



Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique
Option : Télécommunications

Réf:.....

**Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:**

MASTER

Thème

**Réalisation d'un oscilloscope
numérique portatif à affichage
sur écran PC via le port USB**

**Présenté par :
Bekhouché Salah Eddine
Soutenu le : 2 Juin 2013**

Devant le jury composé de :

Mme. Terghini Ouarda

M. Kahoul Nadir

M. Diabi Fathi

MAA

MAA

MAB

Présidente

Encadreur

Examinateur

Année universitaire : 2012 / 2013

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique
Option : Télécommunications

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:

MASTER

Thème

**Réalisation d'un oscilloscope numérique portatif
à affichage sur écran PC via le port USB**

Présenté par :

Bekhouche Salah Eddine

Avis favorable de l'encadreur :

M.Kahoul Nadir

signature

Avis favorable du Président du Jury

Mme. Terghini Ouarda

Signature

Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique
Option : Télécommunications

Thème :

**Réalisation d'un oscilloscope numérique portatif
à affichage sur écran PC via le port USB**

Proposé par : Bekhouche Salah Eddine

Dirigé par : M. Kahoul Nadir

Résumé

L'oscilloscope est un appareil de mesure indispensable dans tous les laboratoires. Actuellement l'oscilloscope numérique remplace presque complètement l'oscilloscope analogique. Pour cela on a pensé à la réalisation d'un oscilloscope numérique portatif. L'oscilloscope réalisé dans ce projet est connectable au PC à l'aide du Bus universel en série (USB). L'interface utilisateur, réalisé sous l'environnement C#, contrôle la communication avec la carte d'acquisition dont le cœur est le PIC 18F4550. La fréquence du signal acquit est limité par le temps de conversion du PIC.

راسم الإهتزازات هو جهاز قياس ضروري في كل المخابر. سم الإهتزاز الرقمي عوض تقريبا راسم الإهتزازات التماثلي. من أجل هذا تم التفكير في إنجاز راسم إهتزاز رقمي محمول. الجهاز المنجز في هذا المشروع هو جهاز موصل ب الحاسوب عن طريق ناقل متسلسل عام (USB). واجهة المستعمل المنجزة في بيئة سي شارب تتحكم في التواصل PIC18F4550. تردد الإشارة الملتقطة محدود بزمن تحويل الـ PIC.

Dédicace

Je dédie ce modeste travail :

À ceux qui m'ont indiqué la bonne voie, À ceux qui attendent patiemment le fruit de leur éducation. À qui n'ont jamais cessé de se donner corps et âme pour ma réussite.

À mes parents : **Abdelhamid** et **Abdallah Fatiha**, pour leurs encouragements, leur affection, leurs conseils et leurs sacrifices.

J'espère que vous trouverez dans ce travail ma profonde reconnaissance et mon grand respect pour vous.

À vous mes grands-mères **Yamina** et **Zohra**, pour vos prières et vos amours.
Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

À mes chers et adorables frères et sœurs :
Yasmina, Khaled, Nawel, Ali, Hakima, Salwa et **Chams eddine**
pour leur précieuse assistante
et pour leur encouragement, quand j'avais besoin de soutien moral.

Je dédie aussi ce travail à toute la famille **Bekhouché** et la famille **Abdellah**, mes oncles et tantes, cousins et cousines.

À tous mes amis de près ou de loin

À tous ceux qui me connaissent et m'aiment

À tous (e) les enseignants (e) et les responsables du département de génie électrique

Enfin, je le dédie à tous ceux que j'aime et j'apprécie.

BEKHOUCHE Salah Eddine

Remerciements

Tout d'abord, j'implore le merci de **Dieu** tout puissant de m'avoir accordé sa bénédiction et sa protection pour accomplir cette responsabilité.

Je tiens par la suite, à exprimer ma gratitude et ma profonde reconnaissance à l'égard de mon directeur de recherche **M. KAHOUL Nadir** qui m'a donné la chance et la confiance d'élaborer mon projet de recherche et pour son assistant, ses conseils et ses suggestions précieuses et enrichissantes durant la réalisation de cette initiative.

Tous mes remerciements s'adressent également aux membres du jury qui me font l'honneur d'évaluer ce travail

J'adresse également mes sincères remerciements et ma reconnaissance à tous les enseignants qui m'ont soutenu, me guider dans mon travail avec un œil critique et avisé, plus précisément: **Dr. OUAFI Abdelkrim, Dr. GUESBAYA Tahar, M.RAHMANI Nacer Eddine** et **M. Allaoui Abdel Hamid** et d'une façon toute particulière, un tout grand merci à mon frère **Dr. BEKHOUCHE Khaled**.

J'ai aussi l'honneur et le plaisir de remercier tous les enseignants qui m'ont formé et pris en charge depuis ma première année universitaire « Tronc commun sciences et techniques ST », jusqu'à ma dernière année universitaire « Electronique ».

Sans oublier, Je remercie sincèrement toutes personnes ayant pu m'aider de près ou de loin dans la réalisation de ce modeste travail, surtout mes amis : **Fares, Soufiane, Aicha et Oumaima**.

Je souhaite remercier très chaleureusement les responsables du laboratoire, en particulier : **M.HAMZA et M. Abdellatif**, pour m'avoir permis d'effectuer mes travaux de recherche et pour tous ce qu'ils font pour gérer parfaitement leur service.

List des Figures

Fig I.1 : Composantes X, Y et Z d'un signal affiché	2
Fig I.2 : Le panneau avant d'un oscilloscope analogique	3
Fig I.3 : Le panneau avant d'un oscilloscope numérique.....	4
Fig I.4 : Données scientifiques capturées par un oscilloscope	4
Fig I.5 : L'oscilloscope analogique visualise le signal	5
Fig I.6 : L'oscilloscope numérique échantillonne le signal et reconstruit une image	5
Fig I.7 : Bloc diagramme de l'oscilloscope analogique	7
Fig I.8 : Le déclenchement stabilise un signal répétitif	7
Fig I.9 : Bloc diagramme de l'oscilloscope numérique	9
Fig I.10 : Fréquence et période	10
Fig II.1 : USB 1 logo	15
Fig II.2 : USB 2 logo	16
Fig II.3 : USB 3 logo	16
Fig II.4 : Câble USB.....	16
Fig II.5 : Composition d'un câble USB.....	17
Fig II.6 : Connecteurs.....	18
Fig II.7 : connecteurs USB type A et B.....	19
Fig II.8 : Full Speed Connections.....	19
Fig II.9 : Low Speed Connections.....	20
Fig II.10 : Exemple de Circuit de sortie	20
Fig II.11 : Code NRZI	21
Fig II.12 : Icône USB	22
Fig II.13 : Architecture d'un système USB avec Hub	22

List des Figures

Fig III.1 : Brochage du PIC 18F4550.....	35
Fig III.2 : Architecture interne du PIC 18F4550	37
Fig III.3 : Exemples de configuration de l'horloge	38
Fig III.4 : Synoptique de la circuiterie d'horloge sur un PIC 18F4550 gérant l'USB	39
Fig III.5 : Synoptique de la circuiterie Reset	40
Fig III.6 : Synoptique général du CAN pour le 18F4550.....	45
Fig III.7 : Les cycles de conversion totale	47
Fig III.8 : Synoptique général des interruptions sur PIC 18F4550	49
Fig III.9 : Architecture simplifiée d'un bus USB	49
Fig III.10 : Architecture interne USB du PIC 18F4550	51
Fig IV.1 : Adaptateur RS232/ICSP pour la programmation des PICs.....	54
Fig IV.2 : Fenêtre principale du Logiciel MikroC pro.....	56
Fig IV.3 : Fenêtre principale du Logiciel PICPgm	56
Fig IV.4 : Bloc diagramme de système	57
Fig IV.5 : Circuit de protection	58
Fig IV.6 : Atténuateur BF	59
Fig IV.7 : Circuit de couplage.....	59
Fig IV.8 : Circuit de sélection de calibre	60
Fig IV.9 : Circuit d'acquisition et de communication.....	61
Fig IV.10 : L'organigramme principal du PIC	62
Fig IV.11 : L'organigramme d'activation de Module USB du PIC.....	63
Fig IV.12 : Le drapeau USBIF	64
Fig IV.13 : L'organigramme de la communication USB entre le PIC et le PC.....	64

List des Figures

Fig IV.14 : L'organigramme du trigger	65
Fig IV.15 : L'organigramme du conversion et communication.....	66
Fig IV.16 : Circuit d'alimentation.....	67
Fig IV.17 : L'interface utilisateur	68

List des Tableaux

Tab II.1 : Le brochage des connecteurs USB	19
Tab II.2 : List des types de descripteur USB	23
Tab II.3 : List des champs de descripteur d'appareil	24
Tab II.4 : List des champs de descripteur de configuration	24
Tab II.5 : List des champs de descripteur d'interface	25
Tab II.6 : List des champs de descripteur de terminaison	25
Tab II.7 : Les différents PIDs	27
Tab III.1 : Caractéristiques principales du pic 18F4550	35
Tab III.2 : Rôle des bits FOOSC0 à FOOSC3 pour le 18F4550	38
Tab III.3 : Bits de contrôle de configuration de port A/D	43
Tab IV.1 : Connexion du PIC au connecteur ICSP	55
Tab IV.2 : RS-232 db9 female	55
Tab IV.3 : Les états de fonctionnement du circuit de couplage	60
Tab IV.4 : Les états de fonctionnement du calibre	61

Liste des Abréviations

- AC : Alternating Current.
- DC : Direct Current.
- GND : GrouND.
- CRT : Cathode Ray Tube.
- CAN : Convertisseur Analogique Numérique.
- USB : Universal Serial Bus
- PC : Personal Computer
- AWG : American Wire Gauge
- PVC : PolyVinyl Chloride
- EOP : End Of Packet
- TTL : Transistor–Transistor Logic
- NRZI : Non-Return-To-Zero
- VID : Vendor ID
- PID : Product ID
- CI : Circuit Intégré
- ENDP : ENDPoint
- CRC : Cyclic Redundancy Check
- SOF : Start Of Frame
- HID : Human Interface Device
- PCI : Peripheral Component Interconnect
- WDM : Windows Driver Model
- CMOS : Complementary Metal–Oxide–Semiconductor
- PIC : Peripheral Interface Controller
- RISC : Reduced Instructions Set Computer
- WDT : Watch Dog Timer
- TOSC : Time of the OSCillator
- FOOSC : Frequency of the OSCillator
- PLL : Phase-locked loop
- IT : Interruption
- SIE : Serial Interface Engine
- ICSP : In-Circuit Serial Programming
- PWM : Pulse-Width Modulation

Résumé

Résumé

L'oscilloscope est un appareil de mesure indispensable dans tous les laboratoires. Actuellement l'oscilloscope numérique remplace presque complètement l'oscilloscope analogique. Pour cela on a pensé à la réalisation d'un oscilloscope numérique portatif. L'oscilloscope réalisé dans ce projet est connectable au PC à l'aide du Bus universel en série (USB). L'interface utilisateur, réalisé sous l'environnement C#, contrôle la communication avec la carte d'acquisition dont le cœur est le PIC 18F4550. La fréquence du signal acqut est limité par le temps de conversion du microcontrôleur.

Mot clés: Oscilloscope, USB, CAN, Microcontrôleur.

Abstract

The oscilloscope is an essential measurement apparatus in all labs. Nowadays the analogic oscilloscope is almost replaced by the numeric oscilloscope. That's why in this project we have thought to carry out a Universal Serial Bus (USB) connectable handheld numeric oscilloscope. The user interface is implemented in the C# environment, controls the communication with the acquisition card which is based on the PIC 18F4550. The frequency of the acquired signal is limited by the conversion time of the microcontroller.

Keywords: Oscilloscope, USB, ADC, Microcontroller.

سم الإهتزازات هو جهاز قياس ضروري في كل المخابر. في الوقت الحالي راسم الإهتزاز الرقمي عوض تقريبا راسم الإهتزازات التماثلي. من أجل هذا تم التفكير في إنجاز راسم إهتزاز رقمي محمول. الجهاز المنجز في هذا المشروع هو جهاز موصل ب الحاسوب عن طريق ناقل متسلسل عام (USB). واجهة المستعمل المنجزة في بيئة سي شارب تتحكم في PIC18F4550. تردد الإشارة الملتقطة محدود بزمن تحويل المتحكم

الدقيق.

كلمات دلالية: راسم الإهتزازات، ناقل متسلسل عام، محول إشارة، متحكم دقيق.

Sommaire

Introduction Générale	A
------------------------------------	----------

Chapitre I : L'oscilloscope

I.1 - Introduction	2
I.2 - L'oscilloscope.....	2
I.2.1 - Types d'oscilloscopes : analogique et numérique	5
I.3 - L'oscilloscope analogique	6
I.3.1 - Principe de fonctionnement	6
I.3.2 - Fonctions de base	7
I.4 - L'oscilloscope numérique.....	8
I.4.1 - Principe de fonctionnement	8
I.4.2 - Bloc diagramme	9
I.5 - Définitions	9
I.5.1 - Fréquence et période.....	9
I.5.2 - Bande passante	10
I.5.3 - Temps de montée	10
I.5.4 - Sensibilité verticale	10
I.5.5 - Résolution verticale	11
I.5.6 - Impédance d'entrée	11
I.5.7 - Couplage	11
I.5.8 - Vitesse de balayage	11
I.5.9 - Base de temps	12
I.5.10 - Précision du gain	12
I.5.11 - Précision de la base de temps ou précision horizontale.....	12
I.5.12 - Fréquence d'échantillonnage.....	12
I.6 - Conclusion.....	13

Chapitre II : Bus universel en série

II.1 - Introduction	15
II.2 - Normes USB	15
II.2.1 - USB 1(Full ou Low speed).....	15
II.2.2 - USB 2.0 (High Speed).....	15
II.2.3 - USB 3.0 (Super Speed)	16
II.3 - Connectique.....	16
II.3.1 - Description du Câble USB	16
II.3.2 - Composition du câble USB	17
II.3.3 - Connecteurs USB	17
II.3.3.a - Connecteur série A	18
II.3.3.b - Connecteur série B	18
II.3.4 - Brochage.....	19
II.3.5 - Caractéristiques électriques.....	19
II.3.5.a - Etats de D+ et D-	19
II.3.5.b - Emetteur sur D+ et D-	20
II.3.5.c - Récepteur sur D+ et D-.....	20

Sommaire

II.3.5.d - Alimentation.....	20
II.3.5.e - La signalisation.....	21
II.4 - Architecture d'un système.....	22
II.5 - Les descripteurs	23
II.5.1 - Définition d'un descripteur	23
II.5.2 - Rôle des descripteurs.....	23
II.5.3 - Les types de descripteur	23
II.5.3.a - Descripteur d'appareil.....	24
II.5.3.b - Descripteur de configuration.....	24
II.5.3.c - Descripteur d'interface.....	25
II.5.3.d - Descripteur de terminaison.....	25
II.6 - Les protocoles USB.....	25
II.6.1 - Les champs de paquet USB ordinaires.....	26
II.6.2 - Les types de paquet USB	27
II.6.2.a - Les paquets jetons.....	27
II.6.2.b - Les paquets de données	28
II.6.2.c - Les paquets poignée de mains	28
II.6.2.d - Les paquets début de trame (SOF)	28
II.7 - Transfert de données	29
II.7.1 - Transfert en mode Contrôle.....	29
II.7.2 - Transfert en mode Interrupt.....	29
II.7.3 - Transfert en mode Isochrone.....	29
II.7.4 - Transfert en mode Bulk.....	29
II.8 - PID/VID	29
II.8.1 - Introduction au Product ID et au Vendor ID.....	29
II.8.2 - Normalisation des PID/VID	30
II.9 - Les fichiers INF/SYS	29
II.9.1 - Les fichiers d'extension INF	30
II.9.1.a - Définition.....	30
II.9.1.b - Fonctionnement des points INF	30
II.9.2 - Les drivers .SYS.....	31
II.9.2.a - Définition d'un driver .SYS	31
II.9 - Conclusion.....	31
Chapitre III : Le microcontrôleur (PIC)	
III.1 - Introduction.....	33
III.2 - Le Microcontrôleur (PIC)	33
III.2.1 - Définition d'un PIC.....	33
III.2.2 - Classification des PICs de Microchip.....	33
III.2.3 - Identification d'un Pic.....	33
III.2.4 - Le choix d'un PIC	34
III.2.5 - Caractéristiques principales du pic 18F4550	35
III.3 - Brochage du 18F4550	35
III.4 - Architecture générale du PIC.....	37

Sommaire

III.4.1 - Architecture interne du PIC 18F4550	37
III.4.2 - Les horloges et le Reset	38
III.4.2.a - Horloge.....	38
III.4.2.b - Type d'horloge externe	38
III.4.2.c - Reset	39
III.4.3 - Les timers.....	40
III.4.4 - Les convertisseurs analogiques/numériques	40
III.4.4.a - Le CAN du PIC 18F4550.....	41
III.4.5 - Les interruptions et leur gestion.....	47
III.4.5.a - L'interruption	47
III.4.5.b - Les différentes causes de déclenchement d'interruption	48
III.4.6 - Le bus USB	49
III.4.6.a - Architecture simplifiée d'un bus USB.....	49
III.4.6.b - Le microcontrôleurs PIC et USB	50
III.4.6.c - Registres utilisable pour la fonction USB	51
III.5 - Conclusion	52
Chapitre IV : Réalisation pratique	
IV.1 - Introduction.....	54
IV.2 - Programmeur PIC	54
IV.2.1 - Les signaux ICSP.....	54
IV.2.2 - Compilateur "MikroC PRO" pour PIC	55
IV.2.3 - Logiciel de programmation.....	56
IV.3 - Partie Hard	57
IV.3.1 - Circuit d'entrée	58
IV.3.1.a - La résistance fusible.....	58
IV.3.1.b - L'atténuateur BF.....	58
IV.3.1.c - Le circuit de couplage	59
IV.3.2 - Circuit de sélection de calibre.....	60
IV.3.3 - Circuit d'acquisition et de communication.....	61
IV.3.3.a - L'organigramme principal du PIC.....	61
IV.3.3.b - L'activation de module USB	63
IV.3.3.c - L'organigramme de la communication USB PC-PIC.....	64
IV.3.3.d - L'organigramme du trigger.....	65
IV.3.3.e - L'organigramme du conversion et communication	65
IV.3.4 - Circuit d'alimentation	66
IV.4 - Partie Soft	67
IV.5 - Conclusion	69
Conclusion Générale	70

Introduction Générale

En physique et en sciences de l'ingénieur, mesurer consiste à comparer une grandeur physique qui caractérise un objet (ou un événement) avec celle de même nature choisie comme unité de mesure. La valeur numérique de la grandeur mesurée est le nombre qui fixe la relation entre la grandeur mesurée et l'unité de mesure choisie.

Pour un oscilloscope le signal mesuré est le plus souvent une tension électrique. L'oscilloscope est un appareil de mesure électronique de représentation graphique sur un écran d'une ou de plusieurs tensions variables en fonction du temps. Le traitement analogique de ces tensions est compliqué et le traitement numérique utilisant les convertisseurs analogique-numérique est devenu l'alternative. Un convertisseur analogique-numérique, est un montage électronique dont la fonction est de traduire une grandeur analogique en une valeur numérique (codée sur plusieurs bits), proportionnelle au rapport entre la grandeur analogique d'entrée et la valeur maximum du signal. Il existe plusieurs techniques pour convertir un signal analogique en signal numérique. Le CAN est l'élément principal de la chaîne d'acquisition.

La chaîne d'acquisition de donnée est l'ensemble des éléments nécessaires à la "capture" des données (analogiques ou numériques) à leur transmission jusqu'au récepteur et à l'utilisateur (homme ou machine) des données capturées, cet utilisateur peut vouloir utiliser ces données immédiatement ou les stocker pour les utiliser ultérieurement.

Pour faire communiquer un PC à une carte d'acquisition de données on utilise plusieurs techniques : bus ISA, bus PCI, port parallèle, port série, port IEEE 488, port USB, ...etc.

Le bus USB est basé sur une architecture de type série. Il s'agit toutefois d'une interface entrée-sortie beaucoup plus rapide que le port série standard. L'architecture qui a été retenue pour ce type de port est en série pour deux raisons principales : l'architecture série permet d'utiliser une cadence d'horloge beaucoup plus élevée qu'une interface parallèle, car celle-ci ne supporte pas des fréquences trop élevées (dans une architecture à haut débit, les bits circulant sur chaque fil arrivent avec des décalages, provoquant des erreurs) ; les câbles série coûtent beaucoup moins cher que les câbles parallèles.

Classiquement, les microcontrôleurs communiquent avec les ordinateurs personnels par le port série (RS232). Or, la tendance actuelle de l'évolution du matériel informatique force aux constats suivants : le port série (RS232) est voué à disparaître, et le port série est trop lent pour certaines applications. Nous nous sommes par conséquent fixés comme objectifs de maîtriser un microcontrôleur possédant un port USB [1, 2, 3], d'effectuer le développement totalement avec

Introduction Générale

des outils libres sous Windows et de développer un logiciel de base permettant de tester le microcontrôleur sous ce même système d'exploitation.

À fin de réaliser ce projet et éclairer l'utilité de ce dispositif, nous nous sommes interrogés sur certaines questions:

Qu'est-ce que l'oscilloscope? Ses types? Quelles sont les avantages de bus USB? Qu'est-ce qu'un PIC et quelles sont ses caractéristiques? Comment fabriquer un oscilloscope numérique portatif ?

Ce travail s'articule autour de quatre chapitres :

- Chapitre 1 : L'oscilloscope.
- Chapitre 2 : Le bus universel en série.
- Chapitre 3 : Le microcontrôleur (PIC).
- Chapitre 3 : Réalisation pratique.

Les références les plus intéressantes sur lesquelles nous avons travaillé sont :

- Patrick LESNE, «Mesures à l'aide de l'oscilloscope», 1998.
- Jan Axelson, «USB Complete The Developer's Guide 4th edition», 2009.
- Pascal Mayeux, « Apprendre la programmation des PIC High-Performance par l'expérimentation et la simulation », Dunod, 2010.

Nous passerons maintenant au premier chapitre dont nous parlerons de l'oscilloscope analogique et numérique.

CHAPITRE

I

L'oscilloscope

CHPITRE I : L'oscilloscope

1 - Introduction	2
2 - L'oscilloscope	2
2.1 - Types d'oscilloscopes : analogique et numérique	5
3 - L'oscilloscope analogique	6
3.1 - Principe de fonctionnement	6
3.2 - Fonctions de base	7
4 - L'oscilloscope numérique.....	8
4.1 - Principe de fonctionnement	8
4.2 - Bloc diagramme.....	9
5 - Définitions	9
5.1 - Fréquence et période.....	9
5.2 - Bande passante	10
5.3 - Temps de montée	10
5.4 - Sensibilité verticale	10
5.5 - Résolution verticale	11
5.6 - Impédance d'entrée	11
5.7 - Couplage	11
5.8 - Vitesse de balayage	11
5.9 - Base de temps	12
5.10 - Précision du gain.....	12
5.11 - Précision de la base de temps ou précision horizontale.....	12
5.12 - Fréquence d'échantillonnage	12
6 - Conclusion	13

I.1.Introduction

Un oscilloscope est un instrument de mesure destiné à visualiser un signal électrique, le plus souvent variable au cours du temps. Il est utilisé par de nombreux scientifiques afin de visualiser soit des tensions électriques, soit diverses autres grandeurs physiques préalablement transformées en tension au moyen d'un convertisseur adapté.

On distingue généralement les oscilloscopes analogiques qui utilisent directement un multiple de la tension d'entrée pour produire la déviation du spot et les oscilloscopes numériques qui transforment, préalablement à tout traitement, la tension d'entrée en nombre. L'affichage est reconstruit après un moment. Il devient alors une fonction annexe de l'appareil qui peut même en être dépourvu, la visualisation du signal étant effectuée par un ordinateur extérieur relié à l'oscilloscope. [1]

I.2.L'oscilloscope

L'oscilloscope est fondamentalement un appareil fournissant une représentation graphique d'un signal électrique. Dans la plupart des applications, ce graphique montre comment les signaux évoluent en fonction du temps : l'axe vertical (Y) représente la tension (ou l'amplitude d'une autre grandeur traduite en tension électrique) et l'axe horizontal (X) représente le temps. L'intensité ou luminosité de l'écran est appelée quelquefois axe Z (Fig I.1). [2]

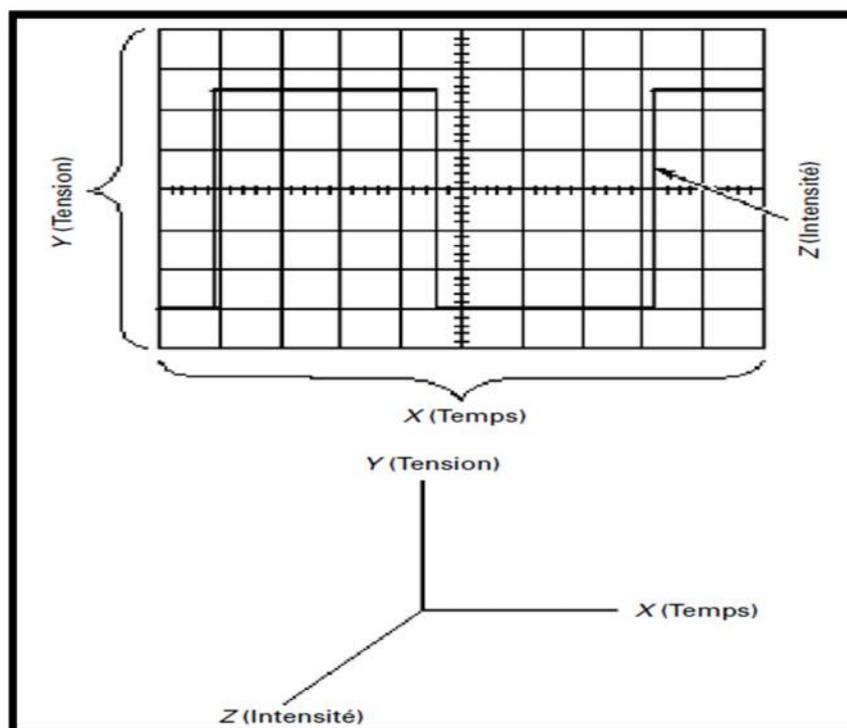


Fig I.1: Composantes X, Y et Z d'un signal affiché. [2]

La représentation graphique représentée sur l'écran de l'oscilloscope fournit beaucoup d'informations au sujet de ce signal, comme :

- la détermination des paramètres temporels et des paramètres d'amplitude du signal
- le calcul de la fréquence du signal
- la visualisation des « variations » du signal électrique
- la visualisation de la composante continue du signal (DC) et de sa composante alternative (AC)
- la mise en évidence du bruit dans le signal et de son évolution dans le temps.

Le panneau avant d'un oscilloscope est normalement divisé en sections : verticale, horizontale, et de déclenchement. Il y a aussi les contrôles de l'écran et les connecteurs d'entrée. On peut localiser ces sections du panneau avant dans les figures : (Fig I.2) et (Fig I.3). Les panneaux avant sont très similaires quel que soit le type d'instrument.

Mais l'oscilloscope numérique dispose de fonctions supplémentaires pour l'acquisition et le traitement de l'information. [3]

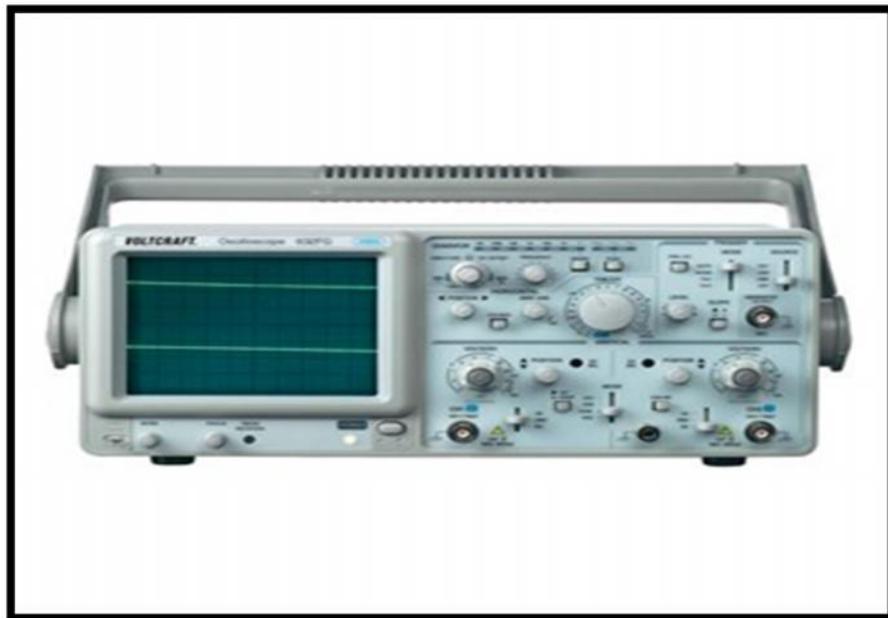


Fig I.2: Le panneau avant d'un oscilloscope analogique. [4]



Fig I.3: Le panneau avant d'un oscilloscope numérique. [4]

Les oscilloscopes sont utilisés par un grand nombre de techniciens, allant des réparateurs de télévision aux électroniciens. Ils sont indispensables pour toute personne travaillant sur du matériel électronique.

L'utilité d'un oscilloscope n'est pas limitée au monde de l'électronique. Avec le capteur adéquat, un oscilloscope peut mesurer tout genre de grandeurs physiques à condition qu'on puisse la convertir préalablement en un signal électrique. C'est dans ce sens qu'on définit un capteur comme un moyen qui convertit la grandeur physique (une pression, la lumière ou la chaleur (Fig I.4)) en un signal électrique. Par exemple, un microphone est un capteur qui transforme l'onde sonore en un signal électrique. En automobile, un électricien-auto utilise un oscilloscope pour mesurer les vibrations du moteur. Un chercheur en médecine utilise l'oscilloscope pour mesurer les signaux du cerveau. Les possibilités sont illimitées. [2]

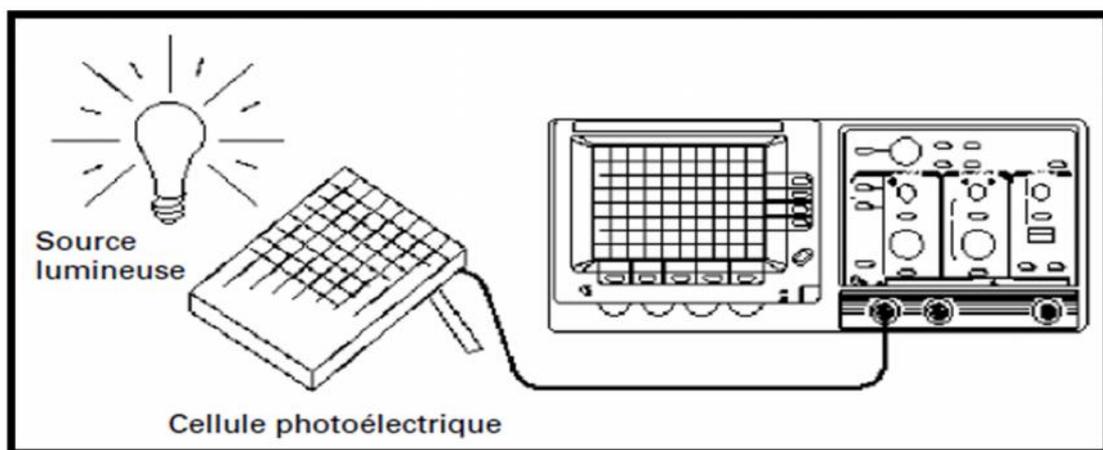


Fig I.4: Données scientifiques capturées par un oscilloscope. [2]

I.2.1.Types d'oscilloscopes : analogique et numérique

Il existe deux types d'oscilloscopes : analogique et numérique. Un oscilloscope analogique fonctionne par déflexion électrique d'un faisceau d'électrons en fonction de la tension mesurée. La tension dévie le faisceau de haut en bas proportionnellement et trace une courbe sur l'écran qui donne une image instantanée (en temps réel) du signal (Fig I.5).

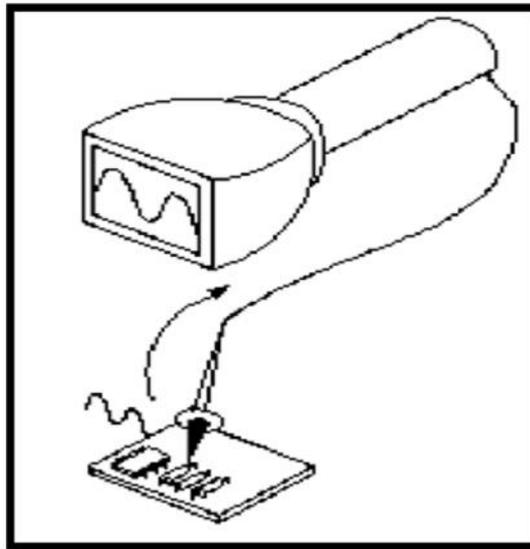


Fig I.5: L'oscilloscope analogique visualise le signal. [2]

Par contre, un oscilloscope numérique échantillonne le signal et utilise un convertisseur analogique-numérique (CAN) pour convertir la tension à mesurer en une information numérique. Il utilise alors cette information numérique pour reconstruire la courbe sur l'écran (Fig I.6).

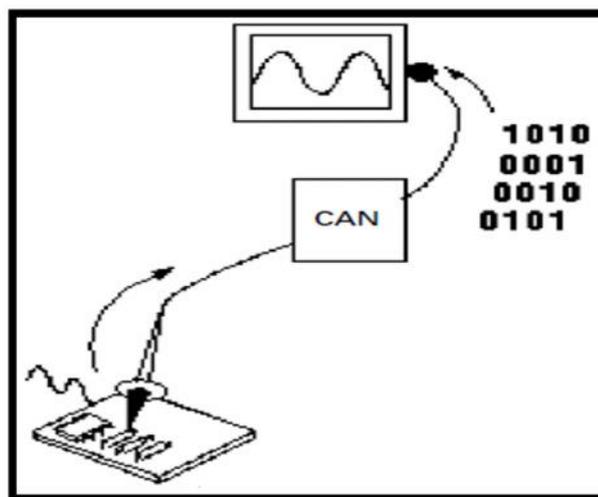


Fig I.6: L'oscilloscope numérique échantillonne le signal et reconstruit une image. [2]

Pour beaucoup d'applications, l'un et l'autre type peuvent être utilisés. Cependant, chaque type possède des caractéristiques spécifiques plus ou moins adaptées aux signaux à mesurer.

L'oscilloscope analogique est souvent préféré à un oscilloscope numérique lorsqu'il est important de visualiser des signaux variables très rapidement dans le temps.

Les oscilloscopes numériques capturent des événements uniques. Ils peuvent traiter les données numériques et les transmettre à un ordinateur pour traitement. Ils peuvent également stocker les données numériques en mémoire pour examen ultérieur ou permettre leur impression sur une imprimante. [2][3]

I.3.L'oscilloscope analogique

I.3.1.Principe de fonctionnement

Lorsqu'un oscilloscope est connecté à un circuit, la tension transite de la sonde à travers le système vertical de l'oscilloscope. La figure I.7 représente le bloc diagramme simple d'un oscilloscope analogique.

En fonction de l'échelle verticale (commande volts/division), un atténuateur réduit la tension du signal ou un amplificateur augmente la tension du signal. Ensuite le signal est appliqué directement aux plaques de déflexion verticales du tube à rayons cathodiques (CRT). La tension appliquée à ces plaques de déflexion provoque le déplacement du spot sur l'écran (le faisceau électronique frappant le phosphore crée le spot lumineux sur l'écran). Une tension positive produit un déplacement vers le haut et une tension négative produit un déplacement vers le bas.

Le signal est également transmis au système de déclenchement pour initier un « balayage horizontal ». Le déclenchement du système horizontal provoque le démarrage de la base de temps horizontale qui déplace le spot au travers de l'écran de la gauche vers la droite dans un intervalle de temps spécifique. Aux vitesses de balayage élevées, le point peut balayer l'écran jusqu'à 500 000 fois chaque seconde.

La répétition du balayage, liée à la persistance rétinienne et à la déflexion verticale, permet la visualisation de la courbe sur l'écran. Le déclenchement est nécessaire pour assurer la stabilité du signal répétitif. Il assure que le balayage commence toujours au même point d'un signal répétitif, il en résulte une image stable comme le montre la figure I.8. Le déclenchement permet d'obtenir une image stable d'un signal, même si sa fréquence de répétition n'est pas constante. [2][3]

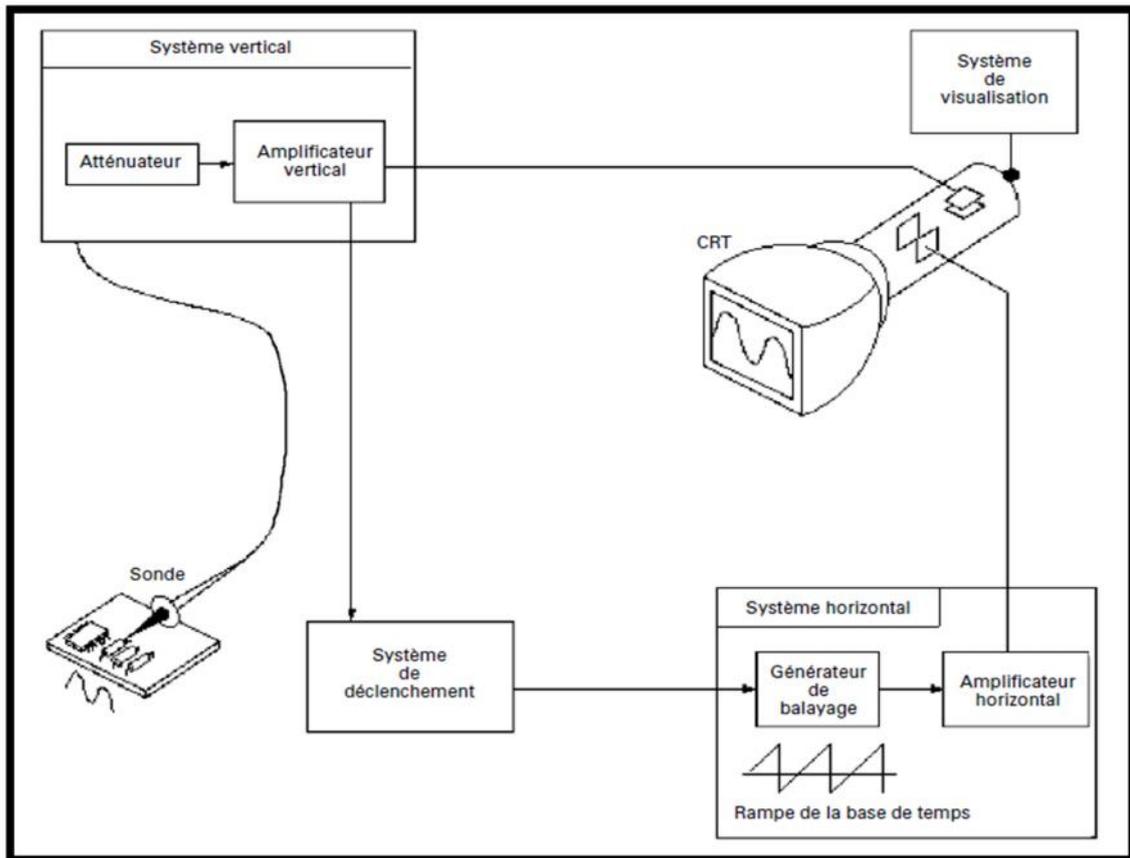


Fig I.7: Bloc diagramme de l'oscilloscope analogique. [3]

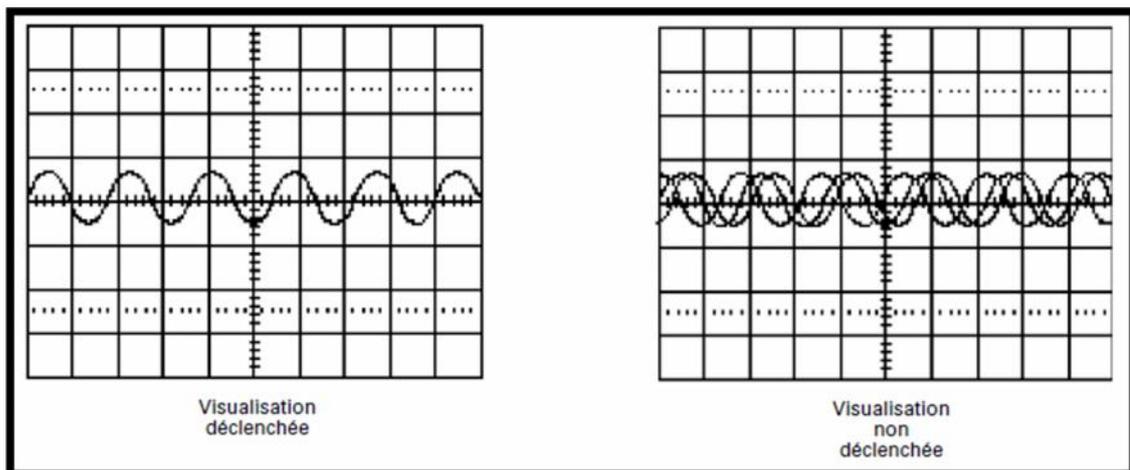


Fig I.8: Le déclenchement stabilise un signal répétitif. [3]

I.3.2.Fonctions de base

En conclusion, avec un oscilloscope analogique, il est nécessaire d'ajuster trois fonctions de base pour visualiser un signal :

1. La base de temps. Elle permet d'ajuster le temps par division (écrit habituellement comme temps/div ou sec/div) représenté horizontalement à l'écran. Le contrôle de l'échelle horizontale sur l'oscilloscope permet d'ajuster la base de temps.

2. L'atténuation (diminution) ou l'amplification (augmentation) du signal. Le contrôle de l'échelle verticale permet d'ajuster le signal avant qu'il ne soit appliqué au système de déflexion verticale.
3. Le déclenchement de l'oscilloscope ; ce contrôle permet de stabiliser un signal répétitif, aussi bien qu'un signal unique.

La mise au point de l'intensité et du focus permet de créer sur l'écran un signal net. [2]

I.4.L'oscilloscope numérique

I.4.1.Principe de fonctionnement

Quelques-unes des parties constituant l'oscilloscope numérique sont les mêmes que celles des oscilloscopes analogiques, cependant, les oscilloscopes numériques contiennent des systèmes de traitement de données additionnels (Fig I.9). Avec les systèmes additionnels, l'oscilloscope numérique rassemble les données numériques pour représenter la courbe entière sur l'écran.

Quand un oscilloscope numérique est connecté à un circuit, le système vertical ajuste l'amplitude du signal de la même façon que dans un oscilloscope analogique.

Ensuite, le convertisseur analogique-numérique (CAN) du système d'acquisition échantillonne le signal à des intervalles de temps discrets et convertit la tension du signal à ces instants en valeurs numériques appelées échantillons. L'horloge du système horizontal détermine l'instant pendant lequel l'échantillonneur du CAN prélève un échantillon. Le rythme auquel l'horloge « déclenche » la numérisation est appelé fréquence d'échantillonnage et est exprimé en échantillons par seconde.

Les échantillons du CAN sont stockés dans la mémoire comme points de la courbe.

Le nombre de points nécessaires pour représenter une courbe complète est appelé la longueur d'enregistrement. Le système de déclenchement détermine le début et l'arrêt de l'enregistrement.

L'écran permet la visualisation des points de la courbe représentant le signal entreposé dans la mémoire.

En fonction des capacités de l'oscilloscope, des traitements additionnels sur les échantillons permettent d'accroître la résolution de la visualisation à l'écran. Le pré déclenchement peut être disponible, permettant de voir des événements précédant le point de déclenchement. [2]

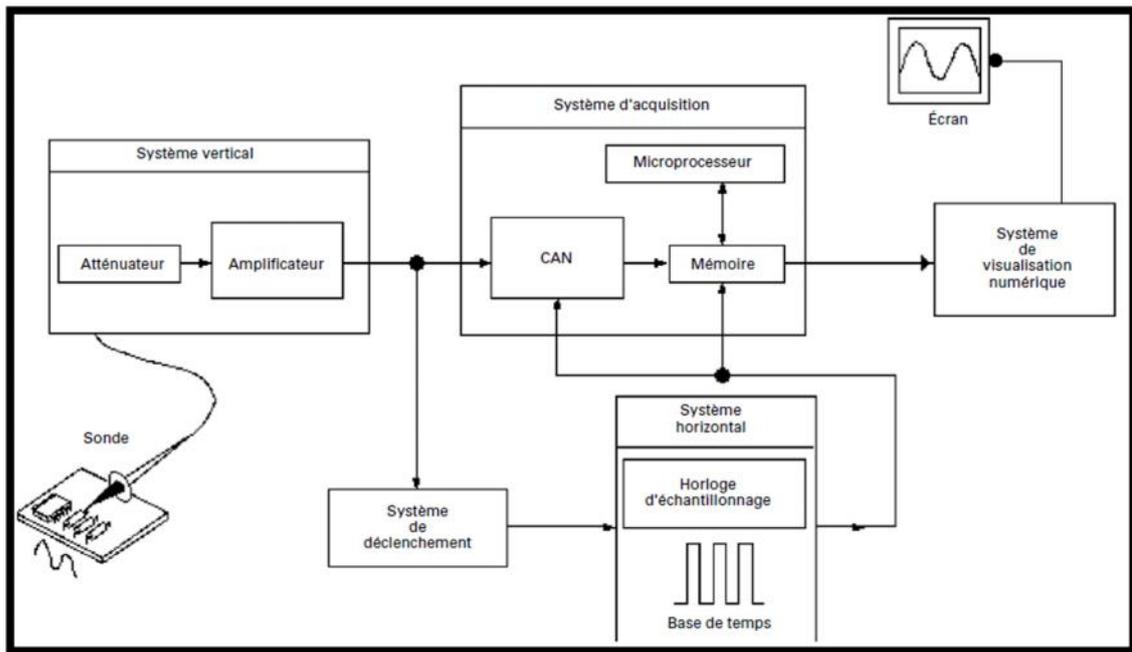


Fig I.9: Bloc diagramme de l'oscilloscope numérique. [3]

I.4.2. Bloc diagramme

Avec un oscilloscope numérique, les réglages suivants sont à ajuster pour effectuer une mesure :

1. la base de temps. Ce réglage permet de sélectionner la fenêtre de temps représentée par la largeur de l'écran. Le contrôle de l'échelle horizontale sur l'oscilloscope permet d'ajuster la base de temps.
2. l'atténuation (diminution) ou amplification (augmentation) du signal. Le contrôle de l'échelle verticale permet d'ajuster l'amplitude du signal avant de l'envoyer au CAN.
3. le déclenchement de l'oscilloscope. Ce contrôle permet de stabiliser un signal répétitif, de capturer un signal monocoup, et de fixer le pré-déclenchement.
4. le mode d'acquisition et le mode d'échantillonnage. Ces réglages dépendent de la fréquence et de la complexité du signal qui doit être mesuré. Les réglages du panneau avant permettent d'ajuster ces paramètres.

I.5. Définitions

I.5.1. Fréquence et période

Un signal répétitif a une fréquence. La fréquence est mesurée en hertz (Hz) et est égale au nombre de fois que le signal se répète lui-même en une seconde (cycles par seconde).

La période est le temps nécessaire pour effectuer un cycle complet. Période et fréquence sont inverses l'un de l'autre. Donc, par exemple, le signal de la figure I.10 a une fréquence de 3 Hz et une période de 1/3 seconde. [2]

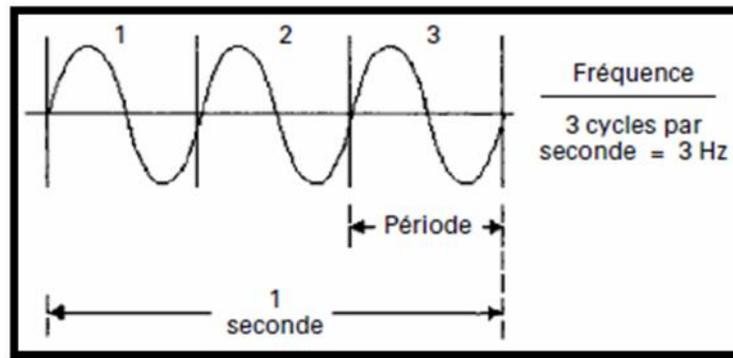


Fig I.10 : Fréquence et période. [2]

I.5.2. Bande passante

La spécification de la bande passante indique la fréquence maximale à laquelle l'oscilloscope mesure correctement un signal avec le minimum d'aberration. Par convention, la bande passante indique la fréquence pour laquelle le signal affiché est réduit à 70,7 % de l'amplitude crête-à-crête appliquée à l'entrée de l'oscilloscope (cette atténuation de 70,7 % est appelée la atténuation à - 3 dB, un terme basé sur une échelle logarithmique). [2]

I.5.3. Temps de montée

Le temps de montée est un autre moyen de décrire la fréquence utile d'un oscilloscope. Le temps de montée peut être une considération de la performance plus appropriée quand il faut mesurer des signaux impulsionnels. Un oscilloscope ne peut pas afficher correctement un temps de montée plus rapide que celui spécifié comme le temps de montée de l'oscilloscope.

Afin de réaliser des mesures avec le maximum de précision, il convient de tenir compte du temps de montée global mesuré :

$$T_m(\text{mesuré}) = \sqrt{T_m^2(\text{signal}) + T_m^2(\text{amplificateur}) + T_m^2 T_m^2(\text{sonde})}$$

I.5.4. Sensibilité verticale

La sensibilité verticale indique de combien l'amplificateur vertical peut amplifier un signal faible. La sensibilité verticale s'exprime habituellement en millivolts (mV) par division. La plus petite tension qu'un oscilloscope d'usage général peut détecter est typiquement d'environ 2 mV par division. Les facteurs d'atténuation sont généralement en séquence 1-2-5. Une possibilité d'ajustage de la sensibilité continûment variable existe sur les instruments. [2]

I.5.5. Résolution verticale

La résolution verticale est limitée d'une part par l'épaisseur de la trace et d'autre part par la sensibilité de l'amplificateur sur un oscilloscope analogique.

Exemple : pour une sensibilité de 1 mV/div, chaque division principale étant subdivisée en 5 petites divisions, l'utilisateur peut apprécier 1/2 division soit : 0,1 mV.

Pour les oscilloscopes numériques, la résolution est liée au nombre de bits du CAN.

Exemple : soit un CAN de 8 bits. La spécification s'exprime pour 8 bits sur la hauteur de l'écran, soit 256 niveaux pleine échelle (10 divisions verticales), donc 25 niveaux par division verticale. Pour 10 mV/div, la résolution est de : $10 \text{ mV}/25 = 400 \text{ mV}$.

Divers procédés numériques permettent d'accroître la résolution verticale. Ils sont décrits ci-après. [2]

I.5.6. Impédance d'entrée

L'impédance d'entrée des appareils dont la bande passante est inférieure à 200 MHz est de l'ordre de 10M Ω en parallèle avec une capacité de 15 pF.

Pour les appareils de bande passante supérieure, la capacité diminue et l'impédance d'entrée est de 50 Ω . [2]

I.5.7. Couplage

L'utilisateur a le choix de filtrer ou non le signal entrant. Il existe trois types de couplage d'entrée : AC, DC, GND.

- En AC (alternatif), la composante continue du signal, bloquée par une capacité, n'est pas transmise.
- En DC (continu), toutes les composantes du signal sont transmises.
- En GND (masse), l'entrée est reliée à la masse de l'instrument pour servir de référence aux mesures d'amplitude. [2]

I.5.8. Vitesse de balayage

Cette spécification indique la vitesse à laquelle le spot se déplace en travers de l'écran pour les oscilloscopes analogiques. Elle indique la fenêtre temporelle d'acquisition pour les oscilloscopes numériques.

La vitesse de balayage la plus rapide d'un oscilloscope est habituellement de 1 nanoseconde/div. [2]

I.5.9. Base de temps

Les oscilloscopes disposent en général de deux bases de temps : la principale (A) et la retardée (B). Le but de cette technique est de permettre de sélectionner un détail dans le signal visualisé (à une vitesse A) pour ensuite le dilater sur tout l'écran à une vitesse $B > A$. [2]

La fonctionnalité reste identique sur les oscilloscopes numériques, bien que le procédé employé soit différent. La mémorisation du signal apporte en outre de nombreuses possibilités, telles que pré et post déclenchement, retards en événement.

I.5.10. Précision du gain

La précision du gain est relative à celle de l'amplificateur vertical. Elle s'exprime habituellement comme un pourcentage d'erreur. Elle est généralement de l'ordre de 1 à 2 %. [2]

I.5.11. Précision de la base de temps ou précision horizontale

Elle indique avec quelle précision le système horizontal affiche un signal à l'écran. Elle s'exprime habituellement comme un pourcentage d'erreur pour les oscilloscopes analogiques.

Elle est exprimée en ppm (partie par million) de l'horloge sur les oscilloscopes numériques. [2]

I.5.12. Fréquence d'échantillonnage

La fréquence d'échantillonnage sur les oscilloscopes numériques indique le nombre d'échantillons par seconde que le CAN (et par conséquent l'oscilloscope) peut acquérir.

Cette fréquence est exprimée habituellement en méga échantillons par seconde (Mé/s soit 10^6 échantillons par seconde). Plus elle est rapide, plus l'oscilloscope peut représenter correctement les détails dans un signal haute fréquence. La fréquence d'échantillonnage minimum peut être aussi primordiale pour regarder des signaux changeant lentement sur de longues périodes de temps.

Typiquement, la fréquence d'échantillonnage se modifie avec la commande sec/div de la base de temps et est choisie de telle sorte qu'il y ait un nombre constant de points pour une longueur d'enregistrement donnée.

Théorème de l'échantillonnage : le lecteur pourra utilement se reporter à l'article Théorie du signal dans le présent traité.

Le théorème de Shannon indique qu'il est nécessaire de prélever au moins deux échantillons par période de la composante la plus élevée présente dans le signal sinusoïdal à numériser, pour le connaître parfaitement. Une autre façon d'exprimer cela revient à dire que,

pour connaître un signal sinusoïdal, sa fréquence doit être au plus égale à la moitié (fréquence de Nyquist) de la fréquence d'échantillonnage.

I.6.Conclusion

On vient de voir le fonctionnement de l'oscilloscope analogique avec les différents blocs. L'oscilloscope numérique ne diffère de l'analogique que par se système d'acquisition.

Le système d'acquisition utilise le bus universel en série (USB : Universal Serial Bus) qui à travers lequel les données du signal a affiché sans transmis ainsi que la configuration désirée, l'importance du bus USB à fais qu'il soit développée dans un chapitre suivant.

Bus universel en série

CHPITRE II : Bus universel en série

1 - Introduction	15
2 - Normes USB	15
2.1 - USB 1(Full ou Low speed)	15
2.2 - USB 2.0 (High Speed)	15
2.3 - USB 3.0 (Super Speed).....	16
3 - Connectique	16
3.1 - Description du Câble USB.....	16
3.2 - Composition du câble USB.....	17
3.3 - Connecteurs USB.....	17
3.4 - Brochage	19
3.5 - Caractéristiques électriques	19
4 - Architecture d'un système	22
5 - Les descripteurs	23
5.1 - Définition d'un descripteur	23
5.2 - Rôle des descripteurs	23
5.3 - Les types de descripteur.....	23
6 - Les protocoles USB	25
6.1 - Les champs de paquet USB ordinaires	26
6.2 - Les types de paquet USB	27
7 - Transfert de données	29
7.1 - Transfert en mode Contrôle	29
7.2 - Transfert en mode Interrupt	29
7.3 - Transfert en mode Isochrone	29
7.4 - Transfert en mode Bulk	29
8 - PID/VID.....	29
8.1 - Introduction au Product ID et au Vendor ID	29
8.2 - Normalisation des PID/VID.....	30
9 - Les fichiers INF/SYS.....	29
9.1 - Les fichiers d'extension INF.....	30
9.2 - Les drivers .SYS	31
9 - Conclusion	31

II.1.Introduction

L'USB est un nouveau type de connectivité sur les machines informatiques, promu par les principaux constructeurs informatiques et éditeurs de logiciels (Compaq, DEC, IBM, Intel, Microsoft, NEC, Nortel), doit permettre le raccordement sur une prise unique de plusieurs équipements divers (imprimante, téléphone, modem, fax, clavier, souris, scanners, écrans...). On désire ainsi éviter la multiplication actuelle des connecteurs sur les PC. Ce BUS permet la transmission de données, de la voix et de l'image compressée.

Cette nouvelle technique se doit d'être rapide, bidirectionnelle, synchrone, de faible coût et l'attachement d'un nouveau périphérique doit être dynamique. De plus la télé alimentation des équipements est possible. [5]

II.2.Normes USB

On va présenter les 3 normes USB utilisée depuis l'apparition de la norme USB 1, en 1995. La différence entre ces dévires réside dans la vitesse qui augmente d'une norme à un autre, tout en ajoutant d'autres fonctions comme la fonction Mass Storage. [6]

II.2.1.USB 1(Full ou Low speed)

La Norme USB 1 regroupe USB 1.0 et USB 1.1 qui permettent d'obtenir le débit de 12 Mbits (c'est la plein vitesse ou Full speed) ou bien le débit de 1.5 Mbits (la basse vitesse ou Low Speed).

Pour identifier la vitesse, l'appareil relié au bus USB précise sa vitesse en reliant le fil D+ à 3.3V à travers une résistance de 15K dans le cas de la pleine vitesse. Dans le cas de la basse vitesse le fil D- a 3.3V à travers résistance 15K . Le logo de l'USB 1 est donné dans la figure II.1. [7]



Fig II.1: USB 1 logo. [8]

II.2.2.USB 2.0 (High Speed)

Le standard USB 2.0 (1999) permet de monter le débit à 480 Mbits/s (haute vitesse ou High Speed) et peut ainsi contrer la célérité du bus Firewire qui possède bien d'autres avantages. Le logo de l'USB 1 est donné dans la figure II.2. [7]



Fig II.2: USB 2 logo. [8]

II.2.3.USB 3.0 (Super Speed)

Le standard USB 3.0 prévoit des vitesses à 4,8 Gbits/s. Les premiers équipements verront certainement en 2010. Le logo de l'USB 3 est donné dans la figure II.3. [7]



Fig II.3: USB 3 logo. [8]

II.3.Connectique

II.3.1.Description du Câble USB

Le câblage USB est relativement simple, il a la même structure quelle que soit la vitesse de transmission. Le câble transporte deux paires de fils :

La paire de signal destinée au transfert de données D+ et D- et une seconde paire qui peut être utilisée pour la télé alimentation GND et Vusb. La première paire est non blindée pour les périphériques lents tels que les claviers, souris fonctionnant à 1.5Mbits/s tandis que caméras, micro et autres ont recours à une paire de fils torsadée blindée pour atteindre les 12Mbits/s. [9]

Le câble USB (Fig II.4) est un ensemble de deux paires de fils : D+, D- et GND, Vusb.



Fig II.4: Câble USB. [8]

II.3.2. Composition du câble USB

Chaque connecteur dispose de deux fils d'alimentation ($V_{usb} = 5V$ et GND) et deux fils destinés au transfert de données (D+ et D-).

Une connexion entre deux PC est aussi possible par l'adjonction d'une interface spéciale qui déjoue la vigilance du PC maître et transforme le second PC en «esclave» 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). [9]

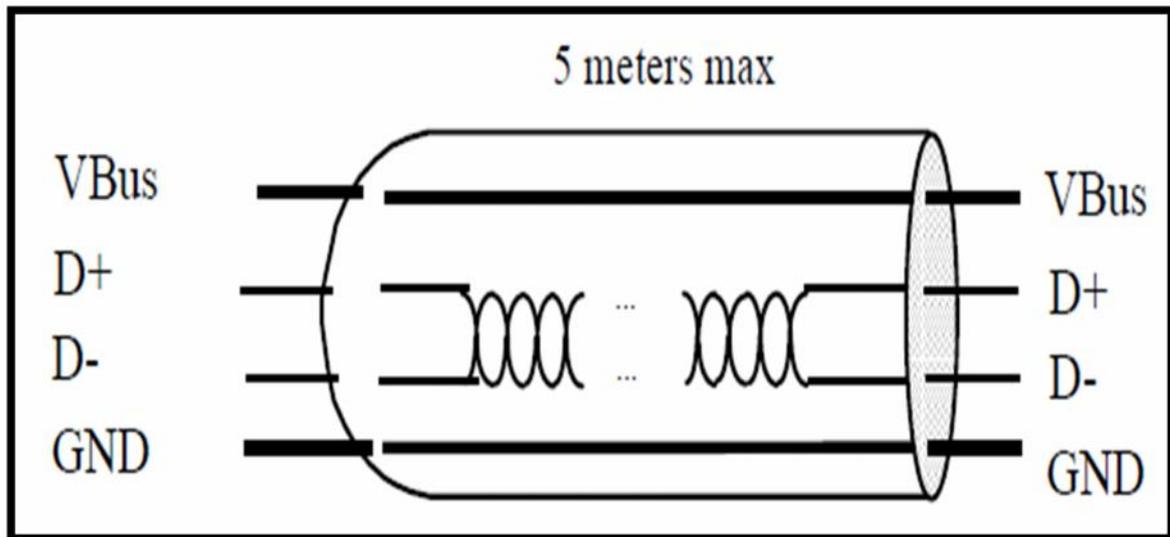


Fig II.5: Composition d'un câble USB. [9]

Longueur maxi = 5 m (3m si 1.5Mbit/s).

Temps de propagation maxi = 30 ns ($< 1/2$ bit).

Diamètre extérieur : 3,4..5,3 mm - couleur = blanc.

- 1 Paire données : torsadée (6..8 cm) (non torsadée toléré si 1.5Mbit/s), 28 AWG (American Wire Gauge) mini, isolant polyéthylène, $Z_c = 90 \pm 15\%$, atténuation = 38 dB/1000 pieds à 10MHz.
- 1 Paire alimentation : non torsadée possible, isolant PVC (Polyvinyl chloride), pour alim. = 5 V, 0.5A max.
- écran : nécessaire si 12Mbit/s ou plus (optionnel si 1.5Mbit/s).

III.3.3. Connecteurs USB

Il existe deux séries de connecteurs pour le bus USB donnée dans la figure II.6 :

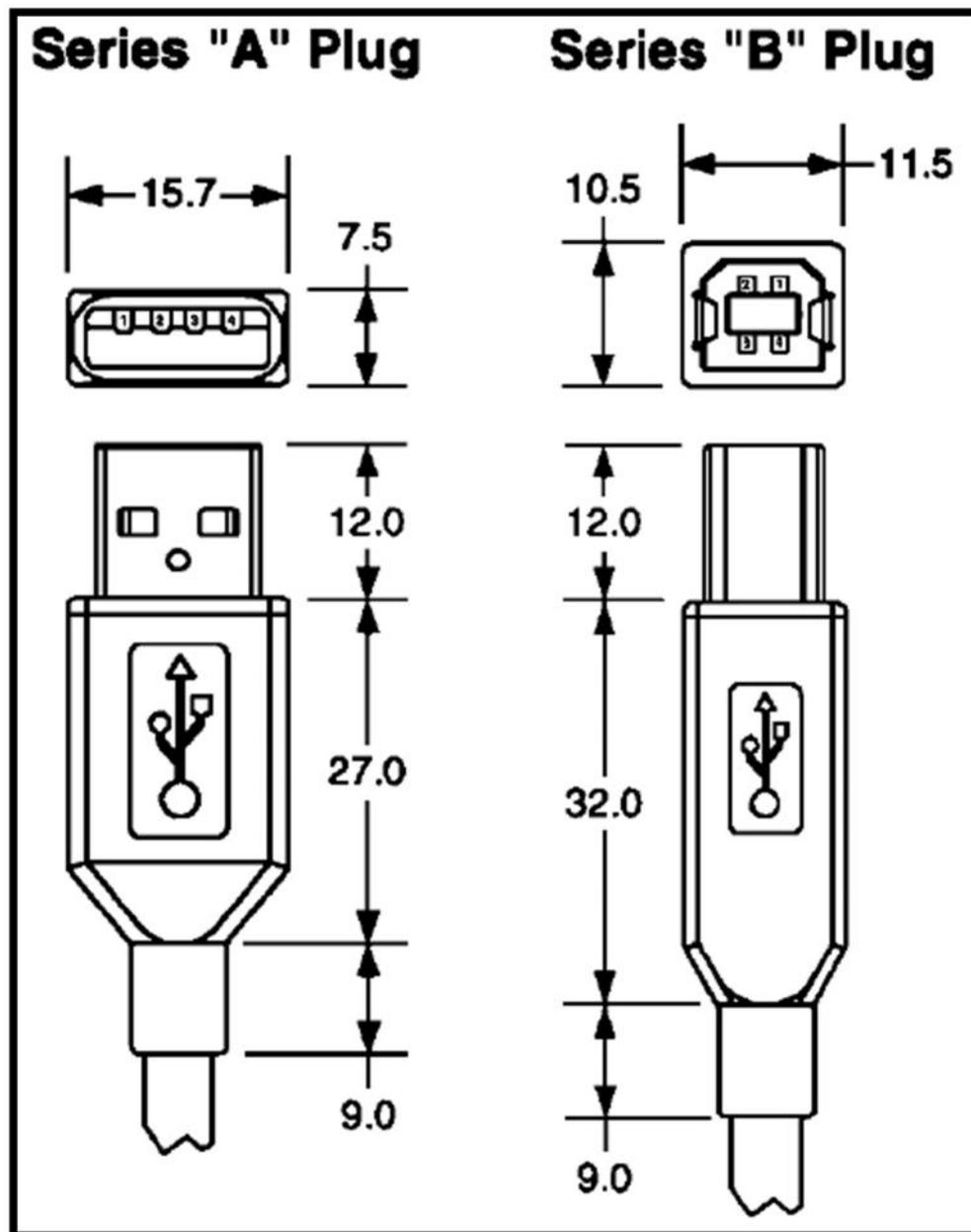


Fig II.6: Connecteurs. [10]

II.3.3.a. Connecteur série A

Destiné au raccordement montant vers le « Host », l'alimentation est sortante sur le réceptacle femelle (châssis). 4 contacts à plat (D+ et D- au milieu et plus courts). [10]

II.3.3.b. Connecteur série B

Destiné au raccordement descendant du « Host » sur les périphériques, lorsqu'il y a un connecteur entrant. L'alimentation est entrante sur le réceptacle. 4 contacts (+v et D- d'un côté, Gnd et D+ de l'autre). [10]

II.3.4. Brochage

Le brochage permet de spécifier le signal de chaque fil relié au connecteur (Fig II.7) comme donné dans le tableau II.1 :

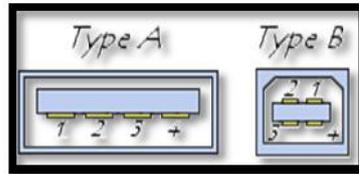


Fig II.7: connecteurs USB type A et B. [11]

Broche	Désignation	Couleur du fil
1	Alimentation +5V (VBUS) 100mA maximum	rouge
2	Données (D-)	blanc
3	Données (D+)	vert
4	Masse (GND)	noir

Tab II.1: Le brochage des connecteurs USB. [11]

II.3.5. Caractéristiques électriques

II.3.5.a. Etats de D+ et D-

Etat repos : $D+ = V_{OH}$ et $D- = V_{OL}$ (inverse si 1,5Mbit/s).

Transmission de données : D+ inverse de D- (mode différentiel).

Fin de paquet ("EOP") : $D+ = D- = V_{OL}$ pendant 2 bits.

Déconnexion : D+ tend vers V_{OL} car R2 débranchée (D- si 1,5Mbit/s).

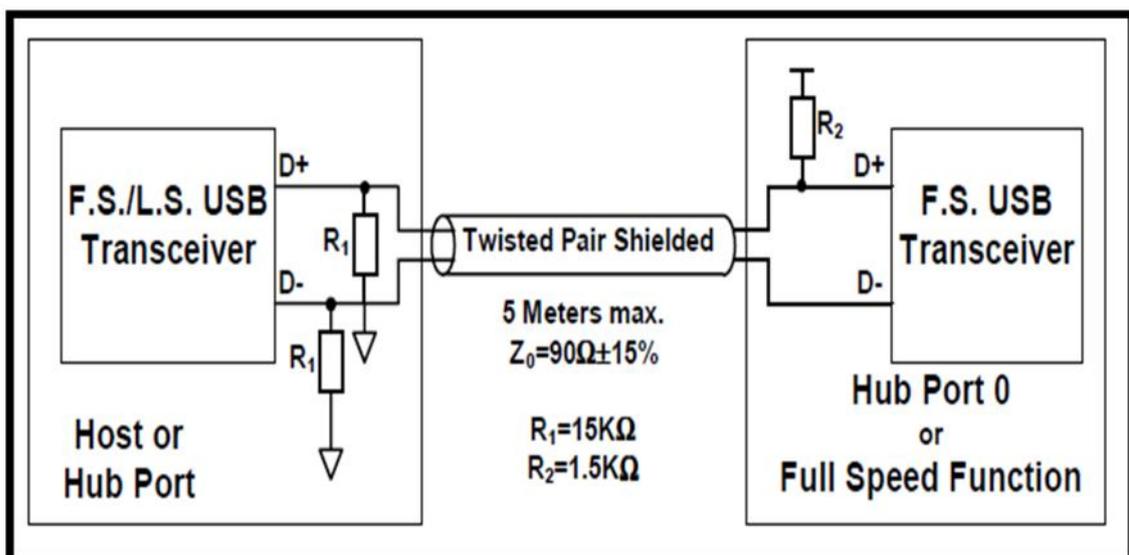


Fig II.8: Full Speed Connections. [5]

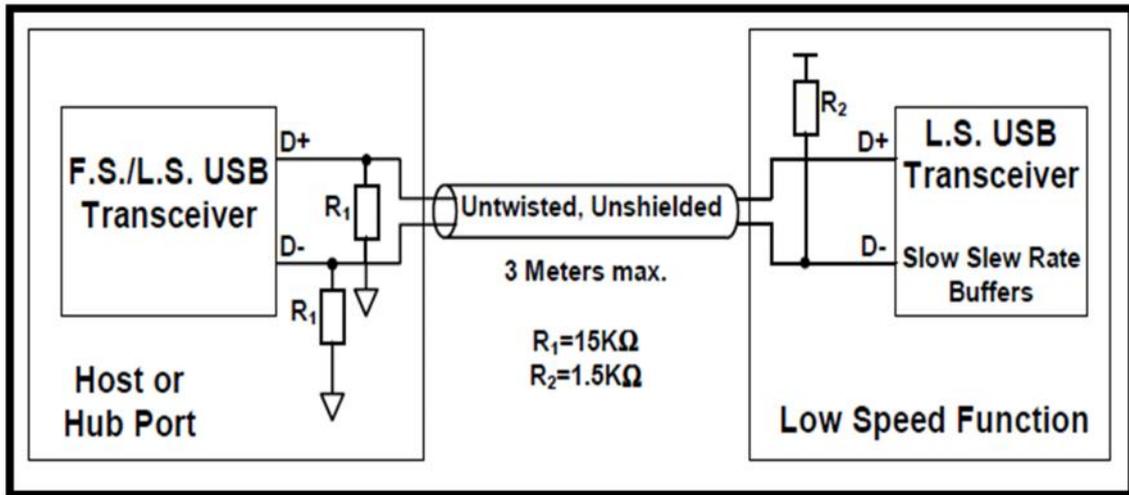


Fig II.9: Low Speed Connections. [5]

II.3.5.b. Emetteur sur D+ et D-

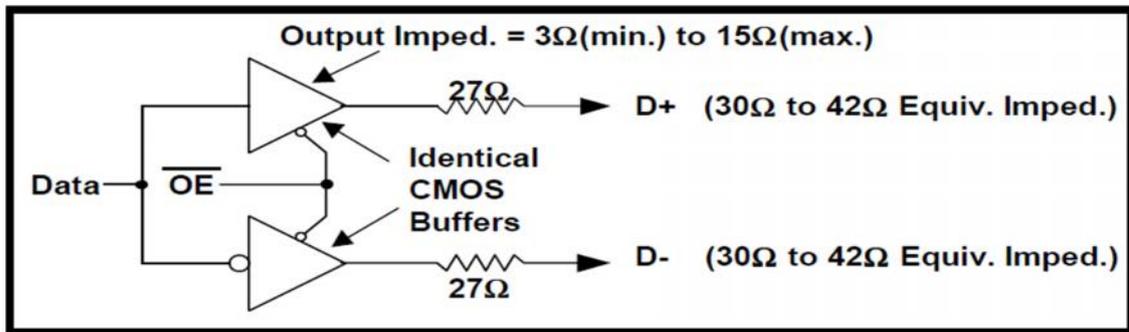


Fig II.10: Exemple de Circuit de sortie. [5]

Impédance de sortie : $30\Omega \dots 42\Omega$ (câble = 90Ω).

Niveau haut : $V_{OH} > 2,8V$ (avec $R=15k\Omega$ sur masse) $V_{OHmax}=3,6V$.

Niveau bas : $V_{OL} < 0,3V$ (avec $R=1,5k\Omega$ sur +V).

Temps de montée/descente : $4..20ns$ ($75..300ns$ si $1,5Mbit/s$).

II.3.5.c. Récepteur sur D+ et D-

- Mode différentiel (réception de données) : Sensibilité $> 200mV$ pour signal entre $0,8v$ et $2,5v$.
- Mode commun (D+ ou D- par rapport à la masse) : Seuil = niveau TTL (Transistor-transistor logic) ($0,8..2V$) avec hystérésis conseillé. [5]

II.3.5.d. Alimentation

Port générateur = $5V$ ($4.75..5.25V$) / $0,5A$. Un HUB alimenté doit pouvoir fournir $0,5A$ sur ses ports (pas forcément tous en même temps). Un HUB télé alimenté doit pouvoir fournir $100mA$ sur ses ports avec une perte maxi de $350mV$. [5]

II.3.5.e. La signalisation

Les lignes de données D+ et D- fonctionnent en mode différentiel. Les circuits doivent avoir un état haut impédance; les lignes doivent résister à un court-circuit avec Vusb ou la masse. Les ports des hubs doivent être compatible faible et pleine vitesse. Dans un appareil pleine vitesse, la ligne D+ est ramenée à une tension comprise entre 3 et 3,6 V par une résistance de 1,5k alors qu'en basse vitesse c'est la ligne D- qui est ainsi polarisée. On définit sur les lignes de données deux états : J et K. En basse vitesse :

- Etat J : D- > D+ d'au moins 200 mV
- Etat K : D+ > D- d'au moins 200 mV

En pleine vitesse c'est le contraire. Un état J avec certaines valeurs minimales et maximales de tension indique un état "inoccupé". De plus il existe un état où D+ et D- sont au potentiel bas appelé SE0. L'état SE0 pendant plus de 10 ms indique un reset, il est émis vers un appareil 100 ms après sa connexion. Dès qu'un appareil voit le signal de reset pendant plus de 2,5 microsecondes, il doit faire son reset et le terminer avant la fin du signal (10 ms).

La polarisation de l'une des deux lignes permet de détecter la connexion d'un appareil et sa vitesse. Les appareils à haut vitesse doivent pouvoir démarrer en pleine vitesse.

Le mode "veille" doit être supporté par tout appareil. Ce mode doit être activé dès qu'il n'y a plus d'activité sur le bus depuis plus de 3 ms; c'est à dire que les trames ne contiennent aucun paquet. (Rappel : une trame dure 1ms) Pendant le mode veille la consommation doit être réduite mais néanmoins la polarisation de D+ ou de D- doit être maintenue. La moindre activité sur le bus doit réveiller l'appareil. Un appareil peut avoir une fonction "réveil" destinée à avertir l'hôte par l'intermédiaire du hub.

L'encodage des données se fait selon la méthode NRZI (Fig II.11). Un "1" est représenté par l'absence de changement d'état et un "0" par un changement d'état.

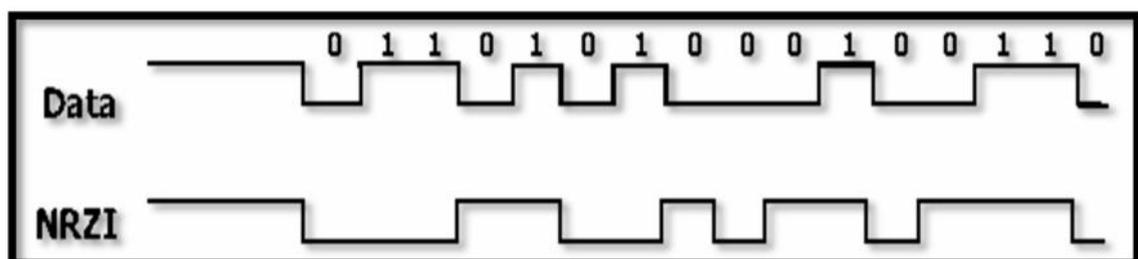


Fig II.11: Code NRZI.

II.4. Architecture d'un système

Les connexions se font point à point. Tous les appareils ont une connexion amont vers l'hôte. Il est possible de connecter jusqu'à 127 périphériques simultanément.

Les ports USB supportent le Hot Plug & Play, c'est à dire qu'un périphérique peut être connecté et reconnu, sans redémarrage de l'ordinateur. Les périphériques qui disposent actuellement de ce type de port sont les imprimantes, scanners, webcams... Ils peuvent être connectés les uns à la suite des autres (en bus) ou reliés à un Hub (en étoile), comme le laisse entrevoir son logo générique (Fig II.12). [12]

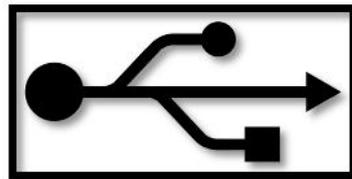


Fig II.12 : Icône USB. [8]

L'USB utilise une topologie en étoile à étages, qui ressemble à celle d'Ethernet. Ceci impose l'utilisation d'un Hub quelque part, mais qui est inclus dans de nombreux appareils. Un hub joue le rôle de multiplexeur (connexion de plusieurs périphériques à un même câble) mais aussi de répéteur, d'amplificateur, de contrôleur du signal et de fournisseur de courant.

Par exemple la figure II.13 clavier peut contenir un Hub qui est connecté à l'ordinateur. La souris et d'autres appareils tels qu'un caméscope numérique peuvent être branchés facilement au dos du clavier. Les moniteurs ne sont que d'autres périphériques sur une longue liste d'appareils qui comportent communément des Hubs intégrés.

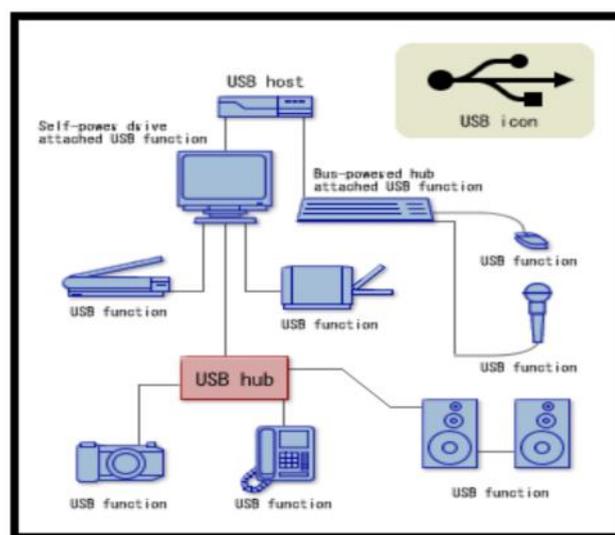


Fig II.13 : Architecture d'un système USB avec Hub. [8]

II.5. Les descripteurs

II.5.1. Définition d'un descripteur

On peut définir les descripteurs comme étant des blocs d'informations pré formatés. Tous composants USB doit obligatoirement posséder les descripteurs standards. Tous les transferts d'informations durant cette phase d'énumération se font suivant le type Control. Il va de soi que tout composant USB doit pouvoir être capable de supporter ce type de transfert. Nous verrons par la suite que ce n'est pas le cas pour tous les autres types de transfert que nous définirons. [9]

II.5.2. Rôle des descripteurs

Il existe sur le marché de nombreux périphériques USB. Il a fallu, lors de la création de la norme USB, trouver un dispositif pour reconnaître chaque composant USB. Cela était indispensable puisque l'USB devait être un dispositif plug & play. Lors du branchement du périphérique, le « host » autrement dit plus communément le PC, doit reconnaître tous les périphériques qui lui sont branché. Tout le processus d'énumération se fait grâce aux descripteurs qui sont rassemblés dans un fichier texte (fichier assembleur par exemple): en général, un fichier assembleur, qui est ensuite programmé dans le système USB.

Lorsque l'on connecte ou déconnecte un périphérique, celui-ci fournit à l'hôte toutes les informations nécessaires à son identification, c'est à dire ces descripteurs. Ils sont très utiles pour l'hôte puisqu'il peut, de ce fait, connaître les caractéristiques périphériques comme par exemple la puissance utile, le type de périphérique, le dispositif de transfert des données, le module de gestion ... etc.

Généralement, dans la plupart des périphériques, 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. [9]

II.5.3. Les types de descripteur

List des types de descripteur USB donné dans le tableau II.2 :

Type	Valeur
Appareil	1
Configuration	2
Chaine de caractères	3
Interface	4
Terminaison	5
Autre configuration de vitesse	7
Interface de puissance	8
Classe	21

Tab II.2: List des types de descripteur USB. [7]

II.5.3.a.Descripteur d'appareil

Chaque appareil doit avoir un numéro de fabricant (VID) et un numéro de produit (PID) Le VID est vendu très cher par le trust USB. Les fabricants de CI (Circuit Intégré). ont souvent un VID et certains peuvent fournir un PID à utiliser avec leur VID. Tous les appareils ne sont pas rattachés à une classe. List des champs de descripteur d'appareil donné dans le tableau II.3 :

Champ	Taille	Description
bLength	1	Taille du descripteur en octets
bDescriptorType	1	Type du descripteur = 1
bcdUSB	2	Version USB codée en BCD
bDeviceClass	1	Code de classe assigné par l'USB-IF
bDeviceSubClass	1	Code de sous-classe dépendant de la classe
bDeviceProtocol	1	Code de protocole dépendant de la classe
bMaxPacketSize0	1	Taille maximum des paquets pour la terminaison 0
idVendor	2	Code fabricant assigné par l'USB-IF (VID)
idProduct	2	Code produit assigné par le fabricant (PID)
bcdDevice	2	Version d'appareil codé en BCD
iManufacturer	1	Index du descripteur de chaine du fabricant
iProduct	1	Index du descripteur de chaine de l'appareil
iSerialNumber	1	Index du descripteur de chaine du numéro de série
bNumConfigurations	1	Nombre de configurations possibles

Tab II.3: List des champs de descripteur d'appareil. [7]

II.5.3.b.Descripteur de configuration

List des champs de descripteur de configuration donné dans le tableau II.4 :

Champ	Taille	Description
bLength	1	Taille du descripteur en octets
bDescriptorType	1	Type du descripteur = 2
wTotalLength	2	Taille totale de tous les descripteurs pour cette configuration
bNumInterfaces	1	Nombre d'interfaces supportées par cette configuration
bConfigurationValue	1	Numéro de cette configuration (A utiliser par SET_CONFIGURATION)
iConfiguration	1	Index du descripteur de chaine décrivant cette configuration
bmAttributes	1	Bit 6 : auto-alimentation, bit 5 : commande de réveil, bit 7 = 1
bMaxPower	2	Consommation maximum en unités de 2 mA.

Tab II.4: List des champs de descripteur de configuration. [7]

II.5.3.c. Descripteur d'interface

List des champs de descripteur d'interface donné dans le tableau II.5 :

Champ	Taille	Description
bLength	1	Taille du descripteur en octets
bDescriptorType	1	Type du descripteur = 4
bInterfaceNumber	1	Numéro de cette interface (Base 0)
bAlternateSetting	1	Numéro de l'interface alternative
bNumEndpoints	1	Nombre de terminaisons excluant les terminaisons zéro
bInterfaceClass	1	Code de classe assigné par l'USB-IF
bInterfaceSubClass	1	Code de sous-classe dépendant de la classe
bInterfaceProtocol	1	Code de protocole dépendant de la classe
iInterface	1	Index du descripteur de chaîne décrivant cette interface

Tab II.5: List des champs de descripteur d'interface. [7]

II.5.3.d. Descripteur de terminaison

List des champs de descripteur de terminaison donné dans le tableau II.6 :

Champ	Taille	Description
bLength	1	Taille du descripteur en octets
bDescriptorType	1	Type du descripteur = 4
bEndpointAddress	1	Adresse de cette terminaison Bits 0 à 3 : Numéro de terminaison Bit 7 : direction (0=OUT, 1=IN) - Ignoré pour canal de contrôle
bmAttributes	1	Propriétés de la terminaison. Bit 0 et 1 : Type de transfert - 00 Contrôle - 01 Isochrone - 10 Bloc - 11 Interruption Bit 2 à 5 : Paramètres pour transfert isochrone
wMaxPacketSize	2	Taille maximum des paquets de données en octets
bInterval	1	Intervalle entre chaque demande de données de l'hôte en nombre de trames. - Transfert isochrone pleine et haute vitesse : 1 à 16 - Transfert par interruption pleine et basse vitesse : 1 à 255 - Transfert par interruption haute vitesse : 1 à 16 - Transfert en bloc OUT haute vitesse : 1 à 255

Tab II.6: List des champs de descripteur de terminaison. [7]

II.6. Les protocoles USB

Contrairement à la RS232 et des interfaces sérielles similaires où le format des données envoyées n'est pas défini, l'USB est composé de plusieurs couches de protocoles. En fait la plupart des CIs contrôleur d'USB s'occuperont de la couche inférieure, la rendant ainsi presque invisible au regard du concepteur final.

Chaque transaction USB consiste d'un:

- Paquet Jeton (Token) (en tête définissant ce qu'il attend par la suite)
- Paquet DATA optionnel (contenant la " charge utile " (payload))
- Paquet d'Etat (utilisé pour valider les transactions et pour fournir des moyens de corrections d'erreurs).

Comme nous en avons déjà discuté, l'USB est un bus géré par l'hôte. L'hôte initie toutes les transactions. Le premier paquet, aussi appelé Jeton est produit par l'hôte pour décrire ce qui va suivre et si la transaction de données sera en lecture ou écriture et ce que sera l'adresse de l'appareil et la terminaison désignée. Le paquet suivant est généralement un paquet de données transportant la " charge utile " et est suivi par un paquet " poignée de mains " (handShaking), signalant si les données ou le jeton ont été reçus correctement ou si la terminaison est bloquée, ou n'est pas disponible pour accepter de données. [9]

II.6.1. Les champs de paquet USB ordinaires

Les données sur le BUS USB sont transmises avec le bit LSB en premier. Les paquets USB se composent des champs suivants :

- Sync : Tous les paquets doivent commencer avec un champ Sync. Le champ Sync fait de 8 bits dans le cas de la basse vitesse (Low speed) et pleine vitesse (Full Speed) ou 32 bits pour la haute vitesse (High Speed) est utilisé pour synchroniser l'horloge du récepteur avec celle de l'émetteur / récepteur. Les 2 derniers bits indiquent l'endroit où le champ PID commence.
- PID : PID (Tab II.7) signifie Paquet ID. Ce champ est utilisé pour identifier le type de paquet qui est envoyé. Le tableau suivant montre les valeurs possibles.
- ADDR : Le champ adresse détermine à quel appareil le paquet est destiné. Sa longueur de 7bits, lui permet de supporter 127 appareils. L'adresse 0 n'est pas valide, tant qu'un appareil qui n'a pas encore d'adresse attribuée, doit répondre aux paquets envoyés d'adresse 0.
- ENDP : Le champ de terminaison est composé de 4 bits, autorisant 16 terminaisons possibles. Les appareils bas vitesse, toutefois peuvent seulement avoir 2 terminaisons additionnelles au-dessus du canal de communication par défaut (4 terminaisons maximales).
- CRC : Les Contrôles à Redondance Cyclique sont exécutés sur les données à l'intérieur du paquet de charge utile. Tous les paquets jetons ont un CRC de 5 bits tandis que les paquets de données ont un CRC de 16 bits.
- EOP : Fin de Paquet. Signalé par une sortie unique zéro (SE0) pendant une durée approximative de 2 bits suivie par un " J " d'une durée de 1 bit.

List des différents PIDs donné dans le tableau II.7 :

PID Type	PID Name	PID[3:0]	Description
Token	0001	OUT	Données de l'hôte pour l'appareil
	1001	IN	Données de l'appareil pour l'hôte
	0101	SOF	Début de trame
	1101	SETUP	Initialisation de transaction de contrôle
Data	0011	DATA0	Paquet de données pair
	1011	DATA1	Paquet de données impair
	0111	DATA2	Paquet de données haute vitesse
	1111	MDATA	Paquet de données haute vitesse pour transactions "Split"
Handshake	0010	ACK	Acquittement positif
	1010	NAK	Acquittement négatif, appareil occupé
	1110	STALL	Terminaison ou canal hors service
	0110	NYET	Acquittement négatif haute vitesse
Special	1100	PREamble	Trafic vers un appareil basse vitesse
	1100	ERR	Erreur transmise par un hub haute vitesse dans une transaction "Split"
	1000	Split	Transaction avec un hub haute vitesse
	0100	Ping	Vérification de liaison
	0000	EXT	

Tab II.7 : Les différents PIDs. [7]

II.6.2.Les types de paquet USB

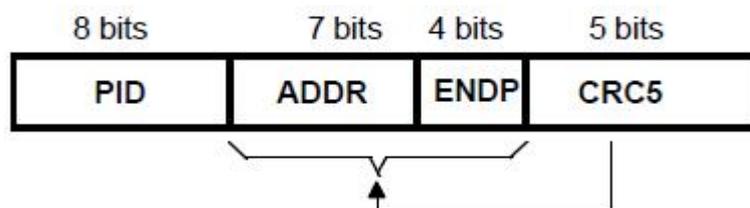
L'USB a quatre types différents de paquet. Les paquets jetons indiquent le type de la transaction qui va suivre, les paquets de données contiennent la charge utile, les paquets " poignée de mains " sont utilisés pour valider les données ou rapporter les erreurs et les paquets début de trame (SOF) indiquent le commencement d'une nouvelle trame. [9]

II.6.2.a.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.

Les paquets jetons doivent se conformer au format suivant :

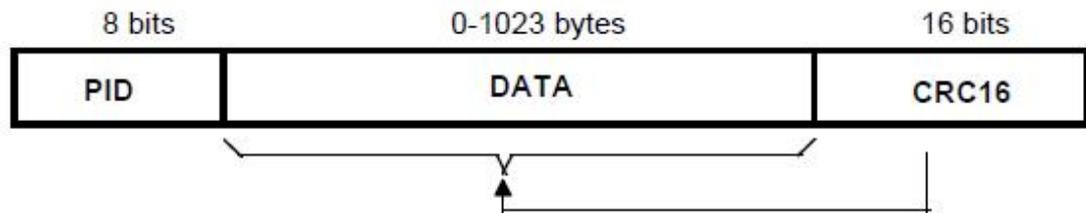


II.6.2.b. Les paquets de données

Il y a 2 sortes de paquets de données, chacun étant capable de transmettre plus de 1024 octets de données.

- Data0
- Data1

Le mode haute vitesse définit 2 autres PIDs de données, DATA2 et MDATA. Les paquets de données ont le format suivant :



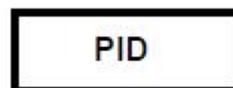
- La taille maximale de données " charge utile " pour les appareils basse vitesse est de 8 octets.
- La taille maximale de données " charge utile " pour les appareils pleine vitesse est de 1023 octets.
- La taille maximale de données " charge utile " pour les appareils haute vitesse est de 1024 octets.
- Les données doivent être envoyées en multiple d'octets.

II.6.2.c. Les paquets poignée de mains

Il y a 3 sortes de paquets " poignée de mains " qui font simplement partie du PID.

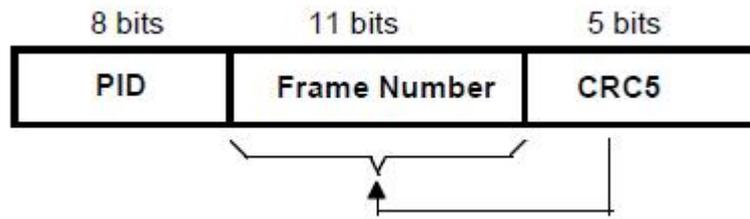
- ACK - validant que le paquet a été reçu correctement.
- NAK - rapporte que temporairement l'appareil ne peut ni envoyer ou recevoir des données. Aussi utilisé pendant les transactions d'interruptions pour avertir l'hôte qu'il n'a pas de données à envoyer.
- STALL (Bloqué) - L'appareil se retrouve dans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

Les paquets " poignée de mains " ont le format suivant :



II.6.2.d. Les paquets début de trame (SOF)

Le paquet SOF composé d'une trame de 11 bits est envoyé par l'hôte toutes les $1\text{ms} \pm 500\text{ns}$ sur un bus pleine vitesse ou bien toutes les $125\mu\text{s} \pm 0,0625\mu\text{s}$ sur un bus haute vitesse.



II.7. Transfert de données

II.7.1. Transfert en mode Contrôle

Ce mode de transfert est compatible avec le Low et Full Speed USB. Il est utilisé pour les opérations d'initialisations et de configurations. Il est éventuellement utilisable pour les transferts standards. Le mode contrôle est aussi utilisé pour tenter d'obtenir un débit Low Speed acceptable, ou pour utiliser le driver de classe HID (Human Interface Device) standard. [9]

II.7.2. Transfert en mode Interrupt

Ce mode de transfert est également compatible avec le Low et Full Speed USB. Il est destiné à des échanges limités et périodiques, il garantit la fréquence de scrutation ainsi que la reprise sur les erreurs. Il est utilisé pour des transferts à l'initiative du périphérique (asynchrones) et pour des transferts périodiques ou permanents comme les claviers. [9]

II.7.3. Transfert en mode Isochrone

Ce mode de transfert est uniquement compatible avec le Full USB. La bande passante est garantie (début, latence), par contre dans ce mode il n'y a pas de reprise sur erreur. Il est utilisé pour des transferts nécessitant un flux régulier de données comme par exemple les caméras ou les téléphones ... La bande passante réclamée et non utilisée est perdue. [9]

II.7.4. Transfert en mode Bulk

Ce mode de transfert est uniquement compatible avec le Full USB. Ce mode est réservé pour les gros transferts de données (ex : imprimantes...) Le débit est variable et dépend de la disponibilité. Ce mode assure la reprise sur les erreurs.

Les échanges isochrones sont les plus privilégiés dans le sens où le host leur réserve une bande passante garantie. Celui-ci peut refuser l'accès au bus à un périphérique s'il juge que les ressources qu'il requiert ne sont pas disponibles. [9]

II.8. PID/VID

II.8.1. Introduction au Product ID et au Vendor ID

Lors de l'installation d'un nouveau périphérique USB, les descripteurs fournissent les informations le concernant à savoir un PID (Product ID) et un VID (Vendor ID). C'est grâce à ces deux valeurs que le PC peut reconnaître l'identité du composant. Comme dit précédemment la réglementation des VID est très stricte, cher et est délivré par le forum USB-IF accessible depuis le site <http://www.usb.org>.

Chaque fabricant possède un VID et c'est grâce à cette valeur codée sur 16 bits que l'on peut retrouver le fabricant du composant. Chaque fabricant ayant plusieurs produits à leurs actifs, ils les différencient avec le PID codé également sur 16bits.

L'allocation des PID, contrairement aux VID, est faite par le constructeur du dispositif. Il n'y a aucune contrainte administrative de la part du forum USB-IF. [9]

II.8.2.Normalisation des PID/VID

Autrement dit chaque couple PID/VID doit être unique sur le marché.

Il existe un site répertoriant la longue liste des VIDs et des constructeurs correspondant. Il est même possible de connaître tous les PIDs des mêmes constructeurs, c'est à dire qu'il est possible de retrouver le modèle exact du périphérique. Voici le site, il pourra vous être utile un jour... <http://www.yourvote.com/pci>. Un petit moteur de recherche vous permettra de trouver rapidement le constructeur en vous réorientant vers son site. Ce site répertorie les PID/VID du bus PCI (Peripheral Component Interconnect). En ce qui concerne l'USB il faut s'adresser au forum. [9]

II.9.Les fichiers INF/SYS

II.9.1.Les fichiers d'extension INF

II.9.1.a.Définition

L'abréviation INF provient du mot « Information » et comme son nom l'indique, renseigne sur les informations de configuration de périphérique utile pour le Plug & Play.

Les points INF ne sont pas seulement utilisés pour les composants USB, les périphériques PCI l'utilisent également. Un fichier INF est un fichier texte, vous pouvez l'éditer avec n'importe quel éditeur de texte standard. [9]

II.9.1.b.Fonctionnement des points INF

En règle générale, les fichiers INF utiles à l'installation sont déjà présents dans les fichiers INF par défaut dans Windows. Les Fichiers INF par défaut sont des fichiers INF génériques c'est à dire qu'ils sont capable de débiter l'installation de plusieurs types de périphérique. Nous verrons par la suite comment est structuré un point INF et quels sont les critères qui permettent aux fichiers INF de reconnaître des périphériques.

Un fichier INF est organisé en plusieurs sections. Chaque section (il en existe une vingtaine) possède une fonction particulière. Ces fonctions vont de la simple manipulation de fichiers jusqu'à la modification d'entrées dans la base de registres en passant par les fichiers INI.

II.9.2.Les drivers .SYS

II.9.2.a.Définition d'un driver .SYS

Comme dit précédemment, un driver a le plus souvent l'extension SYS. Ce driver doit être fourni avec le périphérique à installer... et c'est le plus souvent le cas. A chaque fois que vous installez chez vous votre imprimante ou scanner ou autre périphériques, il vous faut la première fois un CD d'installation. Comme les .INF il existe aussi des .SYS par défaut.

Par contre, contrairement au fichier INF, il n'est pas si évident de concevoir un WDM driver. (Windows Driver Model). Il nécessite des compétences en programmation C et des connaissances sur la communication entre Windows, le hardware et les applications.

Par contre, les connaisseurs disent qu'il est beaucoup plus facile d'écrire un driver USB qu'un driver ISA.

Dans la nature, un fichier SYS est un fichier compilé et il est impossible de le modifier pour l'adapter à un autre type de périphérique. Donc si vous voulez débiter dans le développement de périphérique USB, assurez-vous que vous disposez d'un tel fichier.

Sans celui-ci, le composant ne sera pas installé et ne pourra pas communiquer avec le PC. Bien sûr, dans la plupart des cas, si le périphérique à installer est standard, un point SYS par défaut dans "C:/Windows/SYSTEM32/Drivers" résout bien les problèmes. Il se peut par contre qu'il ne soit pas optimum mais vous donnera une performance suffisante pour une première tentative. [9]

II.10.Conclusion

Dans cette partie, nous avons étudié le bus USB et sa norme, comme nous avons parlé aussi de l'importance et l'avantage du bus USB en comparaison avec le reste.

Dans ce projet nous allons utiliser le bus USB du pic, c'est pour ça nous allons passer à la section consacré au pic et tout ce qui le concerne

Le microcontrôleur (PIC)

CHPITRE III : Le microcontrôleur (PIC)

1 - Introduction	33
2 - Le Microcontrôleur (PIC)	33
2.1 - Définition d'un PIC	33
2.2 - Classification des PICs de Microchip	33
2.3 - Identification d'un Pic	33
2.4 - Le choix d'un PIC	34
2.5 - Caractéristiques principales du pic 18F4550	35
3 - Brochage du 18F4550	35
4 - Architecture générale du PIC	37
4.1 - Architecture interne du PIC 18F4550	37
4.2 - Les horloges et le Reset	38
4.3 - Les timers	40
4.4 - Les convertisseurs analogiques/numériques	40
4.5 - Les interruptions et leur gestion	47
4.6 - Le bus USB	49
5 - Conclusion	52

III.1.Introduction

Les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications grand public ou professionnelles, il en existe plusieurs familles. [13]

La société Américaine Microchip Technologie a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (Peripheral Interface Controller). Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

- Les PIC existent dans plusieurs versions:
- les UVROM qui sont effaçable par une source de rayonnements ultraviolets
- les OTPROM programmable une seule fois
- les E²PROM et flash EPROM qui sont effaçables électriquement.

III.2.Le Microcontrôleur (PIC)

III.2.1.Définition d'un PIC

Un PIC est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes. Les PICs sont des composant dits RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instructions réduit. Le microcontrôleur se trouve, dans plusieurs appareils tels que : les téléphones portables, machines à laver, télévisions les jeuxetc. [14]

III.2.2.Classification des PICs de Microchip

Actuellement les modèles Microchip, sont classés en 3 grandes familles, comportant chacune plusieurs références. Ces familles sont : [13]

- Base-line : les instructions sont codées sur 12 bits.
- mid-line : les instructions sont codées sur 14 bits.
- High-End : les instructions sont codées sur 16 bits.

II.2.3.Identification d'un Pic

Un PIC est identifié par un numéro de la forme suivant : xx(L)XXyy –zz:

- xx : famille du composant, actuellement « 12, 14, 16, 17 et 18 ».
- L : tolérance plus importante de la plage de tension.
- XX : type de mémoire programme :
 - C : EPROM ou EEPROM.

- CR : PROM.
- F : flash.
- yy : constitue la référence du PIC.
- zz : indiquant la fréquence d'horloge maximale que le PIC peut recevoir (Vitesse maximum du quartz).

Donc, un pic 18F4550 est un PIC High-End donc la mémoire programme est de type FLASH, de Identificateur. 45 et capable d'accepter une fréquence d'horloge de 50MHz.

Notez que les PICs sont des composants statiques, c'est à dire que la fréquence d'horloge peut être abaissée jusque l'arrêt complet sans perte de données et sans dysfonctionnement. [13][15]

II.2.4.Le choix d'un PIC

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

- Il faut dans un premier temps déterminer le nombre d'entrées/sorties nécessaires pour l'application. Ce nombre d'entrées/sorties nous donne une idée sur la famille du PIC.
- Il faut ensuite déterminer si l'application nécessite un convertisseur Analogique/Numérique ce qui va orienter un peu plus vers le choix du PIC.
- La rapidité d'exécution est un élément important, il faut consulter les DATA-BOOK pour vérifier la compatibilité entre la vitesse maximale du PIC choisi et la vitesse max nécessaire au montage.
- La taille de la RAM interne et la présence ou non d'une EEPROM pour mémoriser des données sont également importantes pour l'application souhaitée.
- La longueur du programme de l'application détermine la taille de la mémoire programme du PIC recherché.

Pour notre projet, on a choisis d'utiliser le PIC 18F4550 à cause de son prix raisonnable et la facilité de programmation, aussi, le nombre d'entrées/sorties est suffisant pour notre application on a besoin ainsi de convertisseurs analogique-numérique avec une mémoire, ainsi qu'un module USB.[15]

II.2.5. Caractéristiques principales du pic 18F4550

Les caractéristiques principales (Tab III.1) du 18F4550 sont résumées comme suit :

Caractéristiques	Valeur/Type
Type de mémoire de programme	Flash
Mémoire de programme (KB)	32
EEPROM données (octets)	256
Vitesse du processeur (MIPS)	12
RAM octets	2,048
EEPROM données (octets)	256
Périphériques de communication numérique	1-A/E/USART, 1-MSSP(SPI/I2C)
Capture/Compare/ Périphériques PWM	1 CCP, 1 ECCP
Timers	1 x 8-bit, 3 x 16-bit
CAN	13 canaux, 10-bit
Comparateurs	2
USB (ch, speed, compliance)	1, Full Speed, USB 2.0
Température (C)	-40 to 85
Tension de fonctionnement (V)	2 to 5.5
Nombre de broches	40

Tab III.1: Caractéristiques principales du pic 18F4550. [15]

III.3. Brochage du 18F4550

La figure III.1 résume le brochage du PIC 18F4550 :

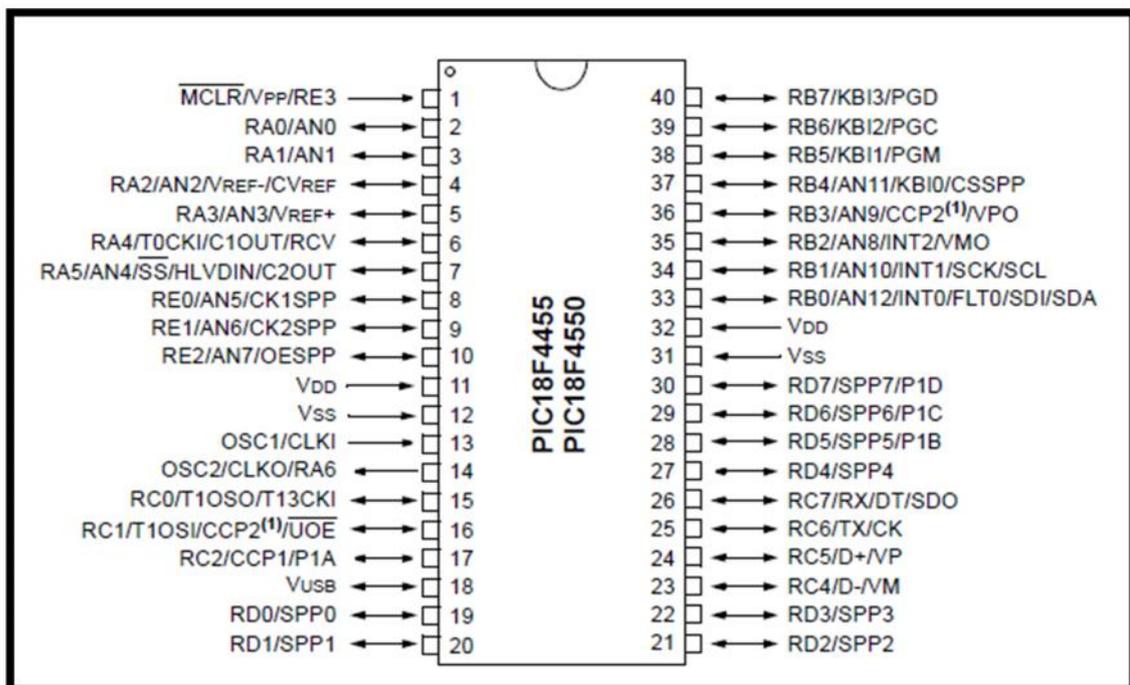


Fig III.1: Brochage du PIC 18F4550. [15]

Ces broches sont définies comme suit :

- **MCLR** : entrée de Reset - entrée haute tension de programmation
- **ANx** : entrée analogique.
- **RAx** : entrée/sortie logique du port A.
- **RBx** : entrée/sortie logique du port B.
- **RCx** : entrée/sortie logique du port C.
- **RDx** : entrée/sortie logique du port D.
- **REx** : entrée/sortie logique du port E.
- **Vref +** : entrée de tension de référence + pour convertisseur analogique.
- **Vref -** : entrée de tension de référence - pour convertisseur analogique.
- **CCPx** : entrée/sortie module CCP (Capture Compare PWM).
- **P1A** : sortie PWM en mode EPWM.
- **P1B** : sortie PWM en mode EPWM.
- **P1C** : sortie PWM en mode EPWM.
- **P1D** : sortie PWM en mode EPWM.
- **CLKin** : entrée d'horloge.
- **CLKout** : sortie d'horloge.
- **T0CKI** : entrée d'horloge externe timer 0.
- **T1CKI** : entrée d'horloge externe timer 1.
- **T1OSI** : entrée oscillateur timer 1.
- **OSC1** : broche oscillateur circuit (quartz).
- **OSC2** : broche oscillateur circuit (quartz).
- **SCK** : horloge en mode SPI.
- **SDI** : entrée de donnée en mode SPI.
- **SDO** : sortie de donnée en mode SPI.
- **SS** : sélection circuit maître/esclave en mode SPI.
- **TX** : sortie émission EUSART.
- **RX** : entrée réception EUSART.
- **Vss** : 0V de l'alimentation.
- **Vdd** : +5V.
- **Vpp** : tension de programmation.
- **Vusb** : sortie 3.3V du régulateur intégré USB.
- **D +** : Data +.
- **D -** : Data -.
- **PSPx** : entrée/sortie logique du port parallèle esclave.
- **RD** : entrée de lecture en mode PSP.
- **WR** : entrée d'écriture en mode PSP.
- **CS** : entrée de sélection de circuit (Chip Select) en mode PSP.
- **OE** : sortie de validation données (Output Enabled) en mode PSP.

III.4. Architecture générale du PIC

III.4.1. Architecture interne du PIC 18F4550

La figure III.2, présente l'architecture interne du circuit. Il est constitué des éléments suivants : [16]

- Un système d'initialisation à la mise sous tension (power-up timer ...).
- Un chien de garde WDT (Watch Dog Timer).
- Un système de génération d'horloge à partir du quartz externe (timing génération).

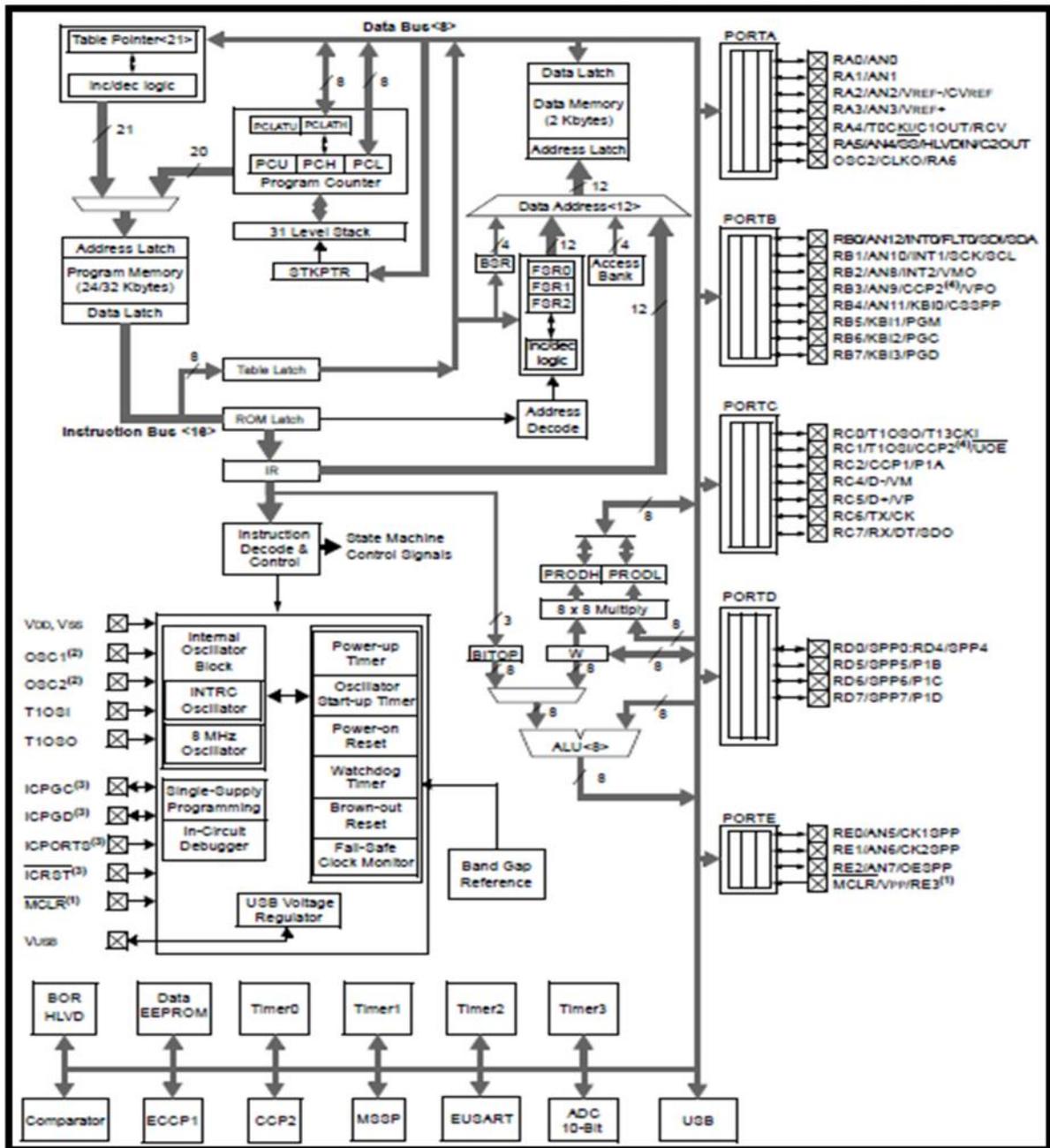


Fig III.2: Architecture interne du PIC 18F4550. [15]

III.4.2. Les horloges et le Reset

III.4.2.a. Horloge

L'horloge va permettre de cadencer tous les échanges internes et externes du PIC (Fig III.4). En cas de problème sur l'horloge, le programme s'arrête. Il existe sur certains PIC une possibilité de basculement sur une deuxième horloge externe ou interne. [16]

III.4.2.b. Type d'horloge externe

Avec la plupart des PIC, elle sera possible d'utiliser trois sortes d'horloge (Fig III.3) :

- La première est réalisée avec un quartz.
- La seconde utilise un circuit RC.
- La troisième utilise un cadencement provenant d'une horloge externe permettant la libération d'une broche.

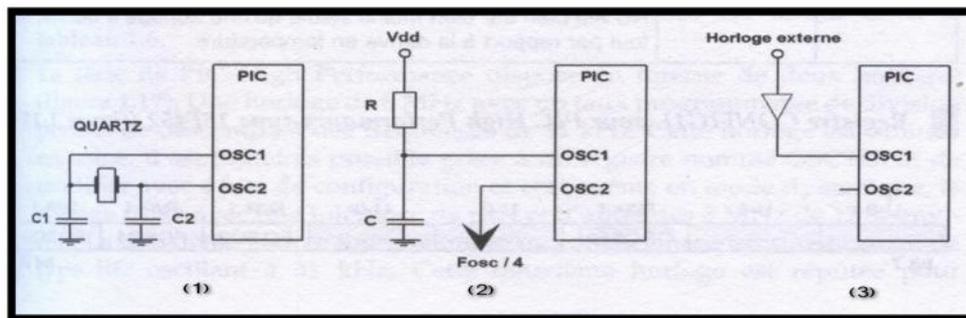


Fig III.3: Exemples de configuration de l'horloge. [16]

Le rôle des bits FOSC0 à FOSC3 pour le 18F4550 (x= 0 ou 1) est donné dans le tableau III.2 :

FOSC3	FOSC2	FOSC1	FOSC0	Type d'horloge
0	0	0	x	XT
0	0	1	x	XT+PLL
0	1	0	0	ECIO
0	1	0	1	EC
0	1	1	0	EC+PLL
0	1	1	1	EC+PLL sortie Clock/4 RA6.
1	0	0	0	INTIO Horloge interne. EC pour cadencement USB.
1	0	0	1	INTCK0 Horloge interne Clock/4 sur RA6. EC utilisée pour cadencement USB.
1	0	1	0	INTXT Horloge interne. XT utilisé pour cadencement USB.
1	0	1	1	INTHS horloge interne. HS utilisé pour cadencement USB.
1	1	0	x	HS
1	1	1	x	HS+PLL

Tab III.2: Rôle des bits FOSC0 à FOSC3 pour le 18F4550. [15]

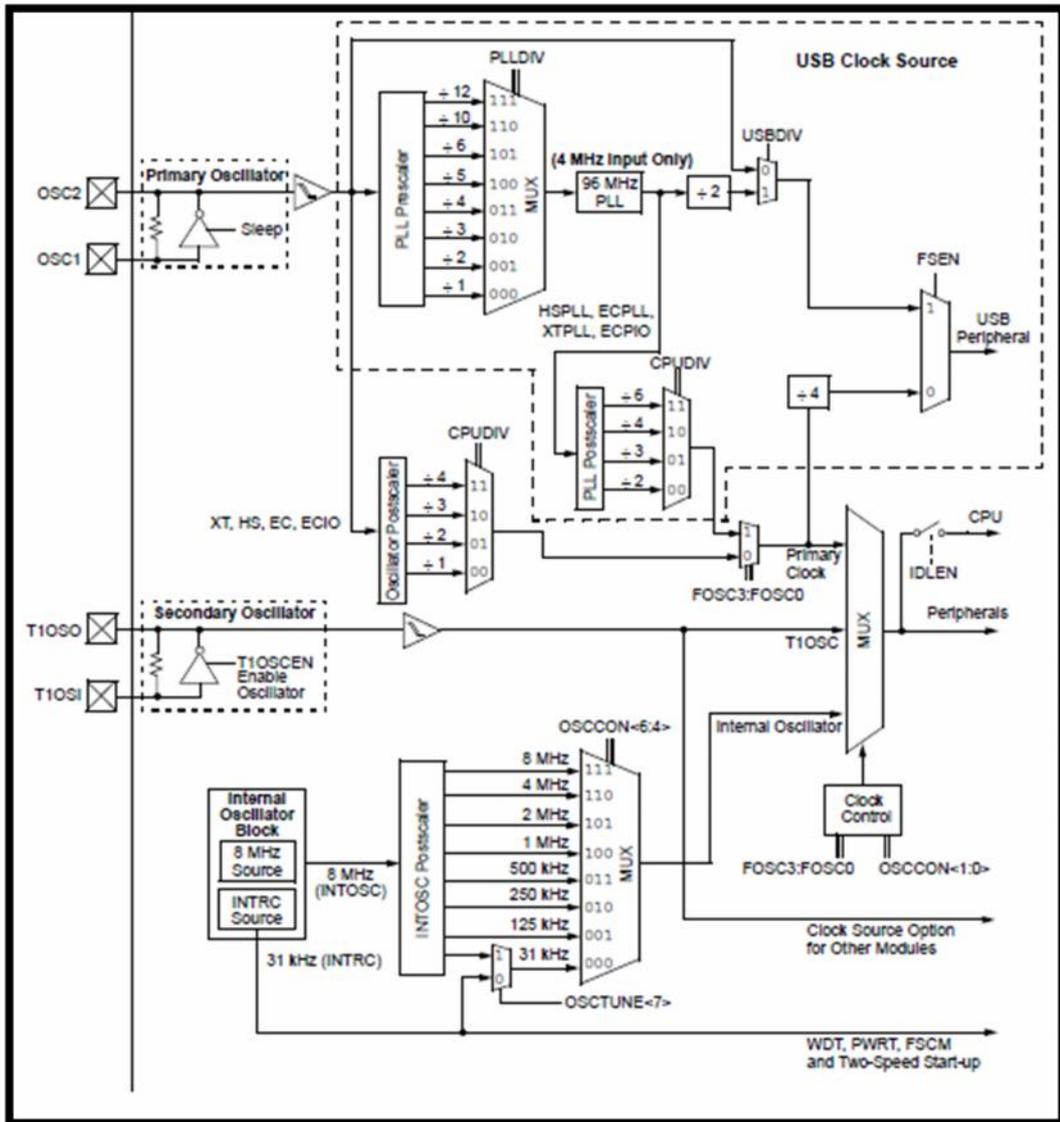


Fig III.4: Synoptique de la circuiterie d'horloge sur un PIC 18F4550 gérant l'USB. [15]

III.4.2.c. Reset

Le Reset d'un microcontrôleur peut être déclenché selon plusieurs événements. Ces événements peuvent être d'origine physique, tel que l'appui sur le bouton Reset du montage (broche $\overline{\text{MCLR}}$) ou encore à cause d'une baisse de tension d'alimentation. Un Reset peut également être déclenché suite à un état logiciel tel que le débordement (ou saturation) de la pile (PIC 18). [16]

Pour résumer, un Reset peut être déclenché par :

- Un appui sur le bouton Reset du montage (broche $\overline{\text{MCLR}} = 0$).
- Une instruction logicielle Reset (PIC 18).
- Une saturation ou un débordement inférieur de la pile (PIC 18).
- Un dépassement de la temporisation du chien de garde (WDT).

- Une atteinte du seuil mini de l'alimentation (BOR).
- Une mise sous tension (POR).

La figure III.5 résumé la circuiterie de la fonction Reset qui peut être déclenché dans les 6 cas possibles :

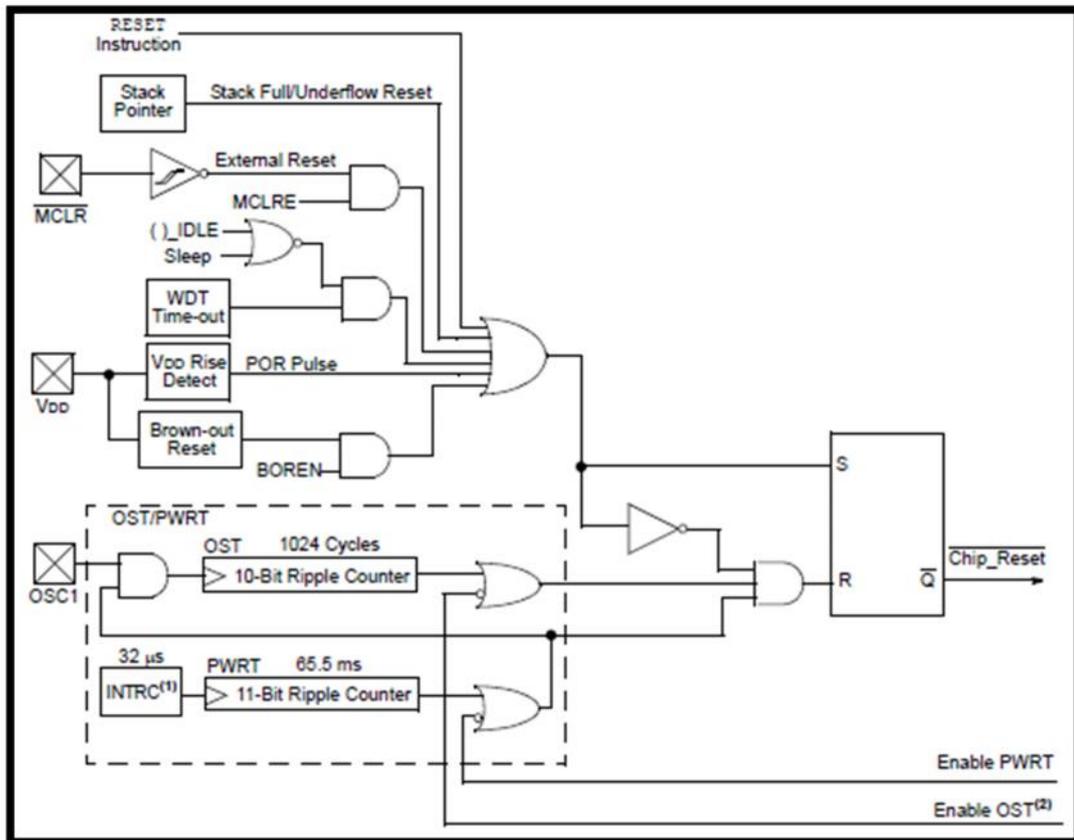


Fig III.5: Synoptique de la circuiterie Reset. [15]

III.4.3. Les timers

Le timer est un élément important dans un PIC. En effet, cette ressource permettra de réaliser de nombreuses fonctions, telle que la temporisation, la génération de signaux PWM (appelée également Modulation de Largeur d'Impulsion, MLI), ou encore les modes Compare ou Capture. Selon le type de PIC rencontré, on trouvera un certain nombre de timers qui seront pour certains associés à des ressources internes du PIC. Un timer est en fait un compteur s'incrémentant au rythme d'une horloge, et ceci en ayant la possibilité de prépositionner une valeur de départ. Certains timers pourront fonctionner en mode 8 bits, 16 bits ou 32 bits. [16]

III.4.4. Les convertisseurs analogiques/numériques

Un convertisseur analogique/numérique (CAN), ou en anglais ADC pour (Analog to Digital Converter), est un montage électronique dont la fonction est de traduire une grandeur analogique en une valeur numérique (codée sur plusieurs bits), proportionnelle au rapport entre la grandeur analogique d'entrée et la valeur maximum du signal.

Pour utiliser un convertisseur analogique/numérique, Il existe trois registres internes nommés ADCON0, ADCON1 et ADCON2 qui permettront de configurer et de sélectionner les différents entrées analogiques du PIC utilisés. Le paramétrage de ces trois registres conduit à choisir la façon dont se comporteront le ou les convertisseurs utilisés. Les entrées du PIC en relation avec le convertisseur analogique/numérique interne sont les broches AN0 à AN12. Les bits CHS0 à CHS3 du registre ADCON0 permettent de sélectionner un canal (AN0 à AN12) qui sera en relation avec le convertisseur. Les bits VCFG0 et VCFG1 du registre ADCON1 permettent quant à eux de définir les références de tension (AN2 ou Vss pour Vref- et AN3 ou Vdd pour Vref+). Les bits PCFG0 à PCFG3 du même registre permettant de configurer chaque broche AN0 à AN12 soit en entrée analogique soit en entrée/sortie logique. Le troisième registre ADCON2 permettra de sélectionner une horloge pour le convertisseur analogique (bits ADCS0 à ADCS2), ainsi qu'un temps d'acquisition avant conversion (temps nommé TAD : bits ACQT0 à ACQT2), allant 0 TAD à 20 TAD (soit 0 top d'horloge à 20 top d'horloge), puis indiquera grâce au bit ADFM le format de la valeur de la conversion en sortie.

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

III.4.4.a. Le CAN du PIC 18F4550

Caractéristiques :

- CAN à approximations successives.
- 10 bits (entre 0 et 5V en entrée au maximum).
- Résolution de 5 mV au maximum.
- Temps de conversion total : 17.5 μ s.
- IT générée à la fin de la conversion.
- Les niveaux bas et haut de la tension de référence (Vref- et Vref+) peuvent être sélectionnés par l'utilisateur.
- Peut-être connecté à une des 12 entrées analogiques et à une seule en même temps.

Les registres associés :

- INTCON, PIR1, PIE1, IPR1 sont liés aux IT.
- TRISx, LATx et PORTx sont liés aux broches utilisées pour le CAN.
- Il reste 3 registres de configuration : ADCON0, ADCON1 et ADCON2.
- Et 2 registres d'utilisation : ADRESH et ADRESL.

Registre ADCON0 :

Le registre ADCON0 permet de sélectionner quelle entrée analogique va être connectée à l'entrée du CAN (bit CHS3 à CHS0). Ce registre contient également les bits permettant de mettre sous tension le CAN et de lancer une conversion.

R/W-0	R/W-0						
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7-6 : Non-implémenté : Lire comme '0'.
- bit 5-2 : CHS3:CHS0: Bits de sélection du canal analogique:
 - 0000 = Channel 0 (AN0).
 - 0001 = Channel 1 (AN1).
 - 0010 = Channel 2 (AN2).
 - 0011 = Channel 3 (AN3).
 - 0100 = Channel 4 (AN4).
 - 0101 = Channel 5 (AN5).
 - 0110 = Channel 6 (AN6).
 - 0111 = Channel 7 (AN7).
 - 1000 = Channel 8 (AN8).
 - 1001 = Channel 9 (AN9).
 - 1010 = Channel 10 (AN10).
 - 1011 = Channel 11 (AN11).
 - 1100 = Channel 12 (AN12).
- bit 1 GO/DONE : Bit de statut du CAN :
 - 1 = Conversion A/D en cours.
 - 0 = Conversion A/D pas en cours.
- bit 0 ADON : Bit de mise en route du module A/D:
 - 1 = Le module convertisseur A/D est en service.
 - 0 = Le module convertisseur A/D est désactivé.

Registre ADCON1 :

Le registre ADCON1 permet de configurer l'état des entrées (analogique ou TOR), mais également de sélectionner les tensions qui serviront de référence au CAN.

R/W-0							
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7-6 : Non-implémenté, Lire comme '0'.
- bit 5 VCFG1 : Bits de sélection de tension de référence (VREF- source) :
 - 1 = VREF- (AN2).
 - 0 = V_{SS}
- bit 4 VCFG0 : Bits de sélection de tension de référence (VREF+ source) :
 - 1 = VREF+ (AN3).
 - 0 = V_{DD}
- PCFG3:PCFG0 : Bits de contrôle de configuration de port A/D (Tab III.3) :

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

Tab III.3: Bits de contrôle de configuration de port A/D. [15]

Registre ADCON2 :

Le registre ADCON2 permet de gérer le timing de fonctionnement du CAN et donc le temps qu'il va mettre à faire une acquisition. Il sert également à choisir la présentation du résultat (justifié à gauche ou à droite)

R/W-0							
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- bit 7 ADFM: Bits de sélection du format de résultat du CAN:
 - 1 = Justifié à droite.
 - 0 = Justifié à gauche.
- bit 6 : Non-implémenté, Lire comme '0'.
- bit 5-3 ACQT2:ACQT0 : Bits de sélection de temps d'acquisition du CAN :
 - 111 = 20 TAD.
 - 110 = 16 TAD.
 - 101 = 12 TAD.
 - 100 = 8 TAD.
 - 011 = 6 TAD.
 - 010 = 4 TAD.
 - 001 = 2 TAD.
 - 000 = 0 TAD.
- bit 2-0 ADCS2:ADCS0 : Bits de sélection d'horloge du CAN:
 - 111 = F_{RC} .
 - 110 = $F_{OSC}/64$.
 - 101 = $F_{OSC}/16$.
 - 100 = $F_{OSC}/4$.
 - 011 = F_{RC} .
 - 010 = $F_{OSC}/32$.
 - 001 = $F_{OSC}/8$.
 - 000 = $F_{OSC}/2$.

Registres ADRESH et ADRESL :

- Les registres ADRESH et ADRESL permettent de stocker le résultat de la conversion.
- Le CAN fournissant 10 bits, il est nécessaire d'avoir deux registres de 8 bits.
- ADRESH contient les 8 bits de poids fort.
- ADRESL les 8 bits de poids faible.

Le Synoptique général du CAN de PIC 18F4550 est donné dans la (Fig III.6) :

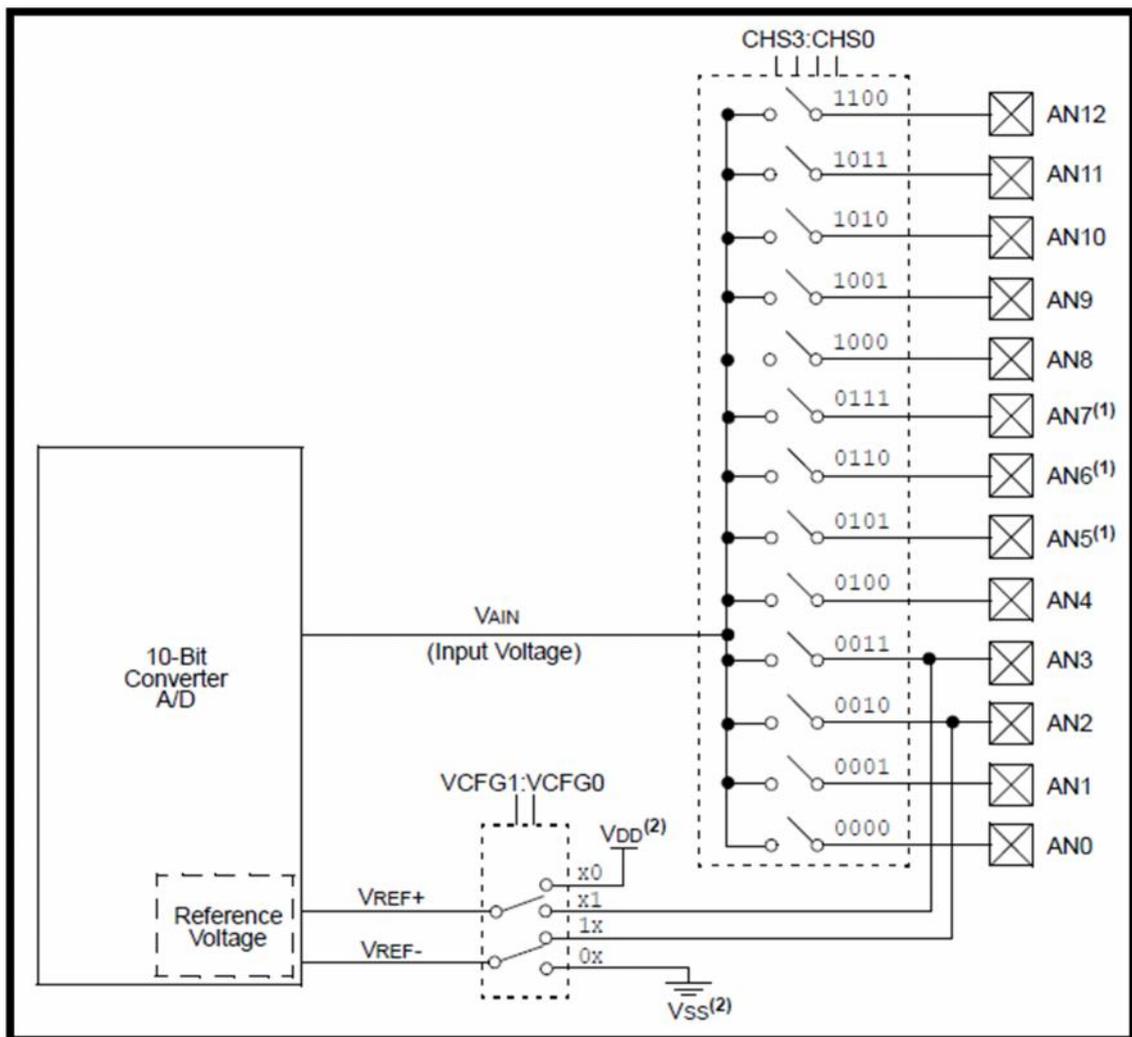


Fig III.6: Synoptique général du CAN pour le 18F4550.

La fréquence d'échantillonnage:

La fréquence d'échantillonnage est importante pour le fonctionnement du CAN du microcontrôleur, elle est choisie parmi plusieurs valeurs qui peuvent générer la circuiterie d'horloge du PIC (Fig III.4).

Pour le calcul de la fréquence d'échantillonnage, on utilise le temps nécessaire pour convertir un bit qui s'appelle le TAD (Time A/D). Le temps minimal du TAD du microcontrôleur utilisé est de 0.8 μ s d'après la notice.

Pour choisir la fréquence d'échantillonnage convenable, on doit choisir le TAD adéquat tel que sa valeur correspond à l'une des valeurs suivantes :

- 2 TOSC,
- 4 TOSC,
- 8 TOSC,
- 16 TOSC,
- 32 TOSC,
- 64 TOSC.

Où le TOSC (Time of the OSCillator) est la période de la fréquence FOSC (Frequency of the OSCillator) 48 MHz de l'oscillateur (c'est la fréquence de quartz 20 MHz qui est divisée par 5 puis multipliée par 22 et enfin la fréquence est divisée par 2 par CPUDIV pour obtenir 48 MHz) Donc :

$$TOSC = \frac{1}{Fosc} = \frac{1}{48} \approx 20.833 \text{ ns}$$

Le TAD minimal est divisé par le TOSC pour obtenir la valeur :

$$TAD \approx \frac{TAD_{min}}{Tosc} = \frac{800}{20.833} \approx 38.4$$

Puisque $32 < 38.4 < 64$, on a choisi 64 TOSC. Le TAD qu'on doit prendre en compte est alors :

$$TAD = 64 \cdot TOSC = 1.33 \mu\text{s}$$

Une deuxième grandeur doit être calculée, le temps d'acquisition Tacq, c'est le temps de lecture de l'échantillon analogique converti en valeur numérique. Ce temps d'acquisition au TAD calculé précédemment, par un facteur.

Le T_{acq} minimal est de $2.5 \mu s$ (d'après datasheet) on a prend :

$$T_{acq} = 2.TAD = 2.66 \mu s$$

Le temps de conversion total (Fig III.7) T_c est la somme entre :

- T_{acq} : le temps d'acquisition ($2.66 \mu s$),
- T_{cnv} : le temp de conversion ($11.TAD = 14.63 \mu s$)
- T_{dis} : le temps de décharge ($0.2 \mu s$)

$$T_c = T_{acq} + T_{cnv} + T_{dis} = 17.5 \mu s$$

La fréquence d'échantillonnage F_e sera calculée à partir du temps de conversion T_c :

$$F_e = \frac{1}{T_c} = 57.15 \text{ KHz}$$

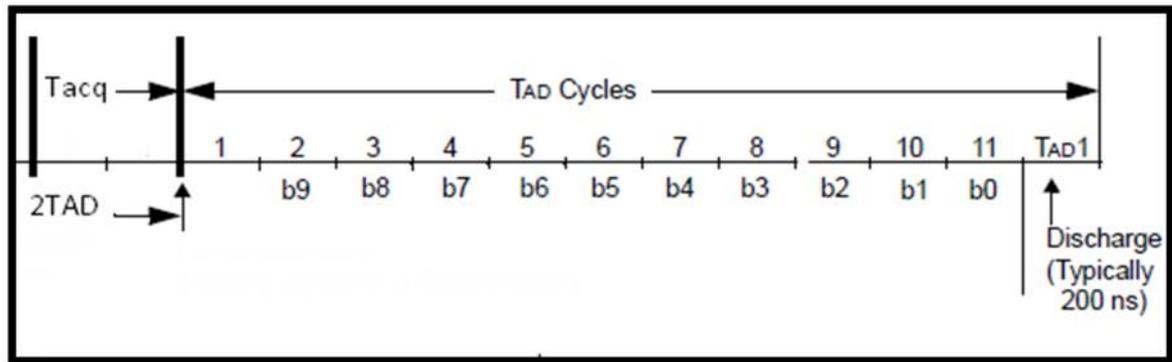


Fig III.7: Les cycles de conversion totale. [15]

III.4.5. Les interruptions et leur gestion

III.4.5.a. L'interruption

Une interruption est un évènement extérieur au programme déclenchée par différentes causes, par exemple un changement d'état sur une broche d'entrée ou encore un débordement d'un registre. Souvent, le mot Interruption est remplacé par IT. Lorsque l'IT est effective, le programme est dirigé vers une adresse spécifique de la mémoire du microcontrôleur et traite alors un sous-programme dit «prioritaire». Lorsque le sous-programme d'interruption est terminé, le PC est de nouveau chargé avec la valeur de l'adresse de retour, adresse qui correspond en fait à l'instruction suivante à exécuter par le programme au moment où l'IT s'est déclenchée. Cet évènement extérieur n'est pas synchronisé avec le programme principal, il peut donc survenir à n'importe quel moment, c'est pour cette raison que l'on dira qu'une interruption est asynchrone.

Sur les microcontrôleurs de la série 18F, il existe deux niveaux de priorité pour les interruptions, on parlera dans ce cas de niveau d'interruption avec gestion des priorités selon la nature de l'évènement qui survient. Chaque source d'IT pourra être déclarée en haute ou basse priorité, Un bit permet d'activer ou non le mode hiérarchique sur les IT. [16]

III.4.5.b. Les différents causes de déclenchement d'interruption

Pour le PIC 18F4550, chaque source d'interruption sera représentée par un bit de mise en service (Enabled), 1 bit d'état (flag) et 1 bit de niveau de priorité (haute ou basse, voir la figure III.8). Nous allons retrouver les sources d'IT suivantes :

- le débordement d'un timer (flag TRMxIF).
- la présence d'un changement de front (paramétrable) sur les broches RBx/INTx (flag INTxF).
- changement d'état d'une des broches RB4 à RB7 (flag RBIF).
- la réception ou l'émission de données avec l'EUSART (flags RCIF TXIF).
- la fin de conversion d'un signal analogique (flag ADIF).
- le transfert de données sur le port parallèle esclave (flag PSPIF).
- la fin de écriture dans l'EEPROM (flag EEIF).
- des événements sur le module MSSP (1²C, SPI) (flags SSPIF BCLIF).
- des événements sur le module Capture Comparaison PWM (CCP) (flags CMIF CCPxIF).
- des événements sur le module USB (flag USBIF).

Dix registres permettent la gestion des interruptions. Le registre INTCON contient quant à lui le bit d'autorisation général des IT (bit GIE pour Global Interrupt Enabled bit), ainsi que le bit d'autorisation des interruptions provenant des périphériques internes (timers, SSP, EUSART, etc.). [16]

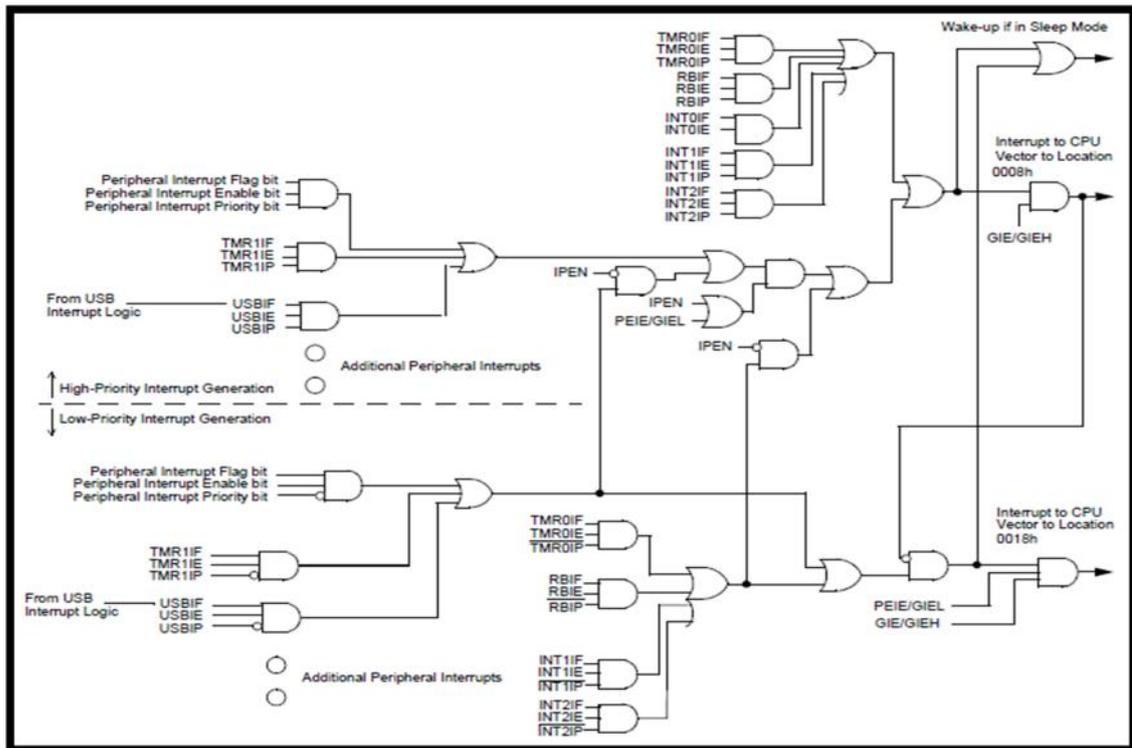


Fig III.8: Synoptique général des interruptions sur PIC 18F4550. [15]

III.4.6. Le bus USB

III.4.6.a. Architecture simplifiée d'un bus USB

L'architecture simplifiée d'un bus USB est donnée dans la figure III.9 :

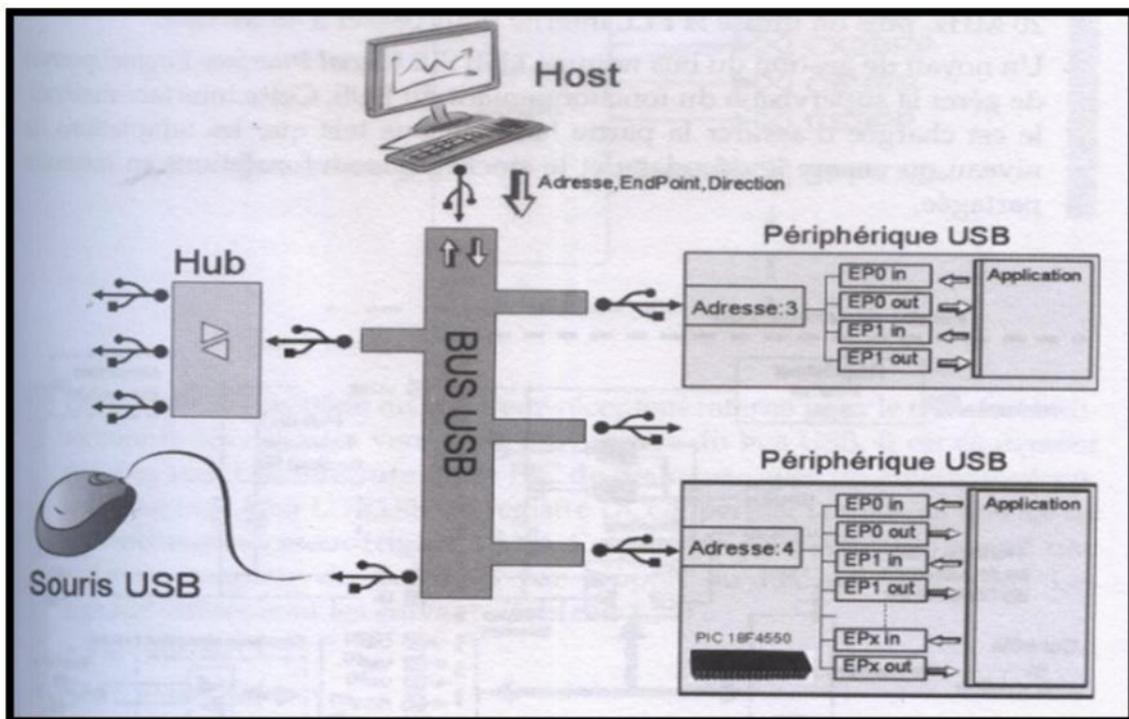


Fig III.9: Architecture simplifiée d'un bus USB. [16]

III.4.6.b. Le Microcontrôleurs PIC et USB

Des PIC de la famille 16 (PIC 16F745-16F765 par exemple) gèrent le bus USB1, d'autres PIC de la famille 18F, tel que le 18F4550, gèrent le bus USB2.

Les broches D+ et D- permettent l'interfaçage avec le bus USB. Comme nous l'avons déjà mentionné, des résistances de pull-up internes permettent la configuration du PIC en tant que périphérique basse ou haute vitesse et également la détection du branchement d'un périphérique sur le bus (Fig III.10). Ces résistances sont pilotées par deux bits de configuration (FSEN et UPUEN du registre UCFG). Dans le cas d'utilisation de résistances externes, les résistances de pull-up internes doivent bien sûr être désactivées. Un régulateur interne de 3,3V est également implémenté, Celui-ci est actif selon la configuration du bit VREGEN (registre CONFIG2L). Dans le cas d'une alimentation 3,3V externe, le régulateur interne devra être désactivé.

On peut également remarquer sur le schéma de l'architecture qu'une mémoire RAM partagée dual port (adresse 400-7FF) de 1 Ko est allouée en interne pour les échanges entre le bus USB (via le SIE) et l'application. Cette allocation mémoire de 1 Ko est divisée en 4 banques. La banque 4 (400-4F11) est chargée du contrôle des tampons des ends points (buffers descriptors EP), les banques 5 à 7 sont utilisées pour les données (USB DATA). Le PIC 18F4550 peut gérer jusqu'à 16 end points, le codage est défini dans le registre USTAT.

L'horloge est issue de l'oscillateur local. En général, on place un quartz de 20 MHz, puis on utilise la PLL interne pour passer à 48 MHz.

Un noyau de gestion du bus nommé USB SIE (Serial Interface Engine) permet de gérer la supervision du fonctionnement du BUS. Cette interface matérielle est chargée d'assurer la partie hard du bus tels que les adaptations de niveau ou encore le décodage et le stockage des informations en mémoire partagée. [15]

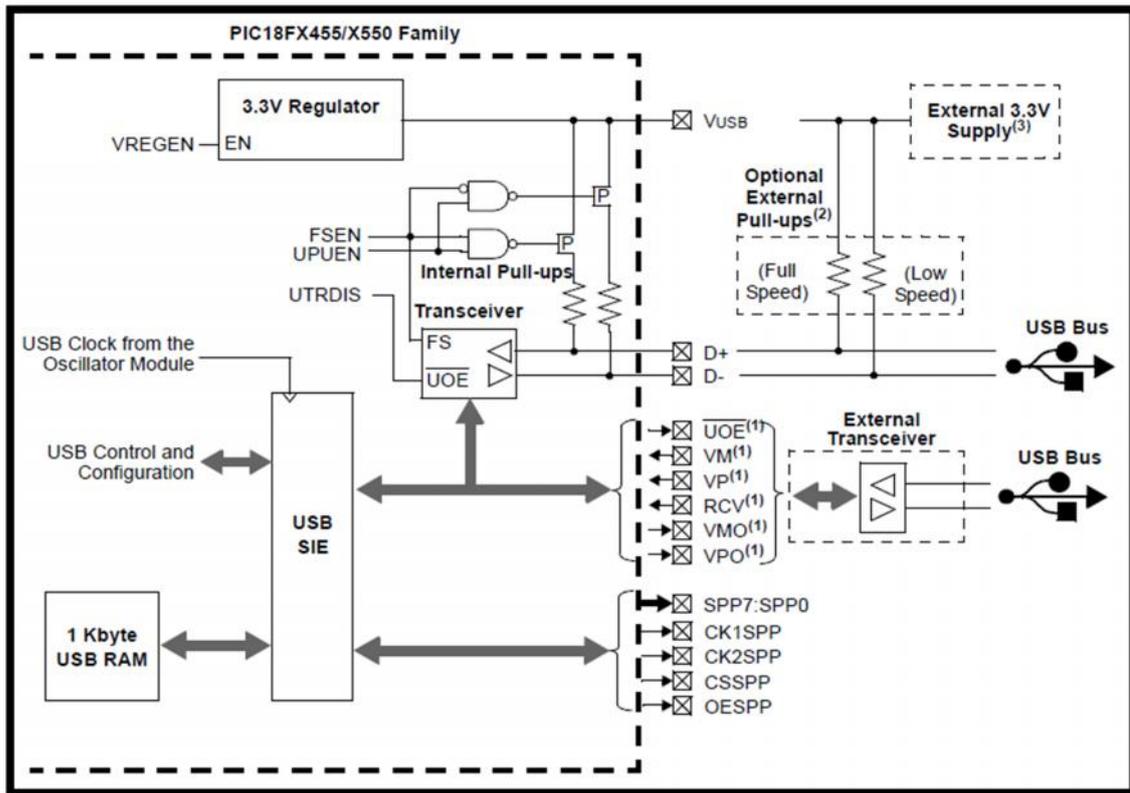


Fig III.10: Architecture interne USB du PIC 18F4550. [15]

III.4.6.c.Registres utilisable pour la fonction USB

Pour le Mode transfert en Interruption, les registres configurables sont les suivants: INTCON, IPR2, PIR2, UIR (Usb Interrupt status Register), UIE (Usb Interrupt Enable register), UEIR (Usb Error Interrupt status Register) et UEIE (Usb Error Interrupt Enable register).

Le registre UCON (Usb CONTROL register) contient les bits nécessaires au contrôle du comportement du module pendant une transaction. Le bit USBEN permet la mise en service ou non du module USB.

Le registre UCFG (Usb ConFiGuration register) permet entre autres de configurer la vitesse du bus en autorisant la mise en service en interne des résistances de pull-up. Un bit de ce registre permet également de valider ou non l'émetteur-récepteur interne.

Le registre US TAT (Usb STATus register) contient les informations sur la dernière transaction du SIE (end point concerné, direction de l'échange, etc.).

Le registre UADDR (Usb ADDRess register) contient l'adresse USB du PIC codée sur 7 bits (0 à 127), cette adresse est égale à « 0 » au Reset. Ce registre est configure pendant la phase d'énumération durant laquelle le host attribuera une adresse au périphérique.

Les registres UFRML et UFRMH (Usb FRaMe number register, L = low, H = high) sont en lecture seule, ils contiennent le numéro de la dernière trame du type SOP. Le maître émet comme nous l'avons déjà précisé une trame nommée SOF (Start Of Frame) toutes les 1 ms pour la version USB1 et toutes les 125 us pour la version USB2. Cette trame permet la synchronisation des échanges et comporte donc un champ Frame Number codé sur 11 bits qui sera logé dans les deux registres UFRM. Les 3 bits de poids fort du numéro de frame sont logés dans UFRMH.

Les registres UEPO à UEP15 (Usb End Point control) permettent la configuration des 16 terminaisons (end points) pour le mode USB2. [16]

III.5. Conclusion

Un PIC est un microcontrôleur ou un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires, unités périphériques et interfaces d'entrées-sorties.

Les éléments qui nous intéressent le plus sont le convertisseur CAN et le bus USB. En effet le signal analogique à visualiser doit être numérisé, puis transmis à travers le bus USB vers le PC.

La partie d'oscilloscope numérique est composée du PIC et de plusieurs blocs : alimentation, atténuateur, protection, calibrage, couplage. Cette partie contrôlée, configurée par une interface réalisée sous l'environnement C#.

Réalisation pratique

CHPITRE IV : Réalisation pratique

1 - Introduction	54
2 - Programmeur PIC	54
2.1 - Les signaux ICSP	54
2.2 - Compilateur "MikroC PRO" pour PIC	55
2.3 - Logiciel de programmation	56
3 - Partie Hard	57
3.1 - Circuit d'entrée	58
3.2 - Circuit de sélection de calibre	60
3.3 - Circuit d'acquisition et de communication	61
3.4 - Circuit d'alimentation	66
4 - Partie Soft	67
5 - Conclusion	69

IV.1.Introduction

L'oscilloscope numérique portatif qu'on va réaliser est un ensemble de circuits liés à PIC. Le PIC exécute les ordres du PC à travers le bus USB.

Dans ce chapitre nous allons expliquer toutes les étapes précédentes pour élaborer ce projet, commençant par le hard et passant au soft.

IV.2.Programmateur PIC

La programmation d'un PIC est l'introduction d'un programme dans sa mémoire Flash. Ce programme à introduire sera exécuté par le PIC lorsqu'il sera en fonctionnement.

La programmation se fait en deux étapes. La première étape consiste à écrire le programme en langage mikroC, puis on utilise le compilateur MikroC PRO pour obtenu le fichier en hexadécimale. C'est fichier que sera par la suite introduit dans la mémoire Flash du PIC. Pour introduire le fichier hexadécimal, on a utilisé l'adaptateur RS232/ICSP qui est contrôlé par le logiciel PICPgm.

Tous les éléments utilisés dans la programmation du PIC seront être détaillés

IV.2.1.Les signaux ICSP

Les PICs sont programmables en utilisant 5 signaux (Fig IV.1). Les données sont transférées à l'aide d'un mode synchrone à double fils, avec l'horloge contrôlé toujours par le programmeur. [17]

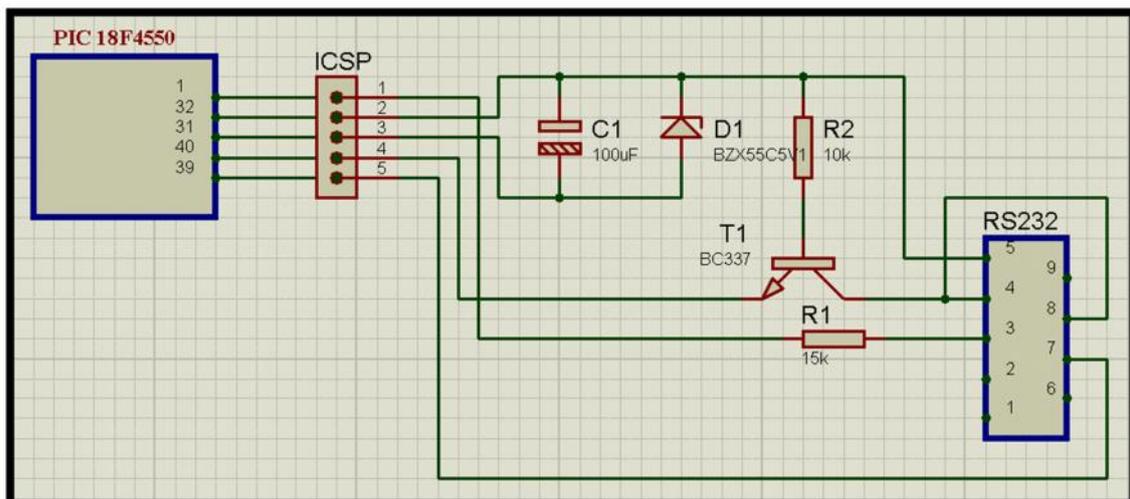


Fig IV.1: Adaptateur RS232/ICSP pour la programmation des PICs.

Les tableaux (Tab IV.1 et Tab IV.2) suivants présentent la connexion du PIC au connecteur ICSP (In-circuit Serial Programming) et la connexion des broches du RS232 :

ICSP		18F4550	
1	V _{PP}	1	$\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$
2	V _{DD}	32	V _{DD}
3	GND	31	V _{SS}
4	PGD	40	RB7/KBI3/PGD
5	PGC	39	RB6/KBI2/PGC

Tab IV.1: Connexion du PIC au connecteur ICSP. [17]

Pin	SIG	Signal Name	DTE (PC)
1	DCD	Data Carrier Detect	in
2	RXD	Receive Data	in
3	TXD	Transmit Data	out
4	DTR	Data Terminal Ready	out
5	GND	Signal Ground	-
6	DSR	Data Set Ready	In
7	RTS	Request to Send	out
8	CTS	Clear to Send	In
9	RI	Ring Indicator	in

Tab IV.2: RS-232 db9 female. [18]

Les signaux ICSP sont :

1. GND : tension de référence (0V) pour tous les signaux mais une négative alimentation pour le PIC.
2. V_{dd} : C'est l'alimentation positive du PIC.
3. V_{pp} : Tension en mode programmation. Elle doit être connectée à la broche MCLR. L'ordre de grandeur de cette tension diffère d'un microcontrôleur à un autre. Elle vaut 3.3V pour le PIC utilisée dans ce projet (18F4550).
4. PGC : Le transfert de donnée se fait au front descendant du signal PGC qui est la ligne d'horloge de l'interface série.
5. PGD : Cette ligne de donnée est bidirectionnelle et varié de GND à V_{dd}. Un bit est transféré à chaque front descendant de PGC.

IV.2.2. Compilateur "MikroC PRO" pour PIC

Le MikroC PRO est un compilateur pour PIC Conçu par la société Mikroelektronika. Le compilateur MikroC PRO (Fig IV.2) qui est une nouvelle génération MikroC pour microcontrôleurs PIC bénéficie d'une prise en main très intuitive et d'une ergonomie sans faille. Ces très nombreux outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code...) associé à sa capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, 1Wire, SPI,

RS485, Bus CAN, cartes compact Flash, signaux PWM (Pulse-Width Modulation), afficheurs LCD et 7 segments...) en font un outils de développement incontournable et puissant. Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète.

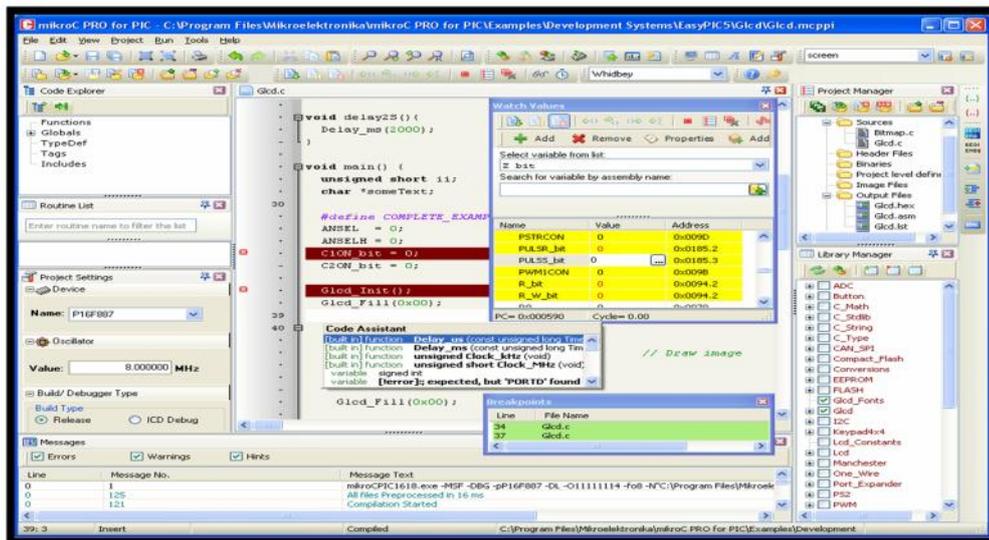


Fig IV.2: Fenêtre principale du Logiciel MikroC pro.

IV.2.3. Logiciel de programmation :

Il existe plusieurs logiciels de programmation des PIC : WinPic PICPgm, MikroC Prog ...etc.

On a choisi PICPgm (Fig IV.3) parce qu'il possède quelques avantages comme la détection automatique de la présence de programmeur et de PIC.



Fig IV.3: Fenêtre principale du Logiciel PICPgm.

IV.3.Partie Hard

Comme tout oscilloscope commercial, mon oscilloscope est composé des éléments suivants :

- Circuit d'entrée : il comporte un circuit de protection contre les sur tensions et un atténuateur pour l'abaissement de la tension d'entrée.
- Circuit de Couplage : il permet à l'utilisateur de choisir le mode du canal d'oscilloscope (AC, DC, GND).
- Circuit de sélection de calibre : il permet à l'utilisateur de choisir le calibre qui convient à la mesure.
- Circuit d'acquisition et de communication : c'est le cœur de l'oscilloscope. Il réalise les deux fonctions principale : la conversion analogique-numérique et le transfert de données.
- Circuit d'alimentation : c'est un convertisseur DC-DC, c'est-à-dire qu'il génère une alimentation symétrique ($\pm 5V$) à partir d'une simple alimentation (+5V).

Le schéma en bloc (Fig IV.4) suivant présente ces éléments.

Remarque : toutes les valeurs des composants se trouvent dans l'annexe J.

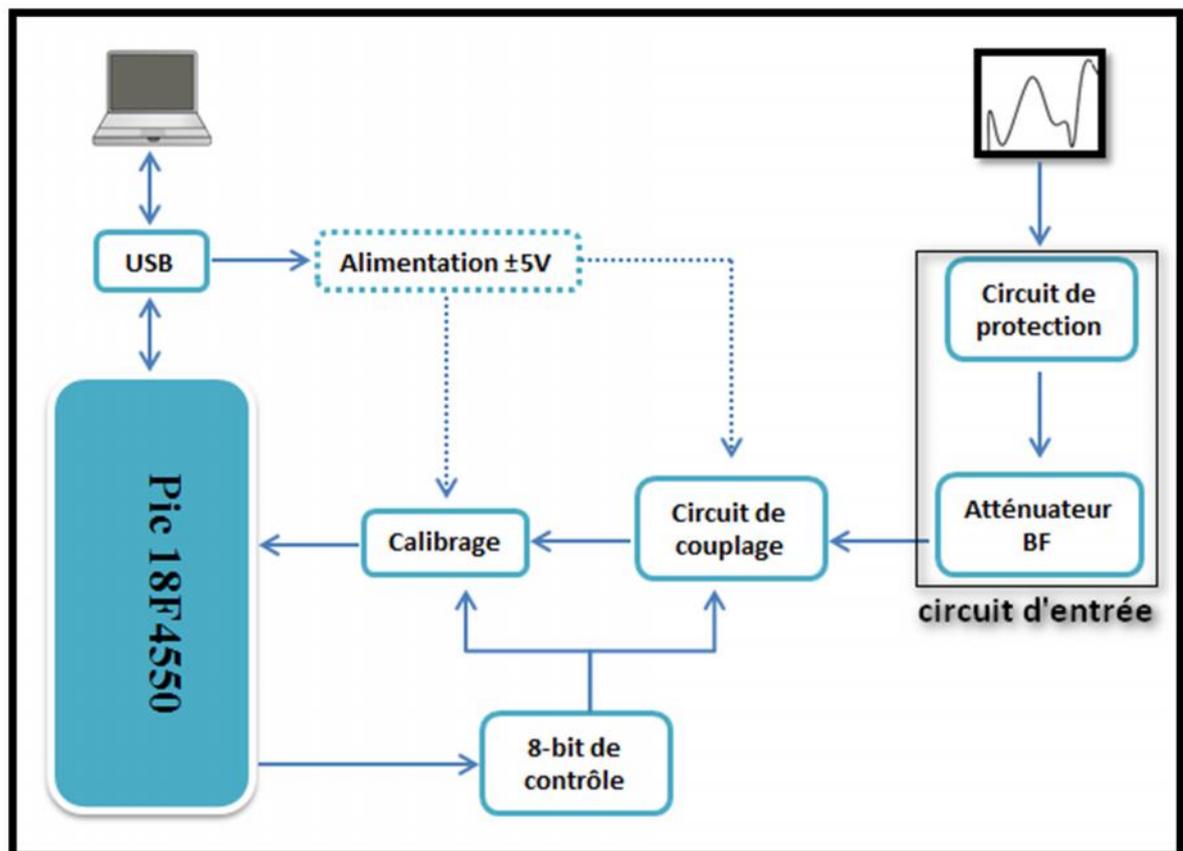


Fig IV.4 : Bloc diagramme de système.

IV.3.1.Circuit d'entrée

Le signal analogique à convertir est tout d'abord passé à travers un circuit de protection (Fig IV.5) contenant une résistance fusible R11 est une deuxième résistance R12 pour limiter le courant dans les diodes. Le rôle des diodes Zener et diode normal est de limiter la tension de fonctionnement de l'oscilloscope à $\pm 25V$. La tension est par la suite et abaissé à l'aide d'un circuit atténuateur (Fig IV.6).

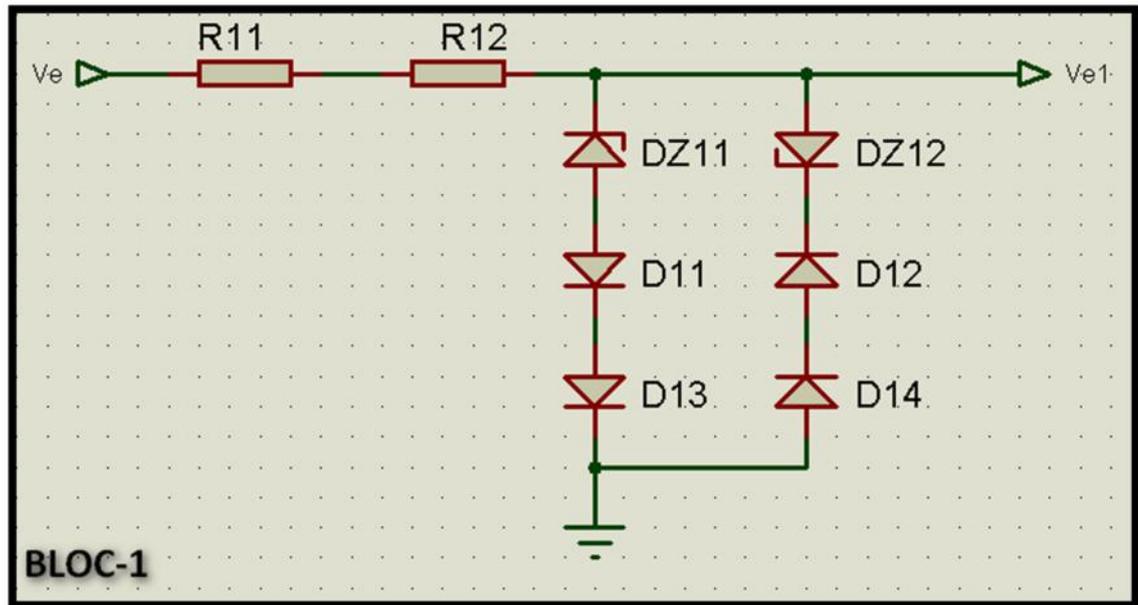


Fig IV.5 : Circuit de protection.

La puissance maximale dissipée dans chaque composant (résistance fusible R11, résistance R12, diode Zener, diode) devra être inférieure à la puissance permise et ceci au fonctionnement normal ($|V_e| < 25V$).

IV.3.1.a.La résistance fusible

Calculé par l'équation suivante :

$$P_{normal} < P_{max} = R_{11} \left(\frac{V_e - V_z - 2V_d}{R_{11} + R_{12}} \right)^2$$

IV.3.1.b.L'atténuateur BF

C'est un diviseur de tension simple, mais il détermine la résistance d'entrée de l'oscilloscope :

$$R_e = R_{21} + R_{22} = 100K\Omega + 900K\Omega = 1M\Omega$$

La tension de sortie de l'atténuateur vaut :

$$V_s = \frac{R_{22}}{R_{21} + R_{22}} V_e = \frac{V_e}{10}$$

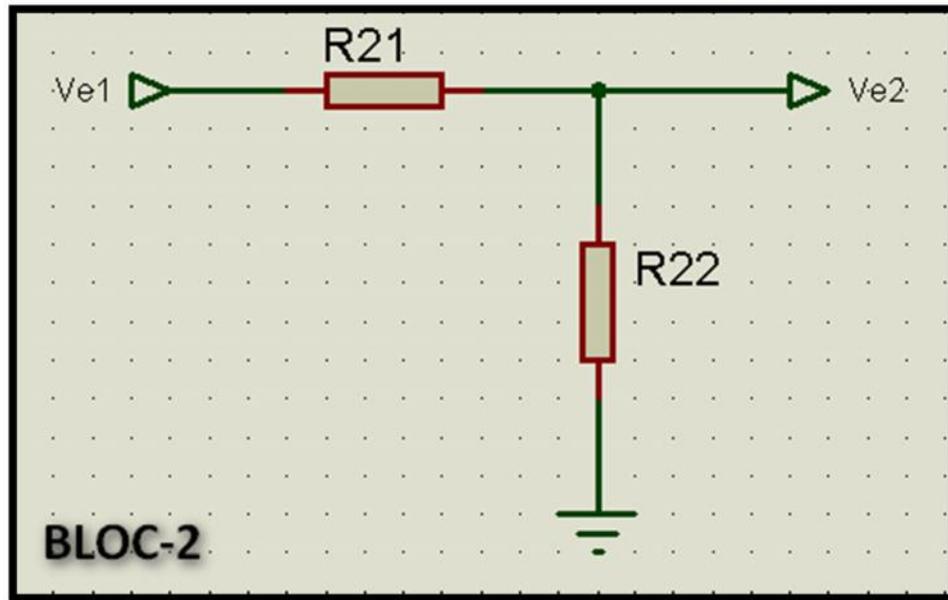


Fig IV.6: Atténuateur BF.

IV.3.1.c. Le circuit de couplage

Il permet la sélection des modes AC/DC/GND, le circuit est présenté dans la figure IV.7 et la table de fonctionnement (Tab IV.3) donne les états possibles des entrées d’adressage du multiplexeur utilisé.

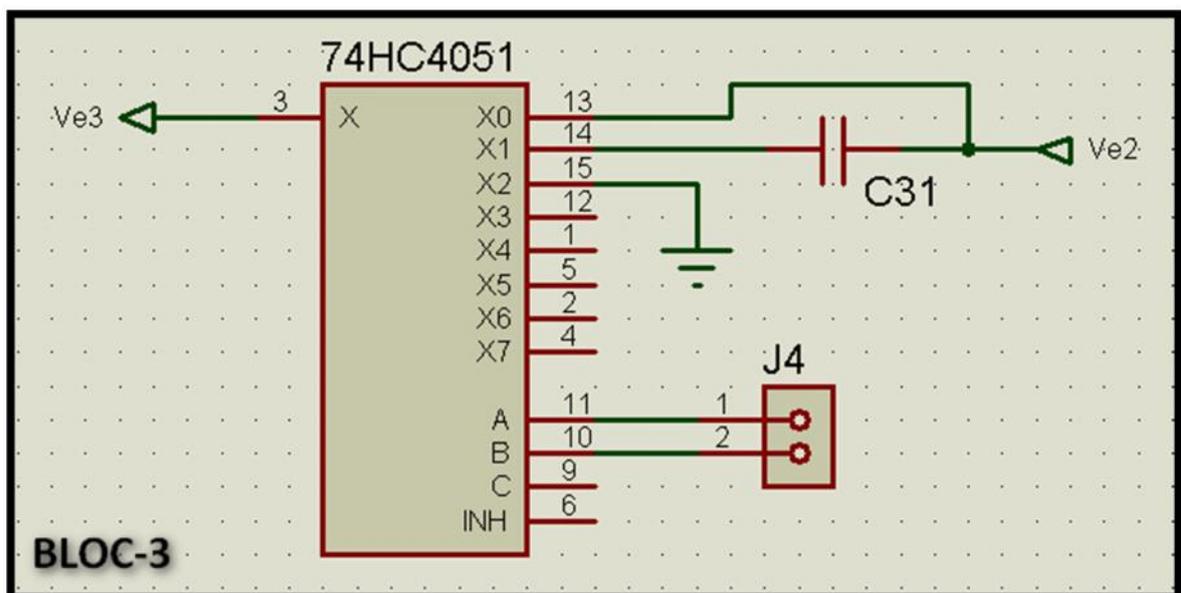


Fig IV.7 : Circuit de couplage.

Le tableau IV.3 présente le fonctionnement du circuit de couplage.

J4	mode
00	DC
01	AC
10	GND

Tab IV.3: Les états de fonctionnement du circuit de couplage.

IV.3.2.Circuit de sélection de calibre

Pour ne pas affecter la résistance d'entrée de l'oscilloscope on utilise un amplificateur opérationnel à haute impédance d'entrée monté en suiveur AOP41 (Fig IV.8).

La plage de conversion du convertisseur du PIC 18F4550 est limitée dans l'intervalle 0V à 5V. Pour cela une tension continue de 2.5V, généré par l'AOP42 monté en suiveur, est ajoutée au signal bipolaire de sortie de l'AOP41.

Le calibre est choisi selon la résistance sélectionnée par le multiplexeur analogique dont la valeur détermine le gain de l'amplificateur AOP43. La table de fonctionnement (Tab IV.4) donne les états possibles des entrées d'adressage du multiplexeur utilisé. [19]

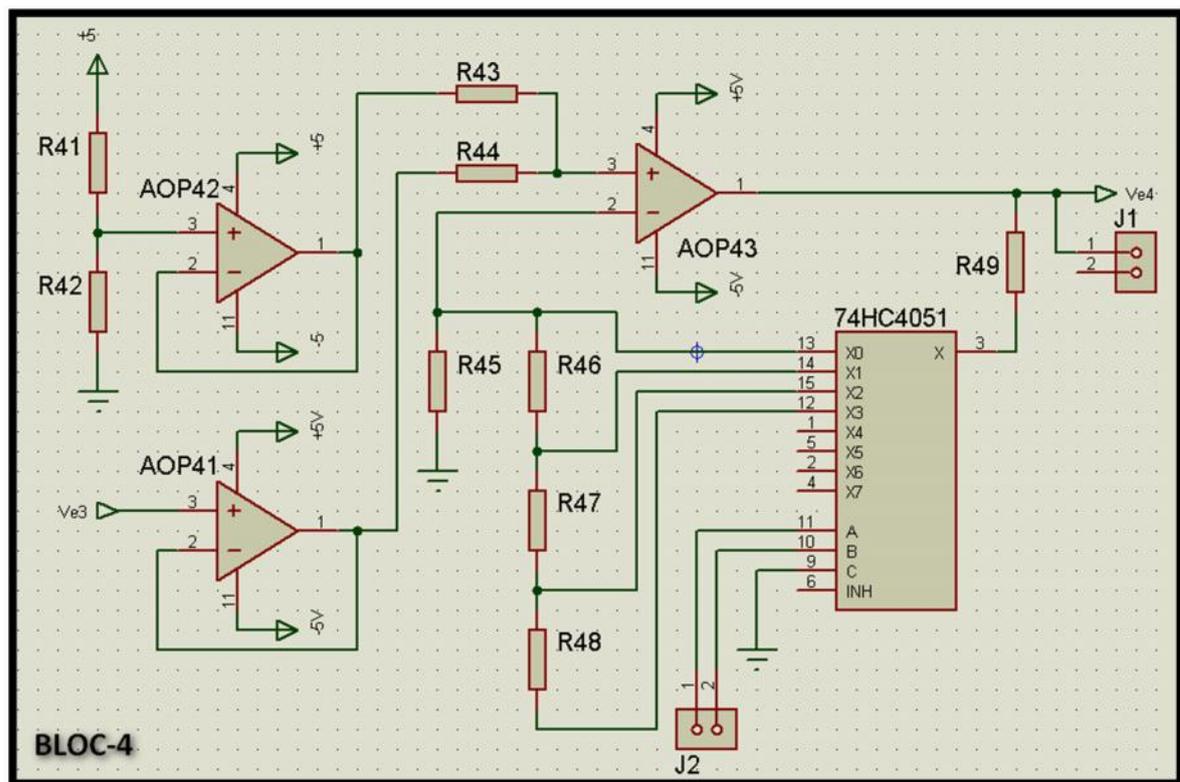


Fig IV.8: Circuit de sélection de calibre.

Le gain :

$$V_{e4} = \left(\frac{R_{45}}{R_{45}} + \frac{R_x}{R_{45}} + \frac{R_{49}}{R_{45}} \right) V_{e3}$$

$$V_{e4} = \left(1 + \frac{R_x}{R_{45}} + 1\right) V_{e3} = \left(2 + \frac{R_x}{R_{45}}\right) V_{e3}$$

Le tableau IV.4 présente le fonctionnement du calibre.

J2	$\frac{R_x}{R_{45}}$	V_{e4}
00	0	2. $\frac{R_x}{R_{45}}$
01	$\frac{R_{46}}{R_{45}}$	10. $\frac{R_x}{R_{45}}$
10	$\frac{R_{46}}{R_{45} + R_{47}}$	50. $\frac{R_x}{R_{45}}$
11	$\frac{R_{46} + R_{47}}{R_{45} + R_{47} + R_{48}}$	250. $\frac{R_x}{R_{45}}$

Tab IV.4: Les états de fonctionnement du calibre.

IV.3.3.Circuit d'acquisition et de communication

Le circuit est connecté au bus USB par un câble USB, dès le branchement au bus USB la LED D51 s'allume pour indiquer que le circuit est alimenté. Lorsque le PC entre en communication avec le PIC, la LED D52 s'allume. Le circuit comporte plusieurs connecteurs : J1 pour les entrées analogiques, J2 et J3 pour le circuit de calibrage, j4 et J5 pour le circuit de couplage, J6 pour le circuit d'alimentation.

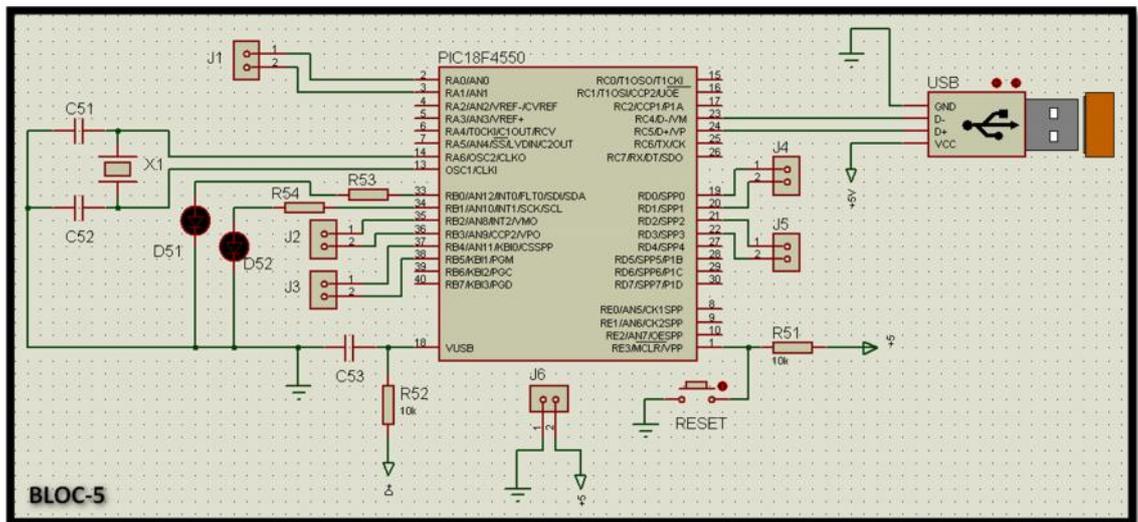


Fig IV.9: Circuit d'acquisition et de communication.

IV.3.3.a.L'organigramme principal du PIC

La figure IV.10 présente l'organigramme principale du PIC. Comme tout programme d'un microcontrôleur on commence par l'initialisation des différents modules constituant le PIC. Après avoir activé le bus USB, ce dernier est configuré selon les besoin de l'utilisateur (trigger, calibre,...etc.). Si le trigger est activé le PIC détecte le point de déclenchement convenable. Le nombre d'échantillons converti à chaque fois, par le convertisseur interne du PIC, est de 32 échantillons.

Ce paquet d'échantillons est transmis directement au bus USB et l'opération se répète 64 fois pour chaque canal. Afin de minimiser la consommation de l'énergie, le PIC reçoit l'ordre « OFF » du PC et met un Reset continu (Sleep Mode).

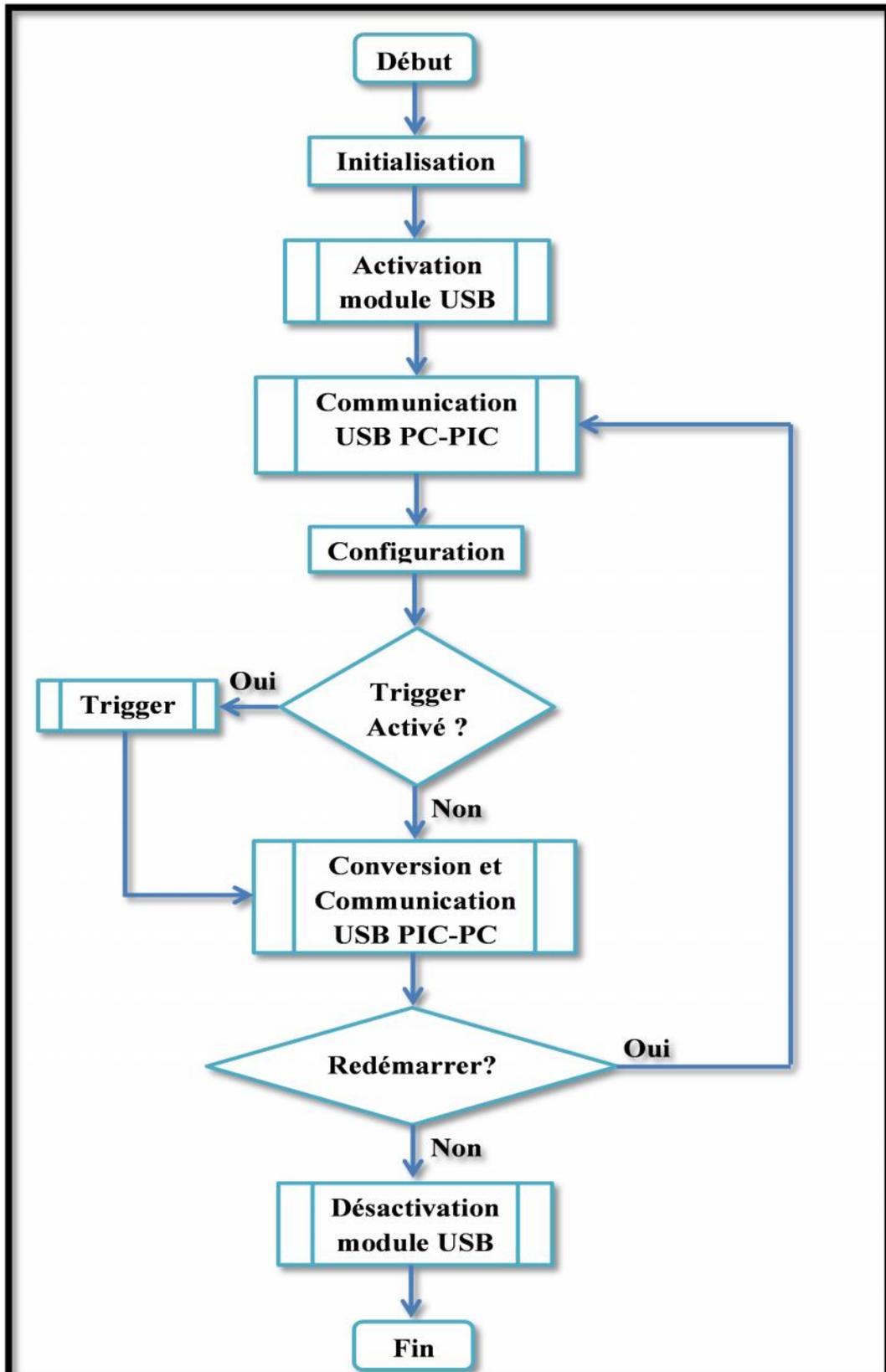


Fig IV.10: L'organigramme principal du PIC.

IV.3.3.b.L'activation de module USB

La figure IV.11 présente l'organigramme d'activation de Module USB du PIC. Le programme d'activation du module USB nécessite d'abord une confirmation de la connexion avec le PC, cette confirmation se fait à travers le drapeau USBIF. Après la confirmation se termine la lecture de la RAM USB puis la transmission du descripteur USB. Le code de ce dernier est donné dans l'annexe I. finalement, le périphérique sera confirmé et identifié par l'ordinateur.

Le drapeau USBIF (USBIF) indique la présence d'un évènement lié au bus USB. La figure IV.12 représente les différents drapeaux d'interruption USB qui génèrent le drapeau final USBIF. Pour plus de détail sur les interruptions USB voir le Datasheet du PIC18F4550. [15]

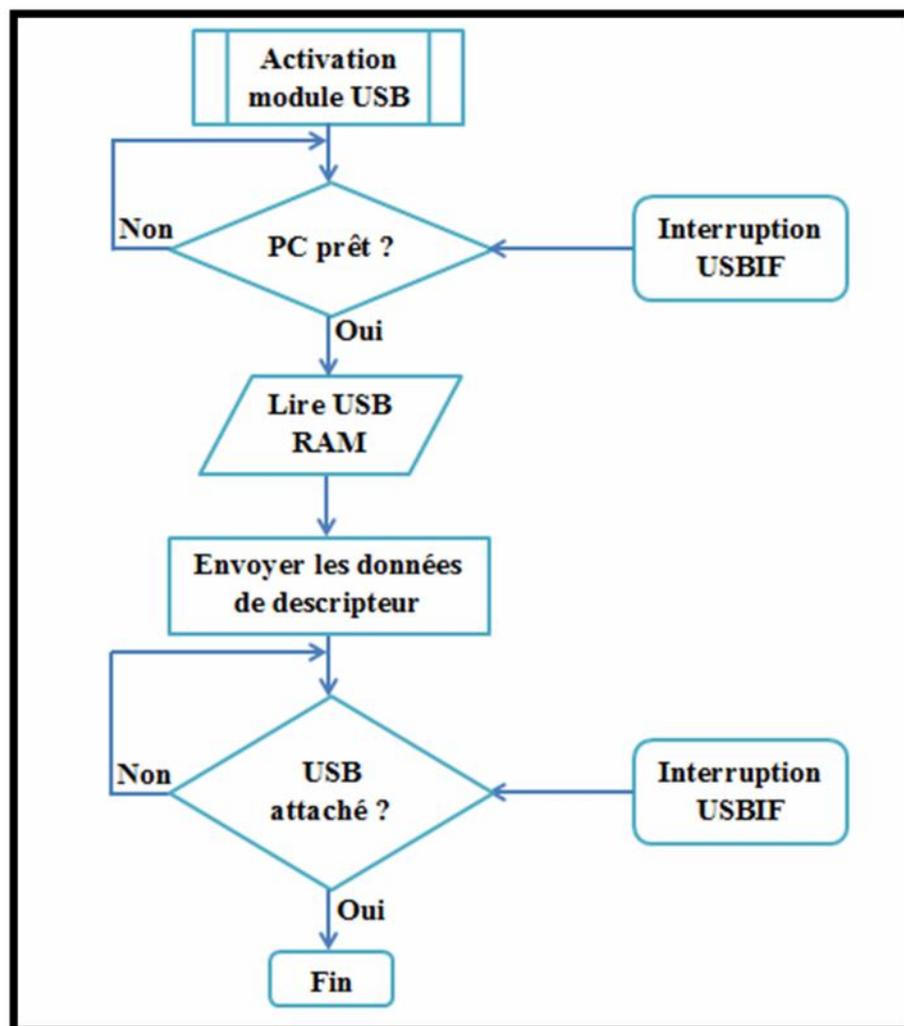


Fig IV.11: L'organigramme d'activation de Module USB du PIC.

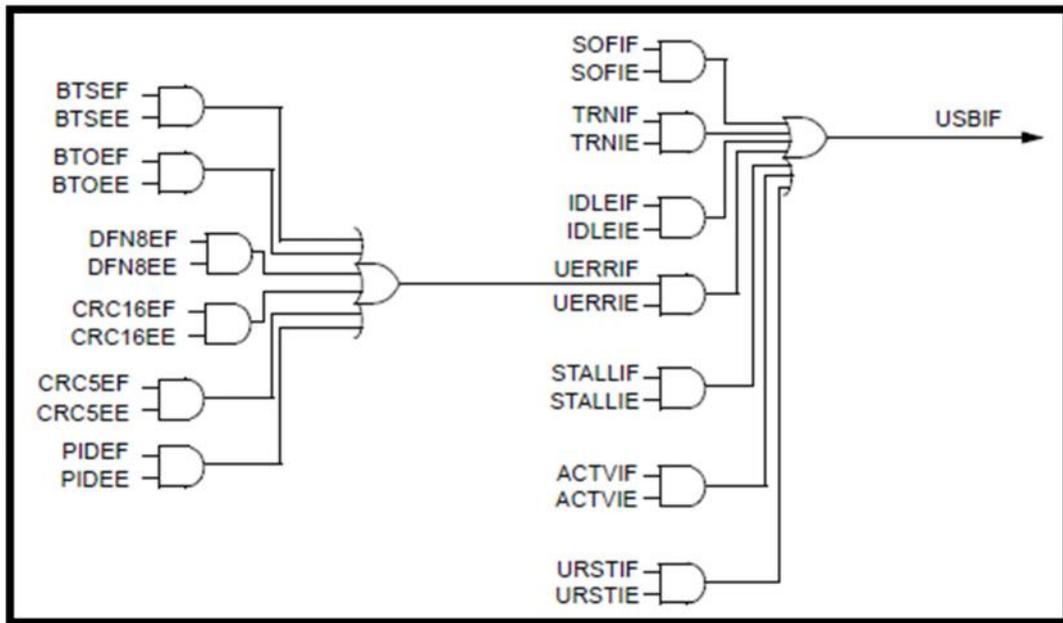


Fig IV.12: Le drapeau USBIF. [15]

IV.3.3.c.L'organigramme de la communication USB PC-PIC

La figure IV.13 présente l'organigramme de la communication USB entre le PIC et le PC. N'importe quelle opération dans le bus USB nous la découvrons par le drapeau USBIF, les données transmises du pc seront stockées dans la RAM USB.

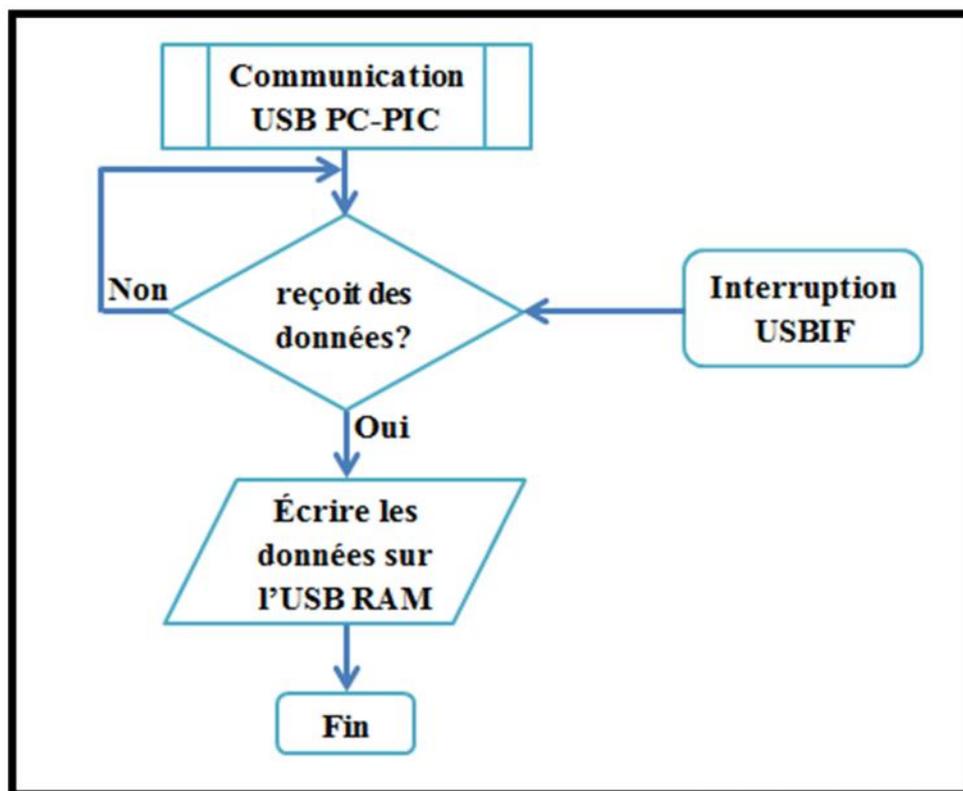


Fig IV.13: L'organigramme de la communication USB entre le PIC et le PC.

IV.3.3.d.L'organigramme du trigger

La figure IV.14 présente l'organigramme du trigger. à travers les configurations réglés par l'utilisateur, nous connaissons le niveau et le front déclenchement. tigger teste le niveau du signal quand il arrive au niveau visé, le rôle du tigger se termine.

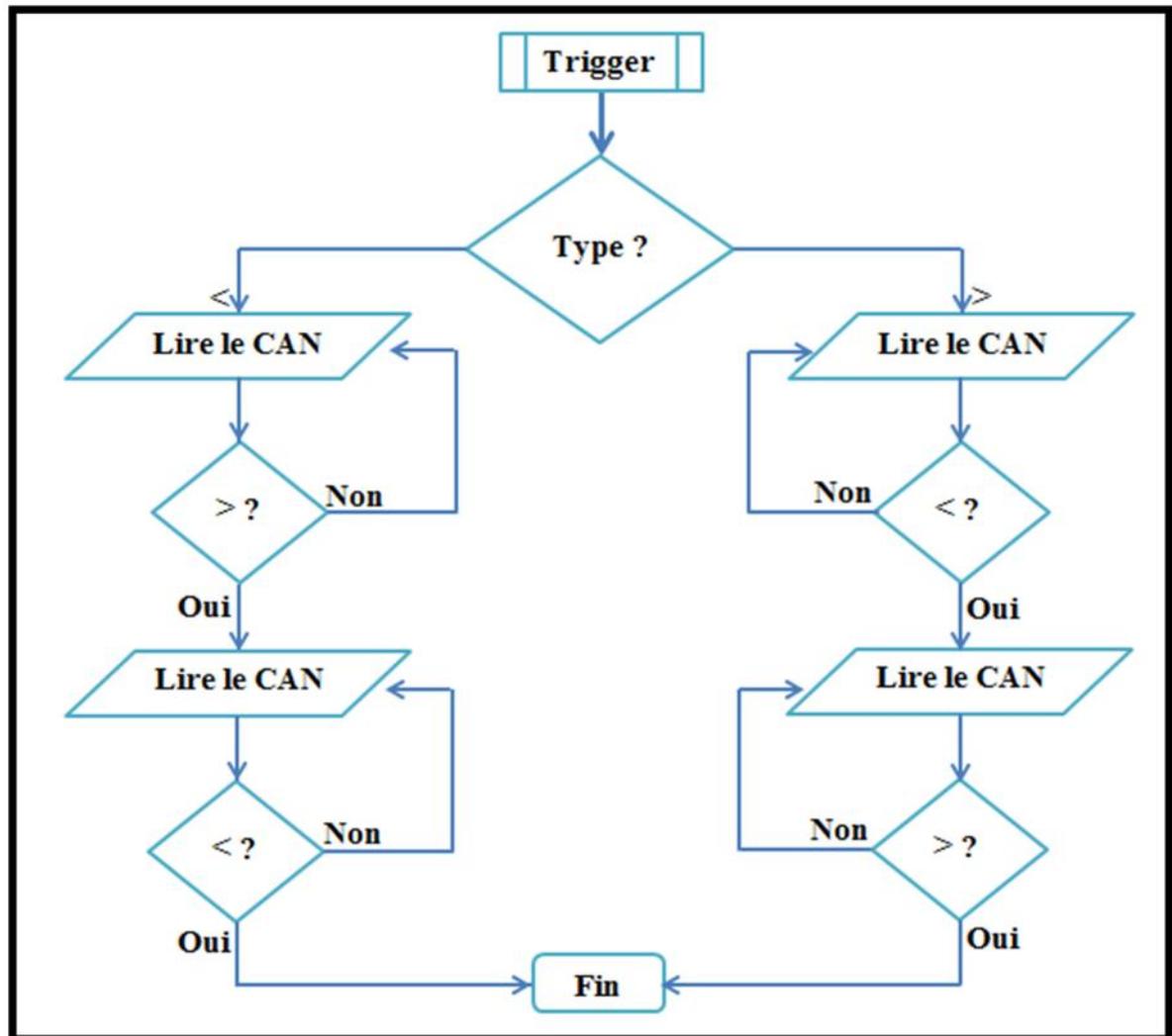


Fig IV.14: L'organigramme du trigger.

IV.3.3.e.L'organigramme du conversion et communication

La figure IV.15 présente l'organigramme du conversion et communication. Le nombre d'échantillons converti à chaque fois, par le convertisseur interne du PIC, est de 32 échantillons. Ce paquet d'échantillons est transmis directement au bus USB et l'opération se répète 64 fois pour chaque canal.

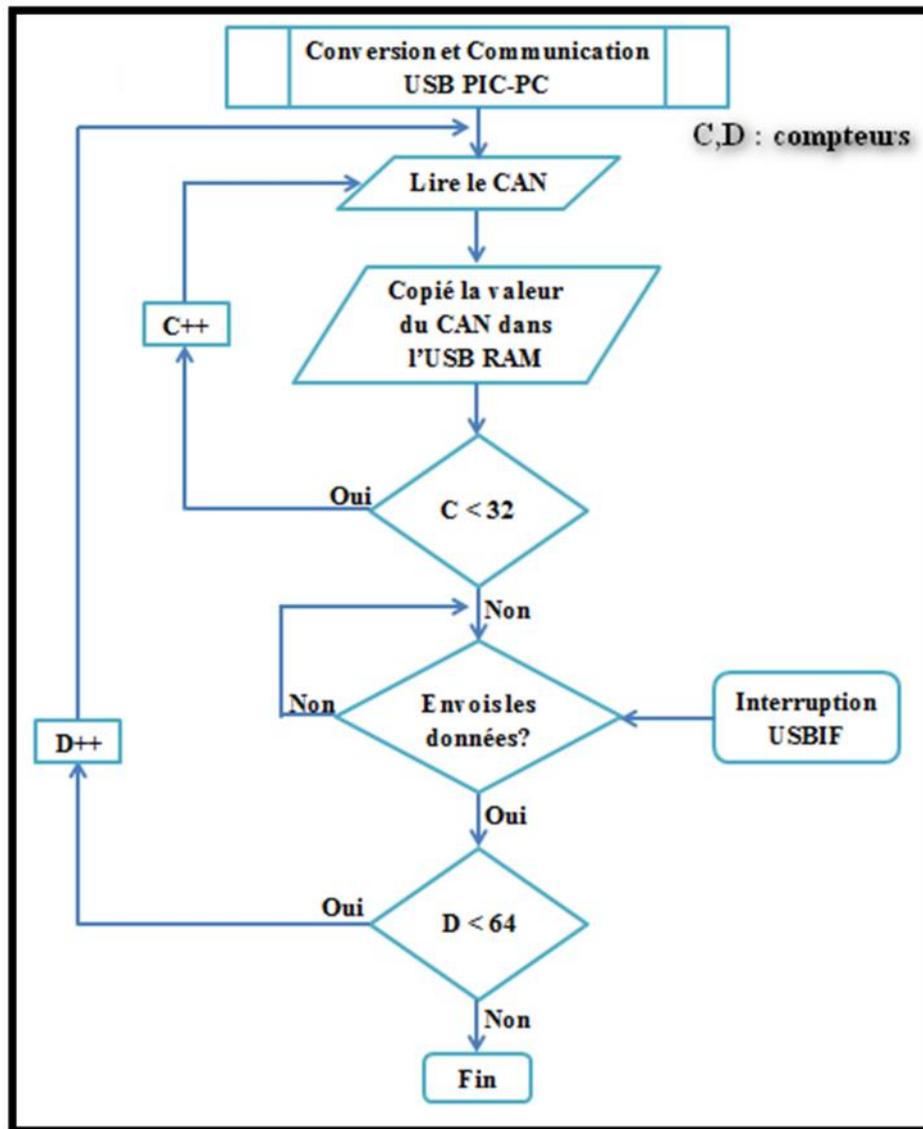


Fig IV.15: L'organigramme du conversion et communication.

IV.3.4.Circuit d'alimentation

Ce circuit comporte un multivibrateur astable NE555, deux circuits clamber et deux redresseurs simple alternance avec deux capacité de filtrage, une pour la tension positive et l'autre pour la tension négative.

Le NE555 comporte un réseau de trois résistances de précision $R = 5k$ montées en diviseur de tension, deux comparateurs, une bascule RS, un amplificateur de sortie et un transistor à collecteur ouvert. La période de l'astable ne dépend pas de V_{cc} . Elle est très stable si le coefficient de température de C est très petit. La fréquence maximale est de l'ordre de 300 kHz. [20]

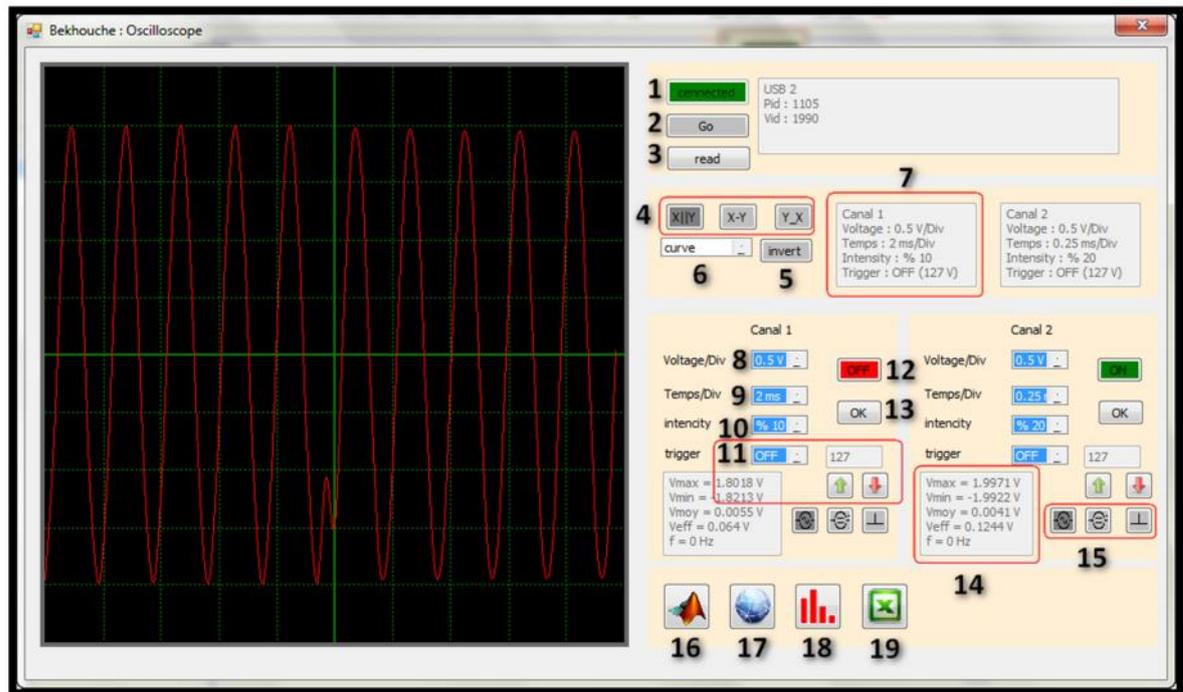


Fig IV.17: L'interface utilisateur.

1. Bouton et indicateur de connexion au bus USB.
2. Lecture et affichage continu.
3. Lecture et affichage une seule fois.
4. Mode d'affichage : $y(t)$, $x(y)$, $y(x)$
5. Inversement de signal.
6. Style d'affichage : ligne, courbe.
7. Indicateur de configuration actuel du canal.
8. Sélection de calibre.
9. Sélection de base de temps.
10. Réglage de l'intensité.
11. Trigger : activation, niveau, front.
12. Mettre le signal visible ou invisible.
13. Validation de la configuration choisie.
14. Information sur le signal : V_{max} , V_{min} , V_{eff} , V_{moy} .
15. Sélection de mode de couplage : AC, DC, GND.
16. Création d'un fichier de données MATLAB.
17. Connexion de l'oscilloscope à une interface de commande et d'affichage sur site web.
18. Analyseur de spectre.
19. Création d'un fichier de données Excel.

IV.5.Conclusion

Dans ce chapitre, nous avons mis en évidence le rôle de tous les blocs d'oscilloscope que nous avons réalisé, comme nous avons expliqué aussi le rôle de l'interface utilisateur sous l'environnement C#.

Conclusion Générale

Pour réaliser l'oscilloscope numérique on a eu besoin de revoir les notions théoriques sur le fonctionnement de l'oscilloscope analogique pour en tenir compte dans sa conception numérique.

La vitesse de transmission des données entre le PC et le PIC, utilisée est de 15 Mbps. Cette dernière ne peut être atteinte qu'avec le bus USB et plus précisément la norme USB Full Speed. Ainsi l'idée d'utiliser le port série est à rejeter à cause des valeurs de la vitesse qui sont permises.

- Les caractéristiques de la carte d'acquisition et de communication sont dictées par le choix du PIC en particulier :
- Son convertisseur analogique/numérique CAN utilisant 10 bits,
- Le nombre de broches dans le port parallèle et pouvant être programmées comme entrée ou sortie,
- Son bus USB, selon la norme USB 2, avec la vitesse Full Speed ou Low Speed,
- Son circuit d'horloge, qui à partir de 20 MHz, synthétise plusieurs fréquences parmi lesquelles on peut choisir une fréquence de travail.

Tenant compte des performances de l'oscilloscope numérique à réaliser, le choix a été fait sur le PIC 18F4550. Ainsi l'oscilloscope réalisé a une gamme de la tension à l'entrée de $\pm 25V$, une résistance d'entrée de $1M$, une fréquence d'échantillonnage de 57.15KHz, un trigger pour déclencher le signal à visualiser.

Le panneau de l'oscilloscope avec ses différentes commandes est représenté dans l'interface affiché à l'écran de PC.

Pour améliorer les performances de la réalisation, on propose :

- L'utilisation des convertisseurs externes mais plus rapide que ceux associés aux PICs,
- La réalisation d'une carte d'acquisition avec un module USB décrit en VHDL et implémenté dans un circuit FPGA,
- L'utilisation de la norme USB 3 Super Speed qui a une vitesse de 4.8 Gbps,
- Etendre la gamme de la tension à une valeur plus importante,
- Augmenter la résistance d'entrée, ce qui nécessite une impédance d'entrée plus importante pour les Amplificateurs opérationnels à utiliser.

Conclusion Générale

Avec ces propositions, on pense présenter un produit compétitif sur le marché des oscilloscopes numériques.

Bibliographies

- [1]: <http://fr.wikipedia.org/wiki/Oscilloscope>, «Oscilloscope», Mai 2013.
- [2]: Patrick LESNE, «Mesures à l'aide de l'oscilloscope», 1998.
- [3]: Ian Hickman, «Digital Storage Oscilloscopes», 1997.
- [4]: <http://www.voltcraft.fr/oscilloscope.html>, «OSCILLOSCOPES VOLT CRAFT», Mai 2013.
- [5]: <http://ylescop.free.fr/mrim/cours/bus-USB.pdf>, «Bus USB - cours Yves LESCOP», Mai 2013.
- [6]: <http://www.usb.org/about/faq/>, «USB Info: Frequently Asked Questions», Mai 2013.
- [7]: Jan Axelson, «USB Complete The Developer's Guide 4th edition», 2009.
- [8]: <http://www.usb.org/> «USB.org», Mai 2013.
- [9]: <http://www.beyondlogic.org/usbnutshell/usb1.shtml>, «USB in a NutShell», Mai 2013.
- [10]: <http://reseaumaroc.com/files/Le%2520bus%2520USB.pdf>, «Le bus USB», Mai 2013.
- [11]: <http://www.commentcamarche.net/contents/773-usb>, «USB», Mai 2013.
- [12]: www.microtel.fr.eu.org/Clubs/CAEN.MTL/Mac/Tout_sur_USB2.pdf, «Tout sur l'USB», Mai 2013.
- [13]: <http://www.technologiepro.com/microcontroleur/>, «microcontroleur», Mai 2013.
- [14]: <http://oumnad.123.fr/Microcontroleurs/PIC16F84.pdf>, «Les microcontrôleurs PIC de microchip», Mai 2013.
- [15]: <http://microchip.com/>, «PIC18F4550 Data Sheet». Mai 2013.
- [16]: Pascal Mayeux, « Apprendre la programmation des PIC High-Performance par l'expérimentation et la simulation », Dunod, 2010.
- [17]: <http://embedinc.com/picprg/icsp.htm>, «In-circuit Serial Programming (ICSP) », Mai 2013.
- [18]: <http://www.lammertbies.nl/comm/cable/RS-232.html>, «RS232 serial cables pinout», Mai 2013.
- [19]: <http://crteknologies.fr/electronique/cours/ampliop.php>, «Amplificateur opérationnel», Mai 2013.
- [20]: <http://fr.wikipedia.org/wiki/NE555>, «NE555», avril Mai 2013.
- [21]: [http://en.wikipedia.org/wiki/Clamper_\(electronics\)](http://en.wikipedia.org/wiki/Clamper_(electronics)), «Clamper», Mai 2013.
- [22]: <http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/electro/redfilt1.html>, «Redressement et filtrage», Mai 2013.

Annexe

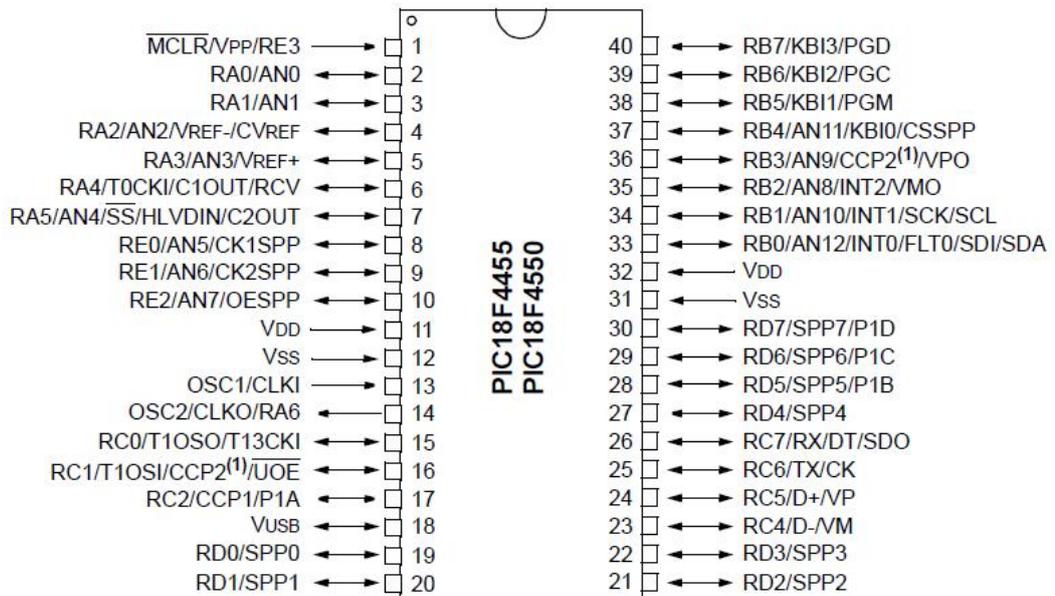
A: Datasheet PIC 18F4550



PIC18F2455/2550/4455/4550 Data Sheet

28/40/44-Pin, High-Performance,
Enhanced Flash, USB Microcontrollers
with nanoWatt Technology

40-Pin PDIP



Annexe

B: Datasheet LM324

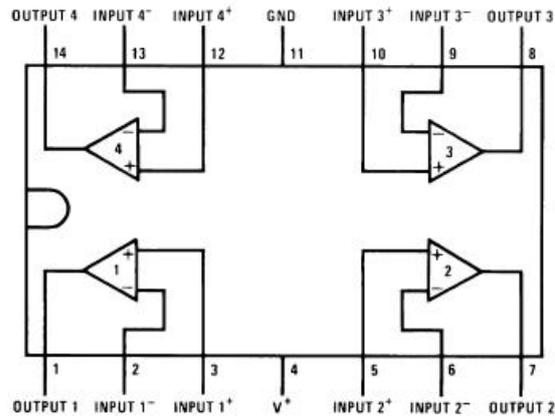
FEATURES

- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain 100 dB
- Wide Bandwidth (Unity Gain) 1 MHz (Temperature Compensated)
- Wide Power Supply Range:
 - Single Supply 3V to 32V
 - or Dual Supplies $\pm 1.5V$ to $\pm 16V$
- Very Low Supply Current Drain (700 μA)—Essentially Independent of Supply Voltage
- Low Input Biasing Current 45 nA (Temperature Compensated)
- Low Input Offset Voltage 2 mV
 - and Offset Current: 5 nA
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Large Output Voltage Swing 0V to $V^+ - 1.5V$

ADVANTAGES

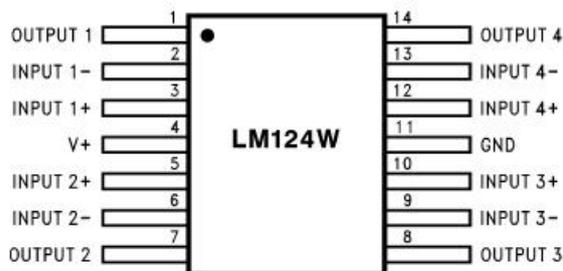
- Eliminates Need for Dual Supplies
- Four Internally Compensated Op Amps in a Single Package
- Allows Directly Sensing Near GND and V_{OUT} also Goes to GND
- Compatible with All Forms of Logic
- Power Drain Suitable for Battery Operation

Connection Diagrams



- Note 1:** LM124A available per JM38510/11006
Note 2: LM124-N available per JM38510/11005
Note 3: See STD Mil DWG 5962R99504 for Radiation Tolerant Device

Figure 1. Dual-In-Line Package - Top View
 See Package Number J0014A D0014A or NFF0014A



- Note 3:** See STD Mil DWG 5962R99504 for Radiation Tolerant Device

Annexe

C: Datasheet 74HC4051

1. General description

The 74HC4051; 74HCT4051 is a high-speed Si-gate CMOS device and is pin compatible with Low-power Schottky TTL (LSTTL). The device is specified in compliance with JEDEC standard no. 7A.

The 74HC4051; 74HCT4051 is an 8-channel analog multiplexer/demultiplexer with three digital select inputs (S0 to S2), an active-LOW enable input (\bar{E}), eight independent inputs/outputs (Y0 to Y7) and a common input/output (Z). With \bar{E} LOW, one of the eight switches is selected (low impedance ON-state) by S0 to S2. With \bar{E} HIGH, all switches are in the high-impedance OFF-state, independent of S0 to S2.

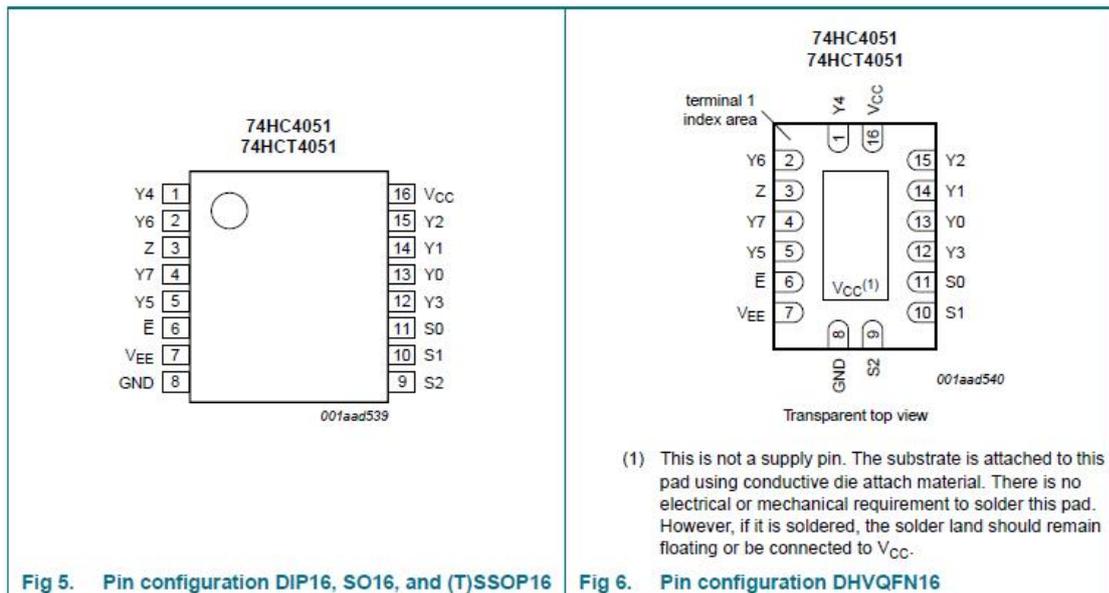
V_{CC} and GND are the supply voltage pins for the digital control inputs (S0 to S2, and \bar{E}). The V_{CC} to GND ranges are 2.0 V to 10.0 V for 74HC4051 and 4.5 V to 5.5 V for 74HCT4051. The analog inputs/outputs (Y0 to Y7, and Z) can swing between V_{CC} as a positive limit and V_{EE} as a negative limit. $V_{CC} - V_{EE}$ may not exceed 10.0 V.

For operation as a digital multiplexer/demultiplexer, V_{EE} is connected to GND (typically ground).

2. Features and benefits

- Wide analog input voltage range from -5 V to $+5$ V
- Low ON resistance:
 - ◆ 80Ω (typical) at $V_{CC} - V_{EE} = 4.5$ V
 - ◆ 70Ω (typical) at $V_{CC} - V_{EE} = 6.0$ V
 - ◆ 60Ω (typical) at $V_{CC} - V_{EE} = 9.0$ V
- Logic level translation: to enable 5 V logic to communicate with ± 5 V analog signals
- Typical 'break before make' built-in
- ESD protection:
 - ◆ HBM JESD22-A114F exceeds 2000 V
 - ◆ MM JESD22-A115-A exceeds 200 V
 - ◆ CDM JESD22-C101E exceeds 1000 V
- Multiple package options
- Specified from -40 °C to $+85$ °C and -40 °C to $+125$ °C

6.1 Pinning



Annexe

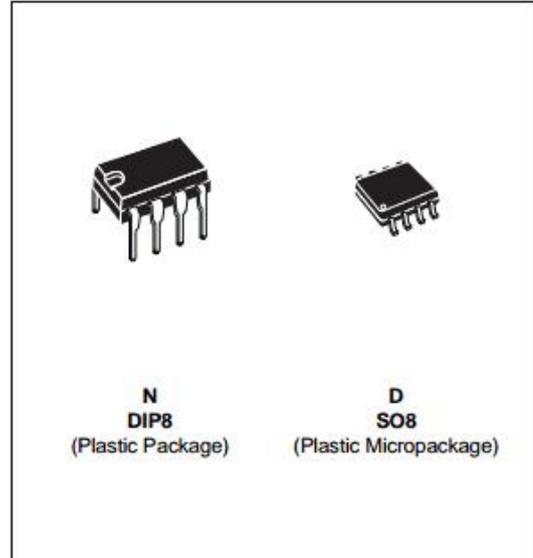
D: Datasheet NE555

GENERAL PURPOSE SINGLE BIPOLAR TIMERS

- LOW TURN OFF TIME
- MAXIMUM OPERATING FREQUENCY GREATER THAN 500kHz
- TIMING FROM MICROSECONDS TO HOURS
- OPERATES IN BOTH ASTABLE AND MONOSTABLE MODES
- HIGH OUTPUT CURRENT CAN SOURCE OR SINK 200mA
- ADJUSTABLE DUTY CYCLE
- TTL COMPATIBLE
- TEMPERATURE STABILITY OF 0.005% PER°C

DESCRIPTION

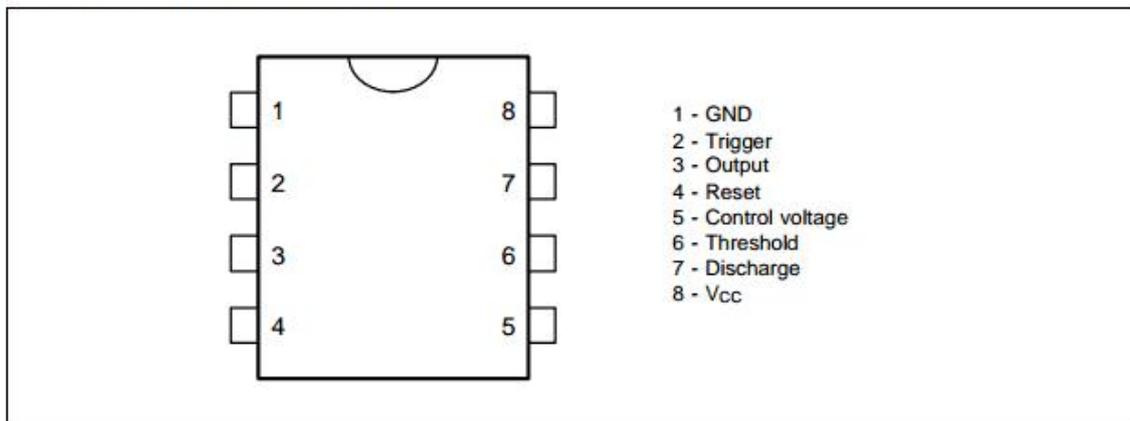
The NE555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays or oscillation. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200mA. The NE555 is available in plastic and ceramic minidip package and in a 8-lead micropackage and in metal can package version.



ORDER CODES

Part Number	Temperature Range	Package	
		N	D
NE555	0°C, 70°C	•	•
SA555	-40°C, 105°C	•	•
SE555	-55°C, 125°C	•	•

PIN CONNECTIONS (top view)



Annexe

E: Proteus ISIS

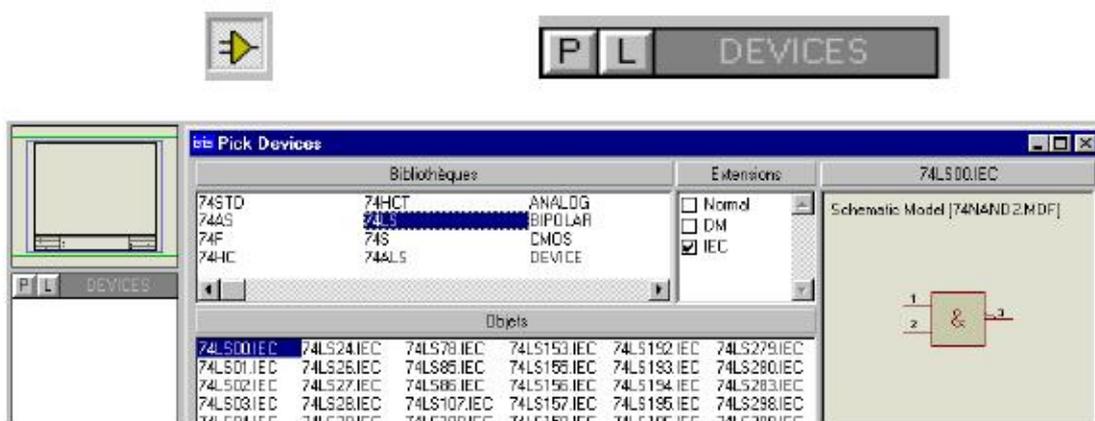
SAISIE DE SCHEMA SOUS ISIS :

Pour la saisie de schéma, on utilise le logiciel ISIS.

- L'écran se compose de plusieurs parties:
- la zone de travail, dans laquelle sera disposé le schéma,
- la zone de contrôle qui permet de se déplacer d'un endroit à un autre de la zone de travail,
- le clavier,
- la liste des objets sélectionnés pour réaliser le schéma.

RECHERCHE DES COMPOSANTS:

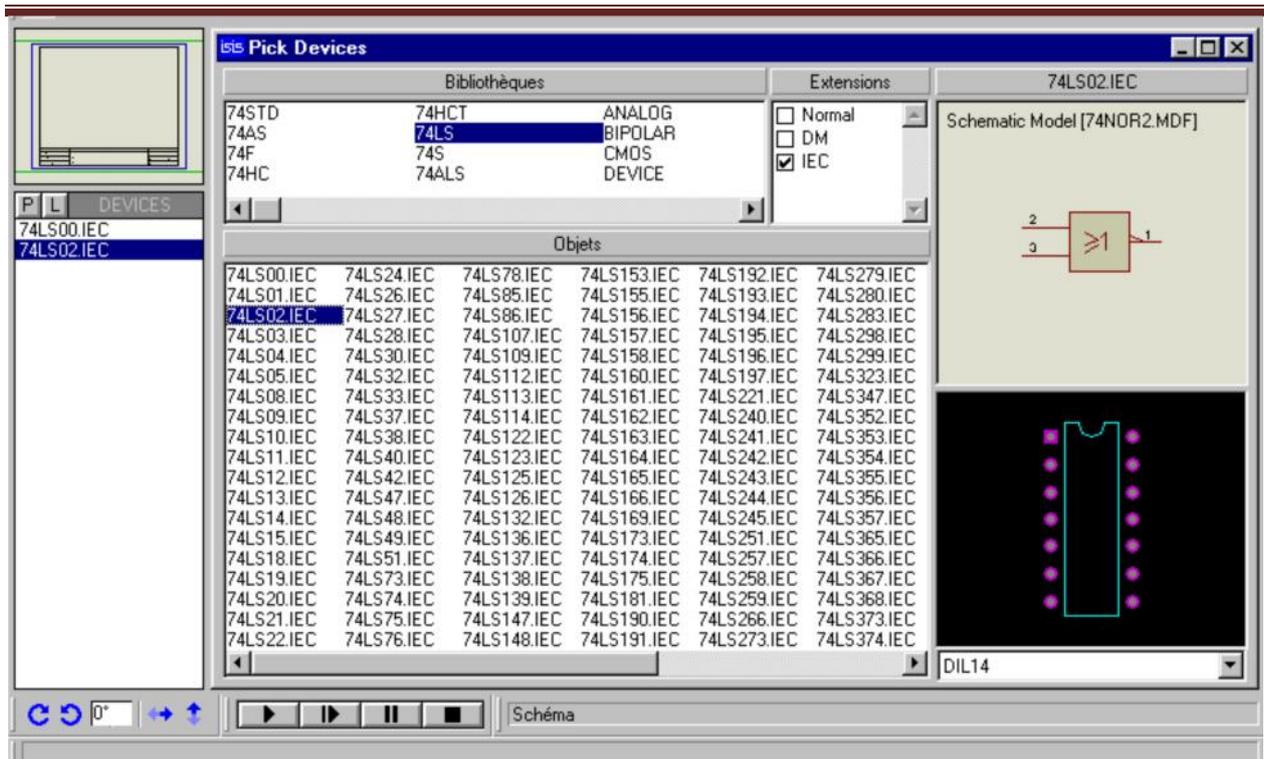
Sélectionnez la touche "Composant" du clavier, puis cliquez sur la lettre P (Prendre composants) les bibliothèques contenant les composants apparaissent alors dans un ordre théoriquement alphanumérique.



SELECTION DES COMPOSANTS:

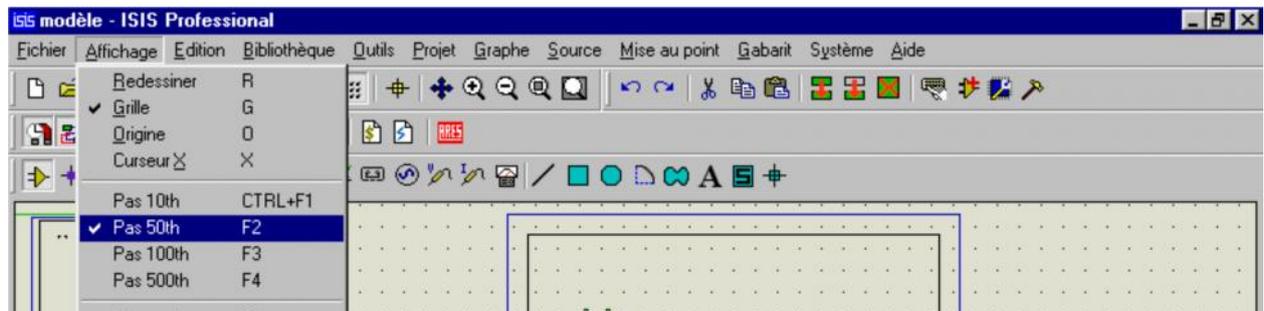
Cliquez (gauche) sur la bibliothèque concernée, puis cliquez (gauche) 2 fois de suite sur les composants souhaités, ceux-ci apparaissent alors dans la liste des composants sélectionnés.

Annexe



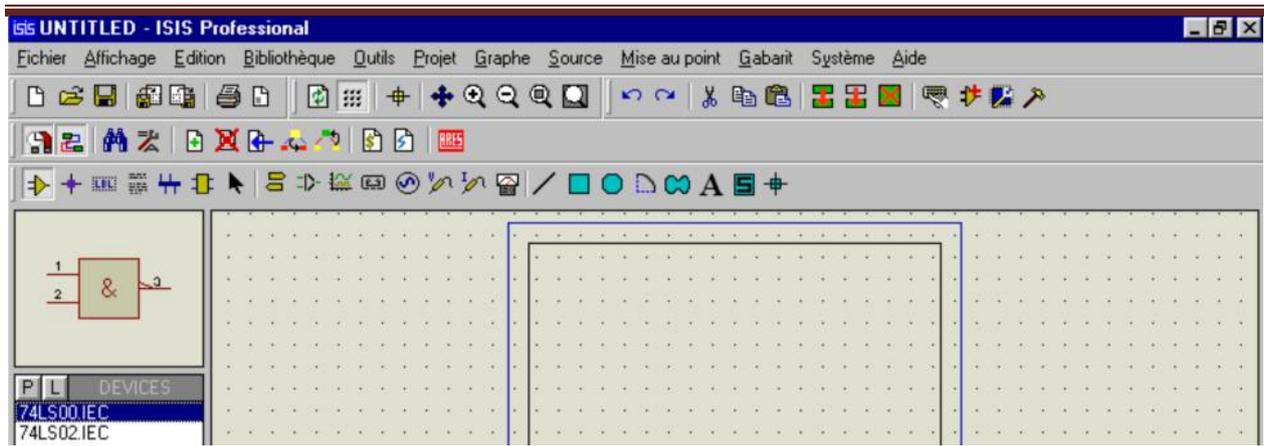
PLACEMENT DES COMPOSANTS SUR LA FEUILLE DE TRAVAIL:

La touche "Composant" doit être préalablement sélectionnée.  De même, il est souhaitable de prendre une grille aux 50 millièmes de pouce. Sélectionnez dans le menu déroulant "Affichage" l'instruction "Pas 50th".



Pour placer un composant, cliquez sur la référence du composant dans la liste des composants sélectionnés, il apparaît alors sur l'écran de contrôle,

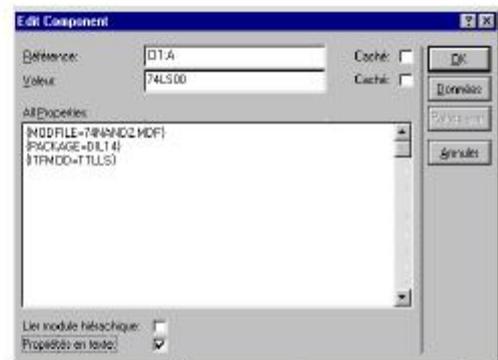
Annexe



Choisissez son orientation en activant les touches suivantes, en bas de l'écran à gauche:



Cliquez alors "gauche" dans la zone de travail où vous souhaitez le placer, puis sélectionnez-le (clic droit), il apparaît alors en surbrillance rouge, enfin éditez-le (clic gauche). Donnez une référence au composant placé, CI1:A, par exemple, CI1:B pour le suivant... Les propriétés de votre composants peuvent également apparaître sous forme de texte comme sur l'écran de droite ci-dessous.



UTILISATION DE LA TOUCHE EDITION INSTANTANEE:

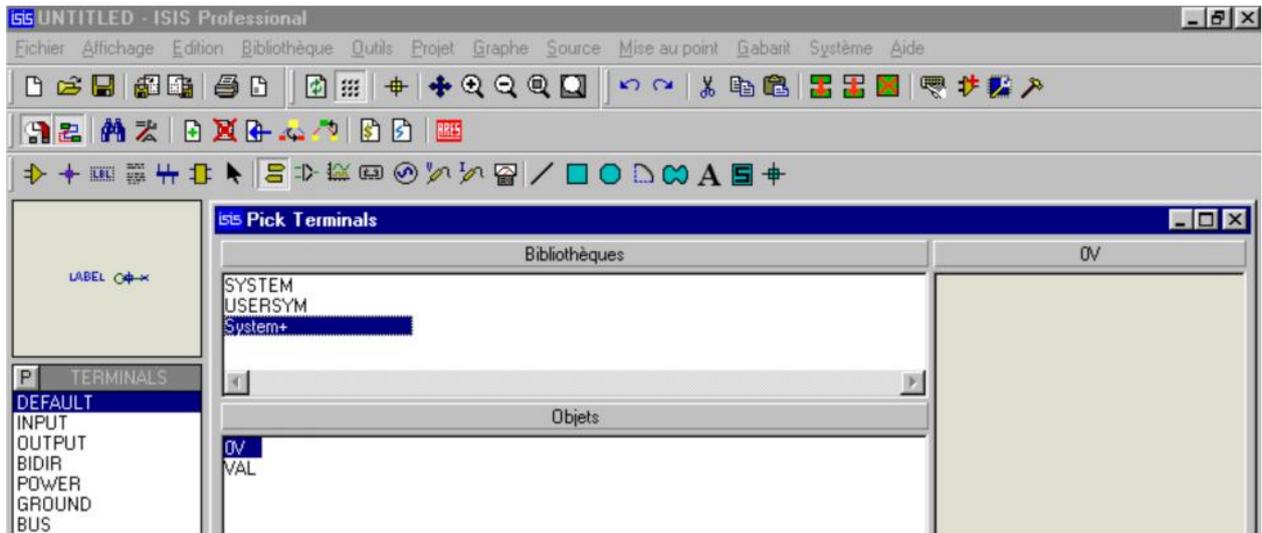
L'opération précédente peut également être réalisée à l'aide de la touche "Edition instantanée" du clavier: un simple clic gauche sur le composant choisi permet d'obtenir l'un des écrans précédents. 

Il convient alors de placer tous les composants et tous les autres objets nécessaires au schéma avant de les éditer, au risque d'en oublier. Il vaut peut-être mieux garder cette option pour la rectification des erreurs constatées in fine.

Annexe

PLACEMENT DES BORNES D'ENTREE, DE SORTIE ET D'ALIMENTATION:

Sélectionnez la touche "Terminal inter-feuille" du clavier:  Les bornes disponibles apparaissent dans la zone des objets sélectionnés. Si vous souhaitez en sélectionner une autre, cliquez sur la lettre P, l'écran suivant apparaît: les bornes disponibles sont dans les bibliothèques System ou System+.



Pour placer une borne, cliquez sur le type de borne souhaitée dans la zone de sélection, elle apparaît alors sur l'écran de contrôle, et ont choisi son orientation en activant les touches suivantes:



Une fois la borne placée, il faut la sélectionner (clic droit) puis l'éditer (clic gauche), ou utiliser la touche "Edition instantanée"; l'écran suivant apparaît.

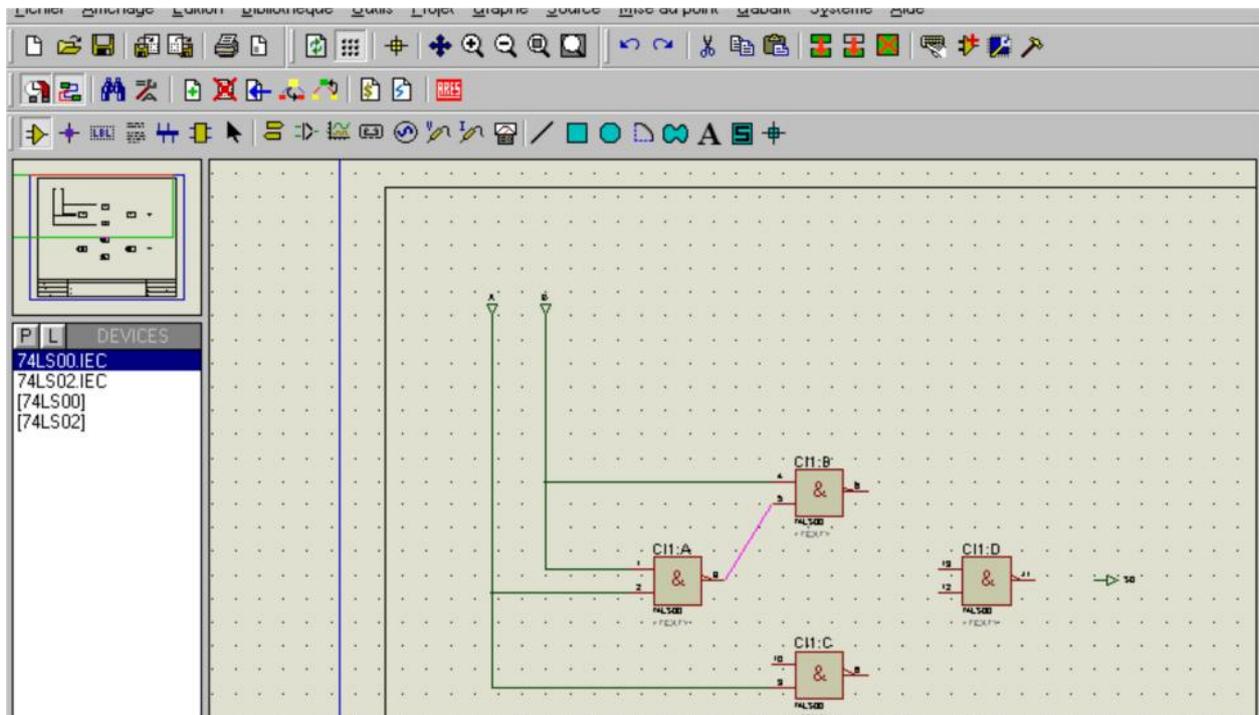


Annexe

Référenciez chacune des bornes placées, et choisissez l'orientation et la disposition de cette référence par rapport à la borne.

REALISATION DES LIAISONS ENTRE COMPOSANTS:

Placez le curseur à l'extrémité de la borne concernée, une croix matérialise la possibilité de faire une liaison; cliquez alors "gauche" pour débiter la liaison, tracez-la en cliquant à chaque fois que vous voulez faire un angle, puis cliquez "gauche" de nouveau à l'endroit où elle se termine, c'est à dire lorsqu'une nouvelle croix matérialise la possibilité de terminer la liaison. On peut également cliquer une première fois "gauche" au point de départ, puis de nouveau au point d'arrivée, et laisser le logiciel chercher un tracé; mais cette solution oblige souvent à reprendre le tracé pour obtenir une solution satisfaisant le concepteur.



DENOMINATION DES LIAISONS:

Sélectionnez la touche "Label de connexion" du clavier:  Cliquez gauche sur la liaison à nommer, puis renseignez la rubrique, comme cela a été fait dans le cas des bornes d'entrée, de sortie et d'alimentation. Ces labels assurent les liaisons équipotentielles entre les liaisons de même dénomination.

Annexe

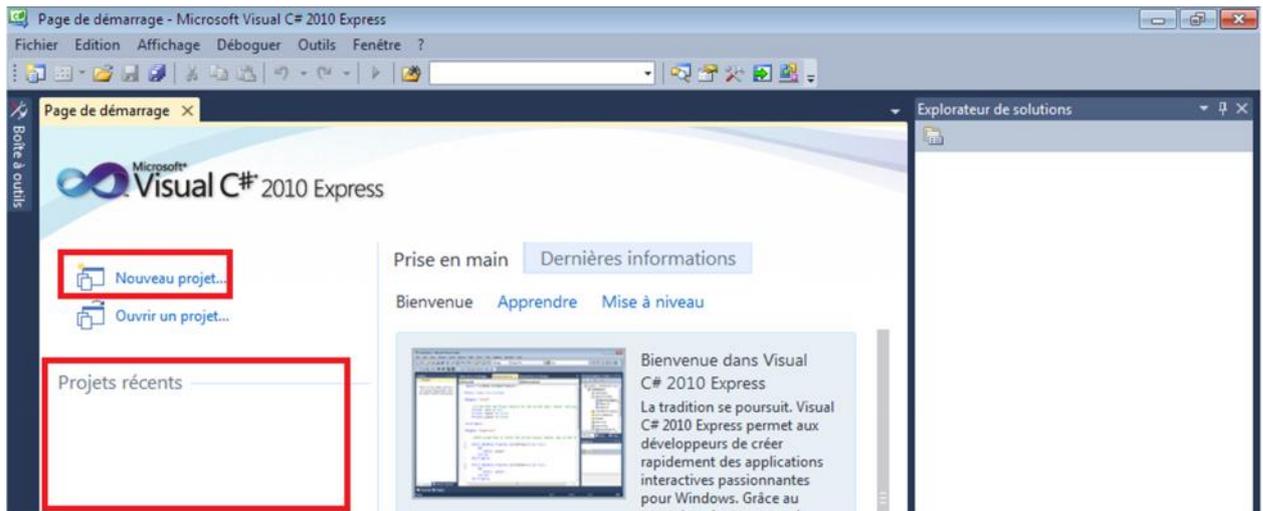
F: Visual C# 2010 Express

Démarrer Visual C# 2010 Express:

Nous allons vérifier que l'installation de Visual C# express a bien fonctionné. Et pour ce faire, nous allons le démarrer et commencer à prendre en main ce formidable outil de développement.

Il vous semblera sûrement très complexe au début mais vous allez voir, si vous suivez ce tutoriel pas à pas, vous allez apprendre les fonctionnalités indispensables. Elles seront illustrées par des copies d'écrans vous permettant de plus facilement vous y retrouver. A force d'utiliser Visual C# express, vous verrez que vous vous sentirez de plus en plus à l'aise et peut-être oserez-vous aller fouiller dans les menus ?

Commencez par démarrer Visual C# 2010 Express. Le logiciel s'ouvre sur la page de démarrage de Visual C# 2010 Express :



Les deux zones entourées de rouge permettent respectivement de créer un nouveau projet et d'accéder aux anciens projets déjà créés. Dans ce deuxième cas, comme je viens d'installer le logiciel, la liste est vide.

Créer un projet :

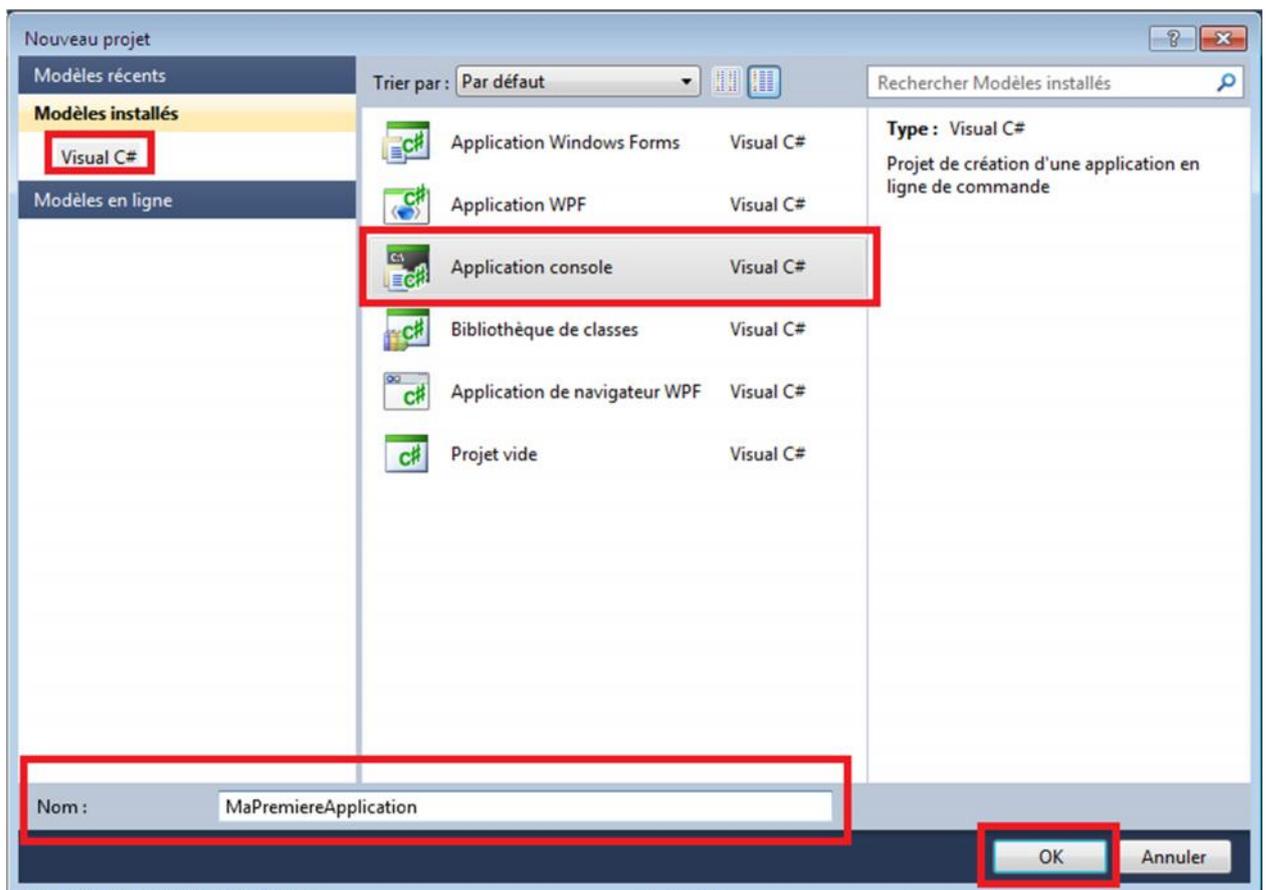
Commençons par créer un nouveau projet en cliquant dans la zone rouge. Cette commande est également accessible via le menu Fichier > Nouveau > Projet

Annexe

Un projet va contenir les éléments de ce que l'on souhaite réaliser. Cela peut être par exemple une application web, une application Windows, etc ...

Le projet est aussi un container de fichiers et notamment dans notre cas de fichiers en langage C# qui vont permettre de construire ce que l'on souhaite réaliser. Le projet est en fait représenté par un fichier dont l'extension est .csproj. Son contenu décrit les paramètres de configuration correspondant à ce que l'on souhaite réaliser et les fichiers qui composent le projet.

Créons donc un nouveau projet. La fenêtre de création de nouveau projet s'ouvre et nous avons plusieurs possibilités de choix. Nous allons dans un premier temps aller dans Visual C# pour choisir de créer une Application console.



Ce que nous faisons ici, c'est utiliser ce qu'on appelle un « **modèle** » (plus couramment appelé par son équivalent anglais : « **template** ») de création de projet.

Si vous naviguez à l'intérieur des différents modèles, vous pourrez constater que Visual C# nous propose des modèles de projets plus ou moins compliqués. Ces modèles sont très utiles pour démarrer un projet car toute la configuration du projet est déjà faite. Le nombre de modèles peut être différent en fonction de votre version de Visual Studio ou du nombre de versions express installées.

Annexe

L'application Console est la forme de projet pouvant produire une application exécutable la plus simple. Elle permet de réaliser un programme qui va s'exécuter dans la console noire qui ressemble à une fenêtre ms-dos, pour les dinosaures comme moi qui ont connu cette époque ... A noter que les projets de type « Bibliothèque de classes » permettent de générer des assemblies de bibliothèques (.dll).

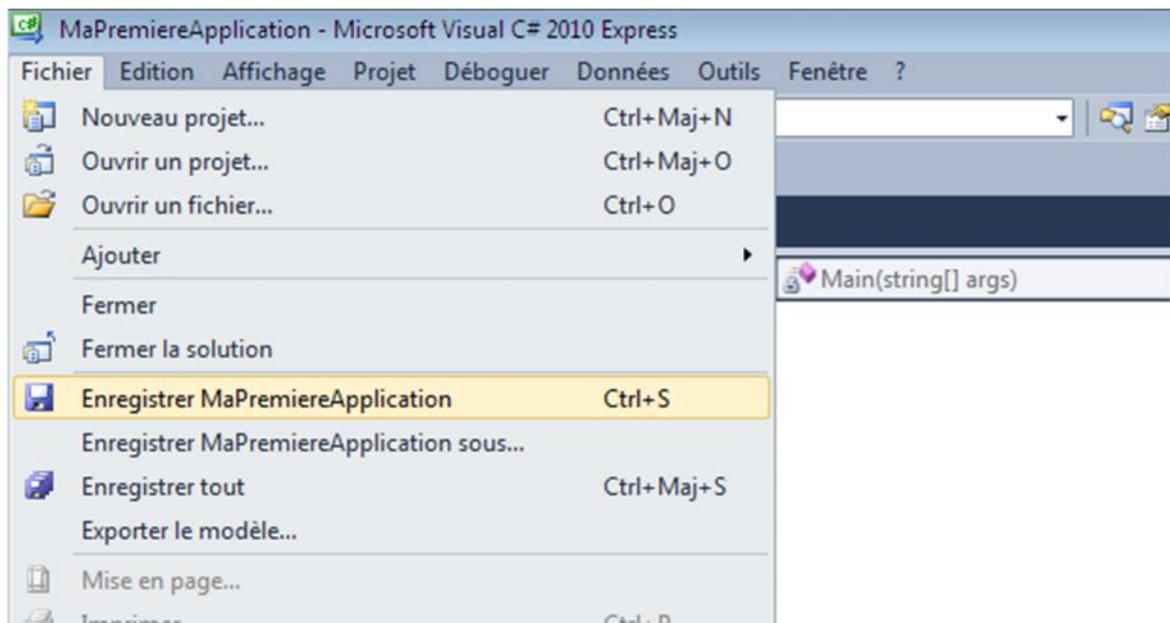
Dans cette console, nous allons pouvoir notamment afficher du texte simple. Ce type de projet est parfait pour démarrer l'apprentissage du C# car il n'y a besoin que de savoir comment afficher du texte pour commencer alors que pour réaliser une application graphique par exemple, il y a beaucoup d'autres choses à savoir.

En bas de la fenêtre de création de projet, nous avons la possibilité de choisir un nom pour le projet, ici ConsoleApplication1. Changeons le nom de notre application, par exemple "MaPremiereApplication", dans la zone correspondante.

Cliquons sur OK pour valider la création de notre projet.

Visual C# Express crée alors pour nous les fichiers composant une application console vide, qui utilise le C# comme langage et que nous avons nommé MaPremiereApplication.

Avant toute chose, nous allons enregistrer le projet. Allons dans le menu Fichier > Enregistrer (ou utiliser le raccourci bien connu `ctrl+s`) :



Annexe

Analyse rapide de l'environnement de développement et du code généré :

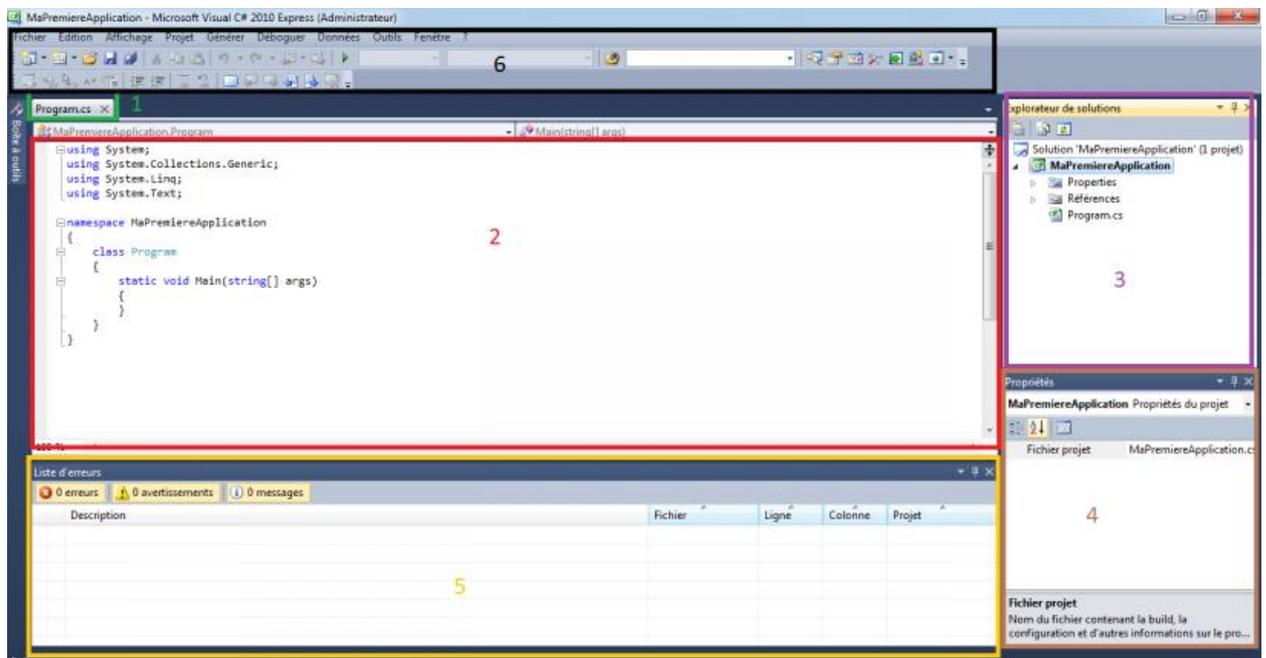
Allons dans l'emplacement renseigné (ici c:\users\ico\documents\visual studio 2010\Projects), nous pouvons constater que Visual C# Express a créé un répertoire MaPremiereApplication, c'est le fameux répertoire pour la solution qu'il nous a proposé de créer.

Dans ce répertoire, nous remarquons notamment un fichier MaPremiereApplication.sln. C'est ce qu'on appelle le fichier de solution ; il s'agit juste d'un container de projets qui va nous permettre de visualiser nos projets dans visual C# express.

En l'occurrence, pour l'instant, nous avons un seul projet dans la solution: l'application MaPremiereApplication, que nous retrouvons dans le sous répertoire MaPremiereApplication et qui contient notamment le fichier de projet : MaPremiereApplication.csproj.

Il y a encore un fichier digne d'intérêt (pour l'instant) dans ce répertoire, il s'agit du fichier Program.cs. Les fichiers dont l'extension est .cs contiennent du code C#, c'est dans ce fichier que nous allons commencer à taper nos premières lignes de code ...

Si nous retournons dans l'interface de Visual C# express, nous pouvons retrouver quelque chose comme ça :



La zone verte numéro 1 contient les différents fichiers ouverts sous la forme d'un onglet. On voit que par défaut, Visual C# nous a créé et ouvert le fichier Program.cs.

Dans la zone rouge numéro 2, c'est l'éditeur de code. Il affiche le contenu du fichier ouvert. Nous voyons des mots que nous ne comprenons pas encore. C'est du code qui a été

Annexe

automatiquement généré par Visual C#. Nous pouvons observer que les mots sont de différentes couleurs. En effet, l'éditeur Visual C# express possède ce qu'on appelle une **coloration syntaxique**, c'est-à-dire que certains mots clés sont colorés d'une couleur différente en fonction de leur signification ou de leur contexte afin de nous permettre de nous y retrouver plus facilement.

La zone numéro 3 en violet est l'explorateur de solutions, c'est ici que l'on voit le contenu de la solution sur laquelle nous travaillons en ce moment. En l'occurrence, il s'agit de la solution « MaPremiereApplication » qui contient un unique projet « MaPremiereApplication ». Ce projet contient plusieurs sous éléments :

- Proprieties : contient des propriétés de l'application, on ne s'en occupe pas pour l'instant
- Références : contient les références de l'application, on ne s'en occupe pas pour l'instant
- Program.cs est le fichier qui a été généré par Visual C# et qui contient le code C#. Il nous intéresse fortement !!

La zone 4 en brun est la zone contenant les propriétés de ce sur quoi nous travaillons en ce moment. Ici, nous avons le curseur positionné sur le projet, il n'y a pas beaucoup d'informations excepté le nom du fichier de projet. Nous aurons l'occasion de revenir sur cette fenêtre plus tard.

La zone 5 en jaune n'est pas affichée au premier lancement, elle contient la liste des erreurs, des avertissements et des messages de notre application. Nous verrons comment l'afficher un peu plus bas.

La zone 6 en noir est la barre d'outils, elle possède plusieurs boutons que nous pourrons utiliser, notamment pour exécuter notre application.

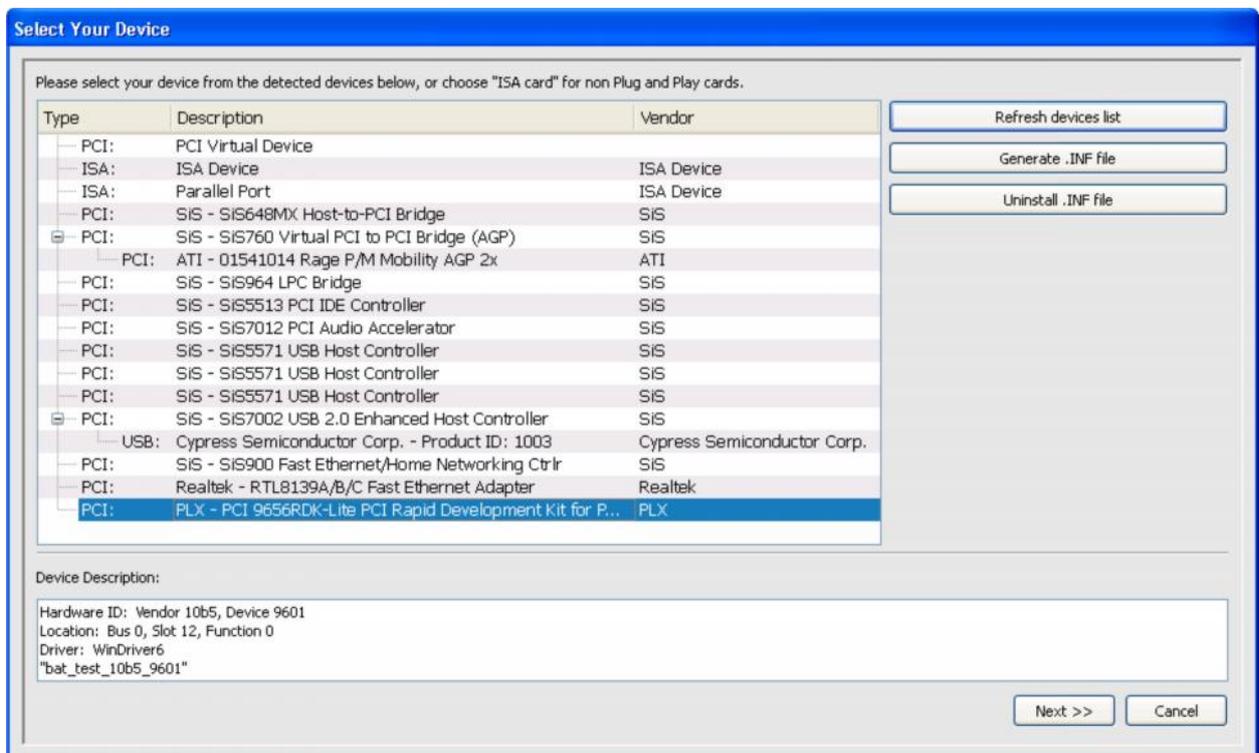
Annexe

G: WinDriver

Créer un nouveau projet “New host driver project”:



Sélectionnez votre carte Plug and Play de la liste des périphériques détectés par DriverWizard.



Annexe

Générer un fichier INF pour DriverWizard

Enter Information for INF File

Please fill in the information below for your device.

This information will be incorporated into the INF file, which WinDriver will generate for your device.

The information you specify will appear in the Device Manager after the installation of the INF file.

Vendor ID: Device ID:

Manufacturer name:

Device name:

Device Class:

WinDriver's unique Class.

Use this option for a non-standard type of device. WinDriver will set a new Class type for your device.

Support Message Signaled Interrupts (MSI/MSI-X)

Automatically install the INF file.

Note: This will replace any existing driver you may have for your device.

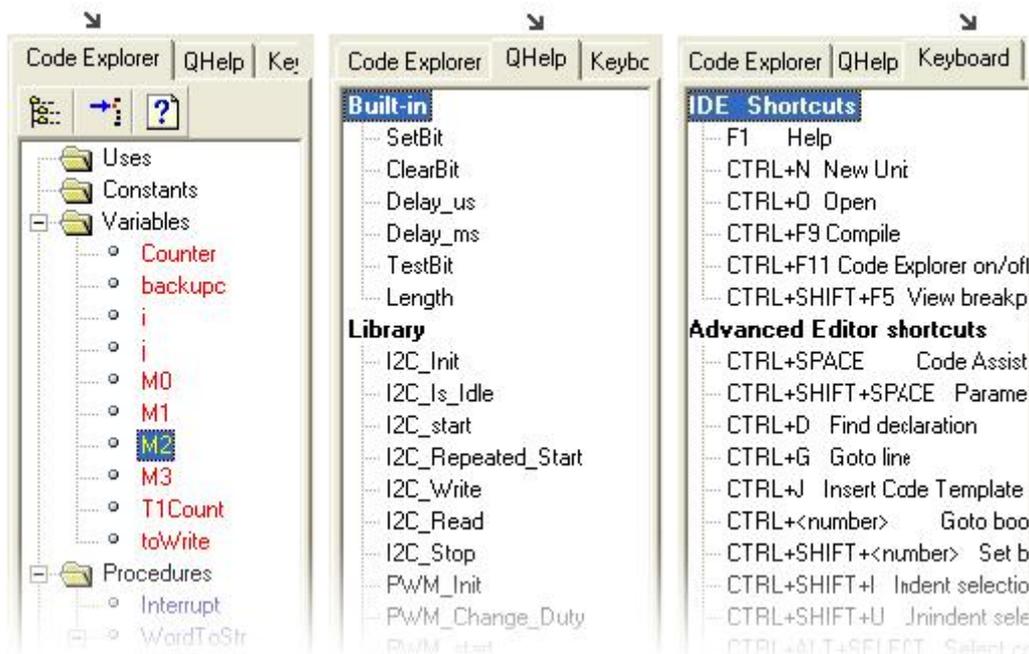
Annexe

H: Compilateur "MikroC PRO" pour PIC

L'explorateur de code :

Cette fonction intégrée au compilateur vous permettra de surveiller et de retrouver rapidement vos variables, fonctions et procédures à l'intérieur de votre programme. Pour ce faire, il vous suffira simplement de cliquer sur un élément pour que l'explorateur vous positionne directement dessus (idéal lorsque vous travaillez sur de "gros" projets).

L'explorateur vous permet également de vous donner la liste complète des instructions ou des touches de raccourci permettant d'activer les différentes fonctions du logiciel.



- L'explorateur de code permet de retrouver aisément tous les éléments.
- L'aide rapide permet en cliquant sur une instruction d'obtenir sa description.
- Le rappel des touches de fonctions simplifie la saisie.

L'éditeur de code :

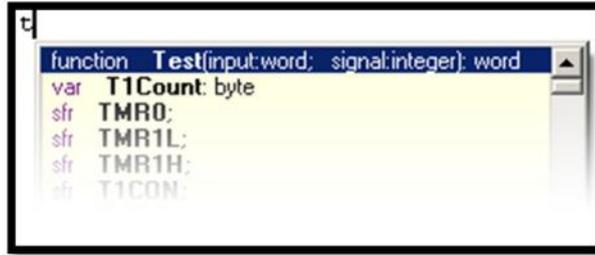
Extrêmement convivial, l'éditeur du "MikroC PRO" pour PIC dispose d'une multitude de fonctions qui permettront aussi bien aux utilisateurs novices comme aux professionnels de travailler très rapidement.

L'assistant de code :

Ce dernier vous permettra après avoir tapé les premières lettres d'une instruction d'ouvrir une fenêtre en tapant CTRL + ESPACE afin d'obtenir la liste de toutes les instructions

Annexe

commençant par les mêmes lettres. Il vous suffira ensuite de cliquer sur celle qui vous convient pour l'importer directement dans l'éditeur.



```
function Test(input:word; signal:integer): word
var T1Count: byte
sfr TMR0;
sfr TMR1L;
sfr TMR1H;
sfr T1CON;
```

L'assistant de paramètres :

Ce dernier vous permettra après avoir saisi le début d'une fonction nécessitant des paramètres, d'afficher la syntaxe de ces paramètres (le premier paramètre à écrire s'affiche en caractères gras). Après avoir saisi ce premier paramètre, le second s'affiche alors à son tours en caractères gras).



```
Test (

```

Correcteur automatique :

Ce dernier vous permettra de corriger automatiquement vos erreurs de frappes les plus courantes. Vous pourrez également ajouter vos propres préférences dans la liste des mots à reconnaître.

Mise en forme automatique:

Afin d'obtenir une lisibilité maximale, il est possible de générer automatiquement la mise en forme de certaines instructions. Par exemple, tapez l'instruction IF, puis CTRL + J. A ce stade, l'éditeur ajoute tout seul les instructions THEN et END IF. A vous ensuite de compléter le programme !

Affichage syntaxique coloré:

Toujours dans le but d'obtenir une lisibilité maximale de votre programme, il vous est possible de configurer entièrement les couleurs de chaque partie du listing. Par exemple les commentaires en "vert", les instructions en "noir", les valeurs numériques en "bleu clair", etc, etc....

Annexe

I: Descripteur USB

```
const unsigned int USB_VENDOR_ID = 0x1105;
const unsigned int USB_PRODUCT_ID = 0x1990;
const char USB_SELF_POWER = 0x80;          // Self powered 0xC0, 0x80 bus powered
const char USB_MAX_POWER = 50;            // Bus power required in units of 2 mA
const char HID_INPUT_REPORT_BYTES = 64;
const char HID_OUTPUT_REPORT_BYTES = 64;
const char USB_TRANSFER_TYPE = 0x03;      //0x03 Interrupt
const char EP_IN_INTERVAL = 1;
const char EP_OUT_INTERVAL = 1;

const char USB_INTERRUPT = 1;
const char USB_HID_EP = 1;
const char USB_HID_RPT_SIZE = 33;

/* Device Descriptor */
const struct {
    char bLength;          // bLength - Descriptor size in bytes (12h)
    char bDescriptorType;  // bDescriptorType - The constant DEVICE (01h)
    unsigned int bcdUSB;   // bcdUSB - USB specification release number (BCD)
    char bDeviceClass;     // bDeviceClass - Class Code
    char bDeviceSubClass;  // bDeviceSubClass - Subclass code
    char bDeviceProtocol;  // bDeviceProtocol - Protocol code
    char bMaxPacketSize0;  // bMaxPacketSize0 - Maximum packet size for endpoint 0
    unsigned int idVendor; // idVendor - Vendor ID
    unsigned int idProduct; // idProduct - Product ID
    unsigned int bcdDevice; // bcdDevice - Device release number (BCD)
    char iManufacturer;    // iManufacturer - Index of string descriptor for the
    manufacturer
    char iProduct;         // iProduct - Index of string descriptor for the product.
    char iSerialNumber;    // iSerialNumber - Index of string descriptor for the serial
    number.
    char bNumConfigurations; // bNumConfigurations - Number of possible
    configurations
} device_dsc = {
    0x12,          // bLength
    0x01,          // bDescriptorType
    0x0200,        // bcdUSB
    0x00,          // bDeviceClass
    0x00,          // bDeviceSubClass
    0x00,          // bDeviceProtocol
    8,             // bMaxPacketSize0
    USB_VENDOR_ID, // idVendor
    USB_PRODUCT_ID, // idProduct
    0x0001,        // bcdDevice
    0x01,          // iManufacturer
    0x02,          // iProduct
    0x00,          // iSerialNumber
    0x01           // bNumConfigurations
};
```

Annexe

```
/* Configuration 1 Descriptor */
const char configDescriptor1[] = {
    // Configuration Descriptor
    0x09,          // bLength      - Descriptor size in bytes
    0x02,          // bDescriptorType - The constant CONFIGURATION (02h)
    0x29,0x00,     // wTotalLength  - The number of bytes in the configuration
descriptor and all of its subordinate descriptors
    1,             // bNumInterfaces - Number of interfaces in the configuration
    1,             // bConfigurationValue - Identifier for Set Configuration and Get
Configuration requests
    0,             // iConfiguration - Index of string descriptor for the configuration
    USB_SELF_POWER, // bmAttributes  - Self/bus power and remote wakeup
settings
    USB_MAX_POWER, // bMaxPower     - Bus power required in units of 2 mA

    // Interface Descriptor
    0x09,          // bLength - Descriptor size in bytes (09h)
    0x04,          // bDescriptorType - The constant Interface (04h)
    0,             // bInterfaceNumber - Number identifying this interface
    0,             // bAlternateSetting - A number that identifies a descriptor with alternate
settings for this bInterfaceNumber.
    2,             // bNumEndpoint - Number of endpoints supported not counting
endpoint zero
    0x03,          // bInterfaceClass - Class code
    0,             // bInterfaceSubclass - Subclass code
    0,             // bInterfaceProtocol - Protocol code
    0,             // iInterface - Interface string index

    // HID Class-Specific Descriptor
    0x09,          // bLength - Descriptor size in bytes.
    0x21,          // bDescriptorType - This descriptor's type: 21h to indicate the HID
class.
    0x01,0x01,     // bcdHID - HID specification release number (BCD).
    0x00,          // bCountryCode - Numeric expression identifying the country for
localized hardware (BCD) or 00h.
    1,             // bNumDescriptors - Number of subordinate report and physical
descriptors.
    0x22,          // bDescriptorType - The type of a class-specific descriptor that follows
USB_HID_RPT_SIZE,0x00, // wDescriptorLength - Total length of the descriptor
identified above.

    // Endpoint Descriptor
    0x07,          // bLength - Descriptor size in bytes (07h)
    0x05,          // bDescriptorType - The constant Endpoint (05h)
    USB_HID_EP | 0x80, // bEndpointAddress - Endpoint number and direction
    USB_TRANSFER_TYPE, // bmAttributes - Transfer type and supplementary
information
    0x40,0x00,     // wMaxPacketSize - Maximum packet size supported
    EP_IN_INTERVAL, // bInterval - Service interval or NAK rate

    // Endpoint Descriptor
```

Annexe

```
0x07,          // bLength - Descriptor size in bytes (07h)
0x05,          // bDescriptorType - The constant Endpoint (05h)
USB_HID_EP,    // bEndpointAddress - Endpoint number and direction
USB_TRANSFER_TYPE, // bmAttributes - Transfer type and supplementary
information
0x40,0x00,     // wMaxPacketSize - Maximum packet size supported
EP_OUT_INTERVAL // bInterval - Service interval or NAK rate
};

const struct {
    char report[USB_HID_RPT_SIZE];
}hid_rpt_desc =
{
    {0x06, 0x00, 0xFF, // Usage Page = 0xFF00 (Vendor Defined Page 1)
    0x09, 0x01, // Usage (Vendor Usage 1)
    0xA1, 0x01, // Collection (Application)
    // Input report
    0x19, 0x01, // Usage Minimum
    0x29, 0x40, // Usage Maximum
    0x15, 0x00, // Logical Minimum (data bytes in the report may have minimum
value = 0x00)
    0x26, 0xFF, 0x00, // Logical Maximum (data bytes in the report may have
maximum value = 0x00FF = unsigned 255)
    0x75, 0x08, // Report Size: 8-bit field size
    0x95, HID_INPUT_REPORT_BYTES, // Report Count
    0x81, 0x02, // Input (Data, Array, Abs)
    // Output report
    0x19, 0x01, // Usage Minimum
    0x29, 0x40, // Usage Maximum
    0x75, 0x08, // Report Size: 8-bit field size
    0x95, HID_OUTPUT_REPORT_BYTES, // Report Count
    0x91, 0x02, // Output (Data, Array, Abs)
    0xC0} // End Collection
};

//Language code string descriptor
const struct {
    char bLength;
    char bDscType;
    unsigned int string[1];
} strd1 = {
    4,
    0x03,
    {0x0409}
};

//Manufacturer string descriptor
const struct{
    char bLength;
    char bDscType;
    unsigned int string[9];
```

Annexe

```
}strd2={  
    20, //sizeof this descriptor string  
    0x03,  
    {'B','e','k','h','o','u','c','h','e'}  
};  
  
//Product string descriptor  
const struct{  
    char bLength;  
    char bDscType;  
    unsigned int string[13];  
}strd3={  
    28, //sizeof this descriptor string  
    0x03,  
    {'B','e','k','h','o','u','c','h','e',' ','U','S','B'}  
};  
  
//Array of configuration descriptors  
const char* USB_config_dsc_ptr[1];  
  
//Array of string descriptors  
const char* USB_string_dsc_ptr[3];  
  
void USB_Init_Desc(){  
    USB_config_dsc_ptr[0] = &configDescriptor1;  
    USB_string_dsc_ptr[0] = (const char*)&strd1;  
    USB_string_dsc_ptr[1] = (const char*)&strd2;  
    USB_string_dsc_ptr[2] = (const char*)&strd3;  
}
```

Annexe

j: Les composants

	Composant	Valeur
BLOC-1	R11 (Résistance fusible)	100mA, 50
	R12 (Résistance)	1 K
	DZ11, DZ12 (Diode Zener)	BZX55C27
	D11, D12, D13, D14 (Diode)	1N4001
BLOC-2	R21 (Résistance)	900 K
	R22 (Résistance)	100 K
BLOC-3	74HC4051 (multiplexeur analogique)	-
	C31 (capacité)	100 nF
BLOC-4	R41, R42, R43, R44 (Résistance)	20 K
	R45, R49 (Résistance)	1 K
	R46 (Résistance)	8 K
	R47 (Résistance)	40 K
	R48 (Résistance)	200 K
	AOP41, AOP42, AOP43 (Amplificateur)	LM324
	74HC4051 (multiplexeur analogique)	-
Bloc-5	PIC18f4550 (Microcontrôleur)	-
	R51, R53, R54 (Résistance)	1 K
	R52 (Résistance)	15 K
	C51, C52 (capacité)	22 pF
	C53 (capacité)	470 nF
	X1 (quartz)	20 MHz
	D51 (Led)	rouge
	D52 (Led)	vert
Bloc-6	NE555 (Timer)	-
	R61 (Résistance)	2 K
	R62 (Résistance)	10 K
	C61 (capacité)	10 uF
	C62 (capacité)	680 nF
	C63, C64, C65, C66	4.7 uF
	D61, D62, D63, D64 (Diode)	1N4001