



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abou Bekr Belkaid – Tlemcen

Faculté de Technologie

Département de Génie Electrique et Electronique

Filière : ELECTRONIQUE

Projet de Fin d'Etudes

Master : Electronique

Option : Instrumentation Electronique

Intitulé : Etude et Réalisation d'une carte d'acquisition de signaux
physiologiques, interface USB

Présenté par :

LARIF Walid

KHATER Mustafa

Jury :

Président : BENABDELLAH Mohamed

Encadreur : NEMMICHE Ahmed

Examineur : BELAID Ali

Examineur : HAMDOUN AEK

Année Universitaire : 2012-2013

Dédicace

Je dédie ce mémoire :

A mes très chers parents pour leurs dévouements, leurs amours, leurs sacrifices et leurs encouragements et sans qui je ne serais pas là aujourd'hui. Ce travail soit pour eux, un faible témoignage de ma profonde affection de tendresse.

A ma sœur Imane

A ma sœur Wiame

A ma petite sœur Nourhane

A ma future femme

A tous ma famille.

A celle qui m'aime.

A Tous mes amis (es) (Amine, Omar, Abdellatif...)

Et à toute ma promotion

A tous mes enseignants

A tous ceux qui j'aime et j'estime .Et à vous aussi.

LARIF Walid

Dédicace

Je dédie ce mémoire :

A mes très chers parents pour leurs dévouements, leurs amours, leurs sacrifices et leurs encouragements et sans qui je ne serais pas là aujourd'hui. Ce travail soit pour eux, un faible témoignage de ma profonde affection de tendresse.

A mon frère Mourad

A ma sœur Amel

A mon frère Amine

A ma nièce Malek

A ma future Femme ;

A tous ma famille.

A Tous mes amis (es) (Abdellatif, Amine, Omar...)

Et à toute ma promotion

A tous mes enseignants

A tous ceux qui j'aime et j'estime .Et à vous aussi.

KHATER Mustafa

REMERCIEMENTS

Nous remercions Allah le tout puissant qui nous a donné durant toutes ces années, la santé, le courage et la foi en nous même pour pouvoir avancer et mener nos études à leurs termes.

Nous ne saurions, réellement, trouver les expressions éloquentes que mérite notre encadreur Monsieur NEMMICHE Ahmed enseignant à l'université Abou Bekr Belkaid pour nous avoir proposé le sujet et offert un cadre de travail agréable et un encadrement de qualité.

Que Monsieur BENABDELLAH Mohamed enseignant à l'université Abou Bekr Belkaid, veuille croire à nos profondes reconnaissances et nos sincères gratitude d'avoir accepté de présider le jury.

Nous adressons nos remerciements aux membres du jury BELAID Ali enseignants à l'université Abou Bekr Belkaid, et Monsieur HAMDOUN Aek enseignants à l'université Abou Bekr Belkaid, qui nous ont fait l'honneur d'évaluer, d'examiner et d'enrichir notre modeste travail.

Nous tenons à remercier également tout le personnel du laboratoire d'électronique et de la Bibliothèque pour leurs gentillesse et leurs disponibilités.

Enfin, nous exprimons nos remerciements à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce modeste travail.

Merci.

SOMMAIRE

INTRODUCTION GENERALE.....	8
Chapitre 1 : PIC 18F2550.....	10
1. Caractéristiques du PIC 18F2550.....	11
2. Fonctionnement du PIC 18F2550.....	12
2.1 Les entrées/sorties	12
2.2 La tension de référence.....	13
2.3 Les comparateurs.....	13
2.4 La PWM	15
Chapitre 2 : PROTOCOLE USB.....	17
1. Introduction	18
2. Naissance de l'USB.....	18
3. Composition du câble USB	19
4. Le bus USB	20
4.1 Principe du bus USB :	20
4.2 Topologie du bus USB	21
5. Les types de paquet USB :.....	21
6. Structure des paquets :.....	22
6.1 Les paquets Jetons :	22
6.2 Les paquets Data :	23
6.3 Les paquets Handshake :	23
6.4 Les paquets SOF :.....	23
7. Explication des différents champs des paquets d'une trame USB :	23
7.1 Synchronisation :	23
7.2 Paquet ID :	23
7.3 Paquet adresse :	24
7.4 Paquet ENDP :	24
7.5 Paquet CRC :	24
7.6 Paquet EOP :	24
8. les types de transferts	24
9. Temps de connexion et de déconnexion :.....	24
10. Les descripteurs :.....	25
10.1 Device descriptor (descripteur d'appareil) :	26

10.2 Configuration descriptor (descripteur de configuration) :.....	27
10.3 Interface descriptor (descripteur d'interface) :	28
10.4 Endpoint descriptor (descripteur de point terminal) :.....	29
11. La classe Human Interface Device (HID) :	30
12. La gestion du dispositif USB/HID :	31
13. Conclusion :.....	31
Chapitre 3 : Programmation et Interface	32
1. Introduction	33
2. Programme du pic 18F2550	33
2.1 Présentation de l'environnement de programmation.....	33
2.2 Pour débiter avec MPLAB	33
2.3 Caractéristiques	33
3. Le fonctionnement général du programme.....	33
4. La configuration des registres	36
a. La conversion analogique / numérique.....	36
b. L'émission et réception des données :.....	36
5. L'interface sur le PC.....	37
a. Partie paramètre.....	37
b. Partie valeur.....	37
c. Partie traceur.....	38
6. Organigramme :.....	38
Chapitre 4 : L'étude Pratique	39
1. Introduction	40
2. Le traitement numérique du signal et la communication avec le PC	40
2.1 Le circuit réalisé :	40
2.2 Le Quartz.....	41
2.3 Le branchement USB	42
2.4 Le circuit de conversion	42
2.5 Le circuit de diviseur de tension.....	44
3. Les étapes de réalisation pratique de la carte	44
3.1 Le typon :.....	44
3.2 L'utilisation pratique du circuit.....	45
4. Le traçage des signaux :	48
5. Les avantages et les inconvénients du projet.....	50
6. Conclusion.....	51

Conclusion générale	52
Liste des composants.....	53
Annexe	54

Introduction Générale

INTRODUCTION GENERALE

Dans le domaine d'électronique les instruments de mesure sont très chers et parfois introuvables. En contrepartie le matériel informatique est plutôt abordable et d'un prix acceptable. Partant de ce constat nous avons tenté de combler un tant soit peu cette lacune concernant les appareils d'acquisition de signaux physiologiques. Notre réalisation se veut un instrument de capture et de traitement de tout signal analogique et ce en mettant à la disposition de l'opérateur 4 canaux indépendants. Les données sont ensuite acheminées vers un PC qui se chargera des opérations nécessaires.

La réalisation électronique tourne autour du microcontrôleur 18F2550 qui représente des avantages dans le domaine d'utilisation d'USB et la communication avec le PC.

Introduction Générale

Ce rapport est une description complète des différentes étapes de notre projet de fin d'étude.

En rédigeant ce rapport nous avons voulu décrire les principales parties de notre projet dont le sujet est l'adaptation des signaux physiologiques et les visualiser sur le PC.

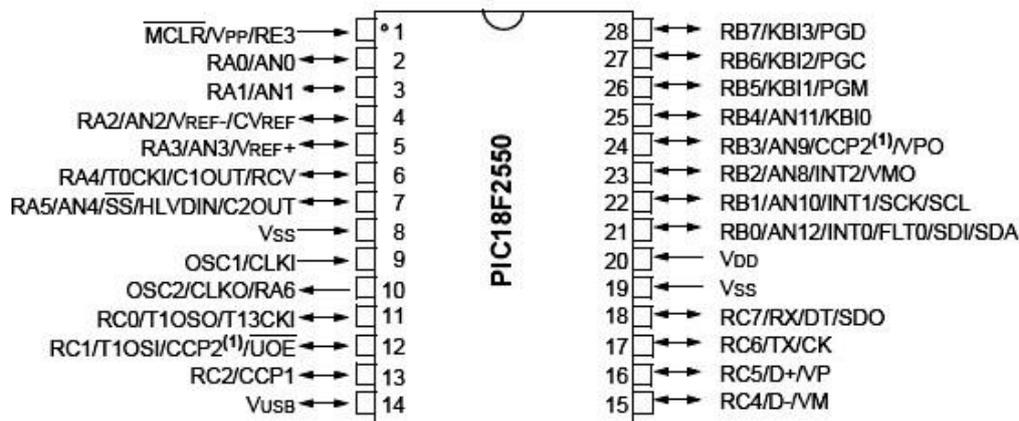
Ce projet contient quatre chapitres décrivant les différentes étapes :

- Dans le premier chapitre on fait une description générale sur les propriétés et les caractéristiques du pic 18F2550.
- Dans le deuxième chapitre on fait une description sur le protocole USB.
- Dans le troisième chapitre on va parler de la programmation et de l'interface de notre projet.
- Dans le dernier chapitre on fait une étude pratique sur notre plaque.

Chapitre 1 :

PIC 18F2550

1. Caractéristiques du PIC 18F2550



Le 18F2550 est un microcontrôleur en boîtier DIL 28 pattes possédant jusqu'à 24 E/S sur 3 ports (A, B, C).

Alimenté de 2 à 6 V continu, il est compatible avec le mode de programmation ICSP (In Chip Serial Programming), et dispose d'un oscillateur interne pouvant monter jusqu'à 8 Mhz. Sur oscillateur externe, le 18F2550 peut monter jusqu'à 48 MHz.

Au niveau mémoire, ce PIC dispose de 16 KO de mémoire flash pour le freeware, de 2 KO de RAM et de 256 octets d'EEPROM.

Côté fonctionnalité, outre les E/S numériques classiques, ce PIC dispose de 10 CAN, de 2 comparateurs, de 2 PWM 10 bits, d'1 module de communication série synchrone et asynchrone, d'un module de communication USB, de 4 timers, ...

Il faut savoir qu'il existe le 18F4550. Ce PIC, pouvant être considéré comme une extension du 18F2550, est en boîtier DIL 40 pattes, et possède jusqu'à 35 E/S. Possédant également un module de communication USB, leur principale différence, outre le nombre de pattes, et donc d'E/S se situe au niveau des CAN. En effet, le 18F4550 en possède 13, soit 3 de plus que le 18F2550.

Chapitre 1 : PIC 18F2550

2. Fonctionnement du PIC 18F2550

2.1 Les entrées/sorties

Dans cette partie, nous allons voir les différentes possibilités de chaque patte, en les désignant par leur numéro. A noter, qu'une seule fonction est disponible à la fois, par patte.

PATTE 1 : E numérique RE3, reset.

PATTE 2 : E/S numérique RA0, CAN0.

PATTE 3 : E/S numérique RA1, CAN1.

PATTE 4 : E/S numérique RA2, CAN2, tension de référence du comparateur, référence basse CAN.

PATTE 5 : E/S numérique RA3, CAN3, référence haute CAN.

PATTE 6 : E/S numérique RA4, entrée comptage timer 0, sortie comparateur 1, E module USB externe.

PATTE 7 : E/S numérique RA5, CAN4, sélection de périphérique synchrone, sortie de comparateur 2.

PATTE 8 : Masse.

PATTE 9 : Entrée oscillateur, entrée horloge.

PATTE 10 : Entrée oscillateur, sortie horloge, S numérique RA6.

PATTE 11 : E/S numérique RC0, sortie horloge timer 1.

PATTE 12 : E/S numérique RC1, entrée horloge timer 1, sortie PWM 2, S module USB externe.

PATTE 13 : E/S numérique RC2, sortie PWM 1.

PATTE 14 : Référence de tension USB 3,3V (condensateur de 470 nF).

PATTE 15 : E/S numérique RC4, E/S – USB, E module USB externe VM.

PATTE 16 : E/S numérique RC5, E/S + USB, E module USB externe VP.

PATTE 17 : E/S numérique RC6, S RS232, S horloge synchrone.

PATTE 18 : E/S numérique RC7, E RS232, E donnée synchrone, sortie donnée SPI.

PATTE 19 : Masse.

PATTE 20 : Alimentation positive +5V.

PATTE 21 : E/S numérique RB0, CAN12, interruption externe 0, entrée d'erreur PWM, entrée SPI, E/S IIC.

PATTE 22 : E/S numérique RB1, CAN10, interruption externe 1, horloge SPI, horloge I²C.

Chapitre 1 : PIC 18F2550

PATTE 23 : E/S numérique RB2, CAN8, interruption externe 2, S module USB externe VM.

PATTE 24: E/S numérique RB3, CAN9, S PWM 2, S module USB extern VP.

PATTE 25 : E/S numérique RB4, CAN11, interruption de changement 0.

PATTE 26 : E/S numérique RB5, interruption de changement 1.

PATTE 27 : E/S numérique RB6, interruption de changement 2.

PATTE 28 : E/S numérique RB7, interruption de changement 3.

Les E/S se configurent via TRISA, TRISB, TRISC

2.2 La tension de référence

La tension de référence est configurable via CVRCON. Toutefois cette source est dédiée au comparateur (d'où le C de CVREF).

CVRCON

CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
-------	-------	------	-------	------	------	------	------

CVREN : active (1) ou non la référence de tension

CVROE : la tension de référence est connectée sur RA2 (1) ou non CVRR : sélection de la formule de calcul de tension

CVRSS : sélectionne la tension de référence : VDD-VSS (0) ou bien (Vref+) - (Vref -)

CVR3-0 : si CVRR=1, alors $CVREF = (CVR3-0/24)*VDD$,

si CVRR=0, alors $CVREF = (VR3-0/32)*VDD + VDD/4$

Attention toutefois, car la tension maximale que peut atteindre cette référence de tension est d'environ 3,6V.

2.3 Les comparateurs

La présence de comparateurs analogiques est fort pratique. En effet, cette fonctionnalité permet de réaliser la comparaison et de connaître au niveau logiciel & hardware le résultat de façon simple.

Le principe d'un comparateur, fort simple, est le suivant :

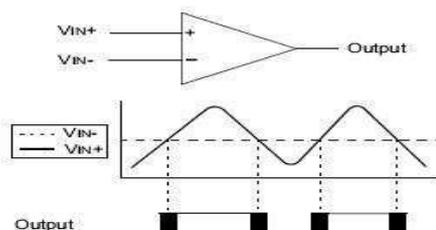


Figure 1.1 : principe d'un comparateur

Chapitre 1 : PIC 18F2550

Si $V^+ > V^-$ alors la sortie = 1, sinon 0. Ce comparateur permet de comparer diverses choses en fonction du mode de configuration.

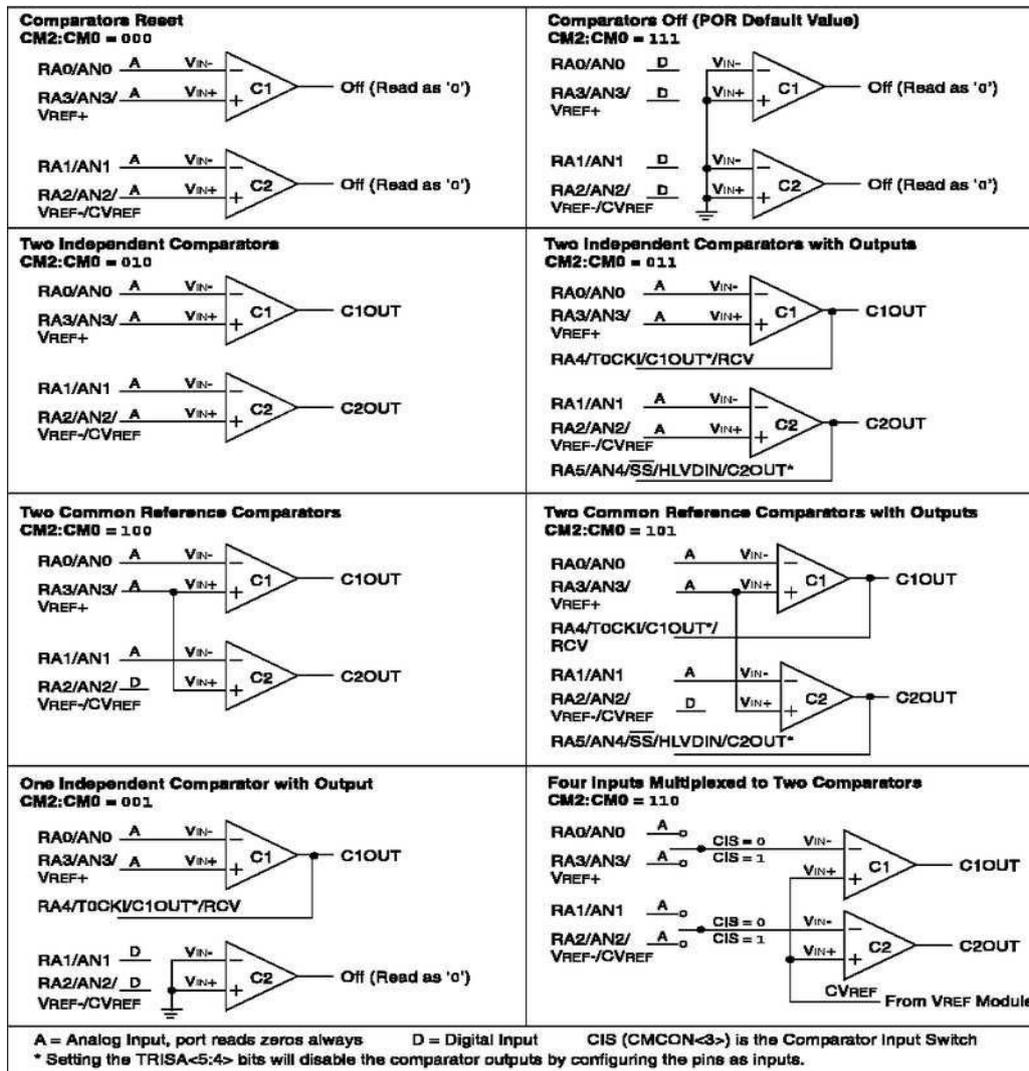


Figure 1.2 : les modes de configuration

Afin de configurer le comparateur dans le mode désiré, il faut configurer le registre CMCON. Le CVREF est une référence de tension.

CMCON

C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
-------	-------	-------	-------	-----	-----	-----	-----

C2OUT : en lecture seulement, permet de connaître logiciellement l'état de la comparaison (sortie du comparateur 2)

Chapitre 1 : PIC 18F2550

C1OUT : en lecture seulement, permet de connaître logiciellement l'état de la comparaison (sortie du comparateur 1) C2INV-C1INV : inverse l'état de la sortie (1) ou non

CIS : sélection d'entrée dans certains modes, voir schémas précédents

CM2-0 : configuration du comparateur, voir schémas précédents.

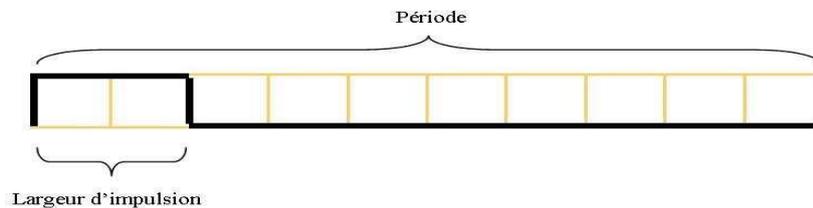
2.4 La PWM

La PWM fait partie d'un bloc interne du PIC : le CCP ; ou Capture/Compare/PWM. Les deux premiers modes assez flous et d'une approche assez difficile ne seront pas vus ici, car très peu utilisés. Nous nous concentrerons sur la PWM, plus répandue.

La PWM pour Pulse Width Modulation, ou en français MLI pour Modulation en Largeur d'Impulsion, permet de gérer l'énergie transmise à l'extérieur. En effet, si un signal continu correspond à 100% d'énergie, un signal carré dont la durée d'état haut égale celle d'état bas correspond à 50 % d'énergie. Le pourcentage d'énergie transmis se calcule en faisant le rapport de la durée d'état haut sur la durée de la période.

La PWM peut permettre de gérer la luminosité d'une lampe, le pilotage d'un servomoteur de modélisme, contrôler la vitesse d'un moteur continu, ...

Voici une période PWM :



Deux formules permettent de calculer la durée de la période, et la largeur de l'impulsion.

$$\text{PERIODE} = (\text{PR}2 + 1) * 4 * \text{Tosc} * (\text{prédiviseur timer } 2)$$

$$\text{LARGEUR} = (10 \text{ bits}) * \text{Tosc} * (\text{prédiviseur timer } 2)$$

Par Tosc, nous comprenons la période de l'oscillateur servant d'horloge au PIC. Les 10 bits correspondent à l'octet de CCPR1L et des 2 bits de DC1Bx. Cependant cette résolution de 10 bits est la valeur maximale. En effet, selon la fréquence désirée, cette résolution peut changer. Pour connaître la résolution maximale, on utilise la formule suivante :

Chapitre 1 : PIC 18F2550

$RESOLUTION = (\log(F_{osc}/F_{pwm}))/\log(2)$ bits.

Par exemple, avec le pré diviseur timer 2 à 16, et un PR2 à 255 (0XFF), nous avons une résolution maximale de 10 bits, et une fréquence de PWM de 1,22 KHz.

Autre exemple, avec un pré diviseur à 1, et un PR2 à 31, la résolution maximale est alors de 7 bits, et une fréquence de PWM de 156,3 KHz.

Ici, les registres utiles sont le CCP1CON, CCP2CON, PR2, CCPR1L, CCPR2L. De plus, comme précisé précédemment, c'est le timer 2 qui est utilisé ici. De fait, il faudra également utiliser T2CON, afin d'activer le timer 2. Cependant, la plupart des compilateurs rendent toutes ces actions transparentes.

CCP1CON

		DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
--	--	-------	-------	--------	--------	--------	--------

DC1B1-0 : contient les 2 bits de poids faibles de la période (PWM 10 bits) CCP1M3-0: Pour la PWM, égal à 11xx

Les 8 autres bits doivent être chargés dans le registre CCPR1L.

CCP2CON

		DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
--	--	-------	-------	--------	--------	--------	--------

DC2B1-0 : contient les 2 bits de poids faibles de la période (PWM 10 bits) CCP2M3-0: Pour la PWM, égal à 11xx

Les 8 autres bits doivent être chargés dans le registre CCPR2L.

ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2
ADR1	ADR0						

Chapitre 2 :

PROTOCOLE USB

Chapitre 2 : Protocole USB

1. INTRODUCTION

La généralisation de l'USB comme interface privilégiée de communication entre l'ordinateur et les périphériques informatiques a entraîné la disparition progressive des interfaces <série> et <parallèle>.

De ce fait, l'USB est devenu le standard pour la connexion câblée de périphérique et de tous les appareils micro-informatiques au PC.

2. Naissance de l'USB

La spécification ou norme USB version 1.0 est sortie en Janvier 1996. A cette époque, le manque de composants USB a retardé la mise à jour de la norme. En Septembre 1998 a apparu la version 1.1.

Puis en Avril 2000 est sortie la spécification 2.0. cette nouvelle norme donne les caractéristiques de l'USB 1.1 : 1,5 Mbps (faible vitesse), 12 Mbps (pleine vitesse), et rajoute une vitesse de 480 Mbps (haute vitesse).

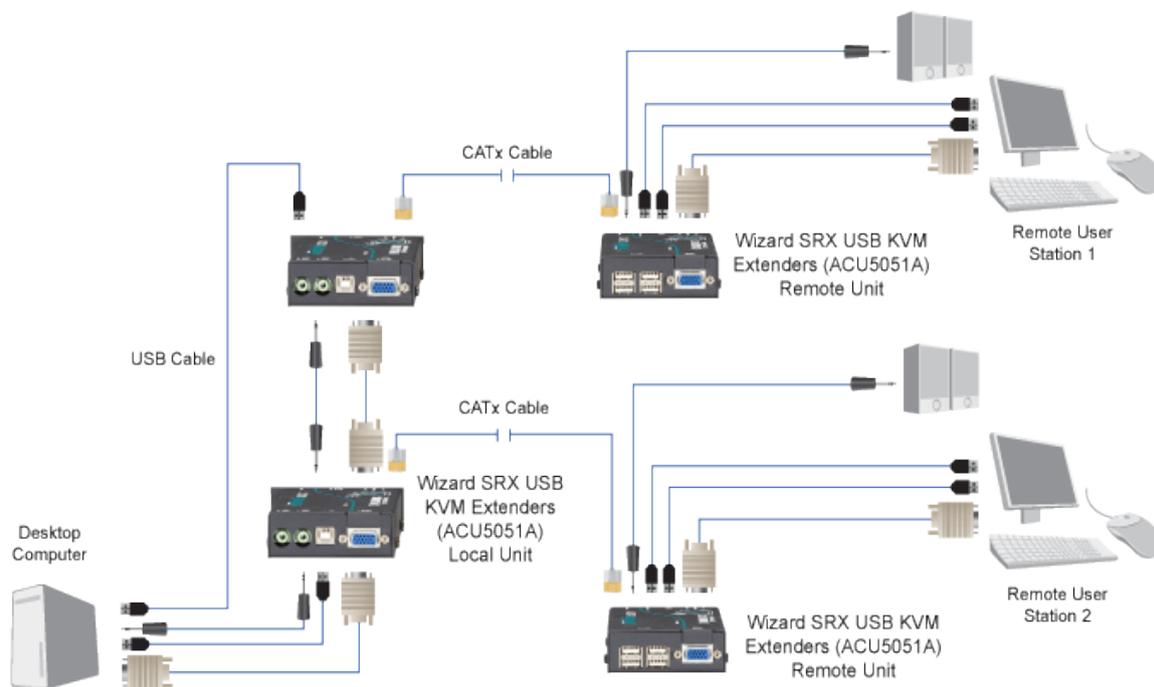


Figure 2.1 : Exemple d'application où l'USB peut remplir un rôle irremplaçable

La dernière norme a été optimisée avec l'apparition de la version 3.0 qui est plus rapide que la précédente (très haute vitesse) plus de 480 Mbps.

Chapitre 2 : Protocole USB

3. Composition du câble USB

La liaison USB est composée d'un câble quad filaire. Deux conducteurs fins servent à la transmission de données (D+ et D-), deux conducteurs de section plus importante assurant l'alimentation des appareils sous 5V et à un courant de 500mA au maximum (GND et Vcc). Il n'est plus nécessaire ainsi de doter certains petits périphériques tels que modems et lecteurs de cartes, de leur propre alimentation. Il existe les applications critiques, une variante de câble blindé. En version Low Speed le blindage n'est pas obligatoire (ce qui assure une plus grande souplesse de manipulation en particulier pour une liaison souris) La figure 2.2, 3 montre les deux versions du câble USB.

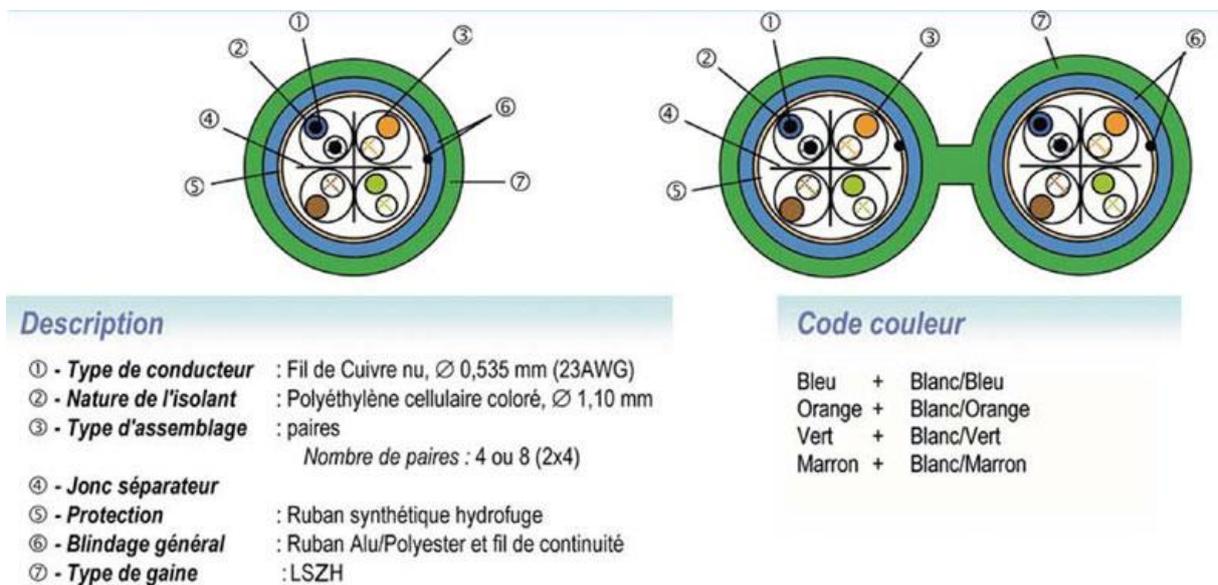


Figure 2.2 : Coupe d'un câble USB.

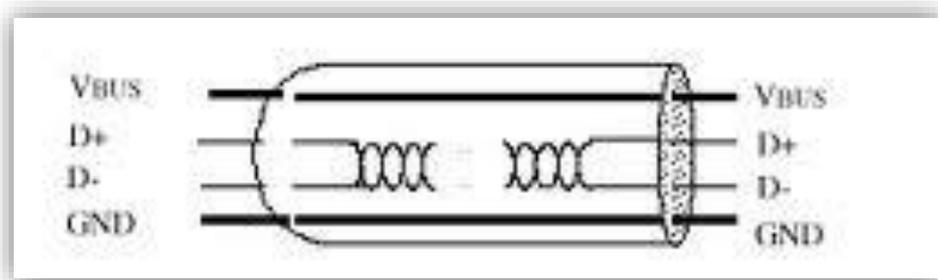


Figure 2.3 : Schéma électrique du câble USB

Chapitre 2 : Protocole USB

Le câble USB est composé de deux fiches bien différentes, en amont d'une fiche appelée connecteur USB de type A, branché au host (PC). L'extrémité en aval par contre peut se retrouver en deux versions : Connecteur USB du type B et un mini connecteur de type B.

Ce dernier est réservé aux dispositifs de très faible dimension tels que les appareils photo numérique. La figure 2.4 montre les différents connecteurs USB.

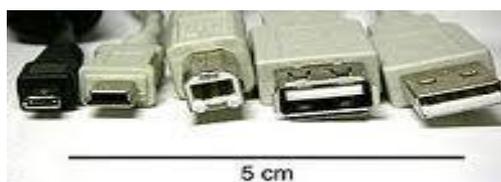


Figure 2.4 : Schéma des différents connecteurs USB

4. Le bus USB

4.1 Principe du bus USB :

Le bus USB est un bus fonctionnant sur la hiérarchie, commandé par un host unique. Le host utilise un protocole Maître/Esclave pour communiquer avec les périphériques USB. Cela signifie que c'est le host qui décide du transfert des données et que les différents périphériques ne peuvent pas établir de connexion entre eux tant que le maître n'a pas donné l'autorisation.

Chapitre 2 : Protocole USB

Le fait que le bus USB fonctionne avec le protocole Maître/Esclave résout les problèmes de collision ou d'arbitrage de périphériques.

4.2 Topologie du bus USB

La topologie est une topologie «étoile série». à l'initialisation d'un périphérique celui-ci reçoit une adresse. Les transferts Full Speed déclenchés par le host parcourent tous les tronçons, mais seule la fonction adressée va évidemment répondre. Ainsi s'il s'agit d'un ordre d'impression il sera reçu par tous les périphériques, mais seul l'imprimante le comprendra. La figure 2.5 montre la topologie du bus USB

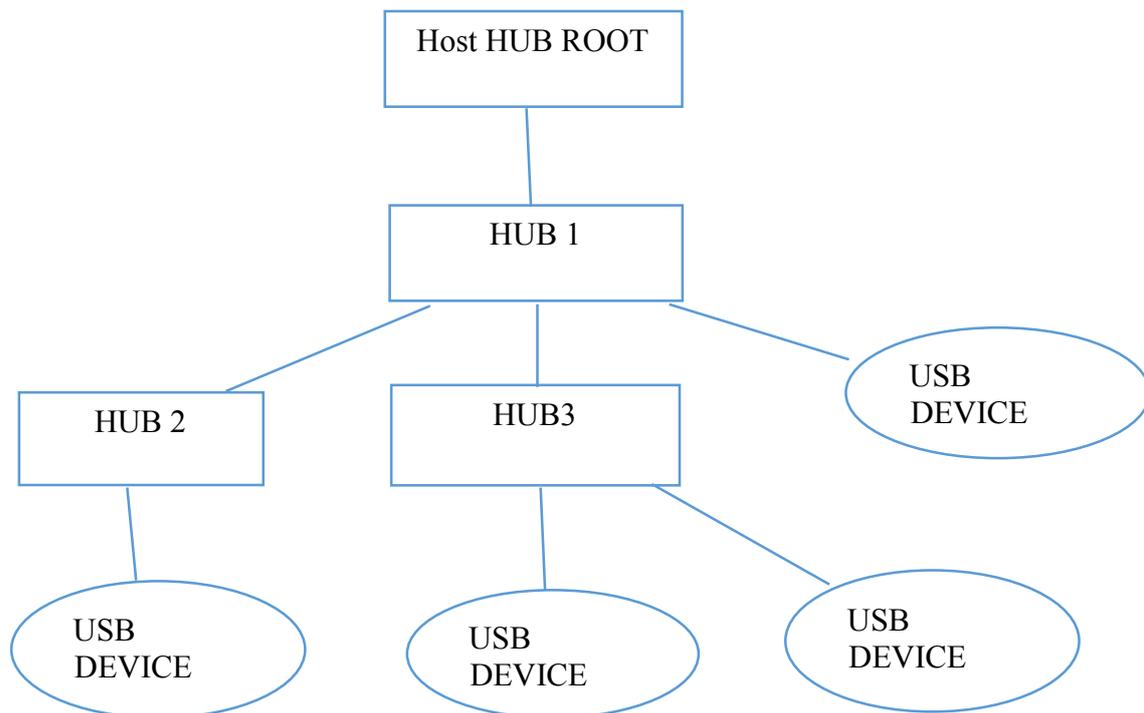


Figure 2.5 : Un exemple de branchement avec la topologie étoile de l'USB

5. Les types de paquet USB :

L'entité de transfert USB est appelée transaction, elle est généralement constituée de paquets juxtaposés, la transaction étant un paquet SOF qui indique le début d'une autre transaction. On dit alors qu'un transfert est composé d'une succession de transaction.

Les transactions USB se font par l'intermédiaire de l'émission de plusieurs paquets. Chaque transaction consiste en la succession de :

- Token (en tête)
- SOF (start of frame)
- Data (donnée)

Chapitre 2 : Protocole USB

- Acknowledge (handshake,ACK)

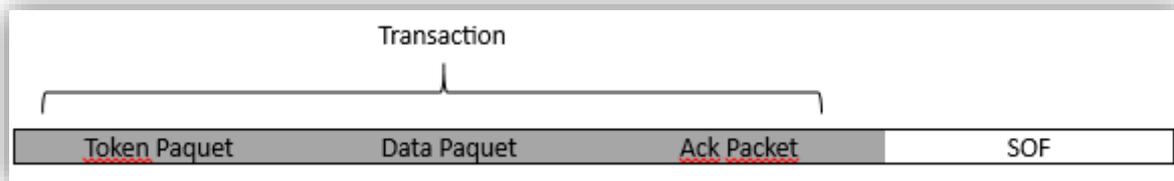


Figure 2.6 : Détail d'une transaction

6. Structure des paquets :

La structure des paquets étant identique, il faut préciser que le format est différent selon la nature du paquet. Les figures 2.7, 8 montrent la structure et le format d'un paquet.

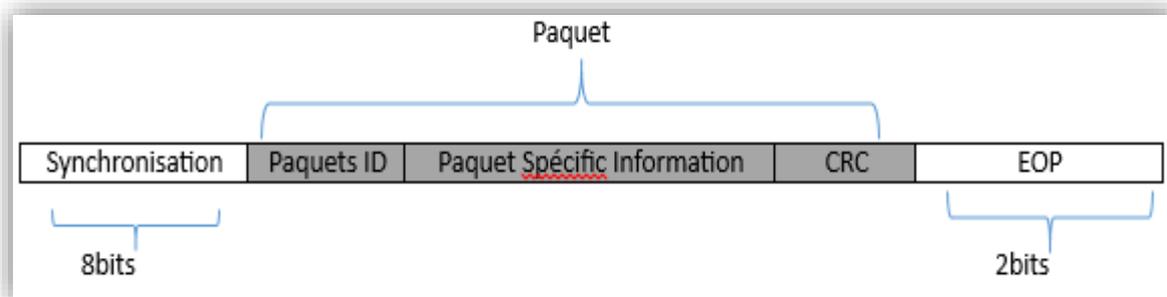


Figure 2.7 : Structure d'un paquet

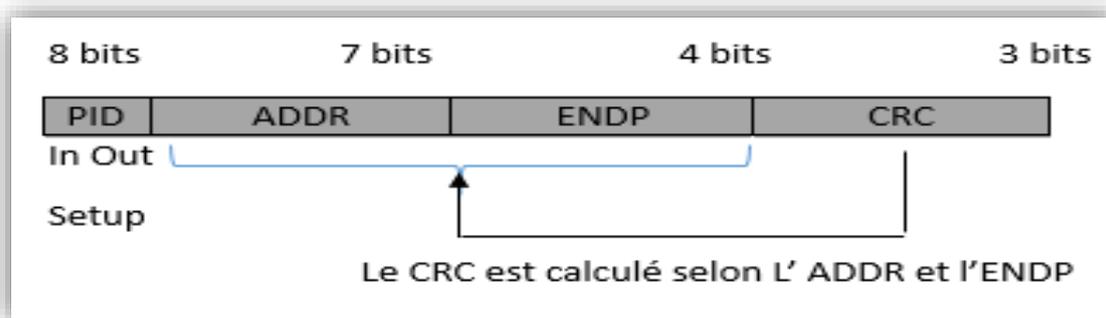


Figure 2.8 : Format d'un paquet

6.1 Les paquets Jetons :

Il y a 3 sortes de paquets Jetons :

- In : Informe l'appareil USB que l'hôte veut lire des informations.
- Out : Informe l'appareil USB que l'hôte veut envoyer des informations.
- Setup : Utilisé pour commencer les transferts de commande.

Chapitre 2 : Protocole USB

6.2 Les paquets Data :

La taille maximale de « données utiles » est respectivement de 8 octets pour les périphériques Low Speed, 64 octets pour les périphériques Full Speed, et 1024 octets pour les périphériques High Speed.

6.3 Les paquets Handshake :

Il y a 3 sortes de paquets Handshake qui font simplement partie du PID :

- **ACK** : validant que le paquet a été reçu correctement
- **STALL** : l'appareil se retrouve dans un état qui va exiger l'intervention de l'hôte.
- **NACK** : Indique que l'appareil ne peut temporairement ni envoyer ou recevoir des données. aussi utilisé pendant les transactions d'interruptions pour avertir l'hôte qu'il n'a pas de données à envoyer.

6.4 Les paquets SOF :

Le paquet SOF composé d'une trame de 11 bits est envoyé par l'hôte toutes les 1ms sur un bus Full Speed ou bien toutes les 125µs sur un bus High Speed. La figure 2.9 montre les champs de paquet SOF.

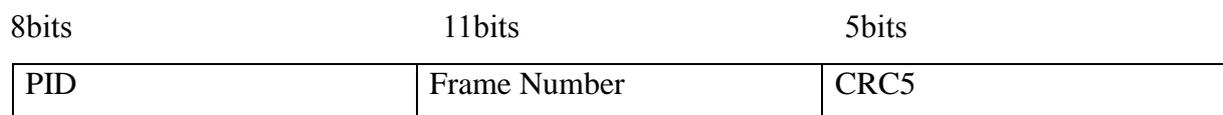


Figure 2.9 : Format des paquets SOF

7. Explication des différents champs des paquets d'une trame USB :

7.1 Synchronisation :

Tous les paquets doivent commencer avec un champ synchrone. Le champ synchrone fait de 8 bits de long pour la basse et pleine vitesse ou 32 bits pour la haute vitesse, est utilisé pour synchroniser l'horloge du récepteur avec celle de l'émetteur /récepteur.

7.2 Paquet ID :

Ce champ est utilisé pour identifier le type de paquet qui est envoyé. Le tableau suivant montre les valeurs possibles : il y a 4 bits pour le PID, toutefois pour s'assurer qu'il a été reçu correctement, les 4 bits sont complémentés et répétés faisant un PID de 8 bits au total. Le format résultant est indiqué à la figure 2.10 ci-dessous :

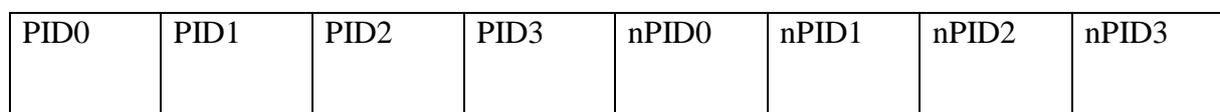


Figure 2.10 : Format des paquets ID

Chapitre 2 : Protocole USB

7.3 Paquet adresse : Le champ adresse détermine à quel appareil le paquet est destiné.

7.4 Paquet ENDP : Le champ de terminaison est composé de 4bits, autorisant 16 terminaisons possibles.

7.5 Paquet CRC : Les contrôles à redondance cyclique sont exécutés sur les données à l'intérieur du paquet. Tous les paquets jetons ont un CRC de 5 bits tandis que les paquets de données ont un CRC de 16 bits.

7.6 Paquet EOP : Fin de paquet.

8. les types de transferts

La spécification de l'USB définit 4 types de transfert entre l'hôte et les périphériques.

➤ **Transfert en mode Contrôle :**

Ce mode de transfert est compatible avec le Low et le Full Speed USB. Il est utilisé pour les opérations d'initialisations et de configurations, ainsi que pour le driver de la classe HID.

➤ **Transfert en mode Interrupt :**

Ce mode de transfert est également compatible avec le Low et Full speed USB. Utilisé par des périphériques qui envoient très peu de données comme les claviers.

➤ **Transfert en mode Isochrone :**

Ce mode de transfert est uniquement compatible avec le Full USB. Il est utilisé pour des transferts nécessitant un flux régulier de données comme par exemple les caméras ou les téléphones.

➤ **Transfert en mode Bulk :**

Ce mode de transfert est uniquement compatible avec le Full USB. Il est réservé pour les gros transferts de données par exemple l'imprimante.

9. Temps de connexion et de déconnexion :

Les connecteurs USB (de type A ou B) ont les deux pattes d'alimentation un peu plus longues que les pattes de données pour que le périphérique USB soit alimenté avant que transitent les données. Il est normal qu'avant de transmettre des données, que le périphérique soit reconnu et que les bons drivers ont été chargés. L'USB permet de faire cette étape automatiquement, et aussi pour protéger les composants USB, car un composant doit d'abord être alimenté avant de recevoir des données. Lorsqu'on branche un composant USB une différence de potentiel se crée et c'est cette différence de potentiel qui démarre le processus d'énumération. Si un composant USB est branché, un câble de données (D+ par exemple pour le Low Speed) est

Chapitre 2 : Protocole USB

proche de Vcc et l'autre proche de GND. On appelle cet état <l'état J>. Si on débranche le composant, les deux lignes de données changent d'état et on appelle cet état <l'état K>. Un périphérique est déconnecté si les deux lignes de données (D+ et D-) chutent sous la valeur de 0.8V DC sur une durée supérieure à 2.5µs. Et de même pour la connexion : si une ligne de donnée (D+ ou D- selon la vitesse) dépasse 2V DC sur une durée supérieure à 2.5µs, le host considère que le composant est attaché.

Il doit être décodé lors de la réception pour pouvoir retraiter les données.

10. Les descripteurs :

On peut définir les descripteurs comme étant des blocs d'informations, toutes ces informations sont stockées dans la ROM des composants, et lors de l'énumération, le périphérique envoie simplement ce fichier pour se faire connaître. La figure 2.11 montre la topologie des différents descripteurs.

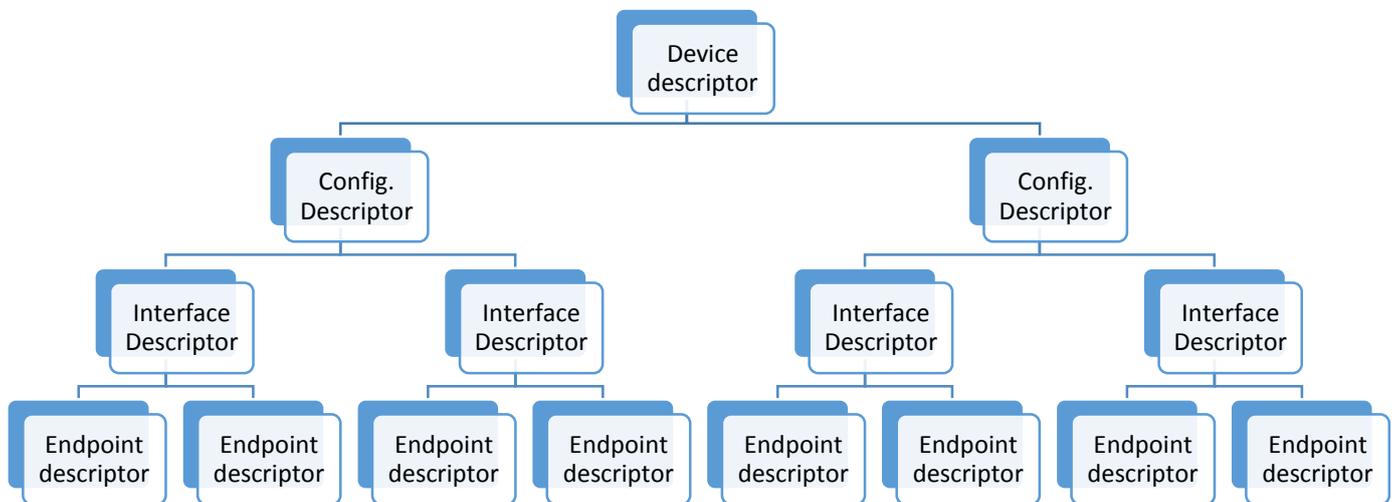


Figure 2.11 : Hiérarchie des différents descripteurs.

Chapitre 2 : Protocole USB

Les descripteurs sont regroupés en 4 catégories :

10.1 Device descriptor (descripteur d'appareil) :

Ce type de descripteur donne les informations générales, comme par exemple le numéro de spécification USB qui est présent dans toutes les configurations, les numéros d'identifiant de produit et de vendeur. C'est le premier descripteur que vient lire le host.

Dans le tableau ci-dessous, un exemple des différents descripteurs faisant parties de la catégorie 'device descriptor'. Dans le champ description il y a la signification de chaque descripteur.

	Nom	Taille	Description
0	Blength	1	Taille du descripteur, toujours égale à 18 pour un devicedescriptor.
1	Bdescriptor Type	1	Type du descripteur, toujours égale à 01h pour une devicedescriptor.
2	bcdUSB	2	C'est la version de la spécification USB, (codé en BCD) exemple : <ul style="list-style-type: none">• Version 1.0 --->0100h• Version 1.1 --->0110h• Version 2.0 --->0200h
4	BDeviceClass	1	Si le composant appartient à une classe, ce champ renseigne sur cette classe. S'il n'appartient à aucune classe ce champ doit être mis à 0. L'adresse FFh est réservée et signifie que la classe est spécifique au vendeur
5	BDeviceSubClass	1	Si un composant appartient déjà à une classe, ce champ permettra de restreindre encore une fois en définissant une sous-classe Si BDeviceClass = 0, BDeviceSubClass doit forcément être égal à 0 aussi.
6	bDeviceProtocol	1	Ce champ renseigne sur le protocole choisi par la classe ou la sous classe. C'est dans ce champ qu'un Hub USB 2.0 dira s'il supporte à la fois le Full et le HighSpeed.

Chapitre 2 : Protocole USB

7	bMaxPacketSize0	1	Taille maximum des paquets de l'Endpoint 0. Low Speed : 8 Full Speed : 8-16-32-64 High Speed : 64
8	IdVendor	2	Ce champ renseigne sur le Vendor ID
10	IdProduct	2	Ce champ renseigne sur le Product ID
12	BcdDevice	2	(Device release number) codé en BCD, ce champ, optionnel, renseigne sur le driver à charger par exemple
14	IManufacturer	1	C'est un index qui pointe sur un string qui donne le nom du constructeur. Ce champ est optionnel, 0 si inutilisé.
15	IProduct	1	C'est un index qui pointe sur un string qui donne des informations sur le produit. Ce champ est optionnel, 0 si inutilisé.
16	iSerialNumber	1	C'est un index qui pointe sur un string qui donne des informations sur le numéro de série du composant. Ce champ est optionnel, 0 si inutilisé. Ce champ peut être intéressant si l'on utilise plusieurs périphériques.
17	bNumConfigurations	1	Nombre de configuration que supporte le composant.

Tableau 2.1 : tableau récapitulatif des différents champs de Device Descriptor

10.2 Configuration descriptor (descripteur de configuration) :

Le descripteur de configuration renseigne sur les différents états dans lequel peut se trouver le composant USB. Ces descripteurs de configuration définissent par exemple l'origine de l'alimentation, elle peut soit provenir d'une alimentation extérieure, soit directement du bus USB.

Dans le **tableau 2.2** ci-dessous, un exemple des différents descripteurs faisant partie de la catégorie 'configuration descriptor'.

	Nom	Taille	Description
0	Blenght	1	Taille du descripteur, toujours égale à 9 pour une configuration descriptor.

Chapitre 2 : Protocole USB

1	Bdescriptor Type	1	Type du descripteur, toujours égale à 02h pour une configuration descriptor.
2	WTotalLenght	2	Longueur total en bytes de toutes les données dans cette configuration. Cette longueur inclus les descripteurs d'interfaces, HIDs, reports et les end points.
4	BNumInterfaces	1	Nombre d'interfaces supportées dans cette configuration.
5	Bconfiguration value	1	Identifiant de Set_Configuration et de Get_Configuration.
6	Iconfiguration	1	C'est un index qui pointe sur un string qui décrit cette configuration. Ce champ est optionnel 0 si inutilisé.
7	BmAttributes	1	Précise comment est alimenté le périphérique. SelfPower ou Bus. Précise également si le périphérique supporte le remoteWakeup.
8	MaxPower	1	Consommation du composant USB (en mA/2).

10.3 Interface descriptor (descripteur d'interface) :

Une interface peut être considérée comme un ensemble de "Endpoint" (point final ou terminaison). Un endpoint est en quelque sorte l'extrémité d'un "pipe". Un Pipe est une sorte de tuyau par lequel transitent les données via le host.

Dans le **tableau 2.3** ci-dessous, un exemple des différents descripteurs faisant parties de la catégorie "interface descriptor".

	Nom	Taille	Description
0	Blenght	1	Taille du descripteur, toujours égale à 9 pour une interface descriptor.
1	Bdescriptor Type	1	Type du descripteur, toujours égale à 02h pour une interface descriptor.
2	BinterfaceNumber	1	Dans un périphérique, une configuration possède plusieurs interfaces qui peuvent être active en même temps. Ce champ spécifie pour pouvoir reconnaître les interfaces de la même configuration
3	BAlternateSetting	1	Nombre d'Endpoint supporté par l'interface. L'endpoint 0 n'est pas compté. Si c'est le seul supporté BnumEndPoints=0

Chapitre 2 : Protocole USB

4	WMaxPacketSize	2	Spécifie la taille maximale des paquets : Bit 0 à 9 : 0 à 1023 Bit 10 et 11 : indique le nombre de transaction supplémentaire par micro frame (uniquement pour high speed) Bit 12 à 15 : réservés
6	Binterval	1	Indique le temps maximal (en ms) entre le polling des Endpoints.

11.La classe Human Interface Device (HID) :

Dans les paragraphes précédents nous avons vu que l'USB peut configurer les périphériques au démarrage ou à leur branchement. Ces périphériques sont divisés en plusieurs classes :

Classe	Exemple
Display	Ecran
Communication	Modem
Audio	Haut-parleurs
Mass storage	Disques durs
Human Interface	Souris, claviers...

TABLEAU 2.5 : Exemples de classes de périphériques USB.

Le bus USB peut soit utiliser un driver personnalisé, soit en utilisant un déjà présent dans le système et qui est basé sur le concept de classe de périphérique. Parmi ces classes, la classe USB HID (Human Interface Devices pour Dispositif d'Interface Homme-machine) qui rassemble des périphériques d'entrée comme les claviers et les souris, et autres périphériques basse vitesse à transfert de données faibles, elle est limitée à 64Kbits/s, la classe pour les périphériques USB Mass Storage (stockage de masse) qui regroupe des périphériques comme les clés USB, les appareils Photos numériques, les lecteurs audio, et la classe USB CDC pour communication devices Class (Classe de Dispositifs de Communication) qui regroupe essentiellement les modems USB et autres périphériques de ce type.

L'inclusion d'une classe USB HID (Human Interface device) dans windows permet de raccorder une carte électronique à un PC plus facilement et offre un chemin rapide pour le développement de l'USB.

Chapitre 2 : Protocole USB

12. La gestion du dispositif USB/HID :

C'est le descripteur interface qui identifie le périphérique comme appartenant à la classe HID. Les informations sur le périphérique USB sont stockées en segments dans la zone consacrée à l'USB en RAM, ces segments sont appelés les descripteurs. La récupération de ces données est faite en examinant les descripteurs du périphérique et les données qui contiennent. La figure 2.14 schématise cette étape.



Figure 2.12 : communication entre le périphérique et le driver HID

13. Conclusion :

Les avantages de l'USB sont nombreux, illustrant la portabilité, la simplicité ainsi que la facilité d'utilisation de l'USB. Ces avantages se traduisent suivant de multiples critères que voici :

- Sa configuration automatique, on l'appelle aussi le "plug &play". Cela signifie que si l'utilisateur connecte un périphérique USB, Windows détecte automatiquement ce périphérique.
- Le bus USB est "Hot pluggable", c'est-à-dire que l'on peut connecter et déconnecter un périphérique USB tout en ayant le PC allumé.
- L'USB possède une caractéristique très intéressante : c'est le mode veille lorsque l'on n'utilise pas le périphérique le bus USB se met en suspend après 3 ms ou il n'est plus utilisé. Pendant ce mode, le composant ne consomme que 500µA.
- Enfin, le dernier point fort pour l'USB est que cette norme permet d'alimenter le périphérique directement avec le PC. Il n'est pas nécessaire d'alimenter le périphérique avec une alimentation extérieure, le PC suffit.

Chapitre 3 :

Programmation et

Interface

Chapitre 3 : Programmation et Interface

1. Introduction

Le programme doit être saisi dans un fichier texte appelé fichier source. L'assembleur permet uniquement de créer un fichier assemblé (comporter l'extension .ASM) à partir du fichier source et aussi de faire la traduction du langage assembleur en langage machine.

2. Programme du pic 18F2550

2.1 Présentation de l'environnement de programmation

Après avoir écrit notre code en assembleur, nous le compilons et le résultat sera stocké dans un fichier .HEX à charger dans le pic grâce à un logiciel spécifique (Mplab).

2.2 Pour débuter avec MPLAB

Le logiciel MPLAB est un outil de développement pour programmer des microcontrôleurs de type PIC de la famille Microchip. Il est mis au point par la société Microchip, et est entièrement gratuit.

Ce logiciel vous permettra de créer un programme, de l'assembler, et de le simuler.

Enfin, vous pourrez transférer votre programme réalisé sous MPLAB pour le mettre sur votre PIC.

2.3 Caractéristiques

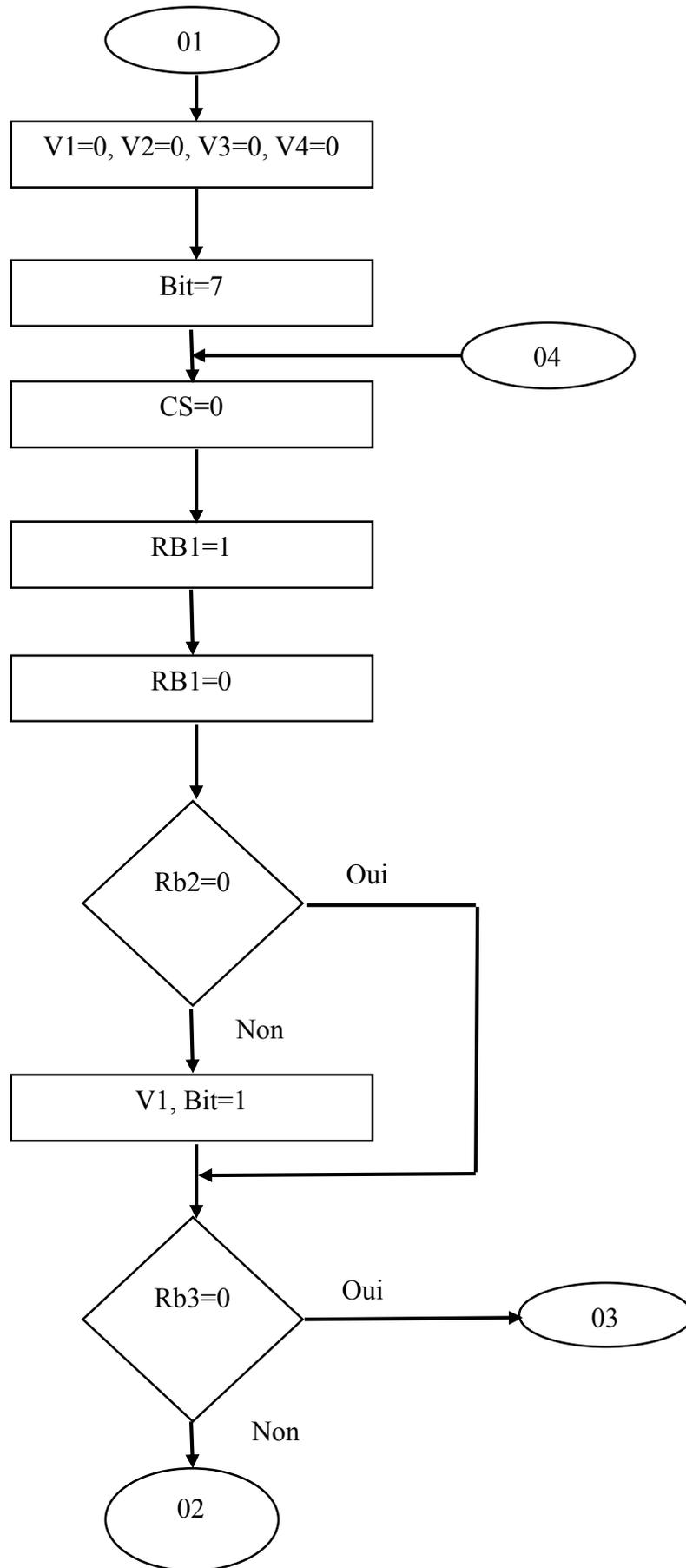
Avec l'environnement de MPLAB il est possible de réaliser un fichier source en langage assembleur (fichier .asm).

Mais l'avantage de MPLAB c'est de réaliser des programmes en langage C. Ainsi MPLAB peut utiliser un langage de programmation évolué pour le développement en électronique.

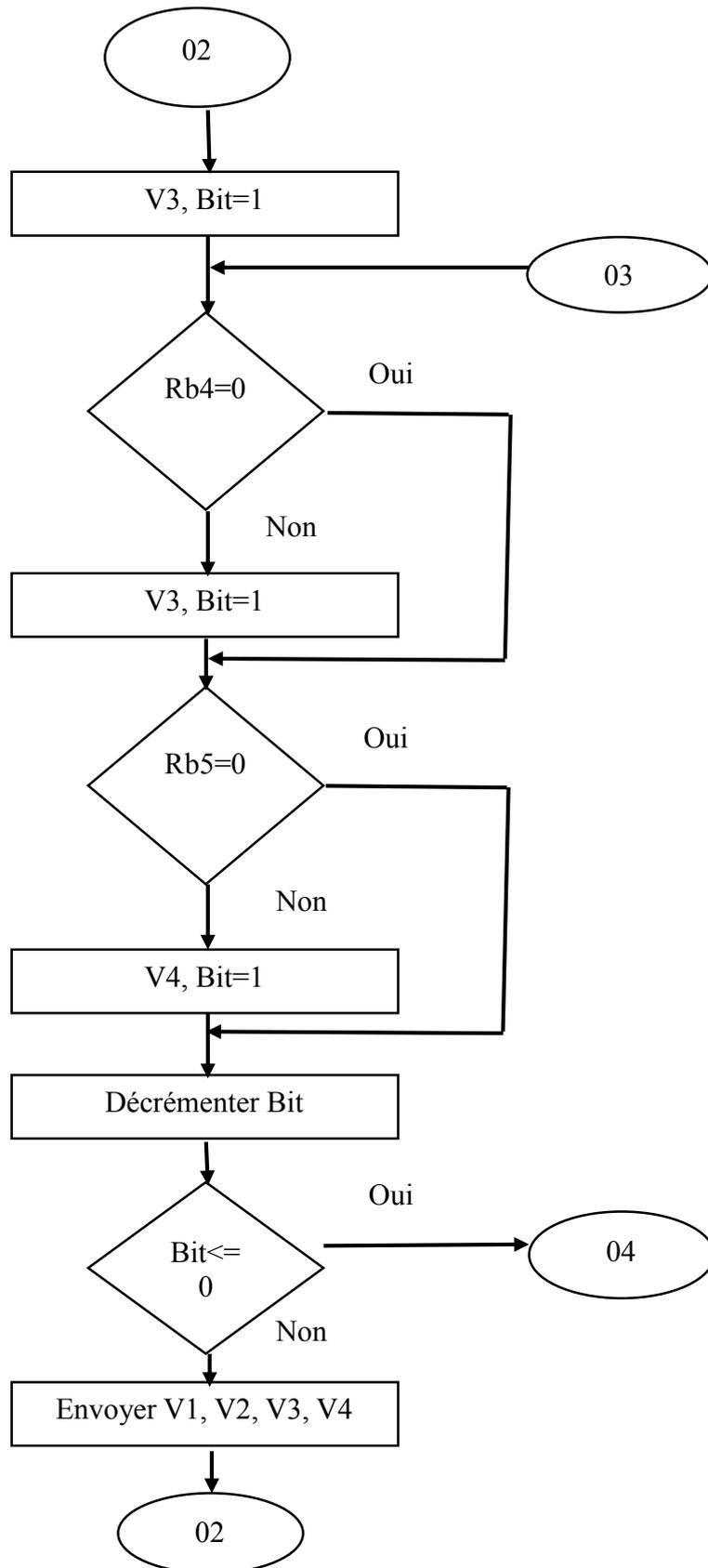
3. Le fonctionnement général du programme

Après avoir alimenté le circuit, on dispose d'une LED qui clignote pour s'assurer du bon fonctionnement du programme. Le choix du canal se fait à partir de l'interface de commande de PC qui envoie ce choix au PIC, ce dernier ne transmettra la conversion que si au moins un canal est sélectionné. Le convertisseur ADC0831 lance la conversion A/N au PIC qui le transmettra à l'interface par USB. Voici l'organigramme de la conversion :

Chapitre 3 : Programmation et Interface



Chapitre 3 : Programmation et Interface



À la fin de cette opération il teste l'état de l'émission de la valeur précédemment envoyée.

Chapitre 3 : Programmation et Interface

4. La configuration des registres

Comme nous avons déjà parlé dans le premier chapitre de notre rapport, la configuration et le contrôle des registres prend une grande partie du programme.

On peut partager le fonctionnement de notre programme en deux parties :

- ✓ **La conversion analogique / numérique**
- ✓ **L'émission et la réception des données**

a. La conversion analogique / numérique

Pour la conversion on a utilisé quatre (4) convertisseurs ADC0831, ces convertisseurs sont utilisés pour la conversion analogique/numérique et permet de :

- 4 entrées analogiques
- Vref de 0 volts et Vref de 5volts.

L'émission des données se fait sur quatre octets car on a envoyé seulement les données qui sont dans le registre ADRESL pour minimiser le temps de transmission.

b. L'émission et réception des données :

La communication entre le PIC et le PC utilisent 5 registres, « TXREG, RCREG » pour le stockage des données en émission et réception, « TXSTA, RCSTA » comme des registres d'état d'émission et de réception et « SPBRG » pour déterminer la vitesse de communication.

Ces registres sont configurés pour avoir :

- Une transmission asynchrone, haut vitesse de 8 bits.
- Une réception de 8 bits continue.

5. L'interface sur le PC

Notre projet a pour rôle de visualiser sur l'écran du PC des signaux physiologiques afin d'en faire une exploitation pratique.

Pour cela on a introduit une interface qui nous permet de commander ; notre PIC reçoit les résultats de la conversion et va les tracer.

Cette interface se compose de trois parties :



a. Partie paramètre

Cette partie contient quatre cases de commande d'entrée de canal 1 à canal 4 plus un échelle qui donne le choix afin de calibrer le tracer.

b. Partie valeur

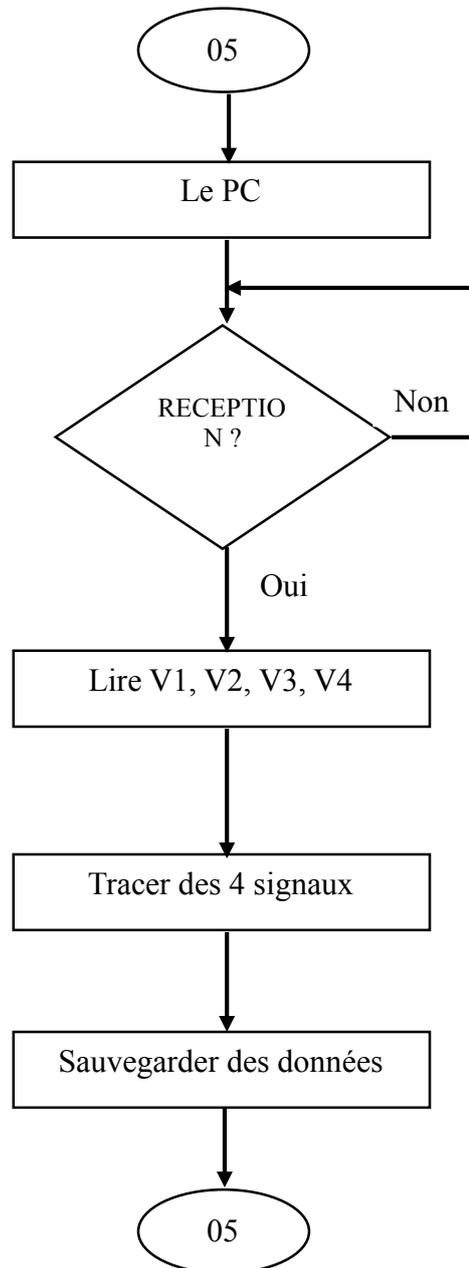
Il y a une autre partie où on peut visualiser le nombre d'octet transmit au PC et les valeurs numériques de la conversion A/N pour chaque échantillon correspondent au canal sélectionné. Ces valeurs ont stockées sur un fichier.

Chapitre 3 : Programmation et Interface

c. Partie traceur

C'est la partie la plus importante dans notre interface. Elle nous permet de visualiser les signaux des canaux sélectionnés dans le menu paramètre. Dans le cas où on a plusieurs signaux à tracer en même temps, l'espace graphique est partagé en conséquence.

6. Organigramme :



Cet organigramme représente le rôle de PC, il va recevoir les données, les analyser puis les tracer sur notre interface. IL sauvegardera ces mêmes données et referra chaque fois ce travail.

Chapitre 4 :

L'étude Pratique

Chapitre 4 : L'étude Pratique

1. Introduction

Dans notre projet de fin d'étude on a réalisé un montage basé sur deux circuits, le premier contient la partie alimentation, traitement de signal et communication avec le PC, et la deuxième partie pour la conversion des signaux utilisant le convertisseur ADC0831.

2. Le traitement numérique du signal et la communication avec le PC

2.1 Le circuit réalisé :

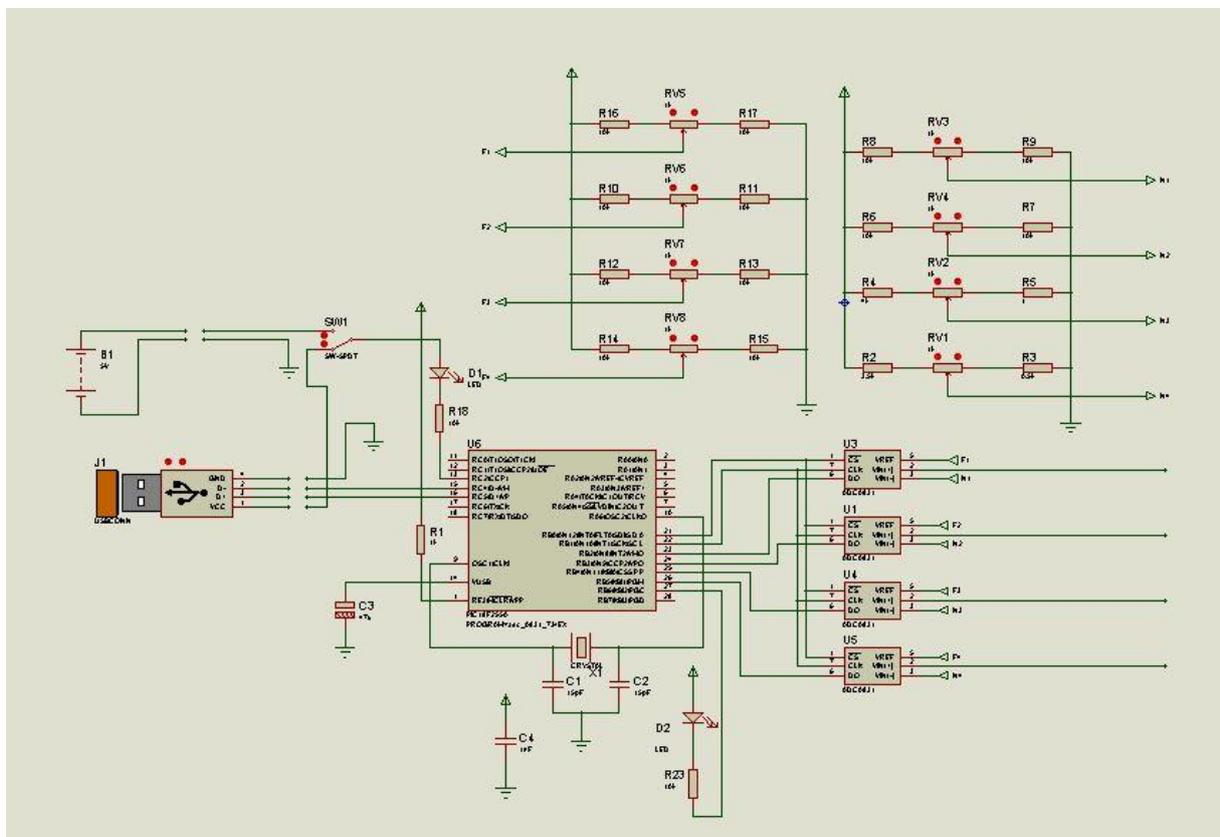


Figure 4.1 : le circuit de traitement

Le signal d'entrée du convertisseur A/D doit être entre 0 et 5 Volts, pour cela on a utilisé le circuit de diviseur de tension pour avoir une adaptation entre les différents types des signaux aux entrées du PIC.

On a utilisé une LED rouge pour indiquer si le circuit d'alimentation fonctionne ou non et une LED verte pour savoir si le programme fonctionne correctement.

Chapitre 4 : L'étude Pratique

2.2 Le Quartz

On trouve le quartz, qui peut être remplacé par un résonateur ou par un simple réseau RC. Les condensateurs de découplage, du fait de la fréquence plus importante du quartz utilisé, sont de valeur environ 15pF.

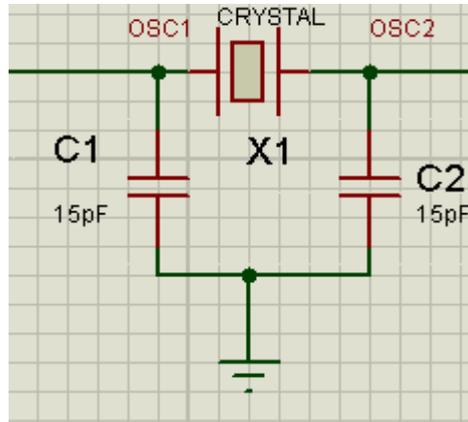


Figure 4.2 : le circuit quartz

Pour assurer une alimentation continue de 5V on a utilisé l'alimentation du PC en option on peut utiliser un chargeur ou une batterie qui génère 5 Volts.

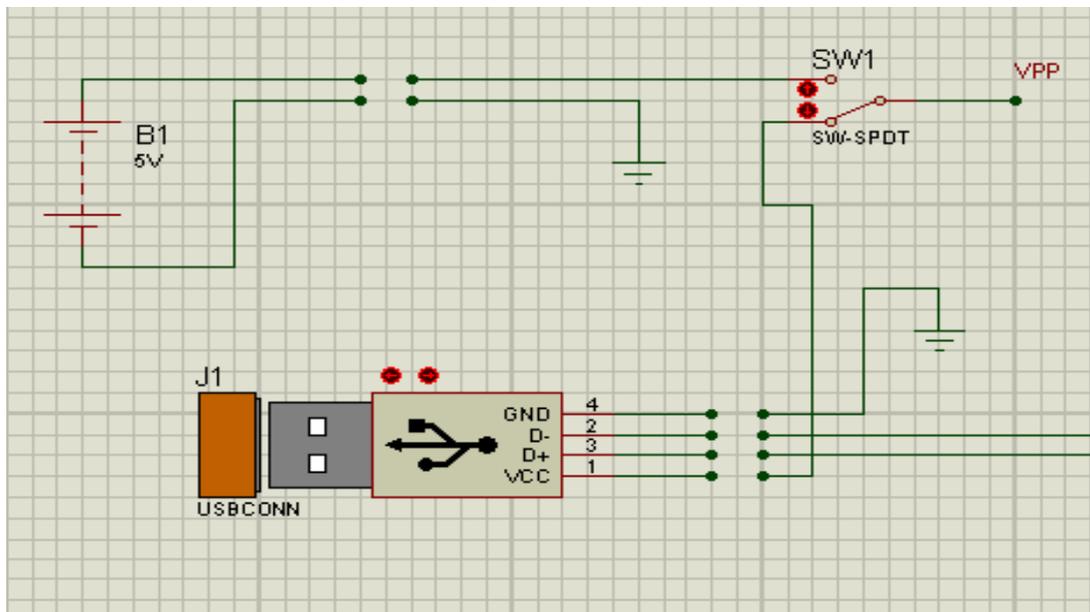


Figure 4.3 Le schéma électronique de circuit d'alimentation

Chapitre 4 : L'étude Pratique

2.3 Le branchement USB

Pour assurer une communication entre notre PIC 18F2550 et le PC on a utilisé un branchement USB relié comme ci-dessous :

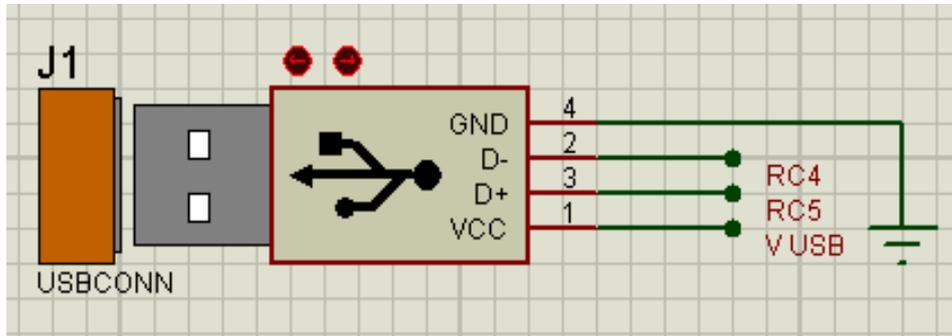


Figure 4.4 : Branchement USB

2.4 Le circuit de conversion

Le branchement de ce convertisseur ADC0831 est décrit ci-dessous :

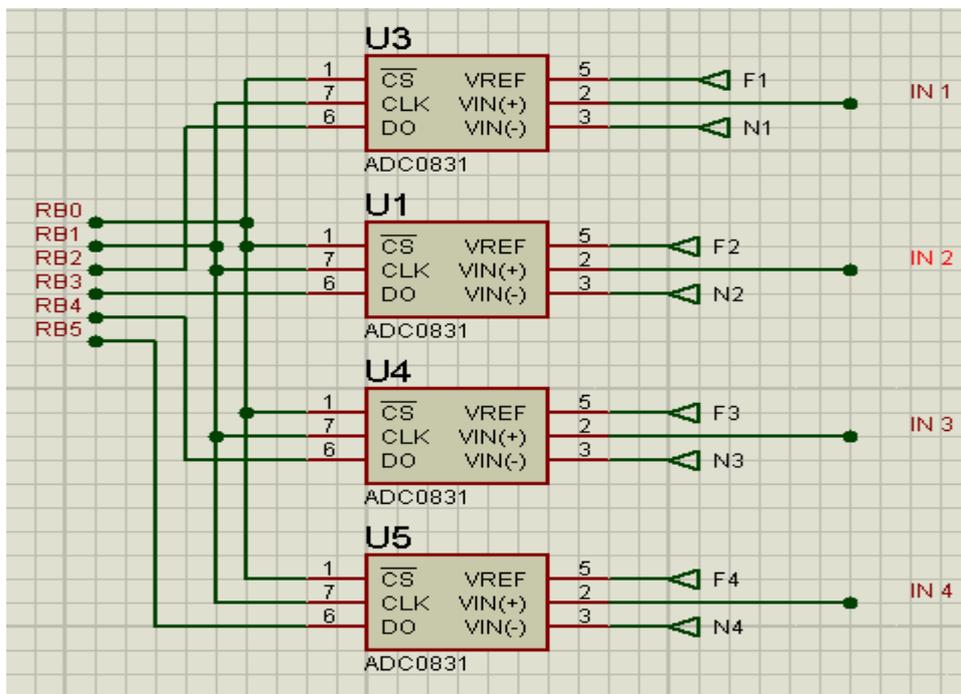


Figure 4.5 : Le schéma électronique de circuit convertisseur

Chapitre 4 : L'étude Pratique

Dans cette partie, nous allons voir les différentes possibilités de chaque patte, en les désignant par leur numéro. A noter, qu'une seule fonction est disponible à la fois, par patte.

PATTE 1 : CS (Chip select) Pour valider l'état et branché avec la patte RB0 de PIC.

PATTE 2 : Vin(+) entrée analogique.

PATTE 3 : Vin(-) branché avec le diviseur de tension

PATTE 4 : La masse du convertisseur

PATTE 5 : Vref du convertisseur branché au deuxième diviseur.

PATTE 6 : D0 la sortie du signal convertie, branché avec PIC (Rb2...Rb5).

PATTE 7 : CLK (Clock) pour générer les impulsions, elle est branchée avec Rb1 du PIC.

PATTE 8 : Le Vcc du convertisseur.

Ce convertisseur peut fonctionner en mode asymétrique ou différentiel. L'entrée de tension analogique différentielle permet d'augmenter la réjection en mode commun et en compensant la valeur de tension analogique nulle d'entrée. En outre, l'entrée de tension de référence peut être ajustée pour permettre de coder un laps de tension analogique inférieure à l'ensemble des 8 bits de résolution, le ADC0831 lit une tension entre 0 et 5 volts et retournera une valeur brute qui correspond à un intervalle de 0 à 255.

On va appliquer cette loi pour avoir la valeur de sortie (0 à 255).

$$V_{ref} \longrightarrow 255 \qquad s = \frac{V_{in(+)} - V_{in(-)}}{V_{ref}} 255$$

$$V_{in(+)} - V_{in(-)} \longrightarrow s$$

Vref	Vin(+)	Vin(-)	Sortie (s) en Hexadecimle
3	3	1	AA
3	2	1	55
5	2.5	0	80

Tableau 4.1 : exemples des calculs de conversion

Chapitre 4 : L'étude Pratique

2.5 Le circuit de diviseur de tension

Cette partie de circuit permet d'assurer une tension de référence qu'elle utilise dans la partie de conversion.

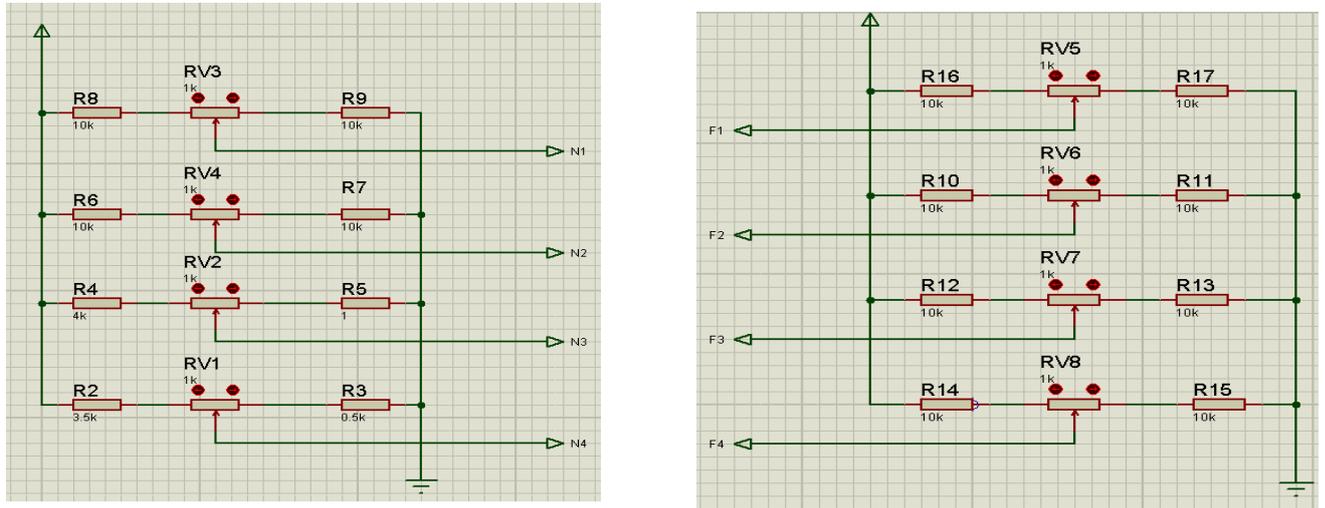


Figure 4.7 : le diviseur de tension

Calcule de N1, N2, N3, N4 pour obtenir la valeur de V_{in} des quatre convertisseurs respectivement, par exemple :

$$N1 = \frac{RV3 + R9}{R8 + RV3 + R9}$$

Dans la deuxième partie en va calculer F1, F2, F3, F4 pour obtenir la valeur de V_{ref} des quatre convertisseurs respectivement, par exemple :

$$F1 = \frac{RV5 + R17}{R16 + RV5 + R17}$$

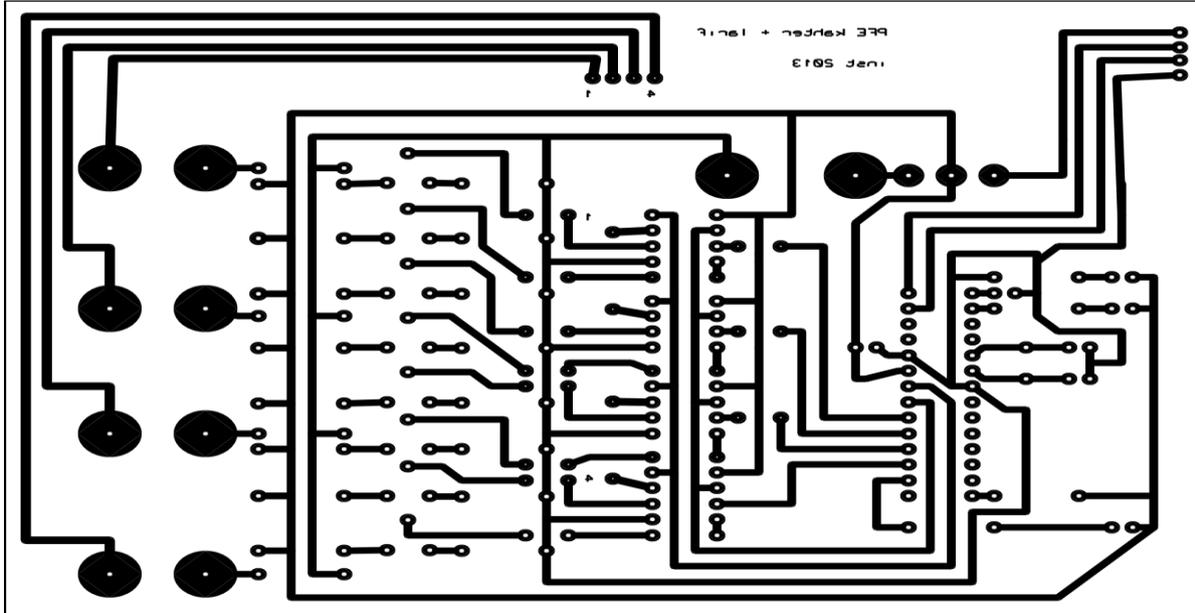
3. Les étapes de réalisation pratique de la carte

3.1 Le typon :

Pour préparer le typon on a recours au même logiciel de simulation qui est ISIS produit par la société LABCENTER ELECTRONICS.

Chapitre 4 : L'étude Pratique

L'ISIS est un éditeur de création de schémas électroniques ; il fait de même toutes les manipulations de simulation analogique et logique afin d'atteindre un résultat bien précis. Un typon est une image du circuit imprimé. Les pistes sont en noir et le reste doit être transparent à la lumière. L'idéal est d'imprimer avec une imprimante jet d'encre sur du papier transparent spécial jet d'encre. Il faut régler l'impression sur 'papier photo épais' pour avoir un maximum d'encre sur le papier et obtenir un noir plus opaque. La photocopieuse et l'imprimante laser convient aussi.

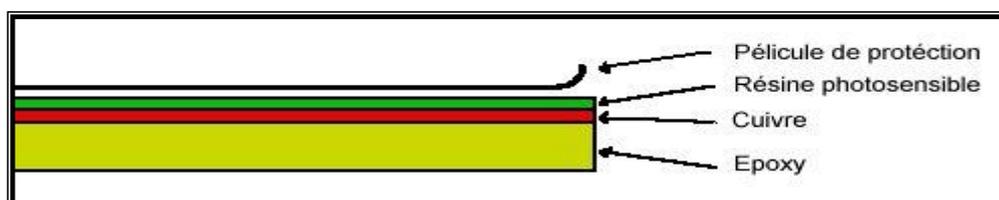


3.2 L'utilisation pratique du circuit

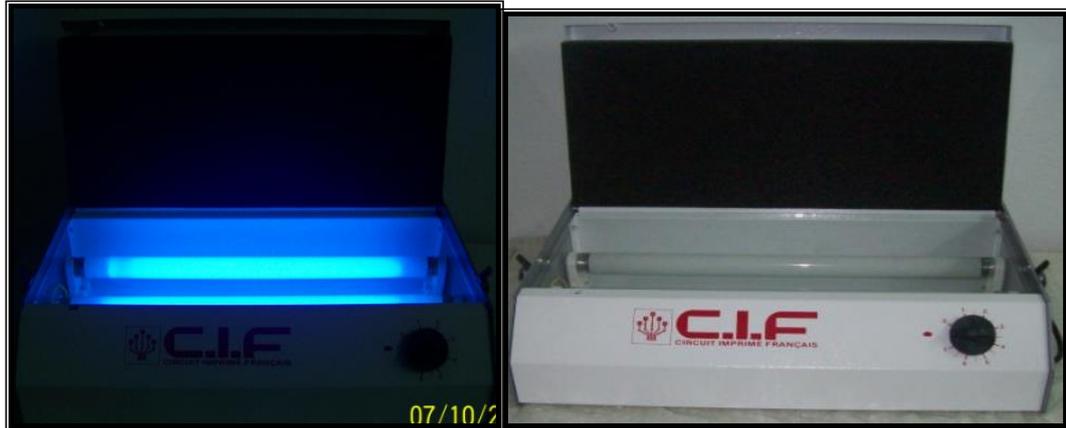
a. Insolation :

Le temps d'insolation est très important. Si ce dernier est trop long les rayons UV passeront au travers les zones noires du typon, et inversement on aura du mal à révéler notre plaque. Les étapes à suivre pour réaliser cette opération on suit la démarche ci-dessous :

- Découper la plaque d'époxy à la taille du typon en laissant une marge de 1 cm au moins.
- Positionner le typon dans le bon sens sur la vitre de l'insoleuse. (vue coté composant)
- Retirer la pellicule protectrice de la plaque (au dernier moment).
- Poser cette plaque avec le coté vert sur le typon dans l'insoleuse et fermer le capot.
- Insoler environ 1 à 5 minutes.



Chapitre 4 : L'étude Pratique



b. Révélation :

Cette étape va dissoudre la résine qui a été exposée aux UV. Ce qui mettra le cuivre à nu et les pistes resteront protégés par la résine verte. Préparation du révélateur : Ce révélateur pourra vous servir pour une bonne dizaine de circuits.

Utilisation du révélateur :

- Verser votre révélateur dans une cuvette en plastique.
- Plonger la plaque qui vient d'être insolé.
- Rincer soigneusement la plaque sous le robinet en frottant avec les doigts jusqu'à ce qu'elle ne soit plus 'grasse' au toucher.
 - Voila... elle est prête à être gravé. Notez que la résine protège encore les pistes.

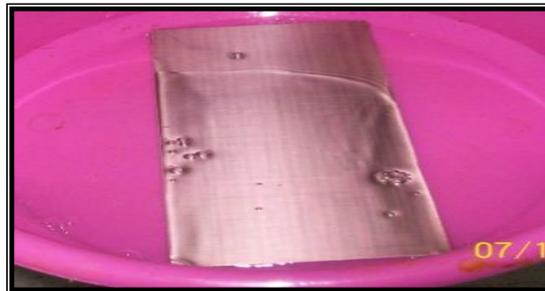


Figure 4.10 : Schéma de la carte dans le révélateur

c. Gravure :

La gravure consiste à plonger le circuit dans un bain d'acide (Perchlorure de fer). Le cuivre mis à nu lors de la révélation sera dissous. Seules les pistes protégées par la résine resteront.

Utilisation de graveur :

- Plonger la plaque d'époxy dans le bain d'acide. Temps : 5 à 30min suivant la graveuse...

Chapitre 4 : L'étude Pratique

- Lorsque tout le cuivre a disparu, sortez le circuit et rincez le sous l'eau.
- Frotter les pistes avec de l'acétone ou de l'alcool à brûler pour retirer la résine et ainsi apparaît les pistes de cuivres.

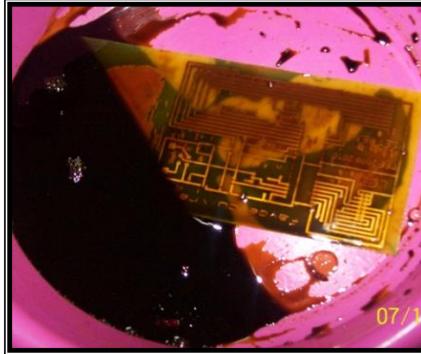


Figure 4.11 : Schéma de la carte dans le Perchlorure de fer

d. Perçage :

L'opération de perçage est l'ultime étape dans la réalisation d'un circuit imprimé, si l'on fait abstraction de la mise en place des composants et du soudage.

- Une simple petite perceuse alimenté en 12 V suffit.



Figure 4.12: Schéma d'une perceuse

e. Test du circuit imprimé:

Si vous doutez de votre carte, utiliser un Ohmmètre ou un testeur de continuité pour vérifier :

- La continuité des pistes.
- L'absence de court-circuit.

f. Soudure :

L'opération de soudure est faite à l'aide d'un fer à souder et l'étain. On commencera par:

Des composants neutres: supports de Circuit .Intégrés, connectiques...

- **Passifs :**

Chapitre 4 : L'étude Pratique

- Résistances. Utiliser un Gabarit de pliage et donner vous un sens de lecture des résistances. Les bandes de couleurs se lisant de la Droite vers la Gauche et de Haut en Bas.
- Condensateurs. Pour les plastiques ou non polarisé, donnez-vous aussi un sens de lecture.
- **Actifs** : Diodes, transistors, circuits intégrés si vous ne mettez pas de support.

4. Le traçage des signaux :

La visualisation des signaux en même temps :

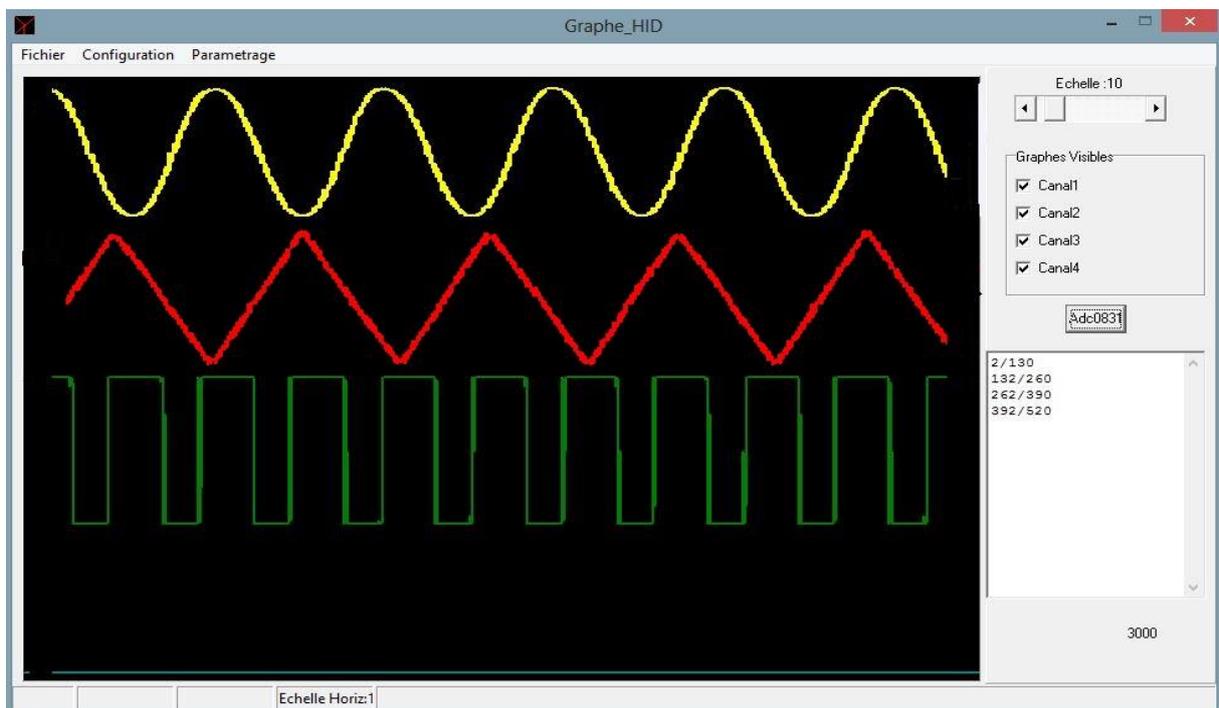
On trace les signaux qui sont différents en forme et en amplitude à l'entrée de notre circuit.

Entrée 1 : signal sinusoïdale d'amplitude 4.1 càd à la fréquence de 663Hz.

Entrée 2 : signal triangle d'amplitude 4.1 càd la fréquence de 663Hz.

Entrée 3 : signal carré d'amplitude 6.5 càd à la fréquence de 671Hz.

Entrée 4 : aucun signal.



Chapitre 4 : L'étude Pratique

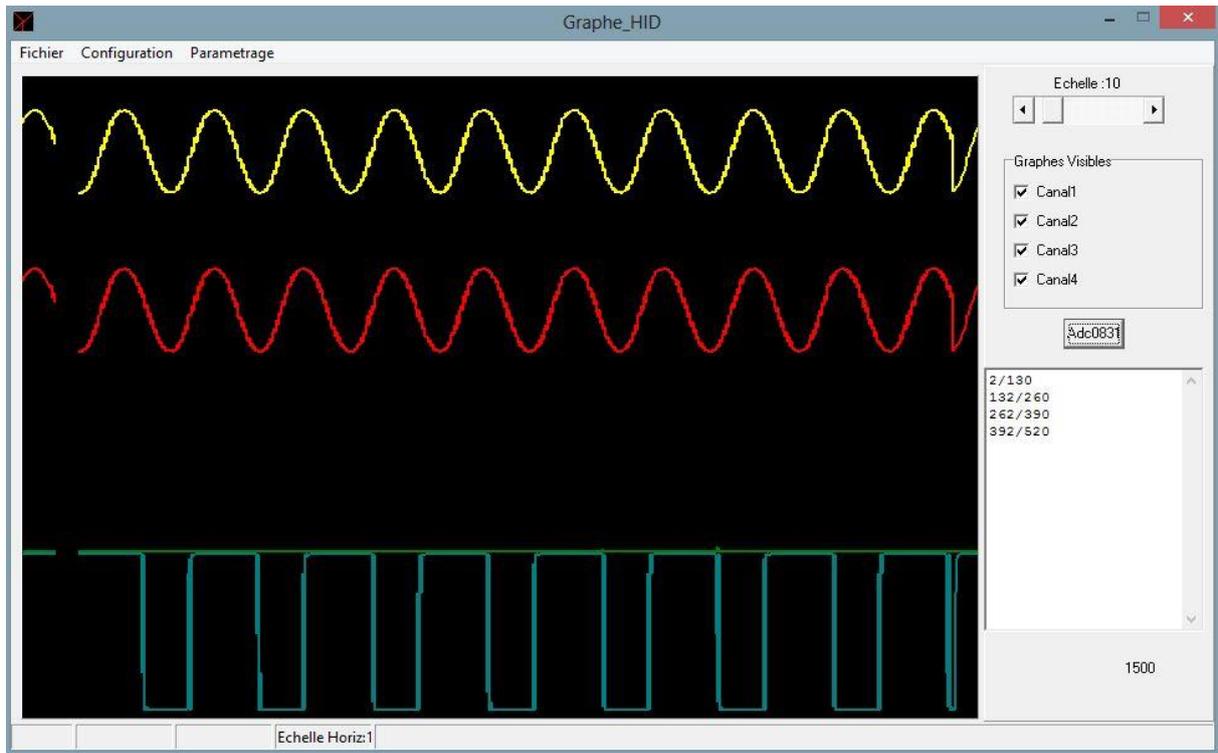
On trace les signaux qui sont différents en forme et en amplitude à l'entrée de notre circuit.

Entrée 1 : signal sinusoïdale d'amplitude 2.6 càd à la fréquence de 660Hz.

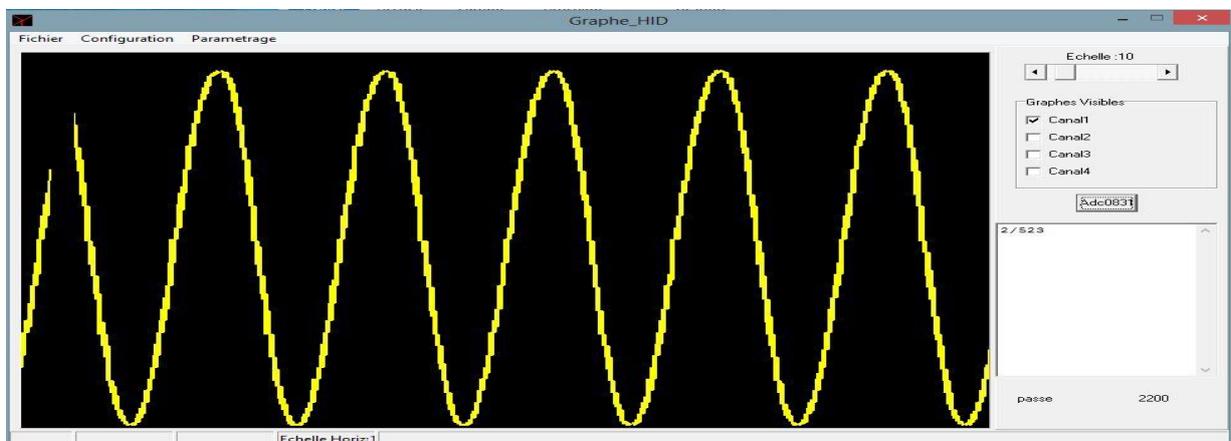
Entrée 2 : signal triangle d'amplitude 2.6 càd à la fréquence de 660Hz.

Entrée 3 : aucun signal.

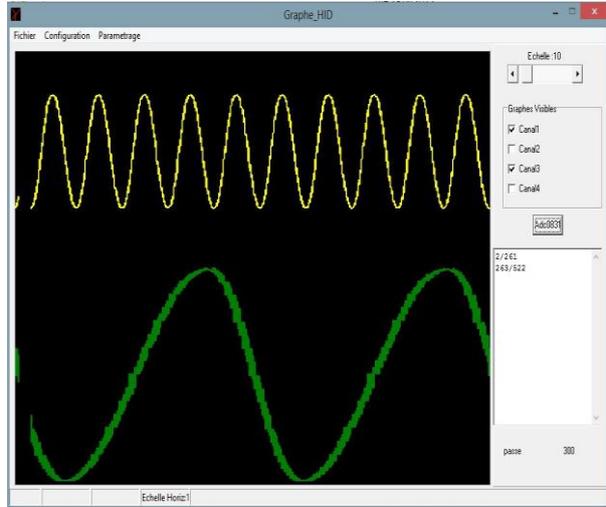
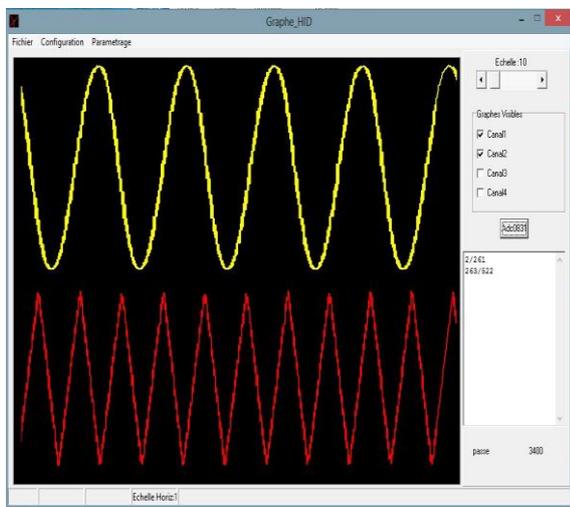
Entrée 4 : signal carré d'amplitude 4.4 càd à la fréquence de 663Hz.



- Les mêmes signaux précèdent mais chacun à la fois :



- **Les mêmes signaux précèdent mais deux signaux à la fois :**



5. Les avantages et les inconvénients du projet

Les avantages :

- ✓ Ce périphérique coûte moins cher par rapport aux appareils qui ont le même rôle.
- ✓ On a la possibilité de stocker les données (oscilloscope à mémoire) et de les envoyer par l'utilisation des moyens de communication (Internet).
- ✓ La taille de ce périphérique est plus petite par rapport aux appareils utilisés dans ce domaine.
- ✓ La possibilité de traiter et de visualiser quatre (4) signaux en même temps.

Les inconvénients :

- ✓ Il faut connaître la caractéristique des signaux avant de les injecter.
- ✓ Pratiquement on ne peut pas traiter des signaux qui dépassent 5V crête à crête et ça revient aux caractéristiques du PIC.
- ✓ Un autre inconvénient c'est qu'on ne peut pas lire la fréquence sur l'interface car il faut plusieurs points.
- ✓ Il faut amplifier les signaux qui sont très petits par un circuit d'amplification pour qu'on puisse les visualiser sur notre interface

Chapitre 4 : L'étude Pratique

6. Conclusion

Ce dernier chapitre a été entamé par une réalisation pratique en exploitant une méthodologie d'analyse hiérarchique. Cette analyse a été suivie d'une explication détaillée des différentes étapes qu'il faut suivre pour la réalisation.

Conclusion Générale

Conclusion générale

Notre projet a consisté à réaliser un périphérique de PC qui permet d'acquérir, de convertir, de stocker, de traiter et de tracer des signaux quelconques en utilisant un PC.

A travers ce travail nous avons pu comprendre une partie du fonctionnement et d'avoir une idée générale sur les domaines d'utilisation des microcontrôleurs.

Ainsi la réalisation pratique des montages durant ce projet nous a montré que l'étude théorique et l'étude par simulation étaient très proches de la réalité pratique, sauf que la théorie ne peut pas tenir compte de tous les aléas relatifs à la pratique.

Liste des Composants

LISTE DES COMPOSANTS

B1 : alimentation 5V

SW1: connecteurs

J1: Porte USB

D1: LED 1 Vert

D2: LED 2 Rouge

D3 : LED 3 Vert

U1=U2=U3=U4 : Des convertisseurs ADC0831

U6 : PIC 18F2550

X1 : Quartz 12 MHz

C1 : Condensateur 15 pF

C2 : Condensateur 15 pF

C3 : Condensateur chimique 47 μ f

C4 : Condensateur 1 nF

RV1= RV2= RV3= RV4=RV5=RV6=RV7=RV8 : Résistances variable 1 K Ω

R1= R5 :Résistance 1K Ω

R2= Résistance 3.5K Ω

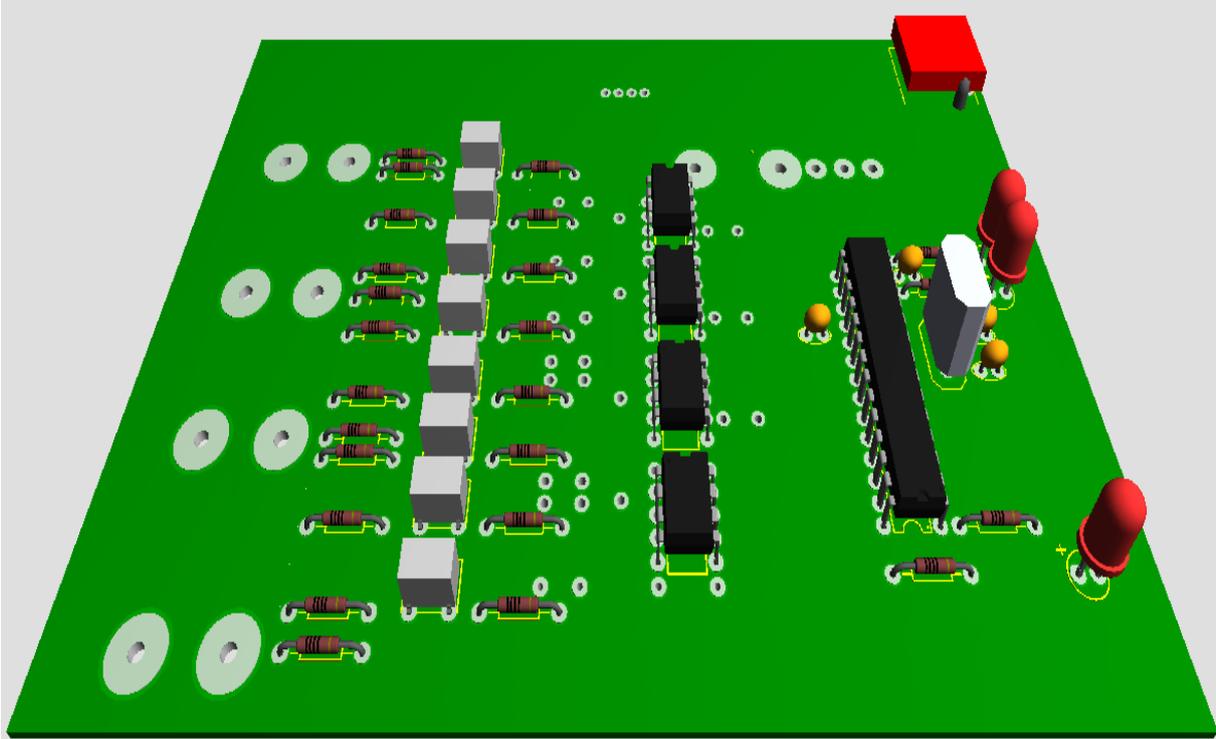
R3= 0.5K Ω

R4= 4K Ω

R6=R7=R8=R9=R10=R11=R12=R13=R14=R15=R16=R17: Resistance 5K Ω

Annexe

Vue 3D de la carte de commande Final



La structure interne du 18F2550

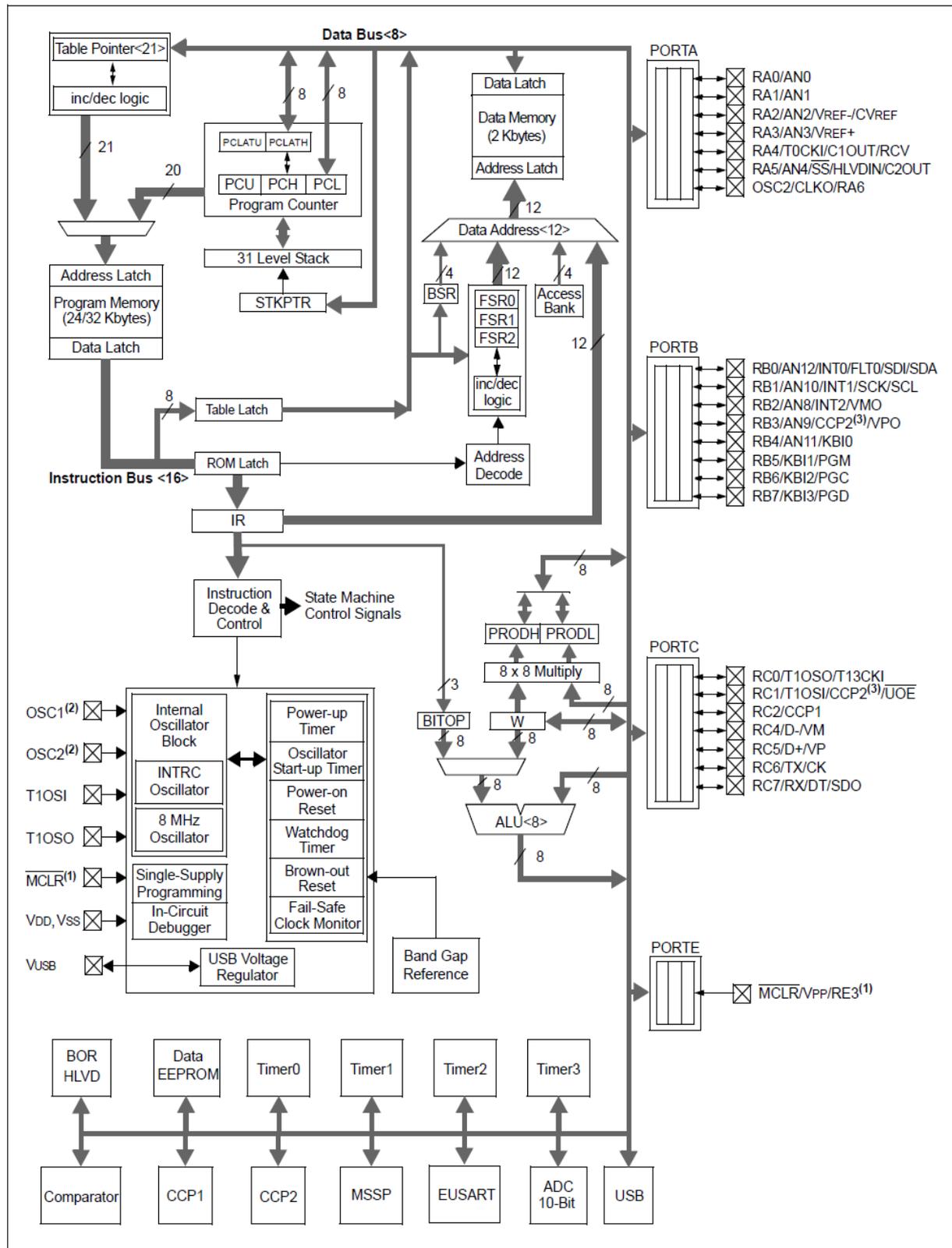


Figure 2 : schéma de la structure interne du 18F2550, issu de la datasheet de Microchip

ANNEXE

Schémas de connexion

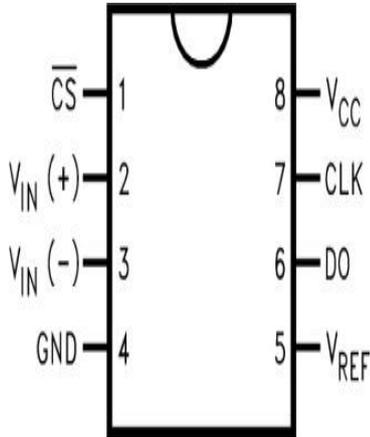


Figure 5.1 ADC0831-N Single DIF

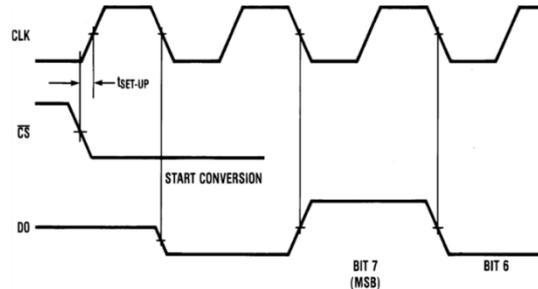


Figure 5.2 ADC0831-N Start Conversion Timing

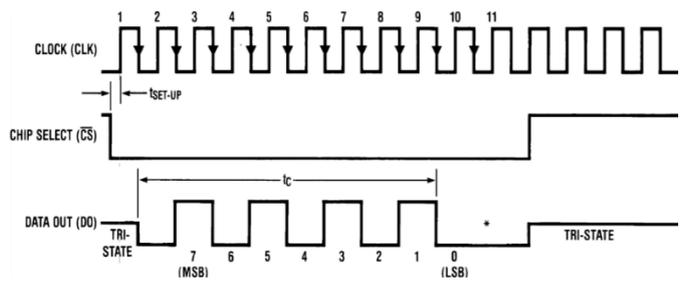


Figure 5.3 ADC0831-N Timing

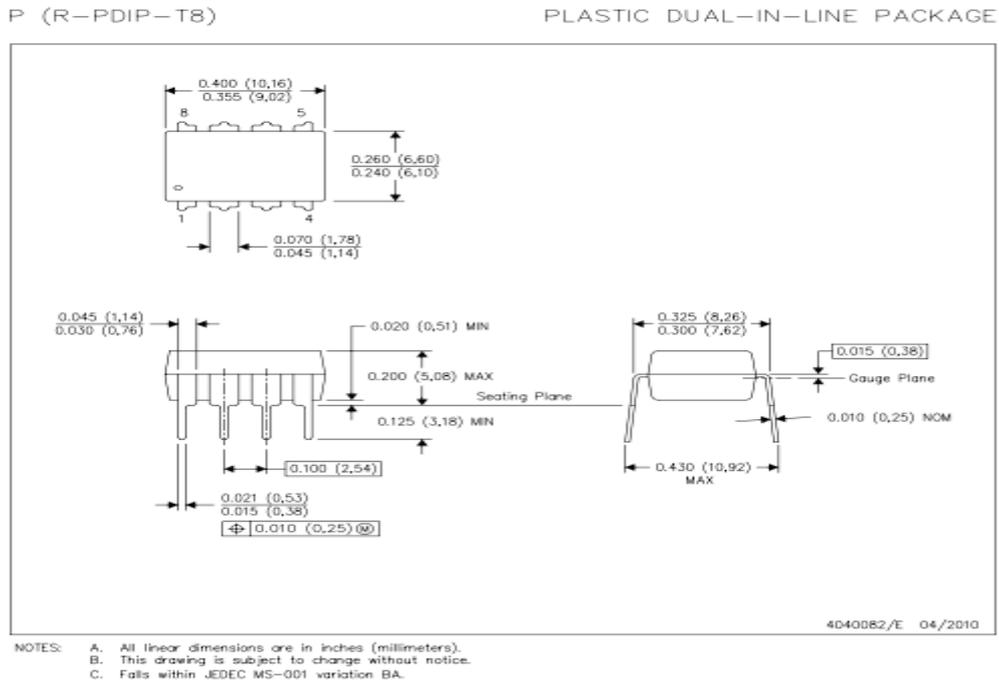


Figure 5.4 Les dimensions d'un convertisseur ADC0831

Bibliographique

Recherche Bibliographique

✓ Les mémoires de Fin d'étude :

1/ ACQUISITION ET TRAITEMENT DE SIGNAUX PHYSIOLOGIQUES présenté par :
LEKEHAL Réda et SOUADJI Mohammed. Encadreur :Nemmiche A.

2/ REALISATION D'UN STORE AUTOMATIQUE A BASE DE PIC 16F876 COMMANDE
PAR LES CONDITIONS CLIMATIQUES présenté par : BOUDALIA Kamel et Touati
Oussama. Encadreur H.BECHAR.

✓ Les ouvrages :

1/ Christian Tavernier, Les microcontrôleurs PIC, recueil d'application 3ème édition, Dunod,
Paris 2005.

2/ Le magazine « Electronique pratique ».

Numéro 176 –décembre 1993.

✓ Les sites internet :

<http://www.tavernier-c.com/>

www.microchip.com/wwwproducts/Devices.aspx

www.protechno.com

www.bigonoff.com

<http://forums.futura-sciences.com/electronique/117415-pic-18f2550-l-usb>