Terrain	250 Ωm	200 Ωm

Tableau 19 : Modèle obtenu

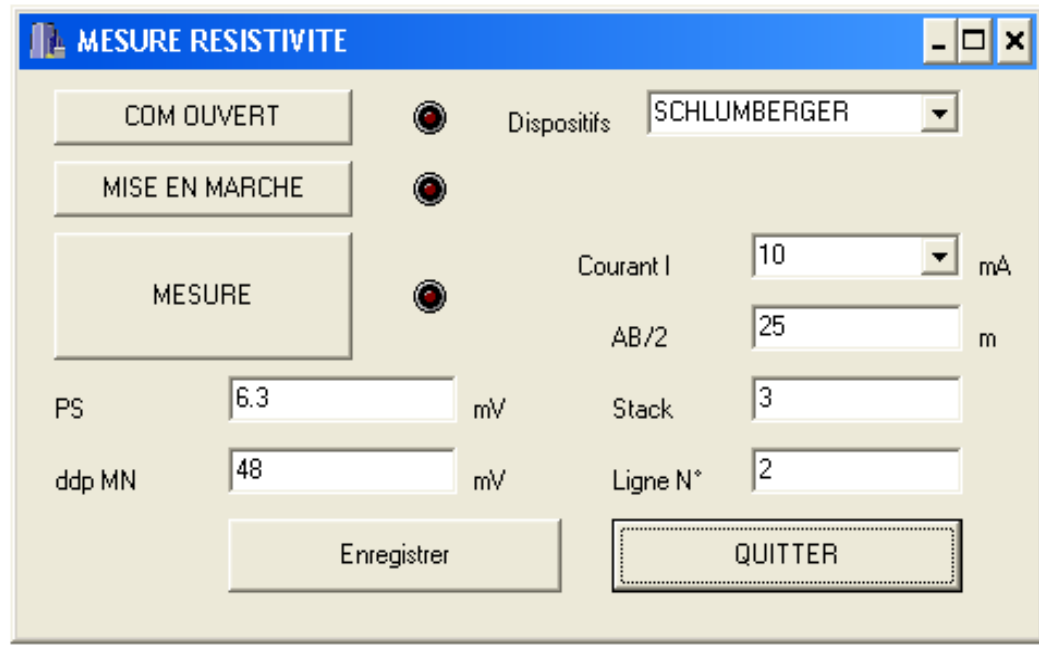


Figure 41 : Interface logiciel – Mesure de la Résistivité

## DISCUSSIONS :

Pour les mesures de la valeur des résistances, le résistivimètre conçu affiche les mêmes valeurs que les deux multimètres ; la mesure et les résultats sont fiables et ne posent aucun problème.

Concernant les mesures sur terrain, mesure de la résistivité, nous constatons une légère différence entre le SYSCAL R2 et notre résistivimètre ; celle-ci est due aux non corrections des phénomènes naturelles (induction électromagnétique, courants telluriques).

Le résistivimètre conçu pourrait être amélioré, en utilisant un convertisseur de tension à base d'une alimentation à découpage (hacheur) ; on obtient ainsi un meilleur rendement et un encombrement plus faible. L'utilisation d'un doubleur de tension nous fournit une tension de l'ordre de kilovolts pour les faibles courants (0.2mA à 10mA), ceci dans le but d'obtenir une grande profondeur d'investigation du sol à étudier. On peut aussi ajouter un module capable de détecter les phénomènes physiques perturbateurs afin de les éliminer pour ne traiter que les signaux dus à l'injection de courant.

## CONCLUSION

Ce travail nous a permis de comprendre et de réaliser le convertisseur de tension 12V de la batterie en une tension beaucoup plus élevée (300V) avec une puissance atteignant 200VA. Nous sommes limités en puissance par l'inexistence des composants électroniques à Madagascar.

Par ailleurs, une partie du temps de ce travail a été consacré à la compréhension de la structure des microcontrôleurs et l'apprentissage du logiciel MPLAB. Nous avons appris alors à programmer un microcontrôleur et le tester. Ainsi, nous avons pu monter quelques applications du microcontrôleur et les réaliser au laboratoire.

Le domaine qui reste à explorer serait d'ajouter un module capable de réaliser un panneau électrique pour que l'utilisateur puisse connaître la structure en 2D du sous sol.



## ANNEXES I

### LA NORME RS232

Cette norme PS 232, appelé aussi CCITT V24 ou encore V24 « tout court », est d'origine américaine et définit deux choses :

- Les niveaux électriques des signaux utilisés pour la transmission
- Un certain nombre de lignes, autres que les lignes d'émission et de réception de données, ayant des fonctions de contrôle.

- Au point de vue niveau, cette norme est très simple : tout signal de niveau compris entre +3 et +25 volts est considéré comme étant au niveau logique A alors que tout signal compris entre -3 et -25 volts est considéré comme étant au niveau logique B. A et B sont quelconques et peuvent être 0 ou 1 selon que l'on travaille en logique positive ou négative.
- Les lignes de contrôle

**RTS** est une ligne de demande d'émission ; elle passe au niveau haut lorsque le terminal veut envoyer des données.

**CTS** est une ligne d'invitation à émettre; elle passe au niveau haut lorsque l'ordinateur attend des données du terminal

**DSR** indique, lorsqu'elle est au niveau haut, que l'ordinateur est prêt.

**DTR** indique, lorsqu'elle est au niveau haut, que le terminal est prêt.

**DCD** n'est utilisée en principe que sur les modems. C'est la ligne de détection de porteuse qui passe au niveau haut en présence de cette dernière.

**TC** et **RC** sont des lignes d'horloges, respectivement d'émission et de réception, pour piloter un équipement par l'horloge de l'autre. Leur utilisation avec des liaisons séries asynchrones est exceptionnelle.

**RI** est encore une ligne utilisée seulement par les modems ; c'est l'indication de sonnerie. Son emploi est peu fréquent sur une liaison série RS 232 classique.

## **ANNEXES II**

### **L'USART**

**EMISSION:** Pin PC6 = TX DATA.

L'émission est autorisée par la mise à "1" du bit 5 de TXSTA: TXEN = 1. Le DATA à transmettre est mis dans le registre TXREG en h'19' page 0. Ce registre prévient qu'il est vide en mettant le flag TXIF à "1" (bit 4 de PIR1).

Ce flag passe à "0" dès que l'on charge un octet dans le registre TXREG. Il repasse à "1" par Hard quand le registre est vidé par transfert dans le registre de sérialisation : TSR. Ce registre n'est pas accessible par l'utilisateur, il n'a pas d'adresse.

Si on charge alors un 2<sup>ème</sup> octet dans le registre TXREG le flag TXIF va passer à "0" et y rester tant que le registre TSR n'aura pas complètement sérialisé l'octet précédent à transmettre. Dès que le STOP de l'octet précédent a été transmis, le registre TXREG est transféré dans TSR et le flag TXIF repasse à "1" signalant ainsi que le registre de transmission TXREG est vide et peut donc recevoir un nouvel octet à transmettre.

Le bit TRMT (bit 1 de TXSTA) informe sur l'état du registre TSR. Quand le registre TSR n'a pas fini de sérialisé, TRMT=0. Ce flag repasse à "1" quand le registre est vide, c'est à dire quand le stop a été émis.

Le flag TXIF permet aussi de générer une interruption, à condition qu'elle soit autorisée par mise à "1" du bit 4 de PIE1 : TXIE = 1. Il faut dans ce cas autoriser les interruptions des périphériques par mise à "1" du bit 6 de INTCON: PEIE = 1, et par la mise à "1" du bit 7 : GIE = 1

#### **PROCEDURE POUR EMETTRE :**

- Initialiser SPBRG pour la vitesse désirée et choix pour BRGH.
- Autoriser mode Asynchrone : SYN = 0 et SPEN = 1.
- Eventuellement faire TX9 = 1 si une émission sur 9 bits est désirée.
- Autoriser l'émission par TXEN = 1.
- Si une transmission 9 bits a été choisie, mettre le 9eme bit dans TX9D.
- Mettre l'octet à transmettre dans TXREG.
- Avant de remettre l'octet suivant à transmettre dans TXREG, il faut tester le flag TXIF qui est à "0" si le registre n'est pas disponible. Dès que le registre est vide ce flag passe à "1" et on peut alors charger TXREG par l'octet à transmettre.
- Pour savoir si le dernier octet a été émis, il suffit de tester le flag TRMT qui signale par son passage à "1" que le dernier bit du dernier octet et son STOP ont bien été sérialisés.
- On peut alors stopper le module émission de l'USART par TXEN=0.

## **RECEPTION: Pin PC7 = RX DATA.**

La réception est autorisée par la mise à "1" du bit 4 de RCSTA: CREN = 1.

Le DATA reçu est mis dans le registre RCREG en h'1A' page 0. Ce registre prévient qu'il est plein en mettant le flag RCIF à "1" (bit 5 de PIR1). On peut autoriser la génération d'une interruption quand RCIF = 1, c'est à dire quand une donnée valide est disponible dans RCREG par mise à "1" du bit 5 de PIE1 : RCIE = 1. Le flag RCIF repasse à "0" par hard quand on vide le registre RCREG par sa lecture.

Si le STOP d'un 2<sup>ème</sup> octet survient alors que le registre RCREG n'a pas été vidé, une erreur OVERRUN se produit. Elle est signalée par le passage à "1" du bit 1 de RCSTA : OERR=1. L'octet dans le registre de dé srialisation est alors perdu.

Le bit d'erreur OERR doit être remis à zéro par soft. Pour cela il faut stopper la réception par CREN=0 puis remettre en service la réception par CREN=1.

En fait le registre RCREG est un double registre FIFO. On peut donc recevoir 2 octets et ne pas les lire avant qu'un 3eme octet ne fasse un OVERRUN.

On doit alors lire deux fois RCREG pour le vider les 2 octets reçus qui sont dans le FIFO.

Si un STOP est trouvé à "0" alors une ERROR FRAMING est générée par mis à "1" du bit 2 de RCSTA: FERR=1

## **PROCEDURE POUR RECEVOIR :**

- Initialiser SPBRG pour la vitesse désirée et choix pour BRGH.
- Autoriser mode Asynchrone : SYN = 0 et SPEN = 1.
- Eventuellement faire RX9 = 1 si une réception sur 9 bits est désirée.
- Eventuellement faire RCIE = 1 si une réception par interruption est désirée.
- Autoriser l'émission par RCEN = 1.
- Test du Flag RCIF (ou attente IT) pour savoir si un octet a été reçu.
- Lire éventuellement le 9eme bit de Data dans RCSTA pour tester la parité.
- Lire les bits FERR et OERR pour déterminer les erreurs éventuelles.
- Si une erreur est survenue il faut faire CREN=0 puis CREN=1 pour RAZ.
- Lecture du registre RCREG pour récupérer l'octet reçu.

## **MODE 9 bits avec DETECTION D'ADRESSE :**

- Faire toutes les initialisations précédentes: SPBRG, BRGH, SYN, SPEN.
- Faire RX9=1 ADDEN=1 et CREN=1
- Dès que registre RCREG est signalé plein, il s'agit de l'adresse (9eme bit =1).
- Le lire et comparer avec l'adresse déterminée. Si OK faire ADDEN=0 pour recevoir maintenant tous les octets, même si le 9eme bit n'est pas à "1". Si pas OK rester avec ADDEN=1 pour ne recevoir que les octets filtrés d'adresse.

## BIBLIOGRAPHIE

1. A. Oumnad. Microcontrôleur famille Mid-Range le PIC16f876/877, Microchip
2. ABEM TERRAMETER, Manuel d'utilisation
3. DATA B, 1987. La bible du PC
4. DATA BOOK Electronic Information Series, Digital Integrated Circuits
5. DATA BOOK, Electronic Information Series, Interface Integrated Circuits
6. DECLERQ M. *Circuits et systèmes électroniques*, Ecole polytechnique fédérale de Lausanne, Département d'électricité, Laboratoire de l'électronique générale.
7. E.GARCIA ; 2002 ; ARCHITECTURE DES ORDINATEURS Cours 5–Hardware,-IUT–GTR, Montbéliard LIFC (Laboratoire Informatique de l'Université de France – Comté)
8. INTEGRATED CIRCUITS DATASHEET (TTL-CMOS), Philips semiconductor
9. IRIS INSTRUMENT; 1999. Résistivimètre (V.9.8++) [CDROM].
10. LEIBSON S.; 1986, traduit de l'Américain par Michel Jacoub. *Manuel d'interface*, Mc Grawhill.
11. Mémento PIC Version 01
12. PADIOLLEAU Jean-Yves, Novembre 2000. Le microcontrôleur, rapport de stage.
13. R. Zaks – A.Lesea – 1977 ; TECHNIQUES D'INTERFACE AUX MICROPROCESSEURS ;
14. RAKOTONDRAFARA Hobiniaina ; Décembre 2004. Conception et réalisation d'une interface logicielle résistivimètre électrique, DEA.
15. RAMBOLAMANANA Mamiharijao, Décembre 2004, Conception et réalisation d'une interface matérielle pour résistivimètre électrique ; DEA
16. S. TISSERANT ; ARCHITECTURE DES ORDINATEURS ; Université de la Méditerranée – Ecole Supérieure d'Ingénieurs de Luminy – Département d'Informatique
17. Stiffler A. Kent; 1992. Design with microprocessors for mechanical engineers, Highstown, NJ, USA.
18. TEXAS Instruments Incorporated; 1984. Linear Circuits Data Book, European Edition.
19. THE TTL DATA BOOK; Advanced Low-Power Schottky; Advanced Schottky TEXAS INSTRUMENTS
20. Thomas LUSSEAU; 2005. MJACQUES Dubois, MICHEL Diement; 1997. Géophysique, Paris, Masson.

## WEBOGRAPHIE

Novembre 2006 au Mars 2007

- <http://www.ac-bordeaux.fr>
- <http://fr.wikipedia.org>
- <http://madchat.org/coding/electro>
- <http://www.univ-savoie.fr>
- <http://www.epsic.ch>
- <http://www.univ-savoie.fr>
- <http://www.microchip.com>
- <http://www.abcelectronique.com/bigonoff>
- [http://www.datasheetcatalog.com/datasheets\\_pdf](http://www.datasheetcatalog.com/datasheets_pdf)
- [http://www.datasheetcatalog.net/datasheets\\_pdf](http://www.datasheetcatalog.net/datasheets_pdf)

Rapport-Gratuit.com

**RESUME :**

Pour mesurer la résistivité du sous-sol, les géophysiciens font appel à un appareil de mesure appelé : « résistivimètre » ; pour générer le courant dans le sol, le résistivimètre a besoin d'un convertisseur, qui transformera le 12V de la batterie par 300V. Le courant injecté varie de 0.2mA à 500mA selon le besoin. La mesure est faite entièrement et automatiquement à travers le contrôle du microcontrôleur PIC16F877 : correction automatique de la polarisation spontanée (Ps), accumulation automatique de mesures pour l'amélioration du rapport signal/bruit, affichage de message d'erreurs en cas de problèmes de mesure. Toutes les valeurs ainsi obtenues seront converties à l'aide du convertisseur analogique numérique du PIC pour être stockées dans un fichier du PC. Le module USART contenu dans le PIC permet de transmettre ou de recevoir les données ou les commandes, dans le PC.

**Mots clés** : Résistivimètre, générateur de courant, microcontrôleur, USART, convertisseur.

**ABSTRACT:**

To measure the basement's resistivity, the Geophysicians call on a measurement apparatus which is called "Resistivimeter". To generate current under the ground, the resistivimeter needs a converter. This one transforms the 12V of the battery into 300V. The injected current varies from 0.2mA to 500mA according to the need. All measurements are automatics with the control the microcontroller PIC16F877: Auto correction of spontaneous polarization (PS), automatic accumulation of measurements to improve the report of the signal/noise, posting up of error messages if any problems of measurement appear.

All of values obtained will be converted by PIC's numerical analogical converter before being stocked in the PC's file. The USART module in the PIC allows the transmission and the reception of the data or the orders in the PC.

**Key words**: resistivimeter, current generator, microcontroller, USART, converter.

**Encadreur :****Pr RANAIVO-NOMENJANAHR Y Flavien****Impétrant :****Nom** : RAZAFIARISERA**Prénoms** : Ralay Tiana**Adresse** : Vontovorona**Tél** : 033 14 288 44**[rrralay@yahoo.fr](mailto:rrralay@yahoo.fr)**