

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

جامعة محمد بوضياف - المسيلة

UNIVERSITE MOHAMED BOUDIAF - M'SILA

كلية التكنولوجيا

FACULTE DE TECHNOLOGIE

قسم الإلكترونيك

DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE MASTER

DOMAINE : SCIENCES ET TECHNOLOGIE

FILIERE : ELECTRONIQUE

OPTION : INSTRUMENTATION ET MAINTENANCE INDUSTRIELLE

THEME

**Etude et réalisation d'une carte d'interface de
commande et de contrôle via le port USB**

Proposé et dirigé par : Mr. H. Oudira

Réalisé par : Nouar kherkhachi Ibrahim

N° D'ordre:

PROMOTION: JUIN 2016

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Au nom de dieu clément et miséricordieux

Dédicaces 

*Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la
volonté nécessaire pour affronter les différentes difficultés de la vie,*

Je dédie ce modeste travail :

*A ceux qui sont les plus chers du monde, Mon père, et ma mère, à qui je
n'arriverai jamais à exprimer ma gratitude et ma reconnaissance, pour ses
amours ses soutiens tout au long de mes études.*

A mes frères et mes sœurs.

A toute ma famille.

A mes amis et mes collègues de la promotion 2016 « Electronique ».

Remerciement

Avant de commencer la présentation de ce travail, je profite de l'occasion pour remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'études.

Tout d'abord on remercie le bon dieu puissant de la bonne santé, la volonté et de la patience qu'il nous a donnée tout au long de notre étude.

Nous remercions Très sincèrement Mr. H. OUDIRA notre encadreur de ce travail, pour ses conseils pertinents, et ses orientations judicieuses sa patience et diligence, et par ses suggestions a grandement facilité ce travail.

Nous tenons à exprimer notre gratitude aux membres de jury qui ont bien voulu examiner ce travail.

Nos remerciements vont aussi à tous les enseignants du département d'électronique qui ont contribué à notre formation.

Enfin nous tenons à exprimer notre reconnaissance à tous nos amis et collègues pour leur le soutien moral et matériel...



Table des matières

Introduction générale	1
------------------------------------	---

Chapitre I

Généralité sur l'acquisition de données

I.1. Introduction	3
I.2. Composition d'un système DAQ	3
I.2.1. Transducteurs (capteurs)	4
I.2.2. Signaux.....	4
I.2.2.1. Signaux analogiques.....	5
I.2.2.2. Signaux numériques	6
I.2.3. Conditionnement du signal	7
I.2.4. Carte d'acquisition de donnée	8
I.3. Structure d'une chaîne d'acquisition numérique	8
I.3.1. Capteur	9
I.3.2. Amplificateur de signal	9
I.3.3. Filtre d'entrée	9
I.3.4. Le convertisseur analogique numérique (CAN)	9
I.3.4.1. Identification de la fonction C.A.N.....	9
I.3.4.2. Symbole d'un C.A.N.....	9
I.4. La fonctionnement d'un CAN	10
I.4.1. Echantillonnage.....	10
I.4.2. Bloqueur.....	11
I.4.3. Numérisation	11
I.4.3.1. Principe	11
I.4.3.2. Résolution	11
I.5. Acquisition de plusieurs grandeurs	12
I.5.1. Acquisition séquentielle décalée	12
I.5.1.1. Le multiplexage.....	12
I.5.2. Acquisition séquentielle simultanée.....	13
I.5.3. Acquisition parallèle	13
I.6. Le Protocol de communication USB	14
I.6.1. Naissance de L'USB	14
I.6.2. Vitesse de transfert de l'USB.....	14
I.6.3. Débits réels.....	14

I.6.4. Le câble USB	15
I.6.4.1. Composition du câble USB	15
I.6.5. Codage de données.....	17
I.6.6. Mode de transfert sur le bus USB	17
I.6.6.1. Transfert en mode Contrôle	17
I.6.6.2. Transfert en mode Interruption	17
I.6.6.3. Transfert en mode Isochrone.....	17
I.6.6.4. Transfert en mode Bulk.....	18
I.6.7. Choix « Low » ou « Full » USB	18
I.6.8. Avantages de l'USB.....	19
I.6.8.1. Avantage de l'alimentation USB	19
I.7. Conclusion	20

Chapitre II

Généralité sur le logiciel de l'interfaçage (LABVIEW)

II.1. Introduction	21
II.1.2. Historique.....	21
II.2. LabVIEW dans le monde de l'instrumentation.....	22
II.2.1. Les instruments virtuels	23
II.3. Environnement LABVIEW	25
II.3.1. Les fenêtres de la face avant du diagramme	26
II.3.2. Face-avant.....	26
II.3.3. Terminaux sur le diagramme	27
II.3.4. Icône et connecteur	27
II.4. Fils de liaison.....	28
II.4.1. Nœuds	28
II.4.2. Barre d'outils du diagramme	29
II.4.3. Barre d'outils de la face avant	30
II.5. Palettes.....	30
II.5.1. Palette d'outils	30
II.5.2. Palette de commande	31
II.5.3. Palette de fonction	32
II.6. Structures de programmation	34
II.6.1. Structures condition	35
II.6.2. Boucle For.....	35
II.6.3. Boucle While	36
II.6.4. Structure Séquence déroulée.....	36

II.6.5. La structure évènement.....	37
II.7. L'avantage de logiciel de LABVIEW.....	37
II.8. Conclusion.....	38

Chapitre III

Notion sur les microcontrôleurs

III.1. Introduction.....	39
III.2. Les microcontrôleurs (PIC)	39
III.2.1. Généralités.....	39
III.2.2. Classification des PICs de Microchip.....	40
III.2.3. Identification des PIC	40
III.3. Le choix du microcontrôleur	40
III.4. Caractéristiques principales du pic 18F4550	41
III.5. Brochage du 18F4550	43
III.6. Architecture générale du PIC	44
III.6.1 Organisation de la mémoire.....	45
III.6.1.1. Mémoire de programme	45
III.6.1.2. Mémoire de données.....	45
III.6.2. Les temporisateurs (timers)	46
III.6.3. Horloges	46
III.6.3.1. Types d'Oscillateur	46
III.6.3.2. Paramètres d'oscillateur pour USB	48
III.6.3.2.1. Le PLL.....	48
III.7. Les convertisseurs analogiques/numériques	50
III.7.1. Registre ADCON0.....	51
III.7.2. Registre ADCON1.....	52
III.7.3. Registre ADCON2.....	53
III.8. Les avantages du microcontrôleur	54
III.9. Conclusion	56

Chapitre IV

Conception et réalisation de la carte

IV.1. Introduction.....	57
IV.2. Application.....	57
IV.3. Description du fonctionnement de la carte réalisée	58
IV.4. Structure générale de la carte	59
IV.5. L'unité de contrôle	59

IV.6. Software et programmation avancé	60
IV.6.1. Programmation du PIC18F4550.....	61
IV.6.2. Configuration de registre de PIC dans la fenêtre Edit Project	63
IV.6.3. Générer le Descripteur de l'interface.....	63
IV.7. Programmation LABVIEW	68
IV.7.1. l'interface de commande et de contrôle	68
IV.7.2. Définition de la carte au LabVIEW.....	68
IV.7.3. Configuration de NI-VISA pour le contrôle d'un bus USB	69
IV.7.4. Test des communications avec VISA Interactive Contrôle et labVIEW	72
IV.8. Réalisation de la carte d'acquisition	75
IV.8.1. Présentation	75
IV.9. Application hardware et software	77
IV.10. Résultats expérimentaux	79
IV.11. Conclusion	81
Conclusion générale	82

Liste des figures

Figure I.1 : Structure d'une chaîne d'acquisition.....	3
Figure I.2 : Principales caractéristiques d'un signal analogique	5
Figure I.3 : Principales caractéristiques d'un signal numérique.....	6
Figure I.4 : Structure d'une carte d'acquisition	8
Figure I.5 : Structure de l'acquisition numérique	8
Figure I.6 : Symbole de la fonction Conversion Analogique / Numérique	9
Figure I.7 : Principe du chaîne de conversion analogique-numérique.....	10
Figure I.8 : Théorème de Shannon.....	10
Figure I.9 : Structure séquentielle décalée	12
Figure I.10 : Structure séquentielle simultanée.....	13
Figure I.11 : Structure parallèle	13
Figure I.12 : Symbole du bus USB.	15
Figure I.13 : Composition d'un câble USB.....	15
Figure I.14 : Brochage des connecteurs USB de type A et B	16
Figure I.15 : Aspect des connecteurs USB de type A et B	16
Figure I.16 : Principe du Codage NRZI.....	17
Figure I.17 : Choix Full Speed – Résistance de Pull Up.....	18
Figure I.18 : Choix Low Speed – Résistance de Pull Up.....	19
Figure II.1 : Les instruments virtuels (VIs)	24
Figure II.2 : La face avant et son diagramme	25
Figure II.3 : L'icône et son connecteur	25
Figure II.4 : Fenêtre de démarrage.....	25
Figure II.5.a : La face avant	26
Figure II.5.b : Le diagramme.....	26
Figure II.6 : L'icône de VI	27
Figure II.7 : Connecteur de VI.....	27
Figure II.8 : Différentes structures de données.....	29
Figure II.9 : La palette d'outils.....	31
Figure II.10 : La palette de commandes.....	32
Figure II.11 : La palette de fonction	32
Figure II.12 : Les structures de programmation.....	34

Figure II.13 : Composantes d'une structure Condition	35
Figure II.14 : Composantes d'une Boucle For	35
Figure II.15 : Composantes d'une Boucle While	36
Figure II.16 : Structure Séquence déroulée.....	37
Figure II.17 : Structure événement.	37
Figure III.1.a : Brochage du 18F4550(40-Pin PDIP)	43
Figure III.1.b : Brochage du 18F4550(44-Pin TQFP) .[6]	43
Figure III.2 : Architecture générale du PIC18F4550 [6]	44
Figure III.3 : Diagramme de block PLL (mode HS) [21].....	49
Figure III.4 : Synoptique de la circuiterie d'horloge sur un PIC 18F4550 gérant l'USB .	49
Figure III.5 : Synoptique général du CAN pour le PIC18F4550.[6].....	51
Figure III.6 : Registre ADCON0 [6].....	52
Figure III.7 : Registre ADCON1[6].....	53
Figure III.8 : Registre ADCON2	54
Figure IV.1 : Le rôle de notre carte d'acquisition.....	57
Figure IV.2 : Schéma électrique de la carte d'acquisition Complet.	58
Figure IV.3 : Le Microcontrôleur 18 F4550.40-Pin PDIP	59
Figure IV.4 : La fenêtre de démarrage de logiciel mikroC PRO for PIC.	61
Figure IV.5 : Structure des fichiers constituant le programme pour le PIC18F4550.	61
Figure IV.6 : L'organigramme principal du PIC.	62
Figure IV.7 : La configuration de registre de PIC 18F4550 dans la fenêtre Edit Project...63	
Figure IV.8 : Générer le descripteur de la carte d'interface	64
Figure IV.9 : Le fichier de programme (.hex).....	65
Figure IV.10 : Simulation du circuit sur le logiciel Proteus 8 Professional.....	65
Figure IV.11 : Le chemin du fichier (.hex)	66
Figure IV.12 : L'emplacement de programme Virtuel USB	67
Figure IV.13 : Identification de dispositif USB.....	67
Figure IV.14 : Propriétés de l'appareil.	68
Figure IV.15 : Propriété de la carte d'interface sur le gestionnaire de périphérique.	68
Figure IV.16 : logiciel de reconnaissance du périphérique connecté avec LabVIEW.....	69
Figure IV.17 : Fenêtre de la sélection du bus matériel de l'Assistant VISA	70
Figure IV.18 : Fenêtre des informations de base sur le matériel de VISA.	70
Figure IV.19: La sélection du dossier d'enregistrement.....	71

Figure IV.20 : Le nouveau périphérique détecté dans le gestionnaire périphérique.....	71
Figure IV.21 : Icône de NI MAX version v2015.	72
Figure IV.22 : Les paramètres de la carte d'interface dans NI MAX de labVIEW.	72
Figure IV.23 : Interface d'utilisateur pour le commande et le contrôle	73
Figure IV.24 : Interface d'utilisateur face (OPTION DE REGLAGE)	74
Figure IV.25 : Interface d'utilisateur face (OPTION AVANCE).	74
Figure IV.26 : Présentation de la carte.	75
Figure IV.27 : La carte d'acquisition sur la plaque d'essai.....	75
Figure IV.28 : La carte d'acquisition final	76
Figure IV.29 : Résultats de signal sinusoïdal appliqué au chenal 1.....	77
Figure IV.30 : Résultats de signal carré appliquée au chenal 2.	77
Figure IV.31 : Résultats de signal continu de 2V appliquée au chenal 5.....	78
Figure IV.32 : Résultats de signal carré de 3V appliquée au chenal 6.....	78
Figure IV.33 : Résultats de commande de 8 sortie numérique et un sortie PWM1 ~8% . .	79
Figure IV.34 : Commande de 4 sorties numériques et 2 sorties PWM1 ~ 10%, PMW2~ 99%.....	79
Figure IV.35 : Commande de 4 sorties numériques et 1 entrée numérique et 2 sorties.... PWM1 ~ 99% ,PMW ~ 99%.....	80
Figure IV.36 : Commande de entrée numérique voie 4..	80

Liste des tableaux

Tableau. I.1 : Fonctions des broches.....	16
Tableau. II.1 : Fils de liaison les plus courants.....	28
Tableau. III.1 : Les caractéristiques générales de PIC18F4550	42
Tableau. III.2 : Rôle des bits FOSC0 à FOSC3 pour le 18F4550.	47
Tableau. IV.1 : Les paramètres de descripteur.	64

Liste des abréviations

USB : Universal Serial Bus.

UART : Universal Asynchronous Receiver Transmitter.

HID : Human Interface Device (USB HID class).

PCMCIA : Personal Computer Memory Card International Association.

RS232 : Recommended Standard 232.

SSP : Synchronous Serial Port.

PIC : Programmable Interface Controller.

ADC : Analog to Digital Converter.

SPI : Serial Peripheral Interface Bus.

IC : Inter – Integrated Circuit.

MSSP : Master Synchronous Serial Port.

SDA : Serial Data line.

SCL : Serial Clock line.

PC : Personal Computer.

PLL : Phase-locked loop.

ICSP : In-Circuit Serial Programming.

PWM : Pulse-Width Modulation.

ISA : Industry Standard Architecture.

PCI : Peripheral Component Interconnect.

PXI : PCI eXtensions for instrumentation.

IEEE : Electrical and Electronics Engineers.

FPGA : Field Programmable Gate Array.

VHDL : Very High Speed Integrated Circuit Hardware Description Language.

NRZI : Non-Retour à Zéro Inversé.

LabVIEW : Laboratory Virtual Instrument Engineering Workbench.

Mac OS : Macintosh Operating System.

GPIO : General Purpose Interface Bus.

DIP : dual in-line pin package.

TQFP : Thin Quad Flat Pack. Typically 1mm thick.

VISA : Virtual Instrument Software Architecture.

Introduction générale

Introduction générale

Dans les milieux industriels, il est nécessaire d'améliorer et optimiser les processus de production. Ainsi pour atteindre cet objectif on doit surveiller continuellement les différents paramètres du système étudié et analyser les données recueillies.

Un système de surveillance de données dans un milieu industriel doit avoir la possibilité d'enregistrer une large gamme de données aussi de commander manuellement ou automatiquement les applications de travail par un micro-ordinateur. Ce dernier est un plate-forme privilégiée pour la réalisation des applications de commande et d'acquisition, de traitement et de restitution des données, avec la visualisation sur l'écran des résultats expérimentaux.

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 (dans une architecture à haut débit, les bits circulant sur chaque fil arrivent avec de décalages, provoquant des erreurs), les câbles série coûtent beaucoup moins cher que les câbles parallèles.

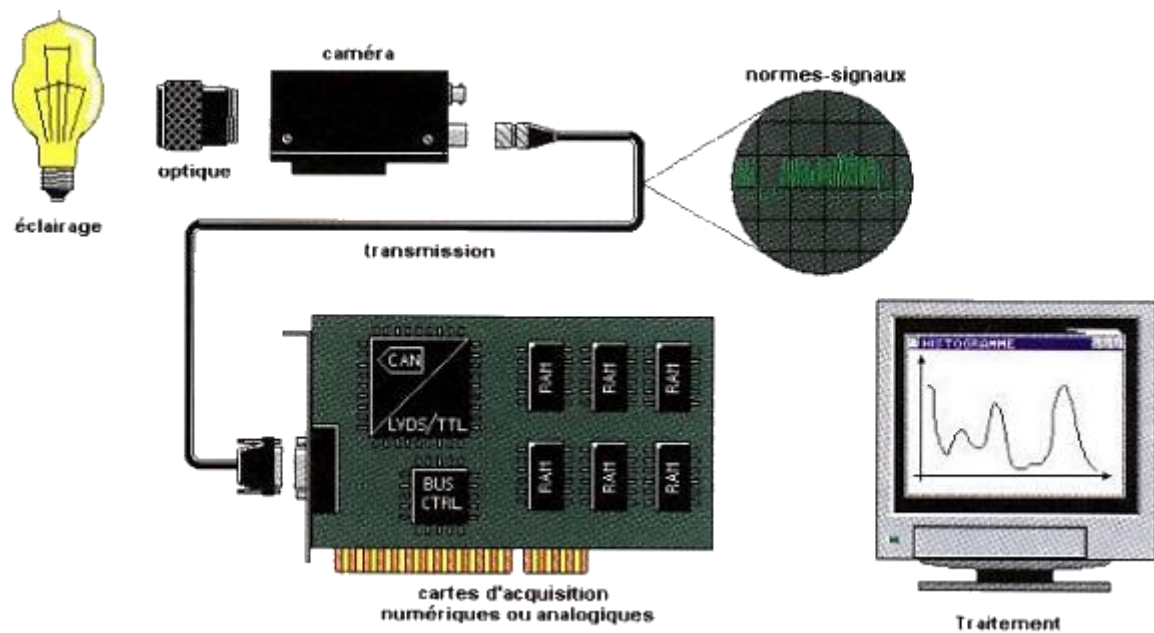
De nos jours d'un part L'USB est très utilisé, et son vitesse de transmission est très rapide par rapport au port série RS232, d'autre part les nouveaux PC ne contiennent pas les ports série et parallèles. Pour cela, l'objectif de notre travail est de réaliser une carte électronique à base de microcontrôleur PIC18F4550 pour faire la liaison entre le PC, avec une interface graphique réalisée par LABVIEW, et le milieu extérieur via le port USB. Cette carte consiste des modules d'acquisition des signaux analogiques (numériques), numérisation, transmission vers l'ordinateur et de restitution des données et les visualisation sous différentes formes.

Pour se faire, nous avons divisé le mémoire en quatre parties. Premièrement, nous avons commencé par une généralité sur l'acquisition de donnée. Ensuite, le deuxième chapitre est axé sur le logiciel de l'interfaçage (LABVIEW), et les instruments virtuels. Le troisième chapitre est axé sur le microcontrôleur PIC18F4550, que nous avons étudié toutes ses caractéristiques et ses descriptions détaillées. La conception et la réalisation pratique de notre carte, fait l'objectif du quatrième chapitre, dans lequel nous présenterons l'environnement de l'application de commande et contrôle, et les divers composants implémentés dans l'architecture de notre système.

Nous finirons ce mémoire par une conclusion générale récapitulative des différentes phases de notre travail, signalant les côtés bénéfiques du projet et énonçant les perspectives du travail élaboré.

Chapitre I

Généralité sur l'acquisition de données



I.1. Introduction

L'acquisition de données consiste à recueillir des signaux de sources de mesure et à les numériser pour les stocker, les analyser et les présenter sur un PC. Les systèmes d'acquisition de données (DAQ) sont disponibles dans de nombreux formats de technologie PC pour vous donner le maximum d'options lorsque vous choisissez votre système. Les ingénieurs et les scientifiques ont le choix entre l'acquisition de données PCI, PXI, PCI Express, PXI Express, PCMCIA, USB, sans file et Ethernet pour leurs applications de test, de mesure et d'automatisation. [1]

I.2. Composition d'un système DAQ

Le système d'acquisition des données est un ensemble d'éléments matériel et logiciel destiné à recueillir des données physiques par l'intermédiaire de capteurs.

Une système d'acquisition est généralement constituée des éléments suivants :

- Les transducteurs et les capteurs
- Les signaux
- Le conditionnement des signaux
- Le matériel DAQ
- Les drivers et logiciels d'application

Une chaîne d'acquisition peut se représenter selon le Schéma bloc suivant : (figure I.1)

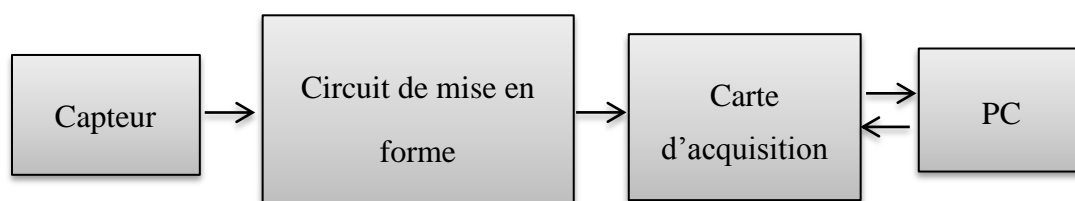


Figure I.1 : Structure d'une chaîne d'acquisition.

I.2.1. Transducteurs (capteurs)

L'acquisition de données commence par le phénomène physique à mesurer. Ce phénomène physique peut être la température d'une pièce, l'intensité d'une source de lumière, la pression à l'intérieur d'une chambre, la force appliquée à un objet, etc... Un système DAQ efficace peut mesurer n'importe lequel de ces phénomènes.

Un transducteur est un appareil qui convertit un phénomène physique en un signal électrique mesurable, comme une tension ou un courant. La capacité d'un système DAQ de mesurer différents phénomènes dépend des transducteurs utilisés pour convertir le phénomène physique en signaux mesurables par le matériel DAQ. Transducteurs et capteurs sont synonymes dans les systèmes DAQ. Il existe des transducteurs spécifiques pour de nombreuses applications, comme les mesures de température, de pression et de débit liquide...etc .[1]

Les spécifications requises pour convertir un phénomène en signal mesurable varient en fonction du transducteur. Certains transducteurs nécessitent un courant ou une tension d'excitation. D'autres requièrent des composants supplémentaires ou des réseaux résistifs pour produire un signal .

La qualité d'une mesure ne dépend pas uniquement du seul capteur, elle est conditionnée par des équipements complémentaires indispensables qui constituent une chaîne de mesure (Préamplificateur, Adaptateur d'impédance, Conditionneur...etc.) ainsi que des éventuels parasites ou perturbateurs qui l'affectent dans son ensemble. [2]

I.2.2. Signaux

Les transducteurs appropriés convertissent des phénomènes physiques en signaux mesurables. Cependant, des signaux différents doivent être mesurés de façons différentes. C'est pourquoi il est important de comprendre quels sont les différents types de signaux et les attributs correspondants. Les signaux peuvent être répartis en deux groupes : [1]

- **Analogiques**
- **Numériques**

I.2.2.1. Signaux analogiques

Un signal analogique peut avoir n'importe quelle valeur en fonction du temps. La tension, la température, la pression, le son et la charge représentent des exemples de signaux analogiques. Les trois principales caractéristiques d'un signal analogique sont le niveau, la forme et la fréquence (figure I.2).

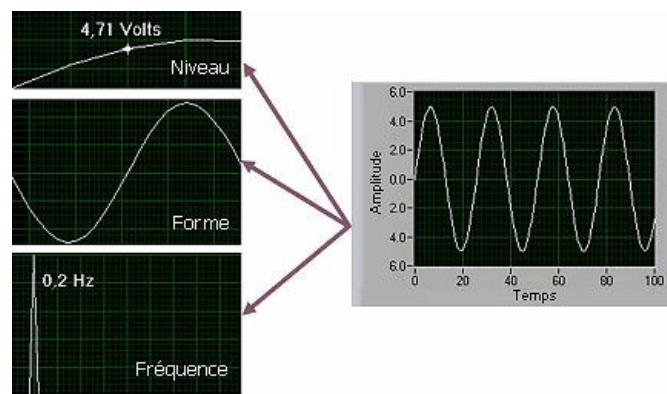


Figure I.2 : Principales caractéristiques d'un signal analogique.

- **Niveau**

Les signaux analogiques pouvant prendre n'importe quelle valeur, le niveau fournit des informations essentielles sur le signal analogique mesuré. L'intensité d'une source de lumière, la température d'une pièce et la pression à l'intérieur d'une chambre sont des exemples qui montrent l'importance du niveau d'un signal. En général, lorsque vous mesurez le niveau d'un signal, ce signal ne change pas rapidement en fonction du temps. Cependant, la précision de la mesure est très importante. Pour effectuer des mesures de niveaux analogiques, vous devez choisir un système DAQ offrant le maximum de précision.

- **Forme**

Certains signaux sont nommés d'après leur forme : sinusoïdal, carré, en dents de scie et triangulaire. La forme d'un signal analogique peut être aussi importante que le niveau car vous pouvez analyser plus précisément le signal en mesurant sa forme et obtenir ainsi les valeurs de pic, les valeurs de CC et la pente. En général, les signaux dont la forme a un intérêt changent rapidement par rapport au temps, mais la précision du système reste importante. L'analyse de battements de cœur, de signaux vidéo, de sons, de vibrations et de réponses de circuit représentent des applications utilisant des mesures de forme.

- **Fréquence**

Tous les signaux analogiques peuvent être catégorisés d'après leur fréquence. Contrairement au niveau et à la forme d'un signal, la fréquence ne peut pas être mesurée directement. Le signal doit être analysé avec un logiciel pour déterminer les informations fréquentielles. Normalement, cette analyse se fait avec un algorithme appelé transformée de Fourier.

Si la fréquence constitue l'information la plus importante, il faut prendre en compte aussi bien la précision que la vitesse d'acquisition. La vitesse d'acquisition nécessaire pour déterminer la fréquence d'un signal est inférieure à celle requise pour obtenir la forme de ce signal ; cependant, le signal doit être acquis à une vitesse suffisante pour que les informations importantes ne soient pas perdues au cours de l'acquisition. Cette vitesse est définie par le théorème d'échantillonnage de Nyquist. L'analyse vocale, les télécommunications et l'analyse sismique représentent des exemples d'applications courantes où il faut connaître la fréquence du signal.

I.2.2.2. Signaux numériques

Un signal numérique ne peut pas avoir n'importe quelle valeur en fonction du temps. Il n'a que deux niveaux possibles : haut et bas. Généralement, un signal numérique doit être conforme à certaines spécifications qui définissent ses caractéristiques. Les signaux numériques sont souvent appelés TTL (Transistor-to-Transistor Logic). Les spécifications TTL stipulent qu'un signal numérique est bas si le niveau se trouve entre 0 et 0,8 V, et haut s'il est entre 2 et 5 V. L'état et la fréquence sont les informations utiles qui peuvent être mesurées pour un signal numérique (figure I.3).

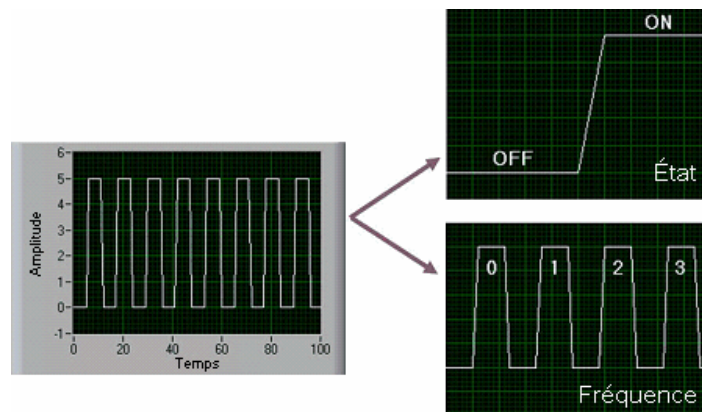


Figure I.3 : Principales caractéristiques d'un signal numérique.

- **État**

Un signal numérique ne peut pas avoir n'importe quelle valeur en fonction du temps. L'état d'un signal numérique est essentiellement le niveau du signal : activé ou désactivé, haut ou bas. La surveillance de l'état d'un commutateur (ouvert ou fermé), est une application courante qui montre l'importance de connaître l'état d'un signal numérique.

- **Fréquence**

La fréquence d'un signal numérique définit comment il change d'état en fonction du temps. La fréquence d'un signal numérique peut par exemple être mesurée en déterminant à quelle vitesse l'axe d'un moteur tourne. Contrairement à la fréquence d'un signal analogique, la fréquence d'un signal numérique mesure à quel intervalle de temps une partie d'un signal se reproduit. Il n'y a pas besoin d'utiliser un algorithme logiciel pour déterminer la fréquence d'un signal numérique.

I.2.3. Conditionnement du signal

Les signaux issus du capteur ne sont pas toujours directement exploitables. Par exemple, les cartes de conversion les plus répandues ont un calibre d'entrée de 0 - 10 volts, avec une résistance de quelques centaines de k Ω . Souvent, il faudra : [1]

- Transformer des courants ou des fréquences en tensions.
- Adapter leur impédance.
- Amplifier les signaux.
- Filtrer pour éliminer des interférences, peut-être aussi linéariser de manière à obtenir un signal proportionnel à la grandeur étudiée.
- Décaler leur origine de manière à exploiter au mieux l'amplitude acceptée par l'organe de mesure électrique.

Lorsque la distance entre le capteur et le système convertisseur analogique-numérique est grande, on a intérêt à transmettre un signal amplifié, à relativement basse impédance, et éventuellement symétrique, pour diminuer l'importance relative des tensions parasites ajoutées le long des câbles de liaison. C'est un autre rôle du conditionnement du signal.

Nous engloberons aussi sous ce terme les techniques nécessaires pour isoler électriquement et amplifier les tensions disponibles cette fois en sortie, de manière à les rendre aptes à

commander les actionneurs des processus externes comme le déclenchement d'un chauffage, le lancement d'un moteur ou l'ouverture d'une électrovanne...etc.[3]

I.2.4. Carte d'acquisition de donnée

Ensemble de circuits électroniques ou interface d'entrée/sortie, destiné à traiter l'information venant des capteurs: échantillonnage, digitalisation, etc.

Le matériel DAQ sert d'interface entre l'ordinateur et le monde extérieur. Sa fonction principale en tant que périphérique consiste à numériser les signaux analogiques entrants pour que l'ordinateur puissent les interpréter. Les autres fonctionnalités d'acquisition de données comprennent : [1]

- Entrées et sorties analogiques.
- Entrées et sorties numériques.
- Compteurs et temporisateur.

Une carte d'acquisition numérique peut se représenter selon la figure (I.4).

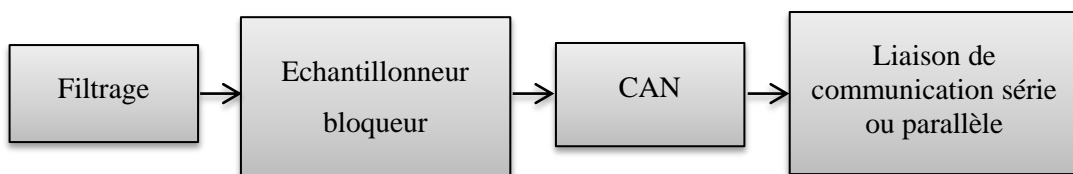


Figure I.4 : Structure d'une carte d'acquisition.

I.3. Structure d'une chaîne d'acquisition numérique

La figure ci-dessous représente une chaîne d'acquisition numérique formée d'un capteur pour la conversion du grandeur physique en grandeur électrique (courant, tension, ..etc) , un amplificateur qui sert a amplifié les faibles signaux ,un système de filtrage pour le débruitage puis la conversion analogique numérique via l'échantillonneur bloqueur.

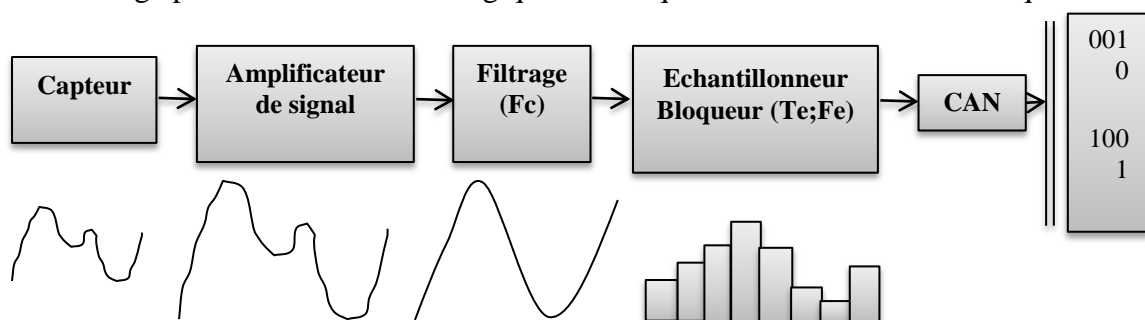


Figure I.5 : Structure de l'acquisition numérique .

I.3.1. Capteur

Il est l'interface entre le monde physique et le monde électrique. Il va délivrer un signal électrique image du phénomène physique que l'on souhaite numériser. Il est toujours associé à un circuit de mise en forme.[4]

I.3.2. Amplificateur de signal

Cette étape permet d'adapter le niveau du signal issu du capteur à la chaîne globale d'acquisition.

I.3.3. Filtre d'entrée

Ce filtre est communément appelé filtre anti-repliement. Son rôle est de limiter le contenu spectral du signal aux fréquences qui nous intéressent. Ainsi il élimine les parasites. C'est un filtre passe bas que l'on caractérise par sa fréquence de coupure et son ordre.

I.3.4. Le convertisseur analogique numérique (CAN)

I.3.4.1. Identification de la fonction C.A.N

On appelle Convertisseur Analogique Numérique (C.A.N.) tout dispositif électronique qui transforme une grandeur analogique d'entrée u_e en un nombre binaire de sortie N proportionnel à cette grandeur u_e . En anglais, le Convertisseur Analogique Numérique est appelé Analogic Digital Conversion (A.D.C.). [5]

I.3.4.2. Symbole d'un C.A.N

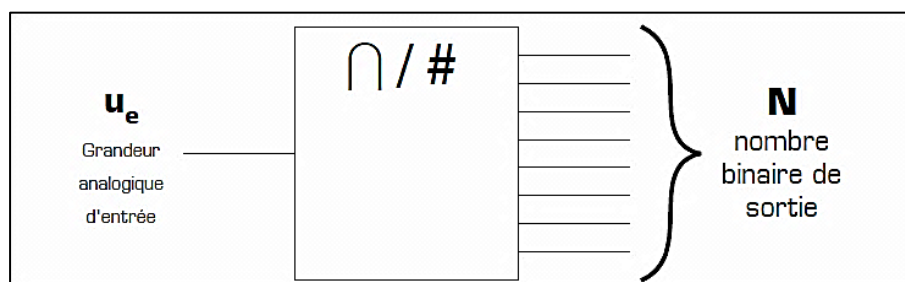


Figure I.6 : Symbole de la fonction Conversion Analogique / Numérique.

- Le signe \cap indique que la grandeur est de type analogique (il s'agit ici de l'entrée).
- Le signe $\#$ indique que la grandeur est de type numérique (il s'agit ici de la sortie).

I.4. La fonctionnement d'un CAN

L'obtention d'un signal numérique à partir d'un signal analogique (exemple : microphone), nécessite de mettre en œuvre une chaîne de conversion analogique numérique. Le convertisseur analogique-numérique a pour fonction de faire correspondre à un signal analogique un signal numérique de sortie.

Cette chaîne comporte typiquement la structure fonctionnelle suivante :

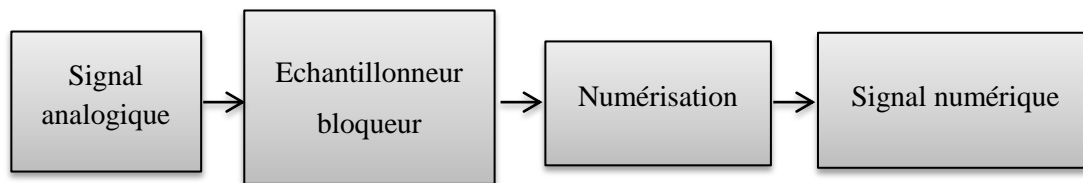


Figure I.7 : Principe du chaîne de conversion analogique-numérique.

Les fonctions réalisées par chaque élément sont les suivantes :

- **Echantillonneur** : Acquérir la grandeur analogique à un instant t .
- **Bloqueur** : Maintenir cette grandeur pendant la durée de la numérisation.
- **Numérisation** : Convertir le signal analogique bloqué en un nombre (sur N bits).

I.4.1. Echantillonnage

L'échantillonnage d'un signal de fréquence f doit se faire avec une fréquence d'échantillonnage f_e supérieure à $2f$. Le non suivi de cette règle essentielle entraîne pratiquement à coup sûr des résultats aberrants (figure I.8).

Un choix judicieux de cette fréquence est imposé par l'application du théorème de Shannon.

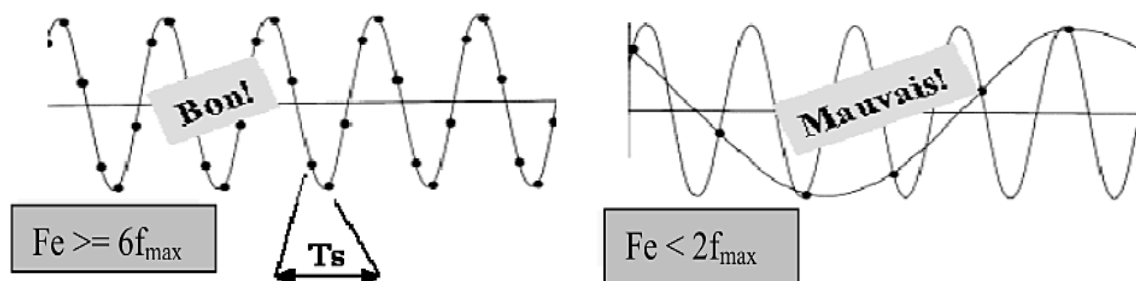


Figure I.8 : Théorème de Shannon.

I.4.2. Bloqueur

Le bloqueur a pour fonction de maintenir constant le signal échantillonné $SE(t)$ afin de permettre au convertisseur analogique numérique de le numériser. Ce signal est maintenu constant à la valeur $SB(t)$ jusqu'à l'échantillonnage de la valeur suivante.

Actuellement les bloqueurs agissent comme une mémoire qui garde constante la valeur échantillonnée ($SB(t) = \text{constante}$), on parle alors de bloqueur d'ordre 0. On trouve aussi des bloqueurs d'ordre supérieur (1, 2..), lorsque l'évolution du signal $SB(t)$ est assimilable à des décroissances exponentielles ou paraboliques.

I.4.3. Numérisation

I.4.3.1. Principe

La numérisation est la transformation d'un signal analogique issu du bloqueur en un signal numérique. Elle est réalisée par un convertisseur analogie numérique (en anglais : ADC) qui convertit le signal bloqué en un nombre. La sortie du convertisseur est raccordée ensuite à un système numérique (ordinateur, analyseur de spectre numérique...etc.)

Cette numérisation réalise la quantification du signal analogique bloqué en lui associant un nombre. Ce nombre est compris entre **0** et 2^{N-1} valeurs possibles où **N** est le nombre de bits en sortie du convertisseur. Ainsi pour **8** bits, on a 2^8 valeurs différentes possibles en sortie soit **256** valeurs différentes. Les nombres associés vont alors de la valeur **0** jusqu'à **255**.

I.4.3.2. Résolution

La résolution du convertisseur, notée r , est l'intervalle de tension analogique permettant de passer du nombre n à $(n+1)$. Pour une tension analogique d'entrée allant de 0 V à une amplitude V_{MAX} , on a $r = \frac{V_{MAX}}{2^N - 1}$. Si cette tension d'entrée varie de 0 à 10 V et que l'on utilise un convertisseur analogique numérique de 8 bits, on a une résolution de 39,2 mV. En d'autres termes, pour que la valeur numérique en sortie du convertisseur varie, il faut la tension d'entrée ait varié de 39,2 mV.

I.5. Acquisition de plusieurs grandeurs

Dans le cadre d'une chaîne d'acquisition traitant plusieurs capteurs (N) vers une même zone de stockage, il existe différentes structures qui diffèrent en terme de performances et de coût. [4]

N Capteurs \Rightarrow 1 zone de stockage (traitement) numérique.

I.5.1. Acquisition séquentielle décalée

Elle se base sur l'utilisation en amont d'un multiplexeur qui va orienter un capteur vers la chaîne unique d'acquisition.

I.5.1.1. Le multiplexage

Le multiplexage est une technique permettant de scruter successivement plusieurs voies d'entrée dans un système d'acquisition. Bien entendu, cela allonge d'autant l'intervalle de temps qui s'écoule entre deux mesures sur un même canal. Le multiplexage peut consister simplement en une commutation sur l'une ou l'autre voie par l'intermédiaire de relais électromagnétiques pilotés par le PC lui-même, mais beaucoup de dispositifs de mesure comportent en entrée un système de multiplexage réalisé entièrement en semi-conducteurs, et bien entendu programmable. Lorsque l'on désire une acquisition plus rapide, ou une mesure quasi-simultanée sur plusieurs voies, on préfère consacrer un convertisseur analogique-numérique à chaque voie.[6]

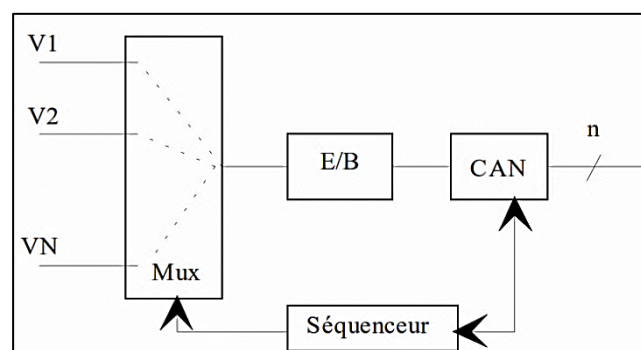


Figure I.9 : Structure séquentielle décalée .

L'avantage de cette structure est bien évidemment son côté économique. Par contre il y a un décalage dans le temps des acquisitions. On réservera donc cette structure ne nécessitant pas une synchronisation entre les données numérisées. De plus le temps d'acquisition complet est à priori élevé car proportionnel au nombre de capteur.[4]

I.5.2. Acquisition séquentielle simultanée

De manière à avoir des acquisitions « synchrones », on utilise la même structure que précédemment mais en utilisant des Echantillonneurs Bloqueurs (E/B) en amont du multiplexeur. On est dans une situation d'E/B en tête.

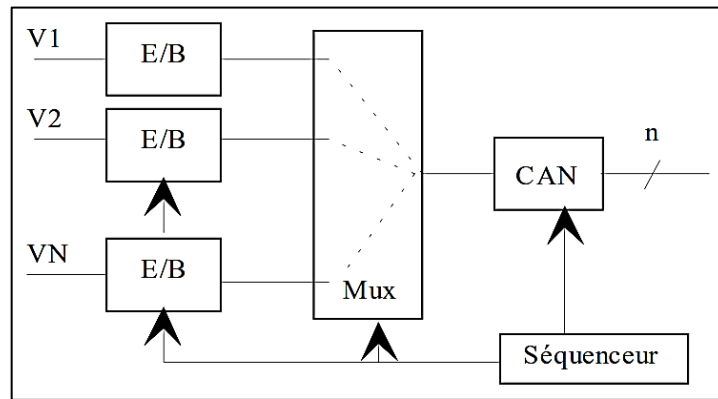


Figure I.10 : Structure séquentielle simultanée.

La prise des échantillons s'effectue au même instant, la conversion est effectuée de manière progressive. Cela signifie que les E/B assurent un maintien de l'échantillon durant les N acquisitions sans introduire de pertes supérieures à la résolution du CAN. Son coût est moyen.

I.5.3. Acquisition parallèle

C'est la structure la plus complète puisqu'elle consiste à disposer N chaînes d'acquisition en parallèle et de les connecter sur un bus de données commun.

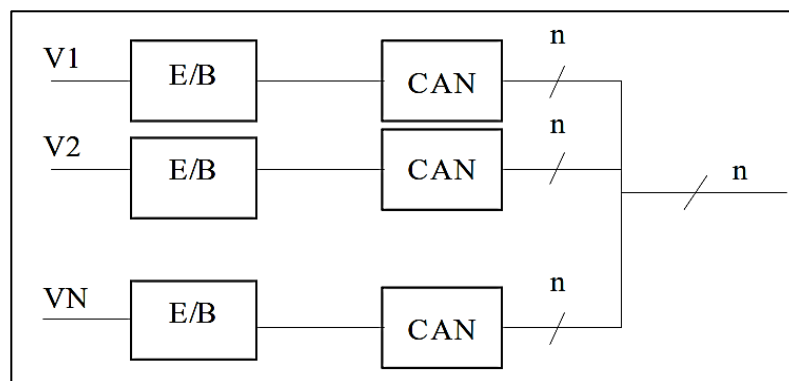


Figure I.11 : Structure parallèle.

Avec cette structure, il est possible d'effectuer en même temps l'acquisition d'une donnée pendant que l'on en stocke une autre. De même, toutes les conversions peuvent être

simultanées, le stockage s'effectuant après. Cela permet un gain de temps sur l'acquisition complète. Mais elle est coûteuse.[4]

I.6. Le Protocol de communication USB

I.6.1. Naissance de L'USB

L'USB (Universal Serial Bus), en français (Bus universel en série), est un protocole de communication série. Le bus USB est donc réellement né de l'alliance en 1994 de sept partenaires industriels (Compaq, DEC, IBM, Intel, Microsoft, NEC et Northern Telecom). C'est eux qui ont commencé à créer la norme USB. Le bus USB a été conçu à l'origine pour faciliter les transferts de données en particulier définir une connectique « universelle » et « Plug & Play », utilisable aussi bien pour une souris que pour un modem ou un moniteur. Conçu également pour répondre au besoin d'intégration entre le monde du PC et celui du téléphone et enfin pour répondre au besoin d'extensions multiples en dehors du PC, en effet on peut brancher jusqu'à 127 périphériques en théorie... cela change de la liaison série habituelle.[7]

I.6.2. Vitesse de transfert de l'USB

L'USB supporte 3 vitesses (Dubé théorique) :

- **Low Speed** à 1.5Mbit/s – (USB 1.1).
- **Full Speed** à 12Mbit/ s – (USB 1.1).
- **High Speed** à 480Mbit/ s – (USB 2.0).

I.6.3. Débits réels

Pour du Low Speed le débit réel est de 800oct/s en mode Interruption. En effet la norme permet 1 interruption toutes les 10ms, dans le meilleur des cas (réglable de 10 à 255ms) donc 100 interruptions/s et comme la taille du paquet est de 8 octets pour cette vitesse, le débit est de 800octets/s

Pour du Full Speed, en mode Interruption la norme permet une interruption au mieux toutes les 1ms (réglable de 10 à 255ms), c'est à dire 1000 interruptions/s, la trame pouvant s'étendre à 64 octets on obtient un débit de 64Ko/s. En mode Bulk on arrive à un débit réel d'environ 1Mo/s. La bande passante est divisée dans ce cas s'il y a plusieurs périphérique qui travail simultanément. [7]

Et pour le High Speed le débit réel de transfert est de 53Mbit/s.

I.6.4. Le 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 Vcc. 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 .

I.6.4.1. Composition du câble USB

Chaque connecteur dispose de deux fils d'alimentation (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» La spécification USB impose une tension maximale de 5V sur les fils et un courant de 500 mA au plus. Pour les 2 fils de données, la tension différentielle entre les deux peut valoir 0, 3,3 ou 5V.



Figure I.12 : Symbole du bus USB.

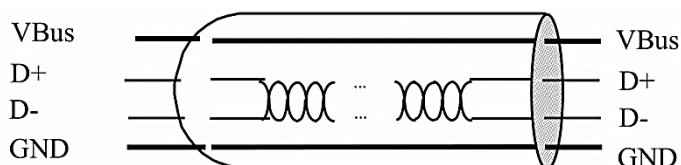


Figure I.13 : Composition d'un câble USB.

La longueur maximale autorisée par la norme est de 3m pour un câble non blindé donc généralement pour un périphérique Low Speed (= 1.5Mb/s) et de 5m pour un câble blindé dans le cas d'un périphérique Full Speed (=12Mb/s).

Le câble USB est composé de deux fiches bien différentes :

En amont d'une fiche appelé connecteur USB de type A, branché au host (PC). L'extrémité aval par contre peut se retrouver en deux versions :

Connecteur USB du type B et un mini connecteur type B (appelé souvent optionnel). Ce dernier est réservé aux dispositifs de très faible dimension (ou de grande intégration), tels les appareils photo numériques.

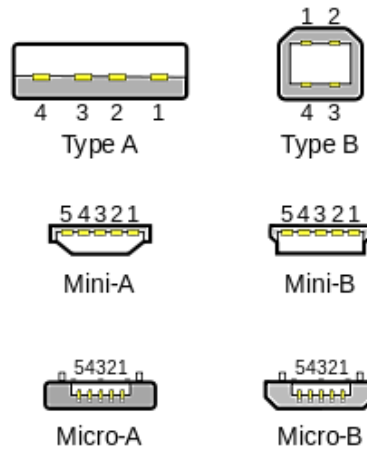


Figure I.14 : Brochage des connecteurs USB de type A et B.

N° de broche	Fonction	Couleur du câble
1	Alimentation	Rouge
2	Donnée D-	Blanc
3	Donnée D+	Vert
4	Masse	Noir

Tableau I.1 : Fonctions des broches.

Et voici l'aspect des connecteurs en trois dimensions :



Figure I.15 : Aspect des connecteurs USB de type A et B.

I.6.5. Codage de données

Les informations qui transitent sur le bus USB sont encodées selon le format NRZI (Non-Retour à Zéro Inversé), voir la figure (I.16), codage qui encapsule les données de la manière suivante : un «1» logique en donnée ne provoque pas de changement d'état en codage NRZI, un « 0 » engendre quant à lui un changement d'état (de 1 vers 0 ou vice versa).

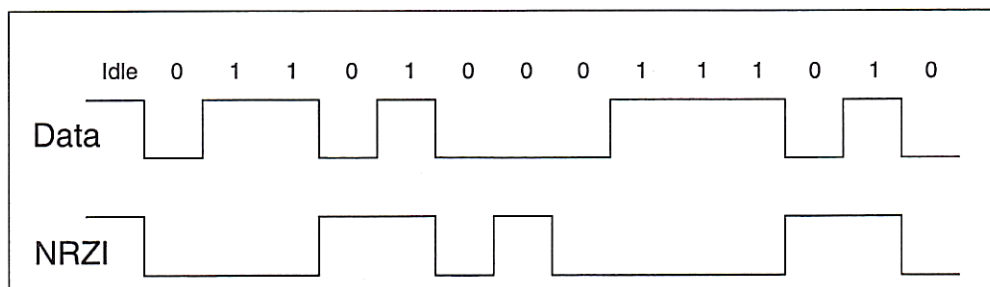


Figure I.16 : Principe du Codage NRZI.

I.6.6. Mode de transfert sur le bus USB

I.6.6.1. 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 standard. Le mode contrôle est aussi utilisé pour tenter d'obtenir un débit Low Speed acceptable, ou pour utiliser le driver de classe HID standard. [7]

I.6.6.2. Transfert en mode Interruption

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.

I.6.6.3. 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.

I.6.6.4. 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.

I.6.7. Choix « Low » ou « Full » USB

Pour faire le choix entre la version USB Low Speed et Full Speed, il suffit de placer une résistance de tirage sur l'interface d'entrée (Fonction ou Hub). Cette résistance de tirage (Pull-Up de 1.5Kohm) est placée soit sur D- dans le cas du Low Speed ou sur D+ dans le cas du Full Speed.

On voit sur le schéma ci-dessous que s'il n'y a pas de résistance de Pull up, les deux câbles de données sont tirés à la masse par les deux résistances de Pull down. Les résistances Pull down étant de 15kOhm et les résistances de Pull up de 1.5kOhm (en fonction de la vitesse choisie), il y aura toujours une ligne de donnée qui sera à 90% de Vcc. Si le HUB détecte qu'une ligne de donnée avoisine les 90% de Vcc il en conclura qu'un composant USB est connecté ainsi que des informations sur la vitesse du composant, Low ou Full Speed. L'énumération peut commencer, c'est parti pour la grande aventure de l'USB.

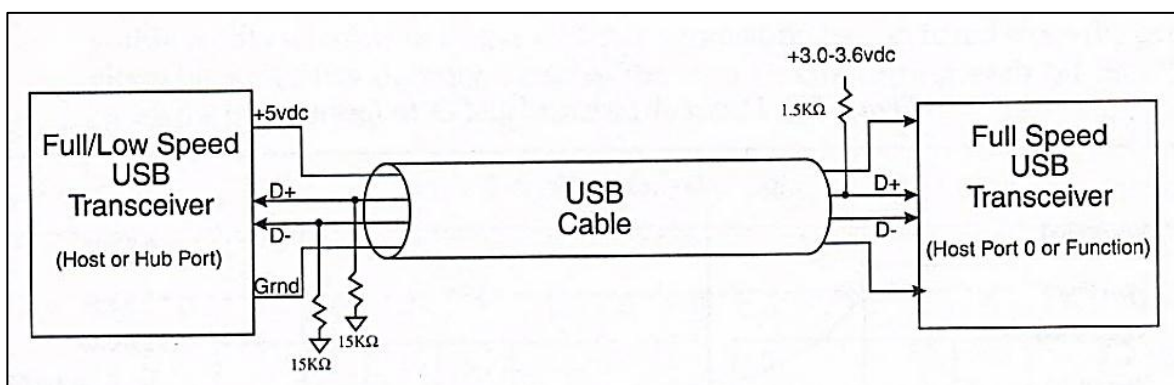


Figure I.17: Choix Full Speed – Résistance de Pull Up.

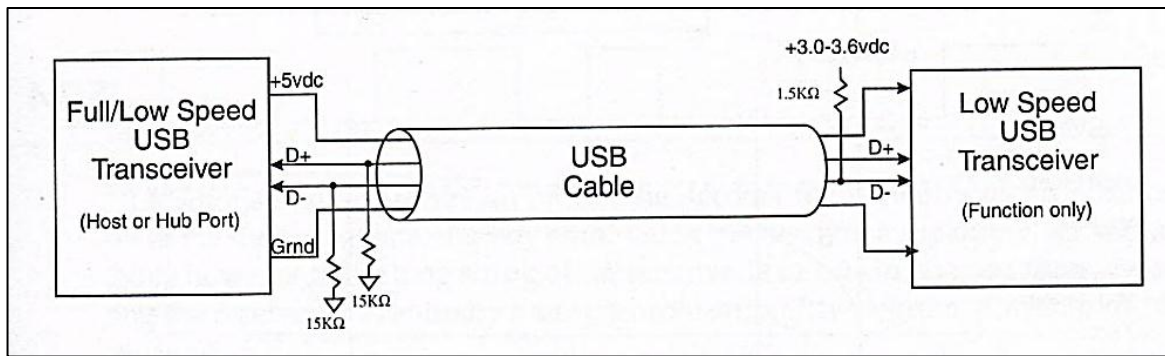


Figure I.18 : Choix Low Speed – Résistance de Pull Up.

I.6.8. Avantages de l'USB

Les avantages de l'USB sont nombreux : faible coût de l'interface, alimentation possible des dispositifs via le câble, indépendance vis à vis des machines hôtes, Hot Plug & Play (c'est à dire branchement et débranchement sans avoir besoin d'arrêter le PC), jusqu'à 127 périphériques possibles, fiabilité et sécurité (détection et correction d'erreurs), plusieurs vitesses possibles et 4 types de transferts.

Un autre avantage du bus USB est la connectique. En effet les câbles possèdent deux extrémités bien différentes pour ne pas se tromper lors du branchement, de plus les câbles USB sont très compacts comparés aux câbles pour les liaisons parallèles ou séries .

I.6.8.1. Avantage de l'alimentation USB

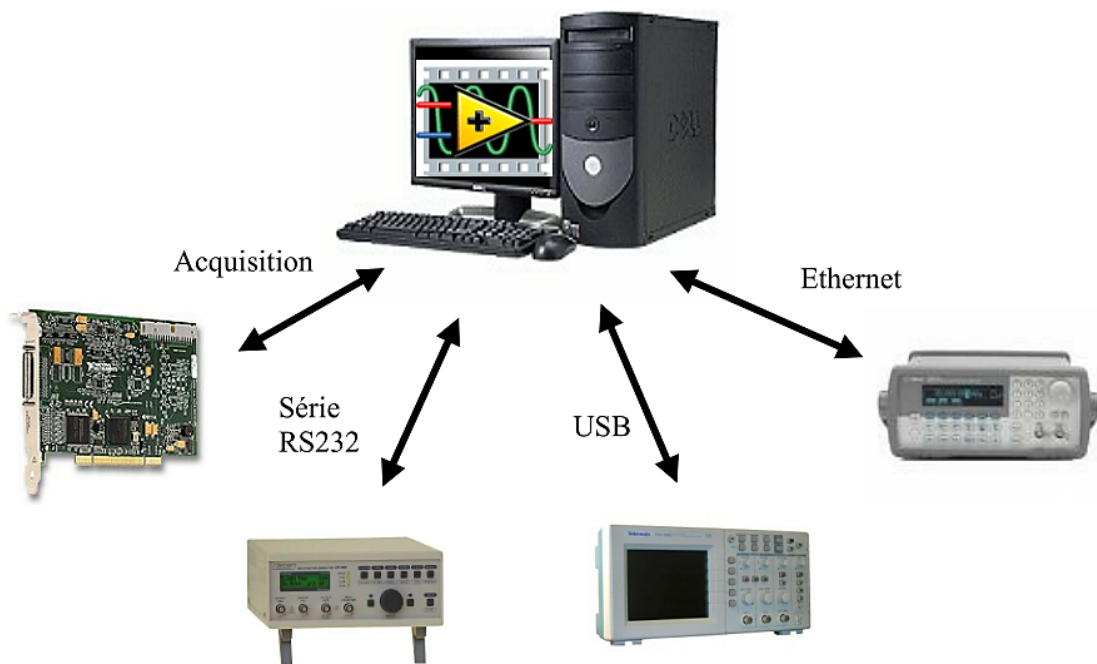
Le fait de pouvoir alimenter un périphérique USB avec le même câble qui transporte les données à de multiples avantages. D'une part cela évite déjà à l'utilisateur de brancher le périphérique sur une prise de courant extérieure, ceci rend déjà le périphérique plus léger et moins encombrant. D'autre part d'un point de vue du concepteur, cela réduit le prix de fabrication.

I.7. Conclusion

Ce chapitre a été consacré essentiellement à la description de manière plus appropriée du système d'acquisition utilisée dans notre projet de fin d'étude et des données échangées ainsi que les phénomènes intervenant dans la partie mise en forme comme le convertisseur analogique numérique et l'échantillonnage...etc. Dans le chapitre suivant nous allons voir plus en détail le logiciel d'interfaçage utilisée.

Chapitre II

Généralité sur le logiciel de l'interfaçage (LABVIEW™)



II.1. Introduction

LabVIEW, contraction de **L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench, est un environnement de développement complet, graphique, compilé et particulièrement bien adapté au domaine de l'acquisition et de la mesure. Son approche totalement graphique offre une souplesse et une dimension intuitive inégalée. Comparativement aux langages textuels il offre la même puissance de programmation mais sans le côté abstrait et complexe lié à la syntaxe. Orienté tests et mesures, il dispose de nombreuses fonctions permettant de piloter facilement des cartes d'acquisition et autres instruments, mais aussi de filtrer, d'analyser et de présenter les données. Ce langage est également appelé code G. Le code est représenté par un schéma composé de fonctions, de structures et de fils qui propagent les données. L'approche visuelle, l'interface entre le code et le développeur s'opposent sur la forme et la philosophie aux langages textuels, comme le C. LabVIEW est redoutable de puissance et n'a rien à envier aux autres langages. [8]

Une des différences fondamentales de LabVIEW est que ce langage suit un modèle de flux de données, et non de flux d'instructions. Cela signifie que pour un langage textuel ce sont les instructions qui ont la priorité, alors qu'avec LabVIEW ce sont les données. Une fonction s'exécutera donc uniquement si elle dispose à ses entrées de toutes les données dont elle a besoin. Lorsqu'un langage classique est ainsi séquentiel, LabVIEW est naturellement prédisposé au parallélisme. Ce qui augmente encore sa puissance et la rapidité d'exécution du code.[8]

Ce langage s'adresse à tous les techniciens ou ingénieurs qui désirent développer une application de mesures, de tests, de supervision ou de contrôle/commande avec une interface utilisateur de très grande interactivité. Mais plus généralement à tous les développeurs qui désirent utiliser un langage intuitif, puissant et ouvert sur la totalité du génie logiciel.

II.1.2. Historique

LabVIEW, a été développé par la société National Instruments à partir de 1983. Initialement conçu pour une plate-forme Mac OS dans la première version distribuée en 1986, le langage devient compilé en 1990. L'environnement de programmation LabVIEW est porté sur plate-forme Windows dès 1992.

La version 3.1 en 1994 intègre le composant « Application Builder », qui permet de générer un exécutable autonome, à partir du code source. En 1998 la version 5 intègre la gestion multitâche, des assistants logiciels pour les cartes d'acquisition (DAQ), et le contrôle d'instruments, le portage sous Linux et le module pour cible Temps Réel (RT). La version 7.1 de 2004 marquera de grandes évolutions avec les modules pour cible FPGA et Windows CE, la structure événementielle et la traduction de la plate-forme de développement en français. La version 8 introduit la gestion par projet, les XContrôles et surtout la programmation objet (OOP). À partir de 2009, National Instruments décide de générer une version de LabVIEW chaque année et dont le nom sera celui de l'année. Dans sa version 2014, le logiciel est actuellement disponible sur différents systèmes d'exploitation : Windows, Mac OS, et Linux. La portabilité des applications développées sous LabVIEW est totale entre les différentes plates-formes. LabVIEW peut générer du code compilé sur ces systèmes d'exploitation mais également pour des cibles Temps Réel (RT), FPGA et embarquées. Possédant une aide en ligne très complète, il est distribué sous forme de DVD-ROM, ou peut être téléchargé. [8]

II.2. LabVIEW dans le monde de l'instrumentation

LabVIEW est un des premiers langages de programmation graphique destiné au développement d'applications d'instrumentation. Un logiciel d'instrumentation pourrait être défini comme un programme permettant de contrôler depuis un ordinateur, un système allant du capteur à la chaîne d'acquisition ou de commande jusqu'à l'édition du rapport final. Couplé à des cartes d'entrées/sorties, il permet de gérer des flux d'informations numériques ou analogiques et de créer ou de simuler des instruments de mesures (oscilloscope, compteur, multimètre, etc.). Le temps nécessaire à l'assemblage des composants de ce type de système est négligeable par rapport à celui nécessaire à sa programmation dans un langage classique (C, Pascal, Ada, etc.).[9]

Les interfaces utilisateurs développées avec ces langages sont le plus souvent obscures et incompréhensibles. Les utilisateurs disposent avec LabVIEW d'un puissant outil intégré d'acquisition, d'analyse et de présentation des données, une solution qui entraîne un gain notable de productivité comparable à celui obtenu par l'introduction des tableurs dans les logiciels financiers. Pour cela, le langage utilise toute la convivialité des interfaces interactives des ordinateurs actuels en fournissant des objets proches graphiquement des objets réels (voyants, curseurs, interrupteurs, boutons, graphes, etc.)

mais aussi des commandes systèmes (pour une représentation plus standard) ainsi que des outils familiers et simples d'utilisation pour la programmation (structures de programmation, fonctions arithmétiques, fonctions logiques, comparateurs, etc.). [8]

Pour l'acquisition de données et le contrôle d'instruments, LabVIEW supporte les standards RS-232/422, USB, IEEE488 (GPIB) et VXI/PXI, ainsi que les cartes d'acquisition de données. Pour l'analyse et le traitement des données, la bibliothèque d'analyse étendue contient les fonctions pour la génération et le traitement de signaux, les filtres, les fenêtres, les statistiques, la régression, l'algèbre linéaire et l'arithmétique matricielle. LabView intègre un grand nombre d'éléments de présentation tels les graphes déroulants, des graphes XY, des abaques de Smith, jauges, cadrans à aiguille... [10]

Dans le langage de programmation du LabVIEW, les lignes de codes sont remplacées par des cases de différentes représentations appelées des VI pour « Virtual Instrument ». [9]

II.2.1. Les instruments virtuels

Les programmes LabVIEW sont appelés instruments virtuels, ou VIs, car leur apparence et leur fonctionnement s'apparentent aux instruments réels, tels que les oscilloscopes et les multimètres. LabVIEW contient une grande gamme d'outils pour l'acquisition, l'analyse, l'affichage et l'enregistrement des données, ainsi que des outils pour vous aider à mettre au point votre programme. [11]

Dans LabVIEW, vous construisez une interface utilisateur, ou face-avant, avec des commandes et des indicateurs. Les commandes sont des boutons rotatifs, des bouton-poussoir, des cadrans et autres mécanismes d'entrée. Les indicateurs sont des graphes, des LED et d'autres afficheurs de sortie. (figure II.1), Après avoir construit l'interface utilisateur, vous ajoutez du code en utilisant des VIs et des structures pour contrôler les objets de la face-avant. Le diagramme contient ce code. [11]

Vous pouvez utiliser LabVIEW pour communiquer avec le matériel comme, par exemple, les périphériques d'acquisition de données ou d'images, et de commande d'axes et les instruments RS-232/422, USB, IEEE488, (GPIB) et VXI/PXI.

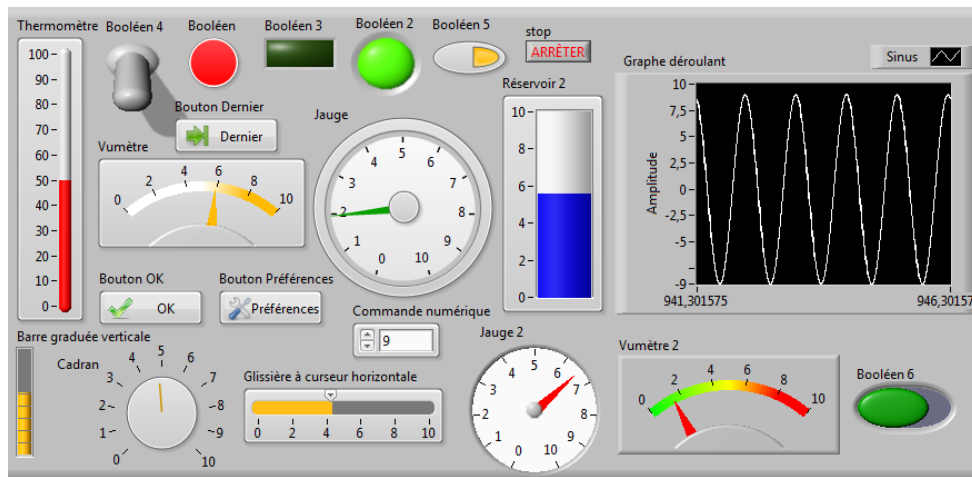



Figure II.1: Les instruments virtuels (VIs).

Chaque objet déposé sur la face avant génère dans le diagramme un symbole appelé Terminal. Ce terminal contient la valeur de l'objet graphique correspondant. Le symbole représente le type de la donnée (par ex. DBL pour double),  le sens du flot, (maigre avec une flèche à droite s'il s'agit d'un indicateur, gras avec une flèche à gauche s'il s'agit d'une commande), Le diagramme contient les terminaux, les sous VIs, les fonctions, les constantes, les structures ainsi que les fils qui relient les différents objets pour leur transmettre les données. [10]

Les programmes LabVIEW portent la dénomination d'instruments virtuels ou VIs, car leur apparence et leur fonctionnement imitent ceux d'instruments réels tels que les oscilloscopes et les multimètres. Chaque VI utilise des fonctions qui manipulent les entrées de l'interface utilisateur ou d'autres sources et qui affichent ces informations ou les déplacent vers d'autres fichiers ou ordinateurs. [12]

Un VI contient les trois composantes suivantes :

- ✓ **Face-avant** : Sert d'interface utilisateur. (figure II.2).
- ✓ **Diagramme** : Contient le code source graphique qui définit les fonctionnalités du VI.
- ✓ **Icône et connecteur** : Identifie l'interface au VI pour que vous (Figure II.3).

puissiez utiliser celui-ci dans un autre VI. Un VI à l'intérieur d'un autre VI est appelé sous-VI. Un sous-VI correspond à un sous-programme dans des langages de programmation textuels.

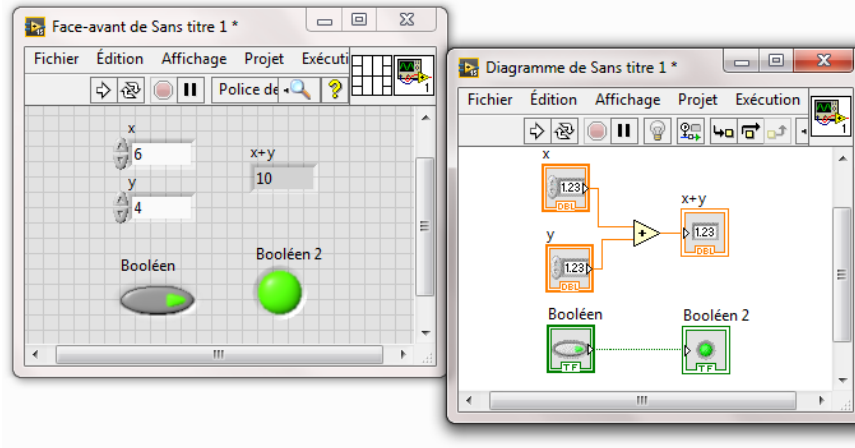


Figure II.2 : La face avant et son diagramme.



Figure II-3. L'icône et son connecteur

II.3. Environnement LABVIEW

Elle permet les actions suivantes :

Dans la zone New il est possible d'ouvrir un nouveau VI ou un nouveau projet. L'option More... permet de créer d'autres types d'objets LabVIEW (des contrôles personnalisés, des variables globales, des VIs polymorphes, des VI pré-remplis à partir de modèles...). La zone Open permet d'ouvrir un VI existant. La zone de droite donne accès à l'aide et aux exemples ainsi qu'aux différentes ressources d'aide disponibles sur la toile.

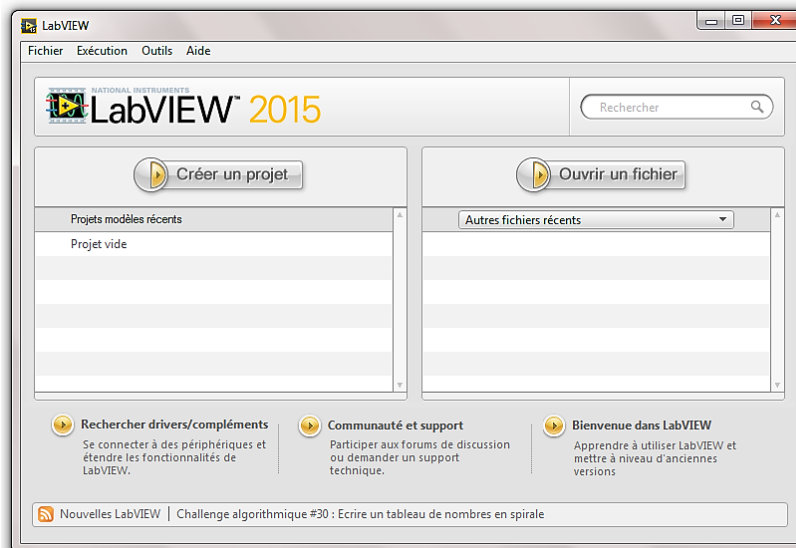


Figure II.4 : Fenêtre de démarrage.

II.3.1. Les fenêtres de la face avant du diagramme

Lorsque l'on ouvre un nouveau VI, une fenêtre de face avant et une fenêtre de diagramme apparaissent.

L'illustration suivante présente les principaux éléments de ces deux fenêtres.

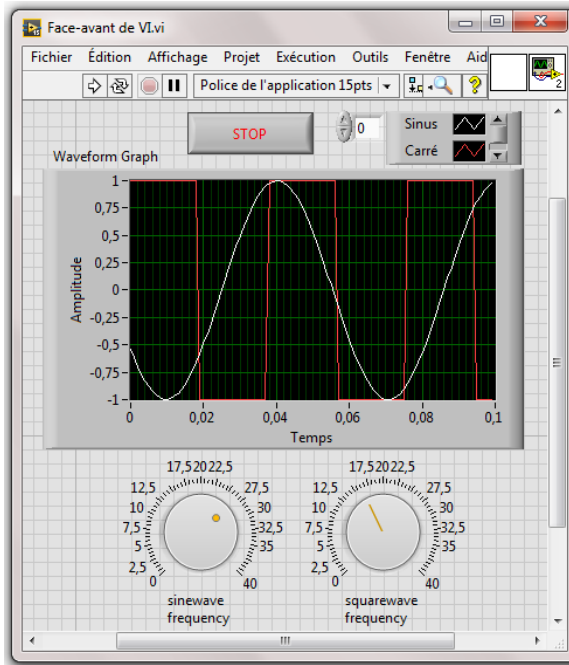


Figure II.5.a : La face avant.

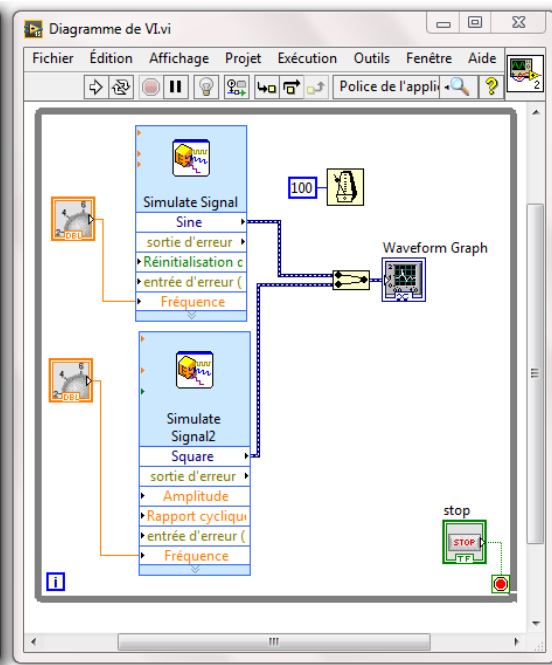


Figure II.5.b : Le diagramme.



II.3.2. Face-avant



La face-avant est l'interface utilisateur du VI, La figure (II .5.a), montre un exemple de face-avant.

Vous construisez la face-avant avec des commandes et des indicateurs qui sont respectivement les terminaux d'entrée et de sortie interactifs du VI. Les commandes sont des boutons rotatifs, des bouton-poussoir, des cadrans et autres mécanismes d'entrée. Les indicateurs sont des graphes, des LED et d'autres afficheurs de sortie. Les commandes simulent les mécanismes d'entrée des instruments et fournissent des données au diagramme du VI. Les indicateurs simulent les mécanismes de sortie d'instruments et affichent les données que le diagramme acquiert ou génère. [12]

Après avoir construit la face-avant, vous devez ajouter le code en utilisant les représentations graphiques des fonctions pour commander les objets de la face-avant. Le diagramme contient ce code source graphique, aussi appelé code G ou code du diagramme. Les objets de la face-avant apparaissent en tant que terminaux sur le diagramme.

II.3.3. Terminaux sur le diagramme

Les terminaux représentent le type de données de la commande ou de l'indicateur. Vous pouvez configurer des commandes ou des indicateurs de la face-avant pour qu'ils s'affichent sous forme d'icônes ou de terminaux de type de données sur le diagramme. Par défaut, les objets de la face-avant sont représentés par des terminaux icônes. Par exemple, un terminal icône de bouton rotatif, représenté dans la figure (II.5.b), représente un bouton rotatif sur la face-avant comme le terminal  → 

La mention **DBL** au bas du terminal représente un type de données numérique à virgule flottante double précision. Un terminal **DBL**, illustré ci-dessous, représente une commande numérique à virgule flottante double précision  ou 

II.3.4. Icône et connecteur

Une fois que vous avez construit la face-avant et le diagramme d'un VI, créez son icône et son connecteur pour pouvoir l'utiliser en tant que sous-VI. L'icône et le connecteur correspondent au prototype de fonction des langages de programmation textuels. Chaque VI affiche une icône, comme celle qui est représentée ci-dessous, dans le coin supérieur droit des fenêtres de la face-avant et du diagramme. [12]



Figure II.6 : L'icône de VI.

Une icône est la représentation graphique d'un VI. Elle peut contenir du texte, des images ou les deux. Si vous utilisez un VI comme sous-VI, l'icône identifie le sous-VI sur le diagramme du VI. Vous pouvez double-cliquer sur l'icône pour la personnaliser ou la modifier.

Vous devez également construire un connecteur, représenté ci-dessous, pour utiliser le VI comme sous-VI.

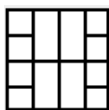


Figure II.7 : Connecteur de VI.

Le connecteur est un groupe de terminaux qui correspond aux commandes et aux indicateurs de ce VI; ce groupe est semblable à la liste de paramètres d'un appel de fonction dans les langages de programmation textuels. Le connecteur définit les entrées et sorties que vous pouvez connecter au VI que vous voulez utiliser comme sous-VI. Un connecteur reçoit des données sur ses terminaux d'entrée et transmet ces données au diagramme par les commandes de sa face-avant. Il reçoit les résultats sur ses terminaux de sortie par les indicateurs de sa face-avant.

II.4. Fils de liaison

Le transfert des données entre les objets d'un diagramme s'effectue par les fils de liaison. Dans la figure précédente, les fils de liaison connectent les terminaux des commandes et des indicateurs aux fonctions Additionner et Soustraire. Chaque fil de liaison a une source de données unique, mais vous pouvez le câbler à de nombreux VIs et fonctions qui lisent les données. Les fils de liaison sont de couleurs, de styles et d'épaisseurs différents en fonction de leur type de données. Un fil de liaison brisé apparaît sous la forme d'une ligne noire en pointillés avec un symbole X rouge au milieu. Les fils de liaison brisés apparaissent pour de multiples raisons, notamment lorsque vous essayez de câbler deux objets dont les types de données sont incompatibles. [12]










Types de fils de liaison	Scalaire	Tableau 1D	Tableau 2D	Couleur
Numérique				Orange (virgule flottante), Bleu (entier)
Booléen				Vert
Chaîne				Rose

Tableau II.1 : Fils de liaison les plus courants .

II.4.1. Nœuds

Les nœuds sont des objets du diagramme qui comportent des entrées et/ou des sorties et qui réalisent des opérations pendant l'exécution du VI. Ils sont analogues aux déclarations, opérateurs, fonctions et sous-programmes écrits dans des langages de

programmation textuels. Les fonctions Additionner et Soustraire de la figure précédente correspondent à des nœuds.

LABVIEW propose des types de données de base (figure II.8), Chaque type est repéré à l'aide de couleur différente. Les arcs d'un type donné ont la couleur de ce type, leur représentation peut varier d'un type à l'autre.

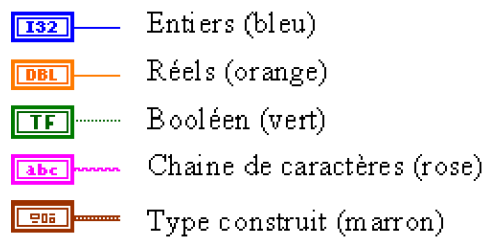
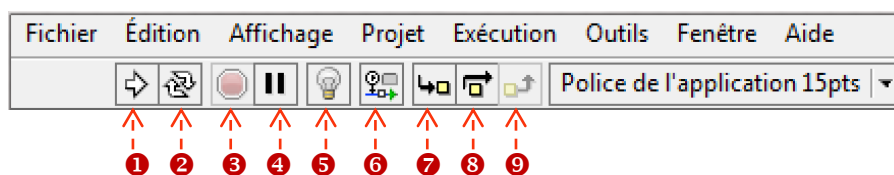


Figure II.8 : Différentes structures de données.

II.4.2. Barre d'outils du diagramme

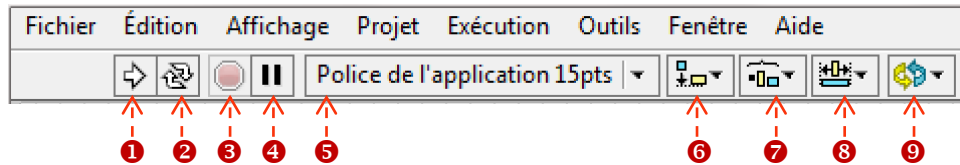
La barre d'outils suivante est présente sur la face avant, elle donne accès aux outils d'exécution et de présentation.



- 1** → Exécution unique.
- 2** → Exécution en continu.
- 3** → Arrêt de l'exécution.
- 4** → Pause : ce bouton prend la couleur rouge pour montrer qu'il est actif, il faut cliquer dessus pour sortir de ce mode.
- 5** → Activation du mode animation : la lampe s'allume pour montrer qu'il est actif, il permet de suivre le flux de données et l'activité des différents VI lors du lancement.
- 6** → Lecture de la valeur des fils de liaisons.
- 7** → Exécution en mode pas à pas.
- 8** → Passage à l'étape suivante sans détailler l'étape actuelle.
- 9** → Fin du mode pas à pas.

II.4.3. Barre d'outils de la face avant

La barre d'outils suivante est présente sur la face avant, elle donne accès aux outils d'exécution et de présentation.



- 1 → Exécution unique.
- 2 → Exécution en continu.
- 3 → Arrêt de l'exécution.
- 4 → Pause : ce bouton prend la couleur rouge pour montrer qu'il est actif, il faut re cliquer dessus pour sortir de ce mode.
- 5 → paramètre de texte.
- 6 → aligner les objets.
- 7 → répartir les objets.
- 8 → redimensionner les objets.
- 9 → réorganiser.

II.5. Palettes

LABVIEW possède trois palettes flottantes respectivement nommées:

Outils, Commandes, et Fonctions.

II.5.1. Palette d'outils

La palette d'outils existe sur le diagramme et sur la face avant. Elle permet de modifier des valeurs, des couleurs, mais aussi de câbler les entrées et les sorties des icônes entre elles, de poser des points d'arrêt, des sondes. On y accède par Fenêtre» Afficher la palette d'outils.[10] (figure II.9).

Palette des outils : disponible dans « Affichage/Palette d'outil »

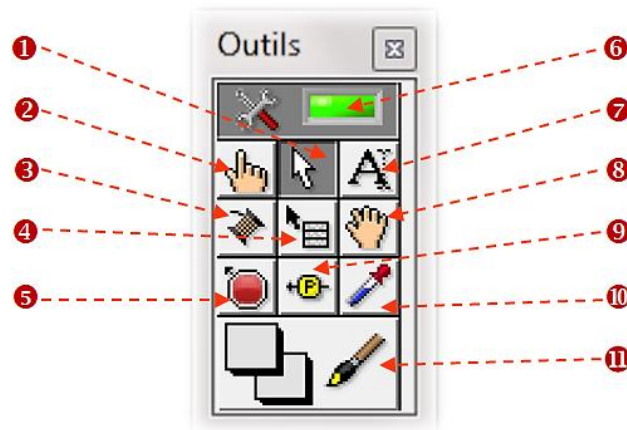


Figure II.9 : La palette d'outils.

- ① → Positionner/Sélectionner/Dimensionner.
- ② → Edition des éléments graphiques.
- ③ → Connexions des éléments graphiques.
- ④ → Menu local de l'objet.
- ⑤ → Placer/supprimer un point d'arrêt.
- ⑥ → Choix automatique de l'outil.
- ⑦ → Insertion de texte.
- ⑧ → Défilement dans la fenêtre.
- ⑨ → Placement de sonde sur un fil.
- ⑩ → Pipette Outil copie couleur.
- ⑪ → Remplissage et choix de la couleur.

II.5.2. Palette de commande

La palette de commandes est disponible dans la fenêtre de face avant. Elle apparaît fugitivement lors d'un clic droit dans un endroit vierge de la fenêtre, ou par Fenêtre »Afficher la palette de commandes. Elle contient tous les éléments graphiques disponibles pour créer l'interface utilisateur. Ceux -ci sont hiérarchisés par type de données ou par grandes familles d'objets. Vous apercevez ci-dessous la palette de commandes.

Palette des commandes : disponible à partir de la fenêtre face avant par un clic droit avec la souris ou dans « Affichage/Palette des commandes » (figure II.10).

II.5.3. Palette de fonction

Elle est accessible dans le diagramme par les mêmes méthodes que celle de commandes. Elle contient l'ensemble des fonctions de LABVIEW regroupées par type de fonctionnalités (Programmation, acquisition, traitement mathématiques, connectivité...).

Palette des fonctions : disponible à partir de la fenêtre diagramme par un clic droit avec la souris ou dans « Affichage/Palette des fonctions » (figure II.11).

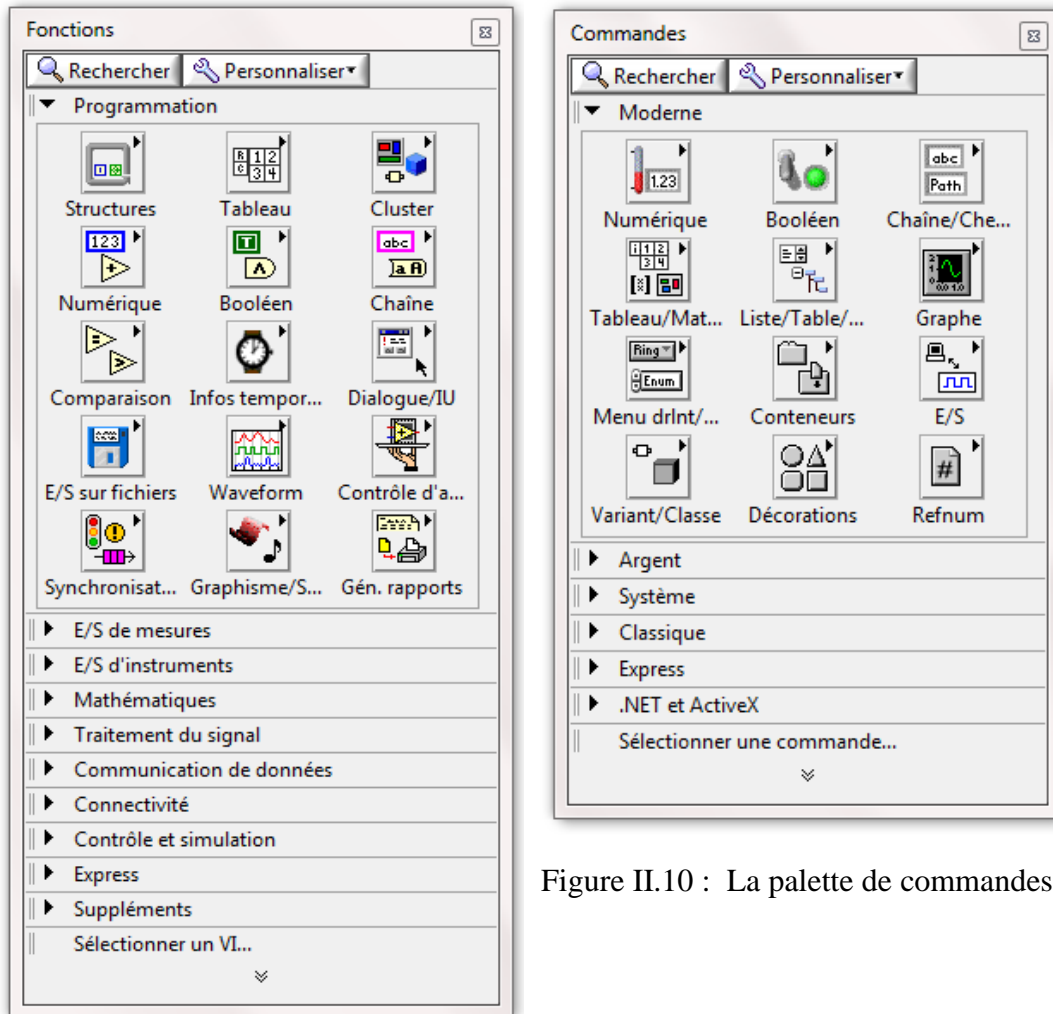


Figure II.10 : La palette de commandes.

Figure II.11 : La palette de fonction.

Dans La palette de commandes : Plusieurs palettes existent (« Moderne », « Système », « Classique », « Express », « Commandes utilisateur », « .NET et ActiveX », les autres palettes dépendant des modules supplémentaires installés). (figure II.10), La plupart du temps nous utiliserons la palette « Moderne ». Cette palette organise les éléments par catégorie : [13]

- **Numérique** : offre des commandes et indicateurs permettant de saisir ou afficher un numérique. Noter que par défaut, ce numérique est en général un nombre réel (par opposition à un nombre entier). Nous parlerons plus tard de la notion de type.
- **Booléen** : commandes et indicateurs booléens (l'algèbre de Boole, dite algèbre booléenne est l'algèbre de la logique, dans laquelle les variables peuvent prendre la valeur vrai ou la valeur faux). Les booléens correspondent aux boutons à 2 états et aux indicateurs à 2 états (LED, etc.).
- **Chaîne et chemin** : commandes et indicateurs permettant de saisir ou d'afficher des chaînes de caractères (par chaîne de caractères, on entend des mots ou phrases, i.e. suite de caractères). De même, on trouve ici ce qui permet de saisir ou afficher un nom de fichier ou de répertoire (chemin).
- **Graphe** : propose différents indicateurs de graphes. Ces quatre sous-palettes sont les plus fréquemment utilisées. Vous êtes invité bien entendu à ouvrir les menus afin de découvrir les éléments que proposent les différentes palettes et sous palettes (par exemple la palette système offre des éléments au « look » du système d'exploitation sous-jacent comme Windows, Mac OS,...). Il est important de comprendre que l'apparence d'une commande, par exemple pour un numérique, n'a pas de réelle importance une « commande numérique » et un « bouton rotatif ».

➤ Dans La palette de la palette de fonctions : (figure II.11), Cette palette peut être obtenue de manière analogue à la palette de commandes (menu déroulant « Affichage » de la fenêtre du diagramme, ou bien click droit dans le diagramme). Chaque fonction (sous-vi) en LabVIEW est représentée par une icône. Les fonctions sont regroupées par catégories : « Structures », « Tableau », « Cluster et variant », « Numérique », « Entrées/Sorties sur fichier », « Booléen », « Chaîne de caractère », « Comparaison », etc. Nous aurons l'occasion de les explorer ensemble. L'une des fonctionnalités importantes de cette palette est le bouton « Rechercher ». En effet, la bibliothèque de fonctions est importante, d'autant plus si différents modules optionnels sont installés.[13]

II.6. Structures de programmation

Les structures sont des représentations graphiques des boucles et des conditions des langages de programmation textuels. Utilisez des structures dans votre diagramme pour répéter des blocs de code et pour exécuter le code de manière conditionnelle ou dans un ordre spécifique.

Les structures de programmation se trouvent dans le menu Structures de la fenêtre représenté ci-dessous.

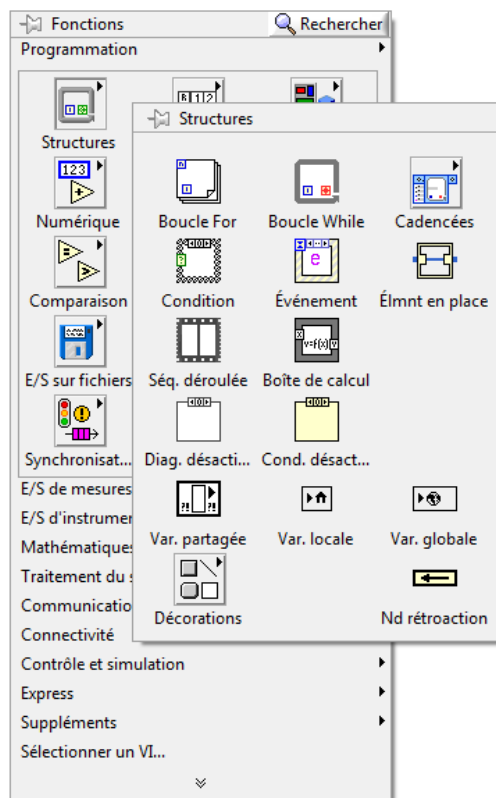


Figure II.12 : Les structures de programmation.

- Il existe 5 principaux types de structures sous LABVIEW :
 - La condition : Si alors sinon.
 - La boucle For : Faire ... N fois.
 - La boucle While : Faire ... tant que .
 - Structure Séquence déroulée.
 - La structure évènement.

II.6.1. Structures condition

Contient un ou plusieurs sous-diagrammes (les "conditions"), dont un exactement s'exécute quand la structure s'exécute. La valeur câblée au sélecteur de condition détermine la condition à exécuter.

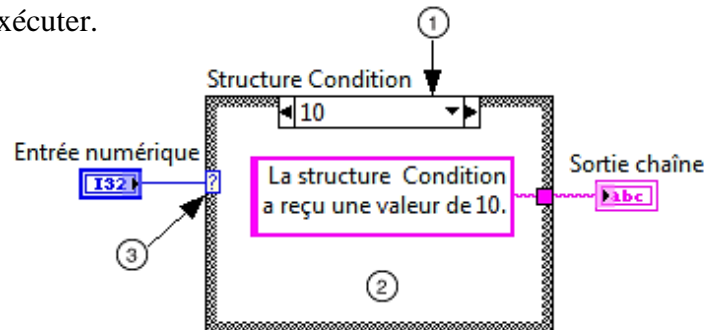


Figure II.13 : Composantes d'une structure Condition.

① → **Étiquette du sélecteur** : Affiche la ou les valeurs pour lesquelles la condition associée s'exécute. Vous pouvez spécifier une seule valeur ou une gamme de valeurs. Vous pouvez également utiliser l'étiquette du sélecteur pour spécifier une condition par défaut.

② → **Sous-diagramme (condition)** : Contient le code qui s'exécute quand la valeur câblée au sélecteur de condition correspond à la valeur qui apparaît dans l'étiquette du sélecteur. Pour modifier l'ordre ou le nombre de sous-diagrammes, cliquez avec le bouton droit sur le cadre de la structure Condition et sélectionnez l'option appropriée.

③ → **Sélecteur de condition** : Sélectionne la condition à exécuter selon la valeur des données en entrée. Les données en entrée peuvent être du type booléen, chaîne, entier, énumération ou cluster d'erreur. Le type de données que vous câblez au sélecteur de condition détermine les conditions que vous pouvez entrer dans l'étiquette du sélecteur.

II.6.2. Boucle For

Exécute son sous-diagramme n fois, (n) étant la valeur câblée au terminal (N) de décompte de la boucle. Le terminal d'itération (i) fournit le nombre d'itérations actuel de la boucle, qui varie de 0 à $n-1$.

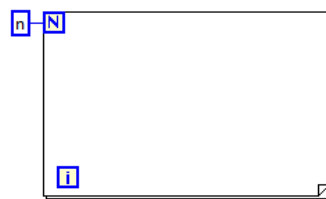


Figure II.14 : Composantes d'une Boucle For.

II.6. 3. Boucle While

Répète le code contenu dans son sous-diagramme jusqu'à ce qu'une condition spécifique se produise. Une boucle While s'exécute toujours au moins une fois.

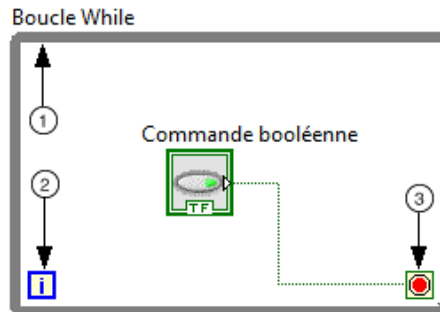


Figure II.15 : Composantes d'une Boucle While.

① ➔ **Sous-diagramme** : Contient le code exécuté par la boucle While une fois par itération.

② ➔ **Terminal d'itération (i)** : Fournit le nombre d'itérations actuel de la boucle. Le comptage des itérations commence toujours à zéro.

③ ➔ **Terminal de condition** : Évalue la valeur booléenne d'entrée pour déterminer si la boucle While doit continuer à s'exécuter ou non. Pour spécifier si la boucle doit s'arrêter pour la valeur booléenne VRAI ou FAUX, configurez la condition d'arrêt de la boucle. Vous pouvez aussi déterminer quand la boucle doit s'arrêter en câblant un cluster d'erreur au terminal de condition.

II.6.4. Structure Séquence déroulée.

Composée d'un ou plusieurs sous-diagrammes ou étapes qui s'exécutent de façon séquentielle. Utilisez la structure Séquence déroulée pour garantir qu'un sous-diagramme s'exécute avant ou après un autre sous-diagramme.

Le flux de données pour la Séquence déroulée diffère de celui des autres structures. Les étapes d'une structure Séquence déroulée s'exécutent de gauche à droite et lorsque toutes les valeurs des données câblées à l'étape sont disponibles. Les données quittent chaque étape lorsque celle-ci finit de s'exécuter. Ceci signifie que l'entrée d'une étape peut dépendre de la sortie d'une autre étape.

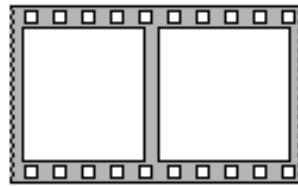


Figure II.16 : Structure Séquence déroulée.

II.6. 5. La structure événement

Attend qu'un événement se produise puis exécute la condition appropriée pour gérer cet événement. La structure Événement possède un ou plusieurs sous-diagrammes, ou conditions d'événement, dont un exactement s'exécute lorsque la structure s'exécute pour gérer l'événement. Cette structure peut être mise hors délai pendant qu'elle attend une notification d'événement. Câblez une valeur au terminal Timeout dans la partie supérieure gauche de la structure Événement pour spécifier le nombre de millisecondes pendant lesquelles la structure Événement doit attendre qu'un événement se produise. Par défaut, la valeur est -1 et il n'y a jamais de timeout.

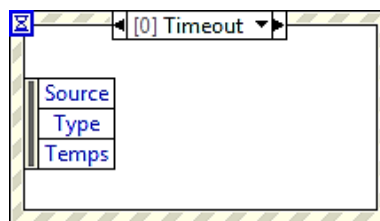


Figure II.17 : Structure événement.

II.7. L'avantage de logiciel de LABVIEW

Nous résumons en quelques lignes les avantages de logiciel LABVIEW :

- ✓ Facile à apprendre et à utiliser .
- ✓ Fonctionnalités complètes.
- ✓ Intégration des fonctions d'E/S.
- ✓ Découverte des concepts.
- ✓ Apprentissage amélioré.
- ✓ Intégration transparente avec le matériel dans les expériences en conception.
- ✓ Plate-forme ouverte pour s'interfacer avec d'autres outils d'ingénierie.
- ✓ Standard industriel.

LabVIEW offre la souplesse d'un langage de programmation puissant sans la complexité des environnements de développement traditionnels.[14]

II.8. Conclusion

LabVIEW est un langage de programmation graphique conçu pour les ingénieurs et les scientifiques désirent de développer des applications de test, de contrôle et de mesure. La nature intuitive de la programmation graphique de LabVIEW permet aux enseignants et aux chercheurs d'intégrer le logiciel dans toute une gamme de cours et d'applications. Avec LabVIEW, les enseignants et les chercheurs peuvent utiliser une approche de conception graphique de systèmes pour concevoir, prototyper et déployer des systèmes embarqués. Il combine la puissance de la programmation graphique avec le matériel pour simplifier et accélérer de façon impressionnante la conception.[15]

L'évolution des systèmes électroniques amène de plus en plus souvent les concepteurs à remplacer l'électronique câblée à base de nombreux circuits intégrés par un circuit programmable qui remplit à lui seul toutes les fonctions. L'objectif du chapitre suivant est la description détaillée du microcontrôleur utilisé dans cette réalisation.

Chapitre III

Notion sur les microcontrôleurs



III.1. Introduction

L'évolution des systèmes électroniques amène de plus en plus souvent les concepteurs à remplacer l'électronique câblée à base de nombreux circuits intégrés par un circuit programmable qui remplit à lui seul toutes les fonctions. Les microcontrôleurs appartiennent à cette famille de circuits.

Le microcontrôleur est implanté sur le système technique et relié aux différents capteurs et actionneurs avec éventuellement des circuits d'interfaçage appropriés. Une liaison avec un ordinateur permet de programmer le circuit et de le tester. [16]

III.2. Les microcontrôleurs (PIC)

III.2.1. Généralités

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires « mémoire morte pour le programme, mémoire vive pour les données », unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique « quelques milliwatts en fonctionnement, quelques nano watts en veille », une vitesse de fonctionnement plus faible « quelques mégahertz » et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels. [17]

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans le domaine d'industrie pour la commande et contrôle des systèmes.[17]

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

Des nombreux fabricants se sont implantés sur le marché des microcontrôleurs, la société américaine Micro-chip a mis au point un microcontrôleur CMO « Complementary Metal Oxide Semi-conductor », appelé PIC ; cette dénomination est sous copyright de Micro-chip, donc les autres fabricants ont été dans l'impossibilité d'utiliser ce terme pour leur propre microcontrôleur. Actuellement ce microcontrôleur est très utilisé.[17]

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 : [18]

- ↳ La famille **Base-Line**, qui utilise des mots d'instructions de 12 bits.
- ↳ La famille **Mid-Range**, qui utilise des mots de 14 bits (et dont font partie la (16FXXX).
- ↳ La famille **High-End**, qui utilise des mots de 16 bits. (18FXXXX).

III.2.3. Identification des PIC

Un PIC est généralement identifié par une référence de la forme suivante :
xx(L)XXyy-zz . [19]

- **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** : Identificateur.
- **zz** : Vitesse maximale du quartz de pilotage.

III.3. Le choix du microcontrôleur

Le choix d'un microcontrôleur est primordial car c'est de lui que dépendent en grande partie les performances, la taille, la facilité d'utilisation et le prix du montage. En fait ce choix est imposé dans le cahier de charge. [20] ,[16]

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 centrer 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 nom d'une EEPROM pour mémoriser des données est également important pour l'application souhaitée.
- La longueur de programme de l'application détermine la taille de la mémoire programme du PIC recherché. [18] , [19]

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.

III.4. Caractéristiques principales du pic 18F4550

Le 18F4550 est un PIC de 40 broches, que l'on peut trouver dans un boîtier PDIP, 4 pins pour l'alimentation, et deux broches de masse (broches 12 et 31). Les deux pins d'alimentation sont situés aux broches 11 et 32. Deux pins pour l'oscillateur au choix par ce qu'il existe un oscillateur interne. Un pins pour le Reset MCLR.

Donc : on a 35 lignes d'entrées/sorties, répartis comme suit : [19]

- ✓ Un port de 7 lignes (port A) .
- ✓ Un port de 8 lignes (port B) .
- ✓ Un port de 8 lignes (port C).
- ✓ Un port de 8 lignes (port D).
- ✓ Un port de 4 lignes (port E) .
- Alimentation sous 5 Volts .
- Une mémoire de programme de type flash (32Ko).
- Une mémoire RAM utilisateur de 2048 Octets .
- Une mémoire EEPROM de 256 Octets emplacements .
- Une interface I²C pour la gestion d'un bus à 2 fils .
- Facilité de programmation comme tous les PICs .
- Universel Serial Bus (USB) , USB V2.0 compatible avec :
 - ✓ Faible vitesse (1.5 Mb/s). ou en English (Low Speed).
 - ✓ Pleine vitesse (12 Mb/s) . ----- (Full Speed).
- Liaison MSSP .
 - ✓ Bus I²C.
 - ✓ SPI
- USART.

- Quatre modes de cristal, y compris de haute précision PLL pour l'USB.
- Courant de Source: 25 mA.
- 13 chenal de 10-Bit pour le convertisseur A/D .
- Quater Temporisateurs.

Le tableau suivant résume tous les caractéristiques générales de PIC18F4550 .[21]

La caractéristique	Valeur
Fréquence Horloge MHz	48MHz
Mémoire programme FLASH	32Ko
Mémoire données (RAM)	2048 Octets
Mémoire EEPROM	256 Octets
Interruptions	20
Ports parallèles (Nombre des lignes)	A(7), B(8) C(8), D(8), E(4)
Timers	4 (3 1× 6bit + 1×8bit) Timer0 à Timer3
CAN 10-bit	13
Instructions	75
Vitesse du CPU (MIPS)	12
Périphériques de communication numérique	1-A/E/USART, 1-MSSP (SPI/I2C)
Capture / Comparateur / Périphériques PWM	1CCP , 1ECCP
Comparateurs	2
USB (canaux, vitesse, respect)	Low Speed, Full Speed USB 2.0
Plage de tension de fonctionnement (V)	de 2 à 5,5
Température (C)	-40 à 85

Tableau. III.1 : Les caractéristiques générales de PIC18F4550.

III.5. Brochage du 18F4550

Les figures suivants résumé le brochage du PIC 18F4550 :

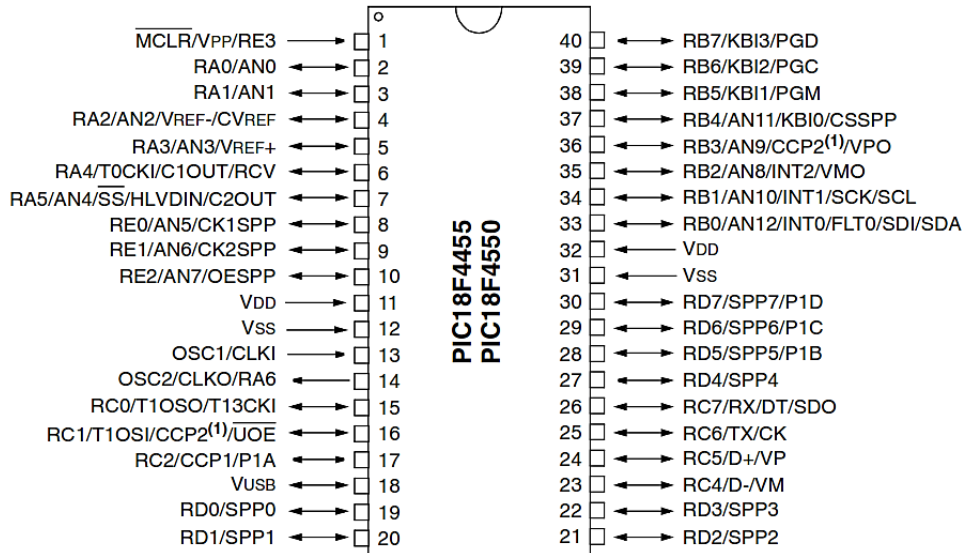


Figure III.1.a : Brochage du 18F4550(40-Pin PDIP) .[21]

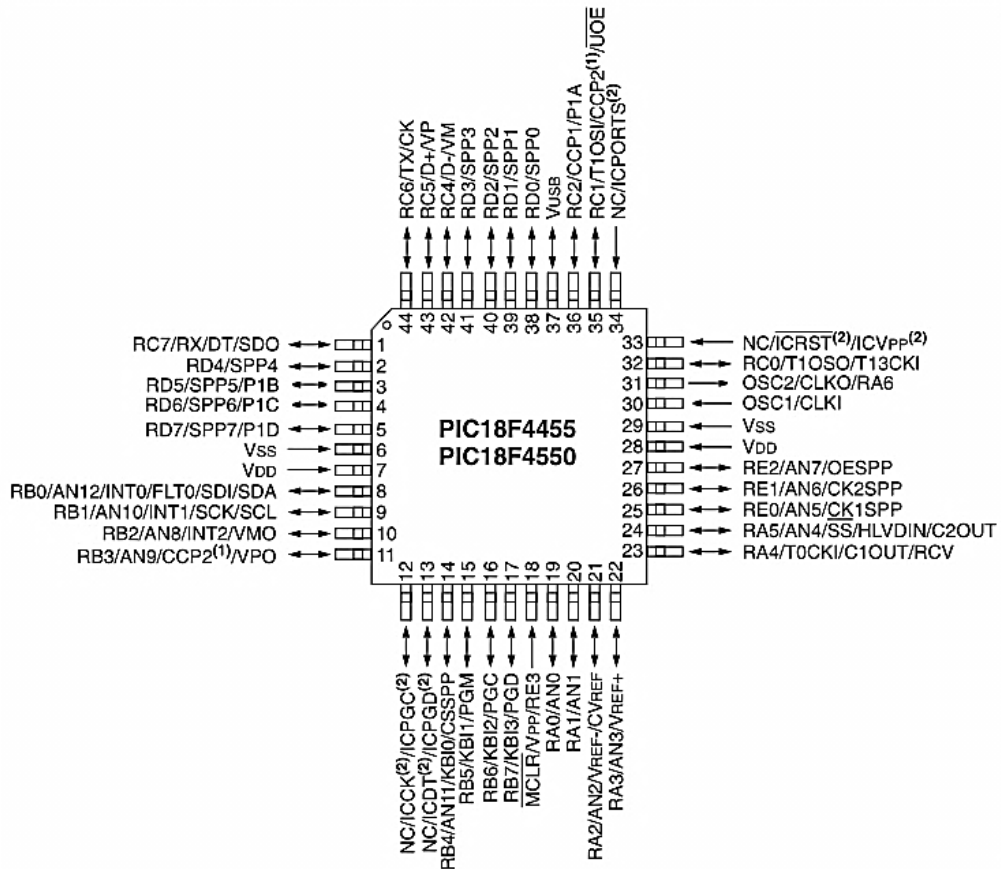


Figure III.1.b : Brochage du 18F4550(44-Pin TQFP) .[21]

III.6. Architecture générale du PIC

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

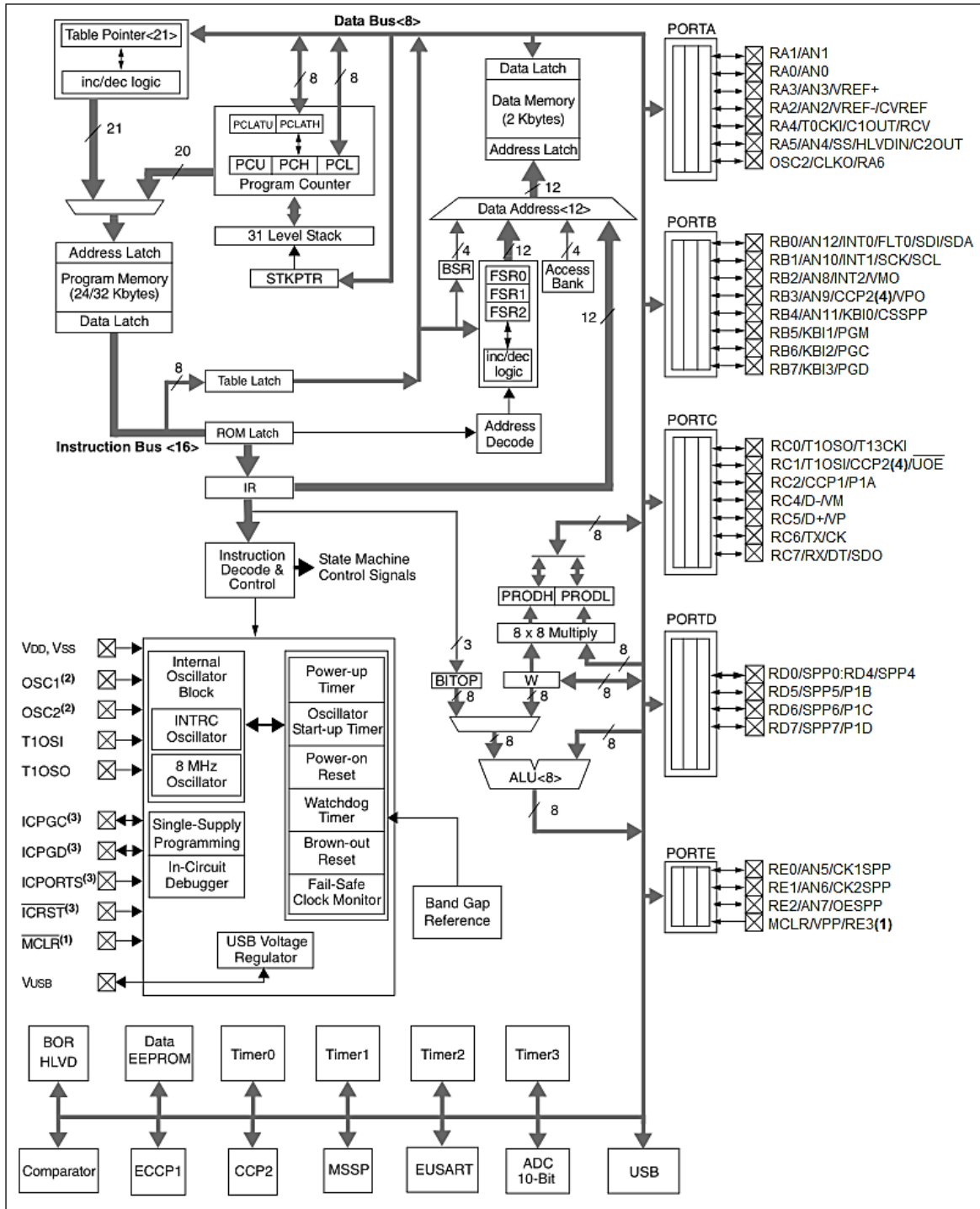


Figure III.2 : Architecture générale du PIC18F4550. [21]

III.6.1. Organisation de la mémoire

Le microcontrôleur PIC18F4550 dispose de deux types de mémoires séparé : [21]

- Une mémoire de programme de 32 ko (mémoire FLASH).
- Une mémoire de données dont :
 - ✓ 2048 octets (SRAM) .
 - ✓ 256 octets (EEPROM).

III.6.1.1. Mémoire de programme

La capacité de la mémoire de programme est de 32 ko (kilo-octets). Cependant, chaque instruction est codée sur 16 bits, ce qui signifie que chaque instruction occupera 2 octets. Ainsi, la mémoire de programme ne pourra stocker que jusqu'a 16384 instructions au maximum [19].

III.6.1.2. Mémoire de données

Elle se décompose en deux parties , une RAM et une zone EEPROM

- **La mémoire RAM (Random Access Memory)**

La mémoire RAM appelée mémoire vive, ce type de mémoire perd toutes ses informations stockées lorsqu'elle cesse d'être alimentée. La capacité totale de la mémoire de donnée est de 2048 octets. Elle est constituée de deux parties :

- ✓ **Les registres SFR (Special Function Register)**

Ce sont les registres de fonctionnement du PIC. L'ensemble de ces registres est souvent appelé fichier des registres.

- ✓ **Les registres GPR (General Propose Register)**

Sont des positions mémoire que l'utilisateur peut utiliser pour stocker ses variables et ces données. On remarque donc que, indépendamment de leur nature, les positions de la RAM sont toujours appelé registres.

Le plan mémoire est découpé en 16 zones (bank), identiques de 256 octets. La sélection de la zone est réalisée par 4 bits du registre BSR (Bank Select Register).

Les registres internes pour les fonctions spéciales (registres généraux et configuration des périphériques → SFR qui signifie (Special Function Registers), se situent aux adresses hautes.[22], [19]

- **Mémoire EEPROM (Electrically Erasable Programmable Read Only Memory)**

Le PIC possède une zone EEPROM de 256 octets accessibles en lecture et en écriture par le programme. On peut y sauvegarder des valeurs, qui seront conservées même si l'alimentation est éteinte, et les récupérer lors de la mise sous tension. Leur accès est spécifique et requiert l'utilisation de registres dédiés. La lecture et l'écriture ne peut s'exécuter que selon des séquences particulières. [22]

III.6.2. Les temporisateurs (timers)

Le temporisateur (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 **MLI**, (appelée également **Modulation de Largeur d'Impulsion**), ou en English **PWM**, (**Pulse-Width Modulation**), encore les modes Compare ou Capture. Selon le type de PIC rencontré, on trouvera un certain nombre de timer qui seront pour certains associés à des ressources internes du PIC. Un timer est en fait un compteur incrémentant au rythme d'une horloge, et ceci en ayant la possibilité de pré-positionner une valeur de départ. Certains timers pourront fonctionner en mode 8 bits, 16 bits ou 32 bits. Dans le PIC18F4550 il existe quatre timers (un timer de 8 bit et le reste de 16 bits). [18],[23]

III.6.3. Horloges

L'horloge va permettre de cadencer tous les échanges internes et externes du PIC. 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. [18]

III.6.3.1. Types d'Oscillateur

Le PIC18F4550 peuvent être exploités en douze modes d'oscillation différents. En contraste avec pré-percé- PIC18 améliorée microcontrôleurs, quatre d'entre eux les modes impliquent l'utilisation de deux types d'oscillateur à la fois.

Les utilisateurs peuvent programmer la Configuration des bits FOSC3 à FOSC0 pour sélectionner un modes d'oscillation. [21], (tableau III.2).

Les lignes suivants représenté les différent Types d'Oscillateur :

1. **XT** : Crystal / Résonateur.
2. **XTPLL** : Crystal / Résonateur avec PLL activé.
3. **HS** : Haut débit Crystal / Résonateur.

4. **HSPLL** : Haut débit Crystal / Résonateur avec PLL activé.
5. **EC** : Horloge externe avec / 4 sorties FOSC.
6. **ECIO** : Horloge externe avec E / S sur RA6.
7. **ECPLL** : Horloge externe avec PLL activé et / 4 sorties sur RA6 FOSC.
8. **ECPIO** : Horloge externe avec PLL activé, I / O sur RA6.
9. **INTHS** : Oscillateur interne utilisé en tant que source d'horloge du microcontrôleur, HS Oscillateur utilisé en tant que source USB d'horloge.
10. **INTXT** : Oscillateur interne utilisé en tant que source d'horloge du microcontrôleur, XT Oscillateur utilisé en tant que source USB d'horloge.
11. **INTIO** : Oscillateur interne utilisé en tant que source d'horloge du microcontrôleur, EC Oscillateur utilisé en tant que source d'horloge USB, E / S numériques sur RA6.
12. **INTCKO** : Oscillateur interne utilisée en tant que source d'horloge du microcontrôleur, EC oscillateur utilisé comme source d'horloge USB, 4 FOSC / sortie sur ER6.

Configuration des bits FOSC3 à FOSC0 pour le mode d'horloge

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

Tableau III.2 : Rôle des bits FOSC0 à FOSC3 pour le 18F4550. [21]

III.6.3.2. Paramètres d'oscillateur pour USB

Lorsque le PIC18F4550 est utilisé pour la connectivité USB, il doit avoir un 6 MHz ou 48 MHz horloge pour USB fonctionnement, selon que Low-Speed ou Mode Full-Speed est utilisé. Cela peut nécessiter un certain prévoyance pour choisir une fréquence de l'oscillateur et programmation de l'appareil. [21]

III.6.3.2.1. Le PLL

Le PIC18F4550 c'est un dispositifs comprennent une Boucle à verrouillage de phase ou en English (**phase Locked Loop (PLL)**). Ceci est fourni spécifiquement pour les applications USB avec des oscillateurs à faible vitesse et peuvent également être utilisés en tant que source d'horloge du microcontrôleur.

Le PLL est activé dans les modes oscillateur HSPLL, XTAL, PLL et ESPIO. Il est conçu pour produire un fixe horloge de référence de 96 MHz à partir d'une entrée fixe de 4 MHz.

La sortie peut alors être divisée et utilisée à la fois pour l'USB et l'horloge de base du microcontrôleur. Parce que le PLL possède une entrée et sortie de fréquence fixe, il y a huit options pré-échelonnage pour relier l'entrée de la fréquence d'oscillateur à la PLL. (figure III.3). [21]

Il y a aussi une option de postscolaire séparée pour dériver l'horloge du microcontrôleur à partir de la PLL. Cela permet au périphérique USB et microcontrôleur à utiliser le même entrée de l'oscillateur et fonctionnent à l'horloge de vitesses différente. Contrairement à la postscolaire pour XT, HS et EC modes, les options de diviseur de fréquence disponibles à la sortie de la PLL sont 1/2, 1/3, 1/4 et 1/6. (figure III.4).

Le modes HSPLL, ECPLL et ECPIO font usage de l'oscillateur en mode HS pour des fréquences jusqu'à 48 MHz. L'étage d'adaptation divise l'entrée de l'oscillateur par 12 pour produire 4 MHz pour la PLL.

Le mode XTPLL ne peut utiliser que une fréquence d'entrée de 4 MHz, ce qui fonctionne directement la PLL. La figure suivant représenté la diagramme de block (PLL).

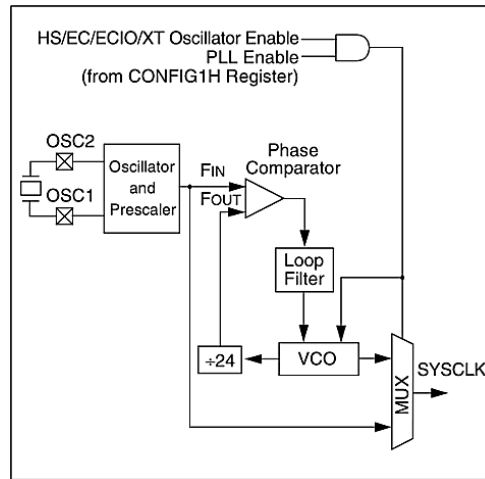


Figure III.3 : Diagramme de block PLL, (mode HS). [21]

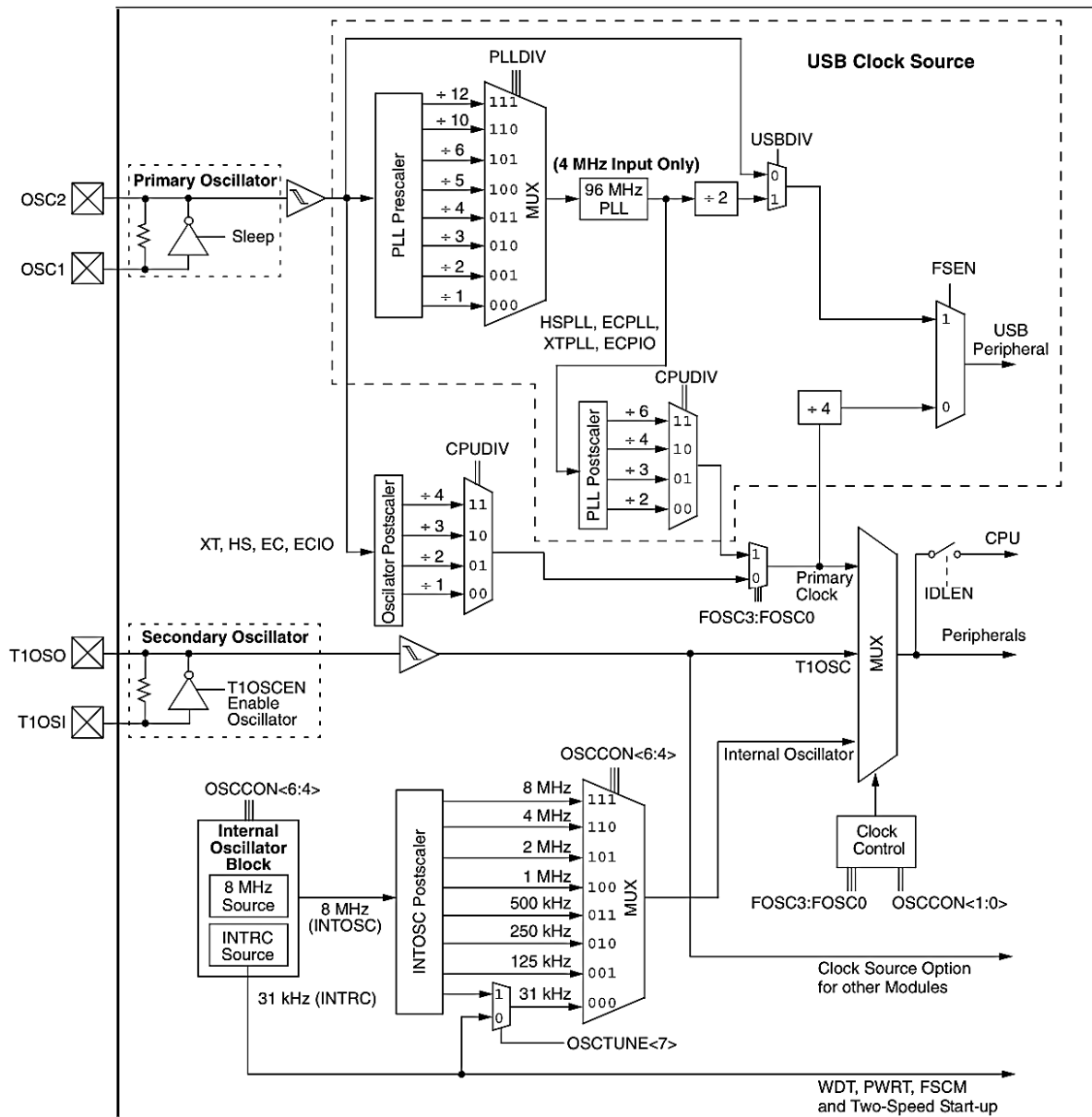


Figure III.4 : Synoptique de la circuiterie d'horloge sur un PIC 18F4550 gérant l'USB.

III.7. Les convertisseurs analogiques/numériques

Pour utiliser un convertisseur analogique / numérique, il existe trois registres internes nommés ADCON0, ADCON1 et ADCON2, qui permettront de configurer et de sélectionner les différentes entrées analogiques, (**ADC** pour **A**nalog to **D**igital **C**onverter), du PIC utilisé. 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/ digital interne sont les broches AN0 à AN12. Les entrées analogiques (RA0 à RA5), transitent via le PORT A du PIC, les entrées AN5 à AN7 font partie du PORT E et les entrées AN8 à AN12 font partie du port B.

Comme vous le voyez sur le schéma général, 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 V_{SS} pour V_{ref-} et AN3 ou V_{DD} pour V_{ref+}). Les bits PCFG0 à PCFG3 du même registre permettent 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 de 0TAD à 20TAD (soit 0 top d'horloge à 20 tops 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/DONE du registre ADCON0 est forcée à «1». Dès que la conversion en cours est terminée, ce bit repasse à « 0 ». On réalise donc une boucle d'attente tant que la conversion n'est pas échue. Notez également que le bit ADON du même registre permet d'alimenter le convertisseur en 5 V : 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 10 bits, 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. [19] (Figure III.5).

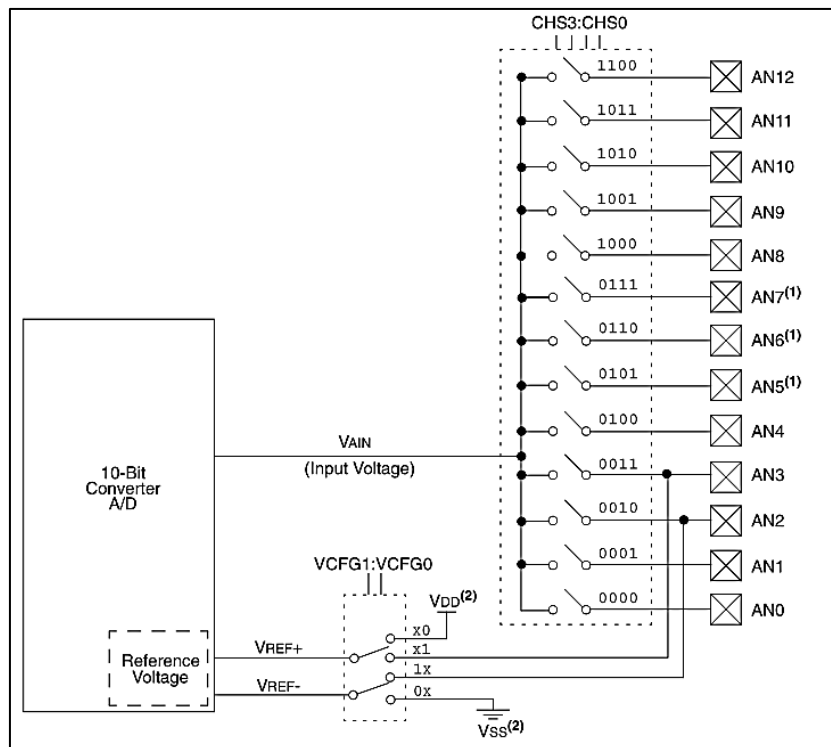


Figure III.5 : Synoptique général du CAN pour le PIC18F4550.[21]

III.7.1. Registre ADCON0

Ce registre permettra: [19]

- ✓ de sélectionner un canal parmi 12 (bits CHS0-CHS3) .
- ✓ d'alimenter le module convertisseur (bit ADON).
- ✓ d'avertir l'utilisateur lorsqu'une conversion en est cours (bit GO/DONE), ou de lancer une conversion.

La (figureIII.6), résume ces différentes fonctionnalités. Les bits CHS0 à CHS2 permettent de sélectionner le canal à choisir pour la conversion (exemple : 0101 → canal 5, soit l'entrée AN5 du PIC).

Le bit 1 (GO / DONE), permet de lancer une conversion ou bien de savoir si une conversion est en cours d'exécution (0 = fin de conversion).

Le bit 0 (ADON), permet l'alimentation du convertisseur : si ce bit est à 0, le convertisseur n'est pas alimenté.

ADCON0							
-	-	CHS3	CHS2	CHS1	CHS0	GO /DONE	ADON
bit 7						bit 0	
bit 7-6 inutilisé : lire come '0'							
bit 5-2 CHS3:CHS0 : bits de sélection de canal							
0000 = Canal 0 (AN0)							
0001 = Canal 1 (AN1)							
0010 = Canal 2 (AN2)							
0011 = Canal 3 (AN3)							
0100 = Canal 4 (AN4)							
0101 = Canal 5 (AN5)							
0110 = Canal 6 (AN6)							
0111 = Canal 7 (AN7)							
1000 = Canal 8 (AN8)							
1001 = Canal 9 (AN9)							
1010 = Canal 10 (AN10)							
1011 = Canal 11 (AN11)							
1100 = Canal 12 (AN12)							
1101 = non implémenté							
1110 = non implémenté							
1111 = non implémenté							
bit 1 GO/DONE : bit d'état conversion							
Quand ADON = 1:							
- 1 = A/D conversion en cours							
- 0 = A/D conversion terminée							
bit 0 ADON : bit de mise en service							
- 1 = A/D Module convertisseur en service							
- 0 = A/D Module convertisseur hors service							

Figure III.6 : Registre ADCON0 [21]

III.7.2. Registre ADCON1

Ce registre (figure III.7), permettra: [19]

- ✓ de définir les tensions de référence V- et V+, ces tensions pourront être l'entrée AN2 ou AN3, ou bien le +Vdd et le Vss de l'alimentation (bits VCFG1 et VCFG0) .
- ✓ de configurer toutes les broches du port A soit en entrée logique soit en entrée analogique avec des options pour la tension de référence (bits PCFG0-PCFG3).

ADCON1													
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0						
bit 7								bit 0					
bit 7- 6 inutilisé : lire come '0'													
bit 5 VCFG1 : bits de configuration de la référence de tension Vref –													
- 1 = Vref – (AN2)													
- 0 = AVss													
bit 4 VCFG0 : bit de configuration de la référence de tension Vref +													
- 1= Vref + (AN3)													
- 0= AVdd													
Bit 3-0 :bit de configuration des entrées du convertisseur													
PCFG: 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
A = entrée Analogique								D = entrée / sortie Digitale					

Figure III.7 : Registre ADCON1. [21]

III.7.3. Registre ADCON2

Ce registre (figureIII.8), permettra:

- ✓ de définir le format de donnée pour la valeur issue de la conversion analogique/numérique (bits ADFM).
- ✓ de configurer le choix de l'horloge utilisée par le convertisseur (bits ADCS0 à ADCS2)

- ✓ de configurer un temps d'attente avant la conversion et d'obtenir ainsi une à éventuelle stabilisation de la tension à mesurer (bits ACQT0 à ACQT2).

ADCON2								
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	
bit 7								bit 0
<p>bit 7 ADFM: bit de configuration du format de sortie</p> <p>1 = justification à droite</p> <p>0 = justification gauche</p> <p>bit 6 Inutilisé</p> <p>bit 5-3 ACQT2:ACQT0: bits de sélection des temps d'acquisition</p> <p>111 = 20 TAD</p> <p>110 = 16 TAD</p> <p>101 = 12 TAD</p> <p>100 = 8 TAD</p> <p>011 = 6 TAD</p> <p>010 = 4 TAD</p> <p>001 = 2 TAD</p> <p>000 = 0 TAD</p> <p>bit 2-0 ADCS2:ADCS0: bits de sélection de l'horloge pour la conversion</p> <p>111 = FRC (Horloge provenant de l'oscillateur RC interne)</p> <p>110 = FO SC/64</p> <p>101 = FO SC/16</p> <p>100 = FO SC/4</p> <p>011 = FRC (Horloge provenant de l'oscillateur RC interne)</p> <p>010 = FO SC/32</p> <p>001 = FO SC/8</p> <p>000 = FO SC/2</p>								

Figure III.8 : Registre ADCON2 . [21]

III.8. Les avantages du microcontrôleur :

L'utilisation des microcontrôleurs pour les circuits programmables à plusieurs points forts. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années.[20]

Nous allons voir que le nombre d'entre eux découle du simple sens.

- Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement de matériel et de circuit imprimé.
- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de donnée d'un composant à un autre.
- L'augmentation de la fiabilité du système puisque, le nombre des composants diminuant, le nombre des connexions composants/supports ou composants/circuits imprimer diminue.
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux :
 - Moins cher que les autres composants qu'il remplace.
 - Diminuer les coûts de main d'œuvre.
- Réalisation des applications non réalisables avec d'autres composants.

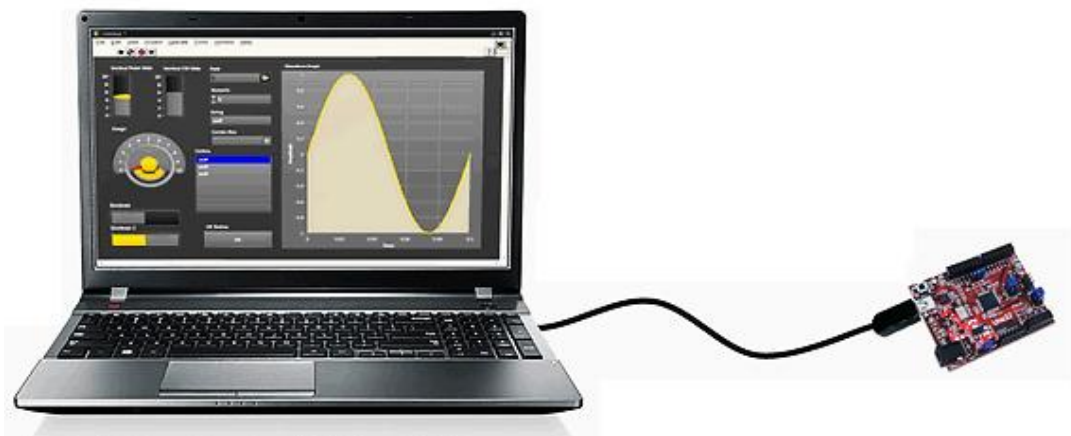
III.9. Conclusion

Ce chapitre est un portail pour notre réalisation et conception matérielle. Ça nous a permis de maîtriser les options de notre PIC et ses caractéristiques afin de les exploiter d'une manière correcte. 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 description de notre réalisation en détail sera l'objet de chapitre suivant .

Chapitre IV

Conception et réalisation de la carte



IV.1. Introduction

Dans tout type d'application électronique, il est courant d'utiliser des capteurs ou des actionneurs pour interagir avec l'environnement. Seuls, ces matériels peuvent être difficiles à utiliser et nécessiter, pour chacun, d'autres composants dédiés à la présentation des mesures ou à la communication. L'objectif de ce projet est donc d'utiliser un microcontrôleur pour centraliser les communications et le contrôle de différents matériels électroniques. Le microcontrôleur est doté d'une faculté de traitement des informations qui lui parviennent et de nombreux moyens de communication ; nous souhaitons donc créer une carte électronique qui servira d'interface de contrôle et de communication entre plusieurs périphériques. Nous choisissons pour notre projet le microcontrôleur PIC 18F4550 pour ses capacités de traitement et ses nombreuses liaisons, dont une liaison USB qui permettra la communication avec un ordinateur.

IV.2. Application

Nous réaliserons donc une carte d'interfaçage à base de microcontrôleur entre un PC et différents capteurs et actionneurs électroniques. La présentation des mesures et le contrôle sera réalisé sur le PC au moyen d'une liaison USB. Même si les matériels pouvant être interfacés à la carte sont multiples, Le schéma des communications de ces différents matériels peut être modélisé comme suit :



Figure IV.1 : Le rôle de notre carte d'acquisition.

IV.3. Description du fonctionnement de la carte réalisée

Cette carte peut être exploitée d'une manière standard comme moyen de surveillance et de contrôle dans différents procédés industriels. Les différents ports du microcontrôleur peuvent être utilisés comme entrée pour les signaux provenant des capteurs après conditionnement et comme sortie pour la commande des systèmes externes en marche arrêt et régulation en fonction des entrées (capteurs et bouton de commande). Les données provenant des capteurs existant dans les procédés industriels sont envoyées vers l'ordinateur central pour afficher les valeurs des capteurs et envoyer des signaux de commande aux actionneurs dans le processus industriel fonctionnant par l'intermédiaire de l'interface (notre carte), et d'être le médiateur entre les capteurs et l'ordinateur. La carte d'acquisition de données a pour tâche la traduction des signaux provenant des capteurs et le stockage temporaire des informations collectées à l'intérieur de la carte, puis il vont être transmis à l'ordinateur via le port série (USB).

La figure (IV.2), représente le schéma électrique de notre carte .

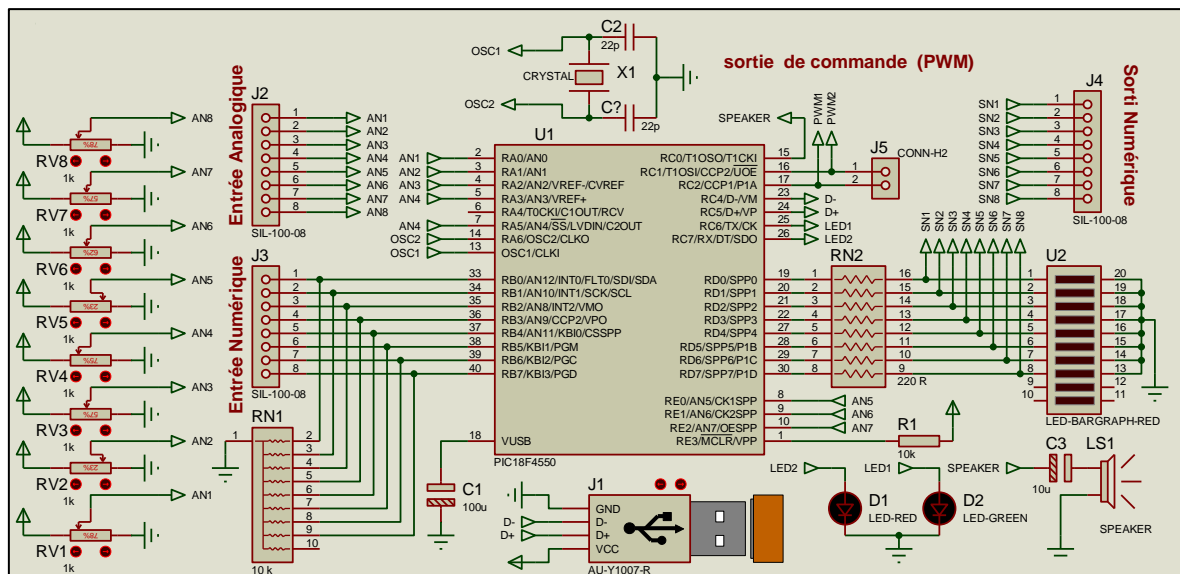


Figure IV.2 : Schéma électrique de la carte d'acquisition Complet.

IV.4. Structure générale de la carte

Pour la gestion de la carte d'acquisition réalisée on a utilisée deux programmes l'un en langage C implanté sur le PIC pour la gestion des ports du PIC, entrées numériques et analogiques et sorties numériques et sorties de commande PWM et l'autre en langage G (LabVIEW), pour la gestion de la commande et du contrôle à travers le PC, et le stockage des données dans fichier Excel , l'affichage sur une interface graphique sur le PC .

IV.5. L'unité de contrôle

Il s'agit d'une carte d'acquisition, permet la communication standard USB-HID, via une interface graphique réalisée par LabVIEW. Cette dernière permet une communication avec le réseau internet, satellite,...etc. L'unité de contrôle est basée sur le microcontrôleur de MICROCHIP PIC18F4550 dont l'avantage est qu'il est équipé d'une interface USB compatible avec le standard USB 2.0.

La figure (IV.3), illustre le type de composant utilisé dans notre projet.

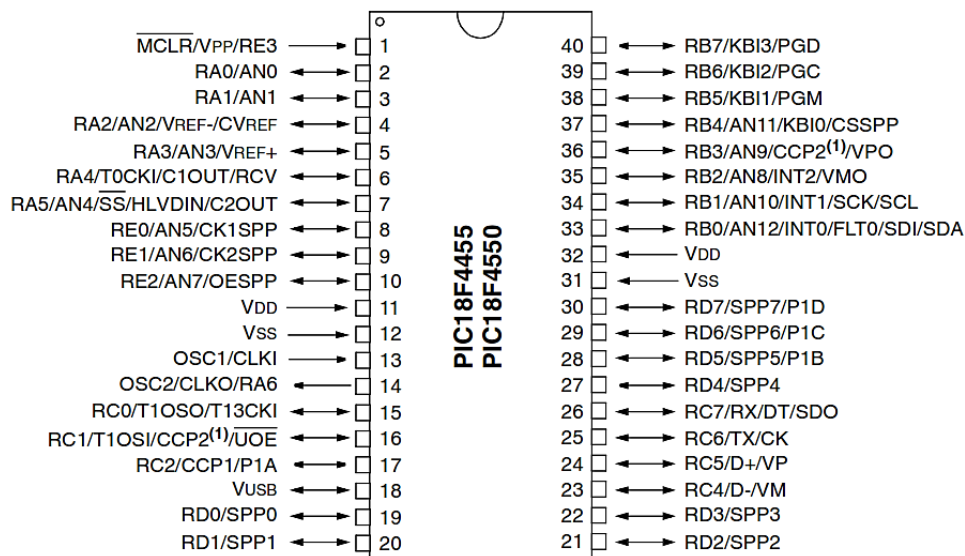


Figure IV.3 : Le Microcontrôleur 18 F4550.40-Pin PDIP .

Nous avons choisi dans notre projet les pins de ce circuit de la manière suivante :

- Les pins de port B (RB0 à RB7), sont des pins d'interface pour les entrées numériques.
- Les pins de port D (RD0 à RD7), sont des pins d'interface pour les sorties numériques.
- Les deux pins de port C (RC1 et RC2), sont des pins d'interface pour la commande PWM.

- Les pins de port A et de port E (RA0 à RA3 et RA5), et (RE0 à RE2), sont des pins d'interface pour les entrées analogiques.
- Le pin de port C (RC0), est relié à l'orateur (speaker), il représente l'état du port USB. Si le port USB est configuré, alors nous allons entendre un bip sonore de 500ms.
- Le pin de port C (RC6), est relié à la led (vert), il représente l'état du port USB. Si le port USB est configuré alors la led s'allume et si le port USB est détaché alors la led s'éteint .
- Le pin de port C (RC7), est relié à la led (rouge), il représente l'état de la connexion avec l'interface exécutable dans le PC. Si l'interface est connecté avec la carte alors la led s'allume et arrêter de clignoter, mais si l'interface n'est pas connecté à la carte la led conserve clignement de période 111ms. Il est équivalent à 9 clignements par seconde .
- Les pins de port C (RC4, RC5), sont reliés respectivement aux broches D- et D+ du connecteur USB.
- Un condensateur chimique de 100 μ F placé sur la borne VUSB (broche 18), pour avoir une tension de 3.3V nécessaire pour le bon fonctionnement du port USB.
- Les pins OSC1 et OSC2 sont reliés à un quartz de 20MHz et de deux condensateurs céramiques pour former l'horloge du microcontrôleur. Dans notre cas on utilise un quartz de 20 Mhz, le microcontrôleur est configuré pour fonctionner à 48 Mhz et ceci afin d'obtenir la vitesse l'horloge permettant d'utiliser le bus USB en mode plein vitesse (96 Mhz).
- Enfin nous avons les pins de programmation PGC, PGD et VPP qui sont utilisés pour programmer le microcontrôleur par la méthode ICSP (In Circuit Serial Programming). Cette méthode permet de programmer le microcontrôleur sans avoir à le retirer de la plaque de circuit imprimé. Toutes ces informations sont illustrées dans le schéma électrique de la carte d'interface.

IV.6. Software et programmation avancé

Dans cette partie de ce chapitre, nous allons évoquer les différents types de programmation qu'on a utilisée dans notre étude. En premier lieu nous aborderons la partie concernant le microcontrôleur PIC18F4550 et ensuite nous verrons la partie concernant le logiciel de commande fonctionnant sur un ordinateur.

IV.6.1. Programmation du PIC18F4550

La partie programme qui gère le PIC18F4550 a été faite en langage C. nous avons utilisé le logiciel mikroC PRO pour PIC v6.6.1 pour écrire notre programme ainsi que le compilateur pour le compiler.

La figure (IV.4), représente la fenêtre de logiciel de programmation mikroC.

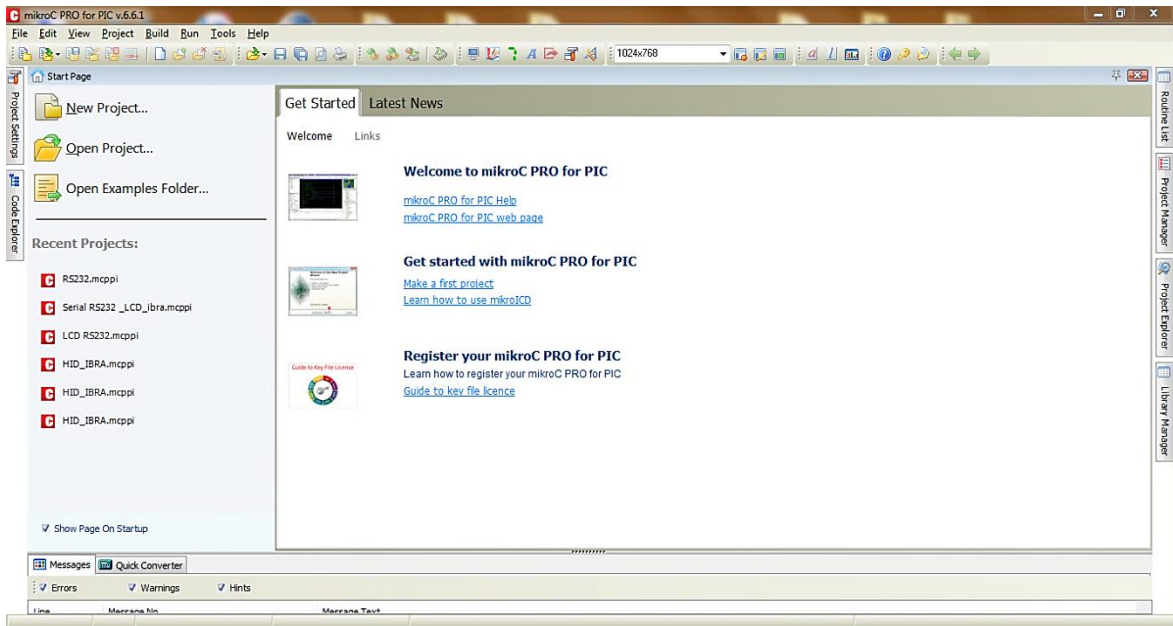


Figure IV.4 : La fenêtre de démarrage de logiciel mikroC PRO for PIC.

Comme vous pouvez le constater sur la figure (IV.5), nous avons utilisé une multitude de fichiers pour constituer le programme qui commande le PIC18F4550, il est un ensemble de procédures permettant d'intégrer la fonction USB dans notre programme.

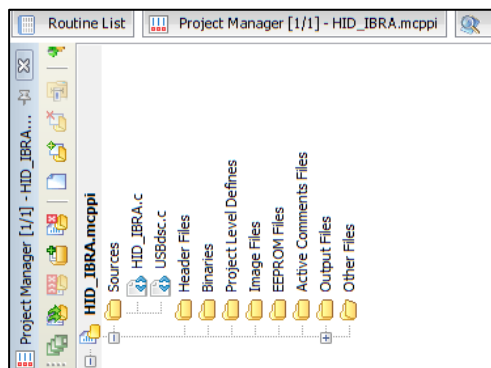


Figure IV.5 : Structure des fichiers constituant le programme pour le PIC18F4550 .

Dans l'organigramme suivant (figure IV.6), nous allons essayer de mettre en évidence les différentes parties de notre programme.

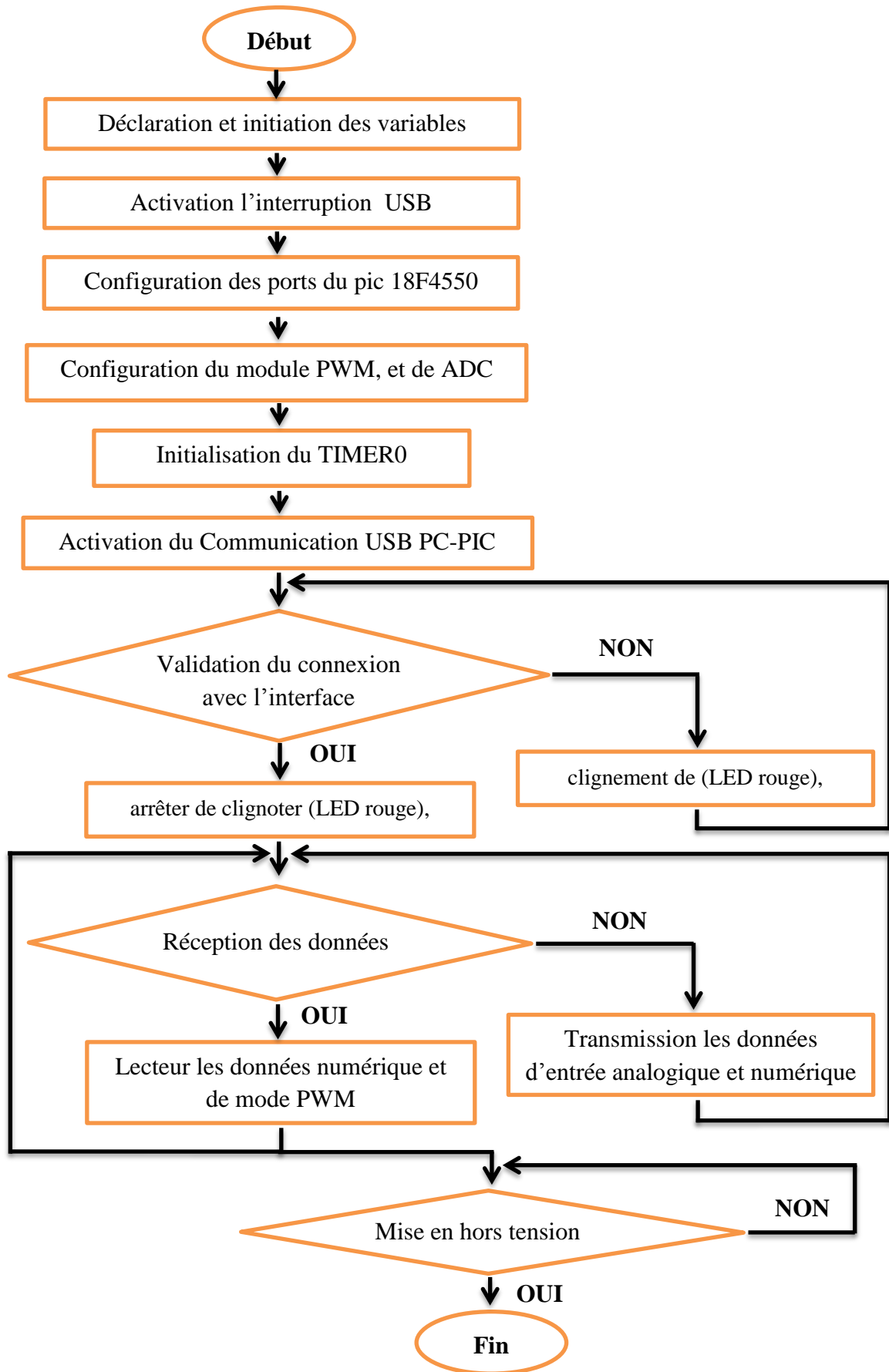


Figure IV.6 : L'organigramme principal du PIC.

Après avoir l'organigramme du programme, nous devrions faire certaines actions en fixant la fréquence de l'horloge du PIC (ce n'est pas la fréquence de résonance du quartz, mais la fréquence de sortie du PLL), fournie par la configuration suivante, (figure IV.7) :

Pour ce faire, logiciel « mikroC PRO for PIC » fournit une interface graphique qui simplifie la configuration du registre de PIC. Nous allons dans le menu (Project -> Edit Project).

IV.6.2. Configuration de registre de PIC dans la fenêtre Edit Project

Pour que l'interface fonctionne correctement, il faut respecter quelques règles:

- ✓ Activer le régulateur de tension USB.
- ✓ Assurer une fréquence d'horloge de 4MHz en sortie de la PLL.
- ✓ Utiliser la sortie de la PLL comme horloge USB .

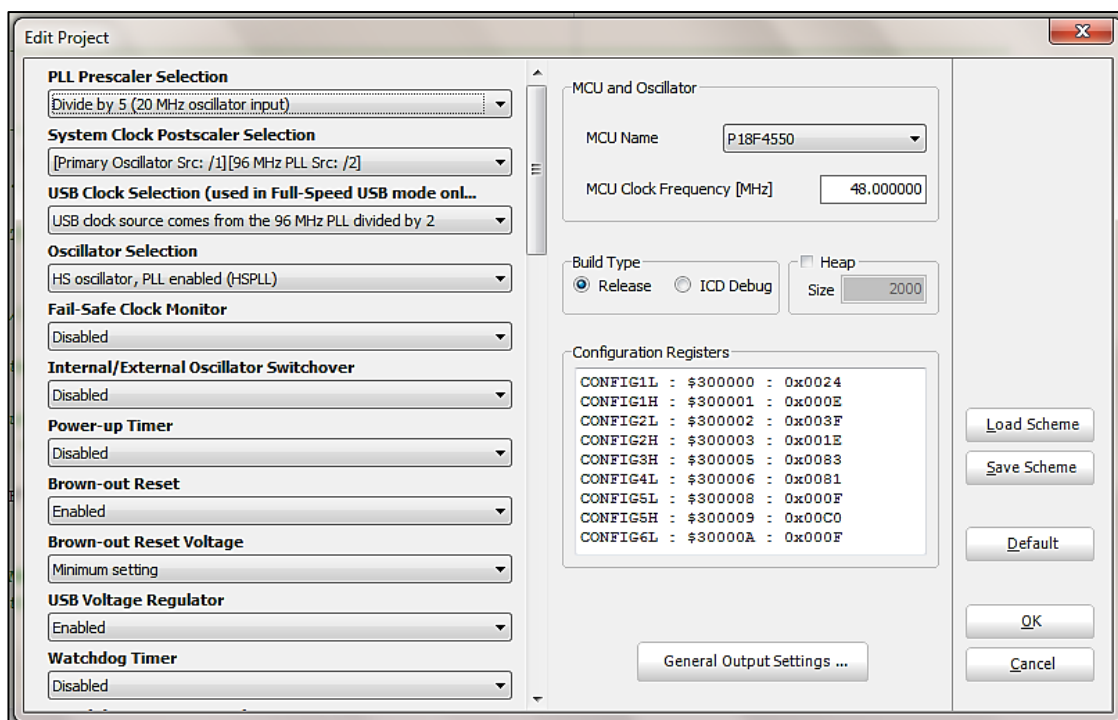


Figure IV.7 : La configuration de registre de PIC 18F4550 dans la fenêtre Edit Project.

IV.6.3. Générer le Descripteur de l'interface

- ✓ Lancer l'utilitaire HID Terminal de mikroC (Tools -> HID Terminal).
- ✓ Onglet Descripteur.
- ✓ Compléter les informations propres à votre interface.
- ✓ Enregistrer le descripteur pour mikroC dans le dossier de votre projet.

La figure (IV.8), représente comment générer le descripteur de la carte d'interface .

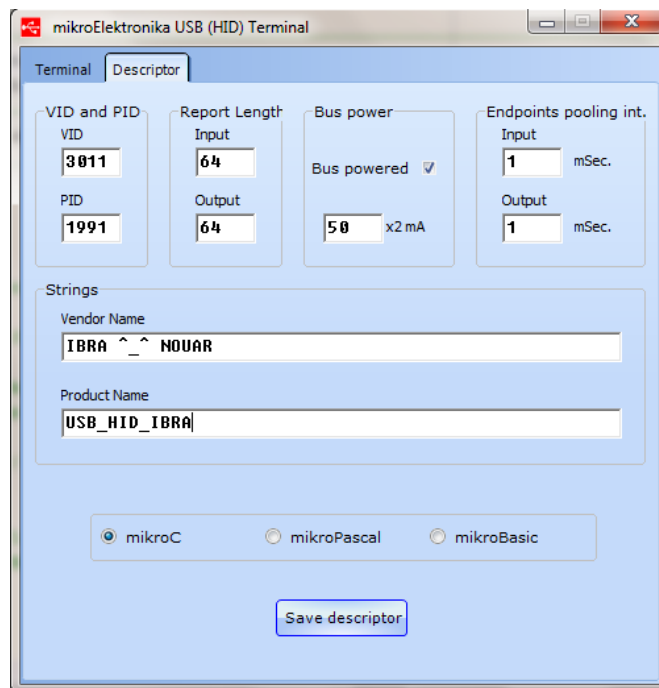


Figure IV.8 : Générer le descripteur de la carte d'interface.

Le tableau suivant représente la description du paramètre de descripteur .

VID : Vendor ID	Numéro de distributeur de machine moyenne si vous avez une société qui vend ces dispositifs il sera la figure de votre entreprise
PID : Product ID	Numéro de produit, si votre société possède plusieurs produits élaborer pour chaque produit, son propre numéro d'identification.
Report length.	Le nombre d'octets qui seront envoyés et reçus, le nombre maximal de 64bits
End points Poolingint.	Il est impératif de prendre le temps d'interruption en (ms), Si ce dernier est de 1 millisecondes la vitesse de transmission et de réception sera 1000 fois par seconde.
Vendor name	Nom de l'entreprise qui apparaît dans le gestionnaire de périphériques
Product name	Le nom du produit dans la barre des tâches lorsque vous connectez le périphérique

Tableau IV.1 : Les paramètres de descripteur.

Après la création du fichier de descripteur, vous l'importez dans votre projet: « uses USBdsc.c » Ensuite, compiler le programme sur logiciel mikroC (génération du programme binaire au format .hex), Ensuite nous allons dans le dossier de votre projet en examinant le fichiers de l'extension (.hex) comme le montre la figure (IV.9),

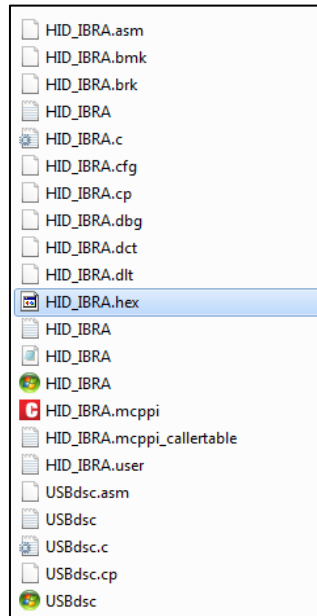


Figure IV. 9 : Le fichier de programme (.hex) .

La partie programmation du PIC18F4550 a été mise au point par le logiciel de simulation Proteus 8 Professional. La figure (IV.10), représente la fenêtre de logiciel Proteus .

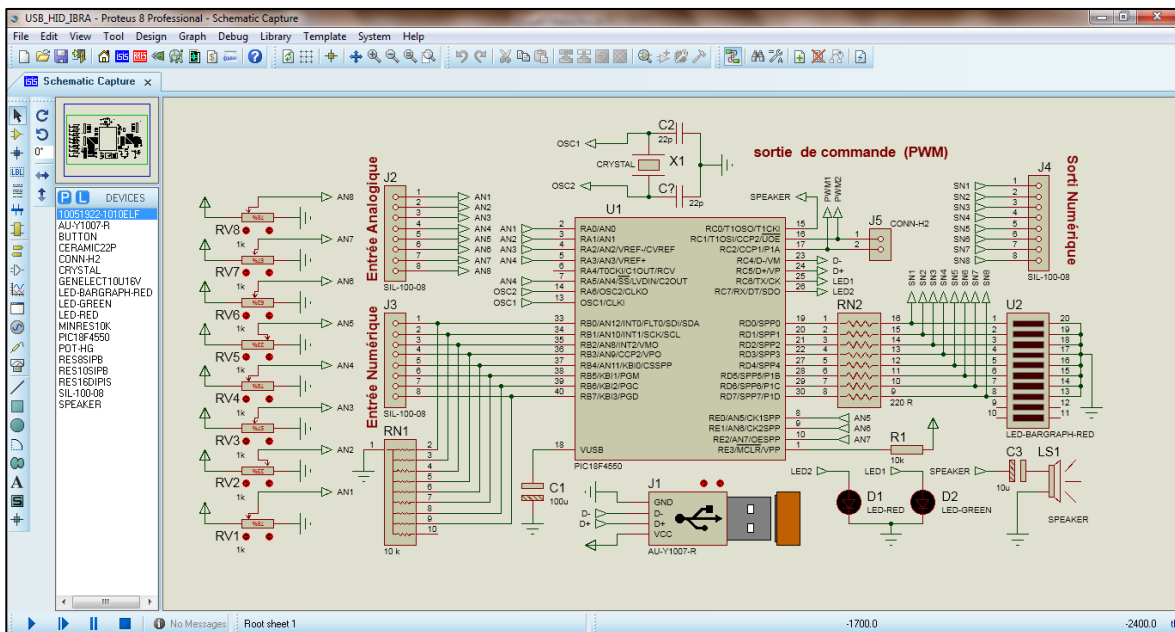


Figure IV.10 : Simulation du circuit sur le logiciel Proteus 8 Professional.

Pour simuler cette carte dans le logiciel de simulation Proteus. Nous avons besoin de deux logiciels et un programme.

- Le 1^{er} est le logiciel mikroC PRO for PIC.
- Le 2^{ème} est le logiciel Proteus Professional.
- Le 3^{ème} c'est un programme appelé Virtuel USB.

Après la compilation du programme sur le logiciel mikroC (génération du fichier .hex), nous utilisons ce fichier (.hex) pour la simulation et pour la programmation du PIC dans la carte. Après les étapes précédentes, on va lancer Proteus et implanter sur le PIC18F4550 le fichier (.hex) avec un double clic sur le PIC, une fenêtre s'ouvre comme le montre la figure (IV.11), pour indiquer le chemin du fichier (.hex) qu'il a implanté.

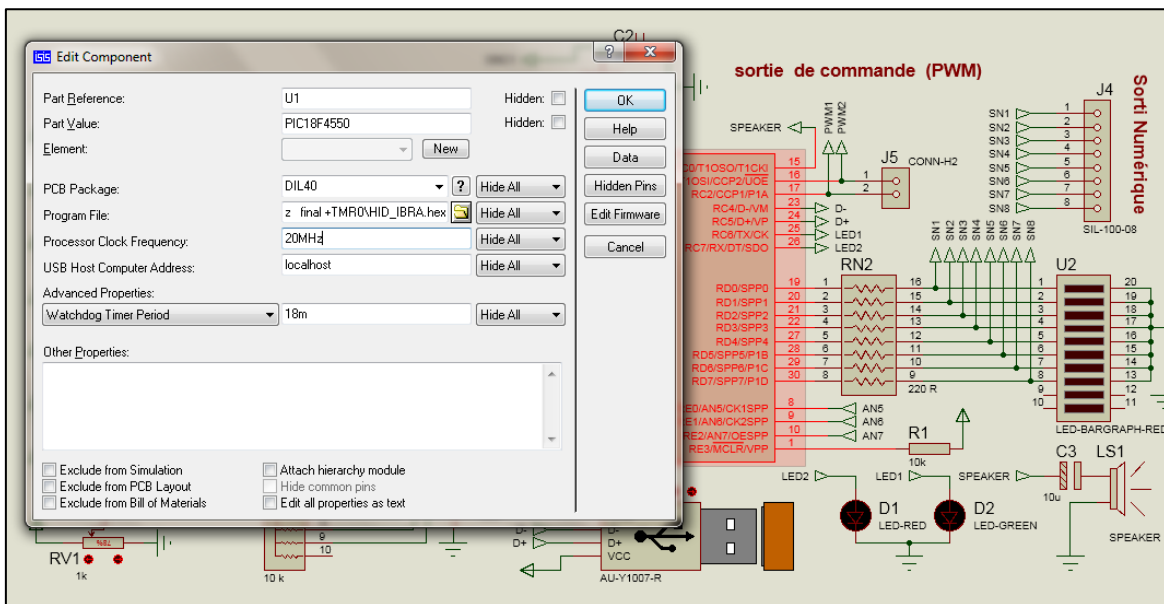


Figure IV.11 : Le chemin du fichier (.hex) .

Lorsque vous avez mis le programme dans le PIC, avant de l'exécuter on a besoin tout d'abord d'installer un programme appelé Virtuel USB, existe sur les fichiers de logiciel Proteus, pour pouvoir détecter le périphérique USB. Pour faire l'installation, nous allons dans le menu Démarrer >Tous les programmes>Proteus>Virtuel USB>Install Virtual USB driver, (figure IV.12).

La figure suivante représente le chemin de programme Virtuel USB .

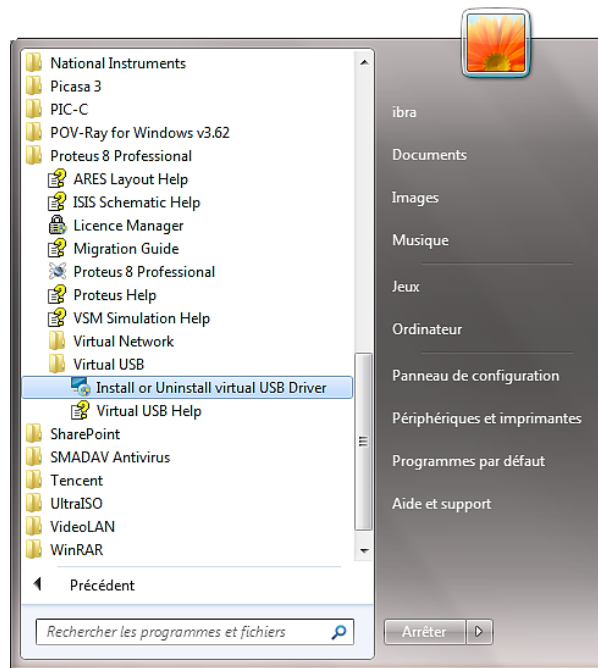


Figure IV.12 : L'emplacement de programme Virtuel USB .

Après l'installation de programme, maintenant on peut lancer la simulation, si la périphérique USB est détecté alors la simulation est bien travaillé, il apparaît dans la barre des tâches, le nom de l'appareil qui a été pré-positionnés dans le descripteur de fichier, dans notre cas USB_HID_IBRA, (figure IV.13).

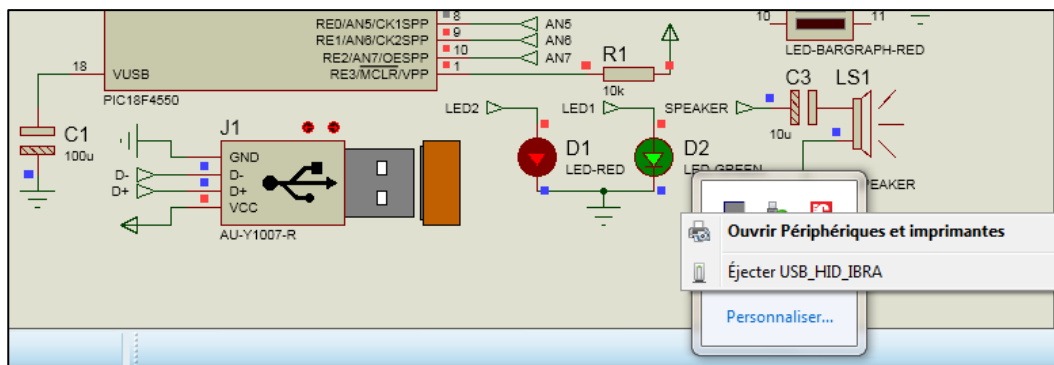


Figure IV.13 : Identification de dispositif USB.

Il apparait aussi sur le gestionnaire de périphérique dans la partie (périphérique d'interface utilisateur), (figure IV.15), comme un périphérique HID (Human Interface Device), Quand on voit les propriétés de l'appareil, on verra son nom comme nous l'avons vu dans la barre des tâches et Comme nous l'écrivions précédemment dans le descripteur de la carte d'interface .

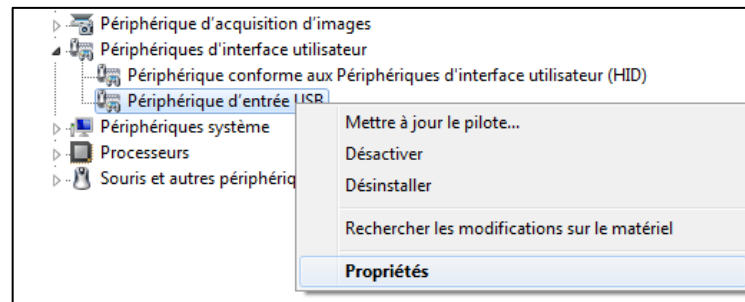


Figure IV.14 : Propriétés de l'appareil.

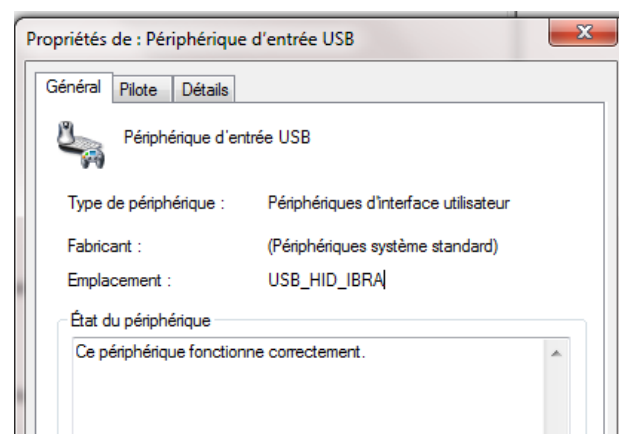


Figure IV.15 : Propriété de la carte d'interface sur le gestionnaire de périphérique.

Après la validation de notre carte dans le logiciel de simulation Proteus , on va passer à la deuxième partie de programmation ce qui bien le logiciel du programmation graphique LABVIEW pour faire l'interface de commande et de contrôle de notre carte .

IV.7. Programmation LABVIEW

IV.7.1. l'interface de commande et de contrôle

Le programme d'interface permettant de contrôler et d'envoyer les différentes commandes depuis l'ordinateur à la carte d'interface ; le programme a été réalisé grâce au logiciel LABVIEW 2015. Après savoir comment communiquer avec la carte d'interface à travers logiciel LABVIEW, nous allons voir en détaille l'interface et toutes les fonctions de commande et de contrôle avec quelque option de réglage.

IV.7.2. Définition de la carte au LabVIEW

La communication entre le LabVIEW et les autres dispositifs virtuels ou réels se fera à travers NI-VISA, (Virtual Instrument Software Architecture), ce dernier est

responsable de la programmation et la reconnaissance des périphériques relié au LabVIEW. Alors Pour reconnaître la carte d'interface comme un dispositif sur LabVIEW, on a besoin d'installer un programme appelé NI-VISA, ce programme est disponible et peut être téléchargé à partir du site officiel de National Instruments et La taille de ce programme comme la Version que nous travaillons sur dans ce projet (visa v5.3.0) est de 667 Mo, après l'installation de ce programme et pour reconnaître la carte, ou n'importe quelle carte comme dispositif sur LabVIEW on doit passer par les étapes suivantes :

Au début on doit ouvrir le logiciel Driver Wizard à partir de :

démarrer >All Programs>National Instruments>Visa>Driver Wizard ,(figure IV.16).

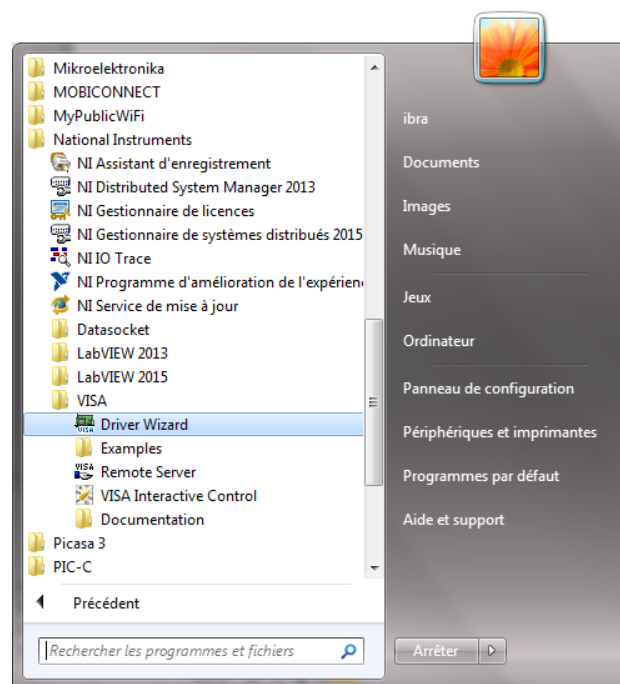


Figure IV.16 : Logiciel de reconnaissance du périphérique connecté avec LabVIEW.

IV.7.3. Configuration de NI-VISA pour le contrôle d'un bus USB

Ce paragraphe résume les étapes de configuration d'un matériel USB pour son contrôle par NI-VISA à partir d'un ordinateur sous Windows.

À ce stade, NI-VISA doit être déjà installé sur l'ordinateur et le matériel USB doit être connecté. De plus, deux étapes sont nécessaires pour configurer un matériel USB pour utiliser NI-VISA :

- ✓ Création d'un fichier INF. (Driver du matériel USB) .
- ✓ Installation du Driver du matériel USB grâce au fichier INF .

Lancer NI-VISA (Driver Wizard) une fenêtre s'ouvre comme suite, (figure IV.17).

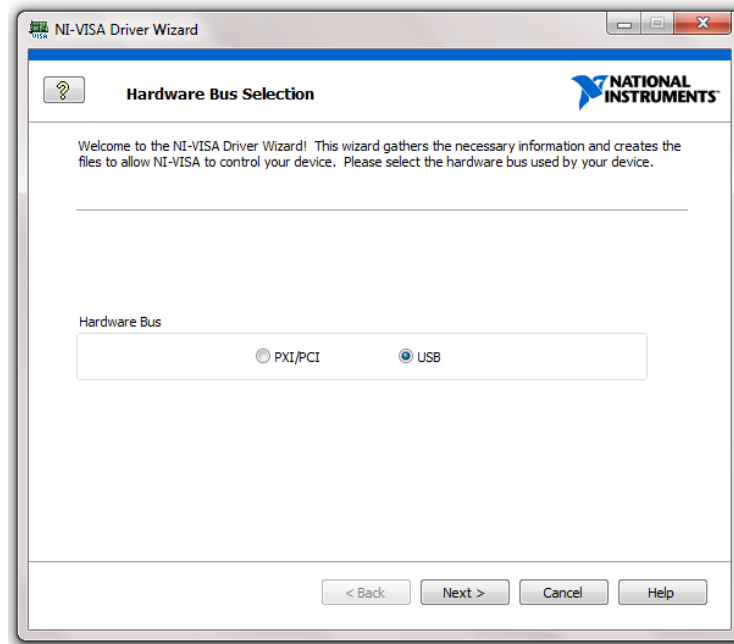


Figure IV.17 : Fenêtre de la sélection du bus matériel de l'Assistant VISA.

Pour la création d'un driver de matériel USB, cliquez sur USB puis sur suivant (Next). La fenêtre des informations de base du matériel de VISA s'ouvre, comme l'indiqué sur la figure (IV.18).

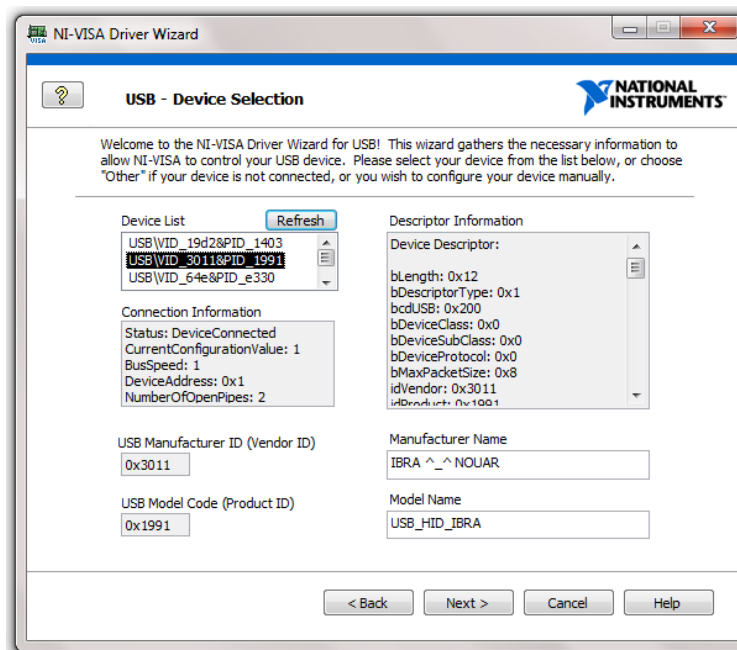


Figure IV.18 : Fenêtre des informations de base sur le matériel de VISA.

Pour la création d'un driver de matériel USB, Choisissez parmi (**Device List**), puis sélectionné le (VID et PID) du matériel, dans notre projet on a (VID_3011 et PID_1991)

puis cliquez sur suivant (Next). La fenêtre des informations de base du matériel de VISA s'ouvre, comme l'indiqué sur la figure (IV.19).

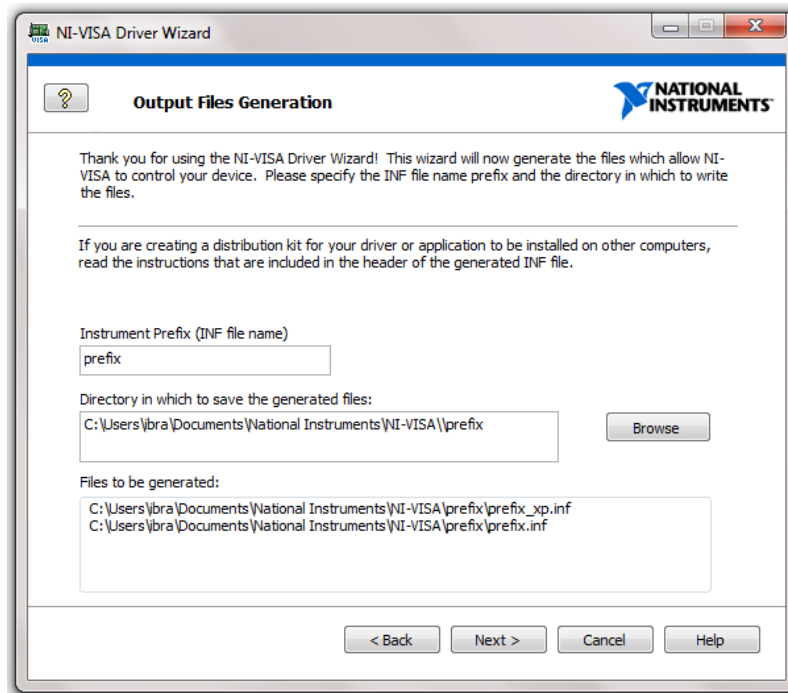


Figure IV.19 : La sélection du dossier d'enregistrement.

Puis cliquez sur suivant (Next) pour installer le driver de ce matériel. Pour s'assurer que le software est installé correctement, nous pouvons le vérifier à partir de gestionnaire de périphériques.

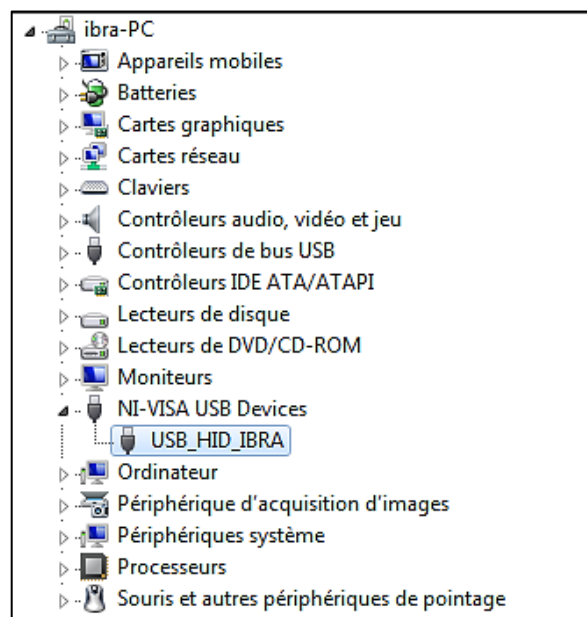


Figure IV.20 : Le nouveau périphérique détecté dans le gestionnaire périphérique.

IV.7.4. Test des communications avec VISA Interactive Contrôle et labVIEW

Ouvrir Measurement & Automation Explorer (NI MAX).

par le chemin << C:\Program Files\National Instruments\MAX. >> ce programme existe sur les fichiers de National Instruments ou sur le bureau sous la forme (icône) après l'installation de logiciel labVIEW .



Figure IV.21 : Icône de NI MAX version v2015.

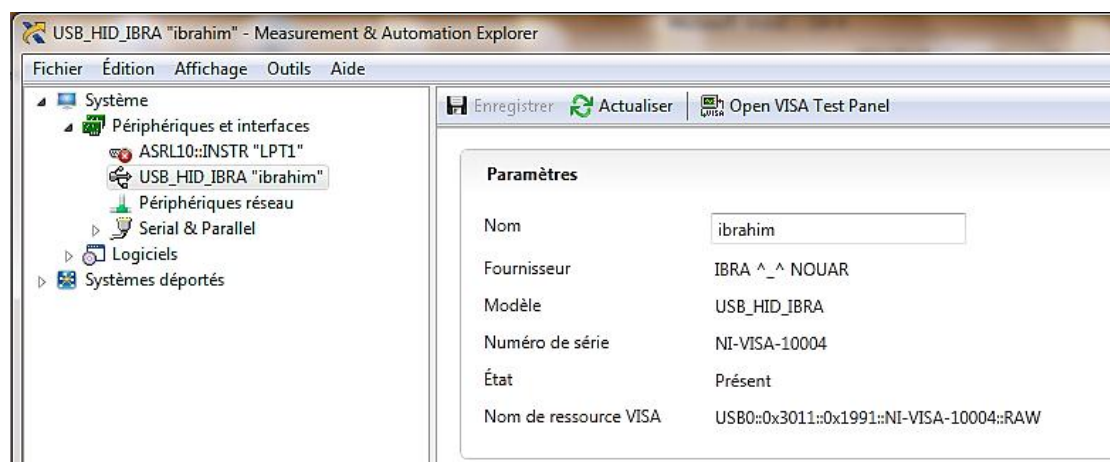


Figure IV.22 : Les paramètres de la carte d'interface dans NI MAX de labVIEW.

Après avoir tous les configurations qui nous permettent d'utiliser notre carte d'interface avec notre programme de logiciel labview (interface d'utilisateur exécutable), la figure suivante représente l'interface d'utilisateur .exe pour la commande et le contrôle dans notre projet.

L'interface d'utilisateur de ce programme est illustrée par la figure (IV.23).



Figure IV.23 : Interface d'utilisateur pour le commande et le contrôle .

Ce dernier contenant trois faces : la face principale (face de COMMANDE ET CONTROLE), nous voyons maintenant et les deux autres face, l'un pour option de réglage , et l'autre pour option de réglage avancer, on va les voir plus tard.

La face principale capable de contrôler jusque 8 entrées analogiques et 8 entrées numériques à la fois ; pour les entrées numérique on a deux possibilité de surveillance visuellement ou acoustiquement ou ensemble. Aussi bien que de registrer les donnés de chaque entrée analogique dans un fichier Excel, Aussi, nous avons différentes façons d'affichage des signaux analogiques, par exemple, comme indicateur ou affichage comme l'oscilloscope ...etc.

Ainsi que la commande de 8 sortie numérique et 2 sortie de mode PWM .

Il pourrait aussi faire le décalage de chaque signal d'entrée analogique, Ainsi que le multiplié pour faire le calibrage des signaux analogiques, ainsi que la commande de chaque sortie manuellement ou automatiquement comme nous le verrons dans les deux faces (d'options avancés et options de réglage).

Pour afficher l'interface d'utilisateur face (OPTION DE REGLAGE), il suffit un clic sur le bouton (OPTION DE REGLAGE), comme nous le pouvons voir sur la figure (IV.23), en haut.

La face (OPTION DE REGLAGE), est illustrée par la figure suivante :



Figure IV.24 : Interface d'utilisateur face (OPTION DE REGLAGE) .

Pour afficher l'interface d'utilisateur face (OPTION AVANCE), juste un clic sur le bouton (OPTION AVANCE) comme nous pouvons le voir sur la figure (IV.24), en haut .

La face (OPTION AVANCE), est illustrée par la figure suivante :



Figure IV.25 : Interface d'utilisateur face (OPTION AVANCE).

IV.8. Réalisation de la carte d'acquisition

IV.8.1. Présentation

La figure suivante représente notre carte sur la plaque d'essai.

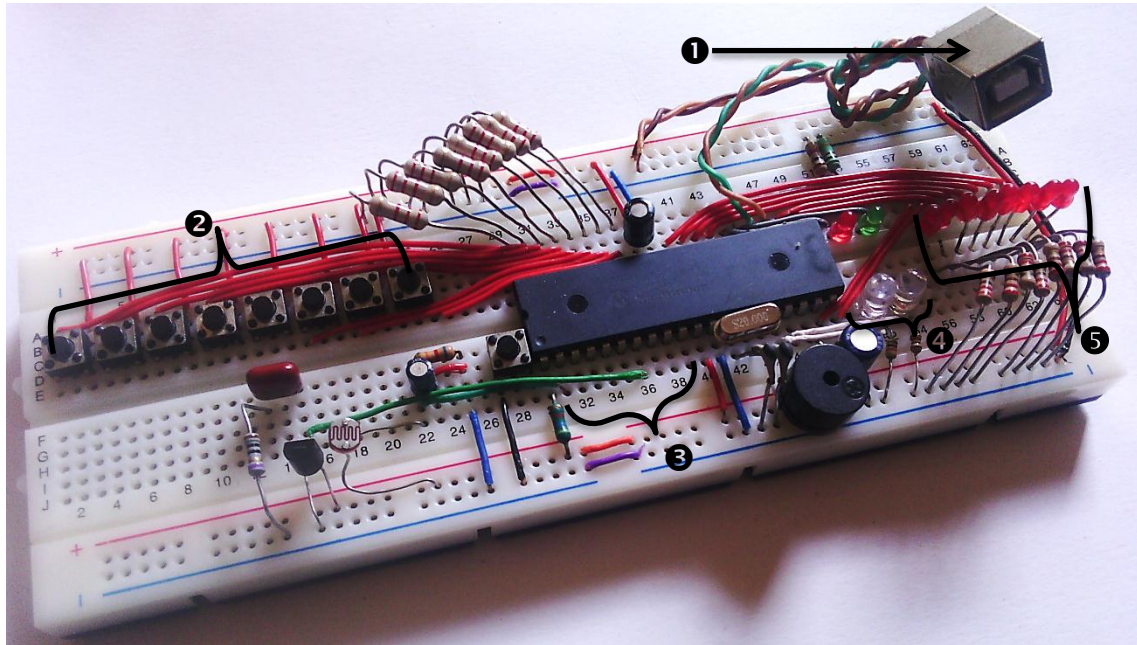


Figure IV.26 : Présentation de la carte.

- ❶ Connecteur USB type B.
- ❷ 8 entrées numériques.
- ❸ 8 entrées analogiques.
- ❹ 2 sorties pour de mode PWM.
- ❺ 8 sorties numériques.

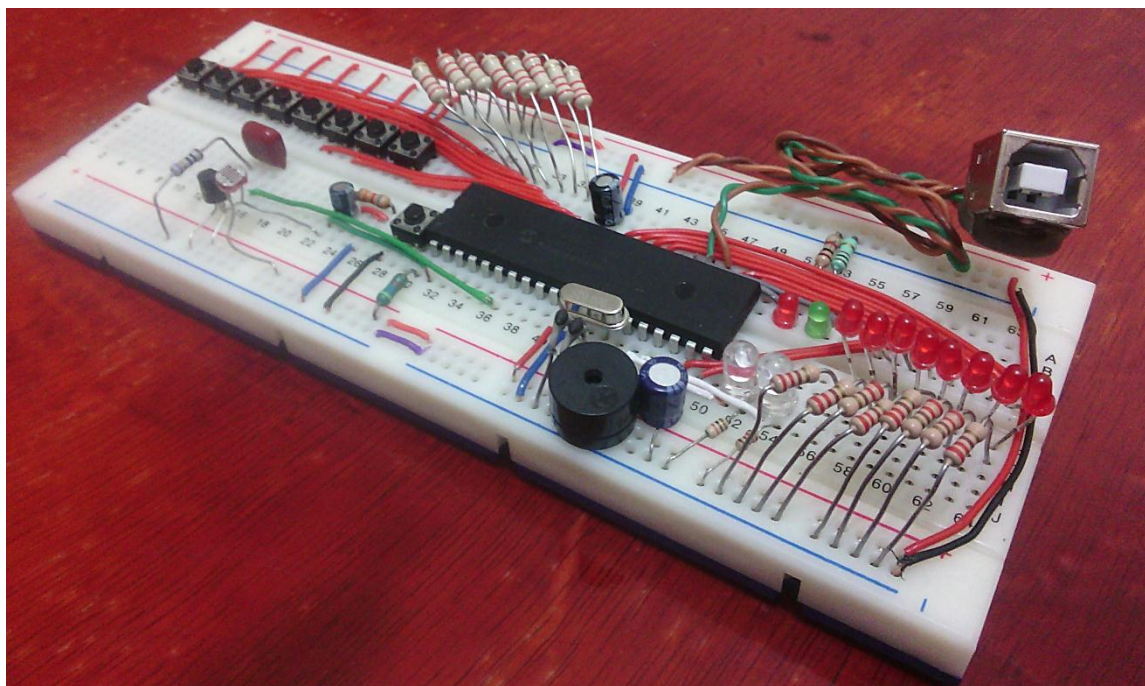


Figure IV.27 : La carte d'acquisition sur la plaque d'essai .

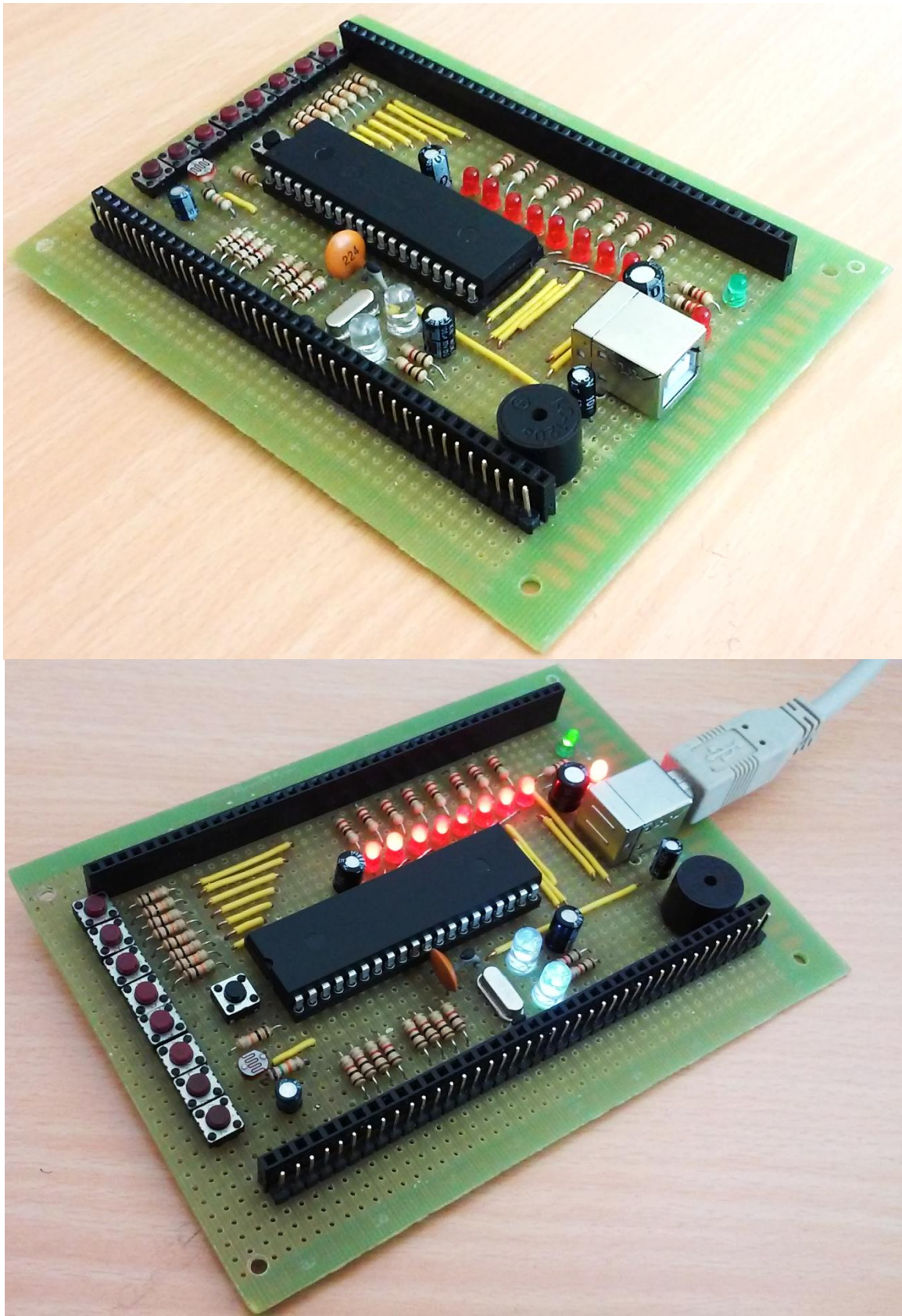


Figure IV.28 : La carte d'acquisition final .

IV.9. Application hardware et software

Après l'application de la tension continue et variable aux différentes entrées de notre carte d'acquisition les résultats seront représentés sur les petites affiches dans l'Interface d'utilisateur sur le côté droit comme suit :

Dans ce cas par exemple, on applique seulement sur la chaine N°1(signal sinusoïdal), et N°2 (signal carré), avec un 'amplitude qui peut varier de 0 à 5V.

Pour voir le signal de chaque chaine on clique sur le bouton de **chenal 1** ou **chenal 2**.

Dans la figure suivant nous avons cliqué sur le bouton de **chenal 1** pour voir le signal (sins)

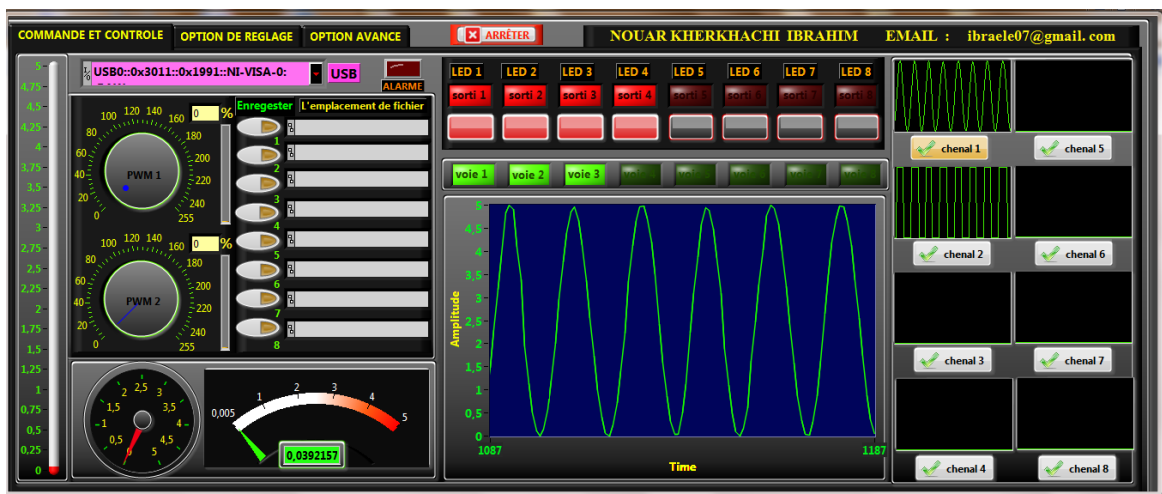


Figure IV.29 : Résultats de signal sinusoïdal appliqué au chenal 1.

Ensuite, nous avons cliqué sur le bouton **chenal 2** pour voir le signal (carré) .

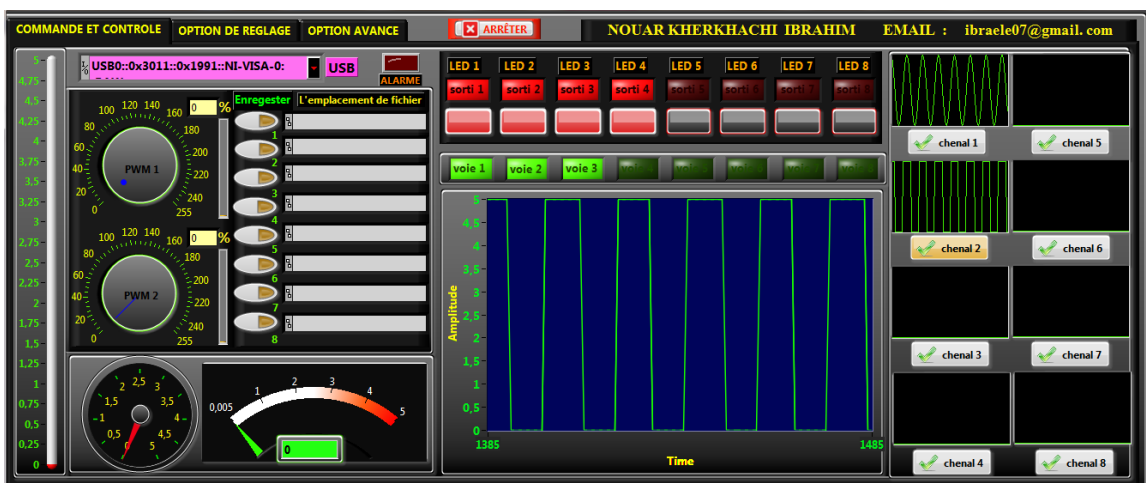


Figure IV.30 : Résultats de signal carré appliquée au chenal 2.

Ensuite, nous avons appliqué des tensions continue et variable de différent amplitudes de 0 à 5V aux différentes entrées de notre carte d'acquisition, les résultats sont représentées comme suit :

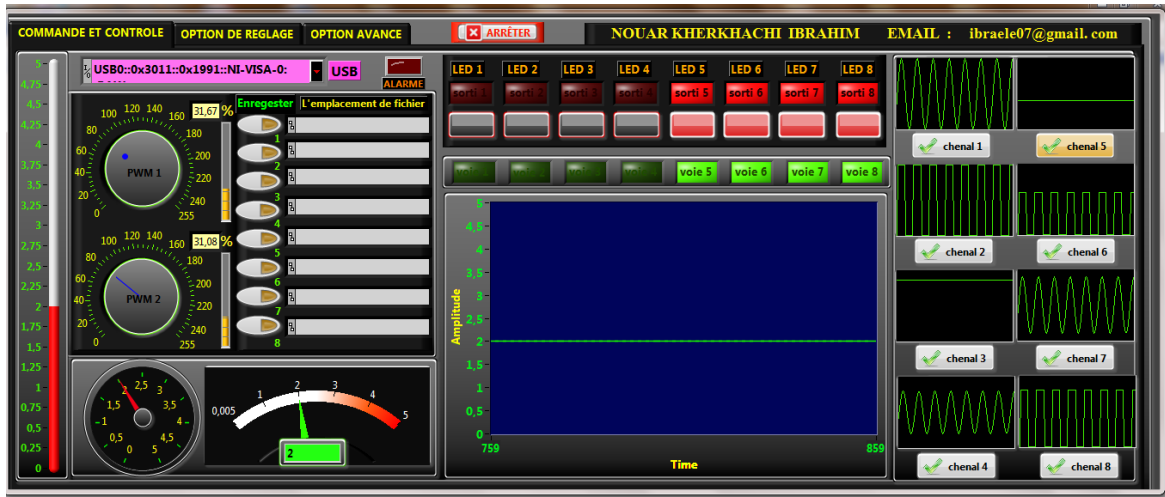


Figure IV.31 : Résultats de signal continu de 2V appliquée au chenal 5.

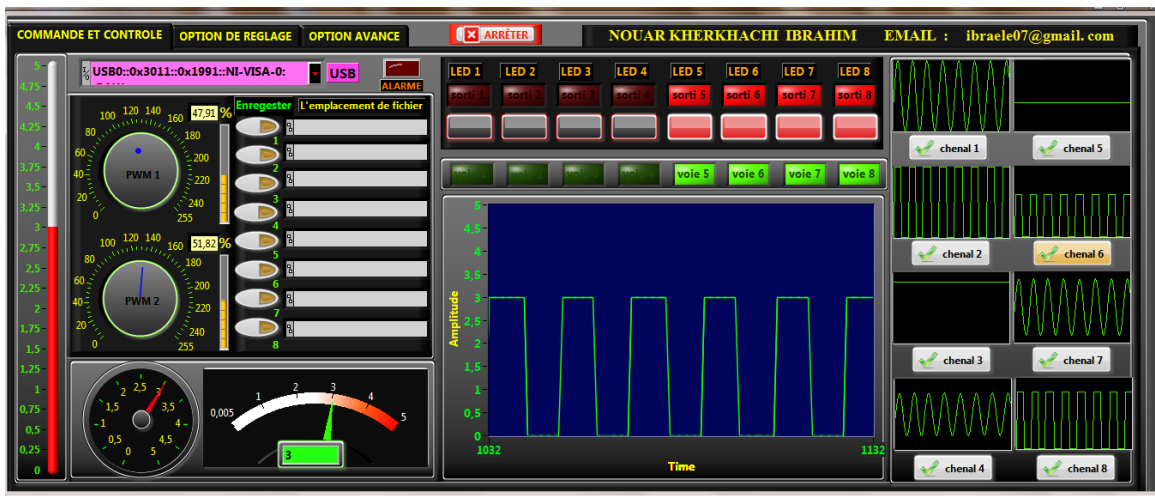


Figure IV.32 : Résultats de signal carré de 3V appliquée au chenal 6.

Pour voir les autres signaux des autres chaînes on doit suivre la même méthode de la sélection des chaînes et on peut aussi enregistrer les valeurs de signal de chaque chaîne dans un fichier (Excel) à travers un clic sur le bouton enregistrer dans l'interface d'utilisateurs. Ainsi, comme nous avons dit précédemment, on peut utiliser deux autres facettes pour faire le calibrage des signaux analogiques ainsi que la commande de chaque sortie manuellement ou automatiquement.

IV.10. Résultats expérimentaux

Ces images sont tirées de l'application pratique de notre carte

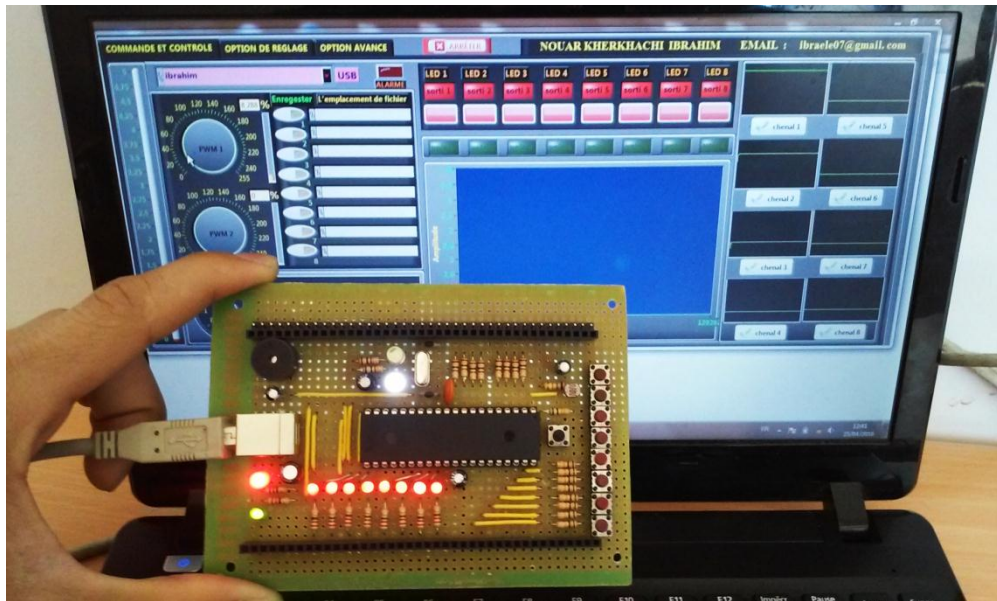


Figure IV.33 : Résultats de commande de 8 sortie numérique et un sortie PWM1 ~8%.

Remarque : Si nous avons vu le côté droit de l'Interface d'utilisateur (les petites affiches), vous remarquerez qu'ils prennent des valeurs aléatoires de quelque volt parce que dans notre cas les entrée analogique en l'air.

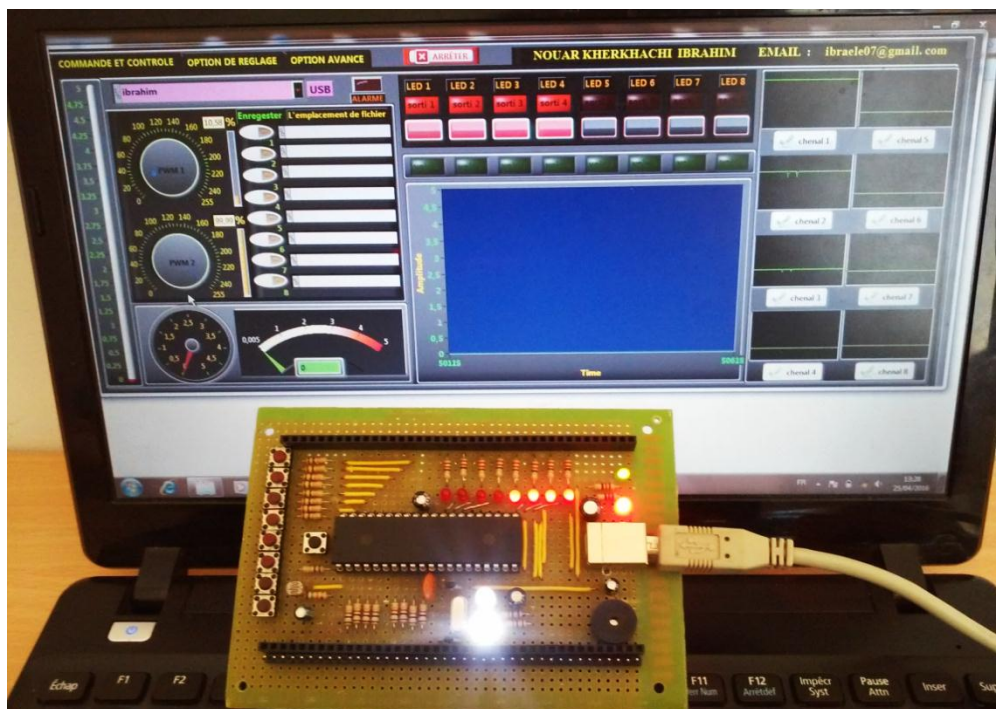


Figure IV.34 : Commande de 4 sorties numériques et 2 sorties PWM1 ~ 10%, PMW~ 99%.

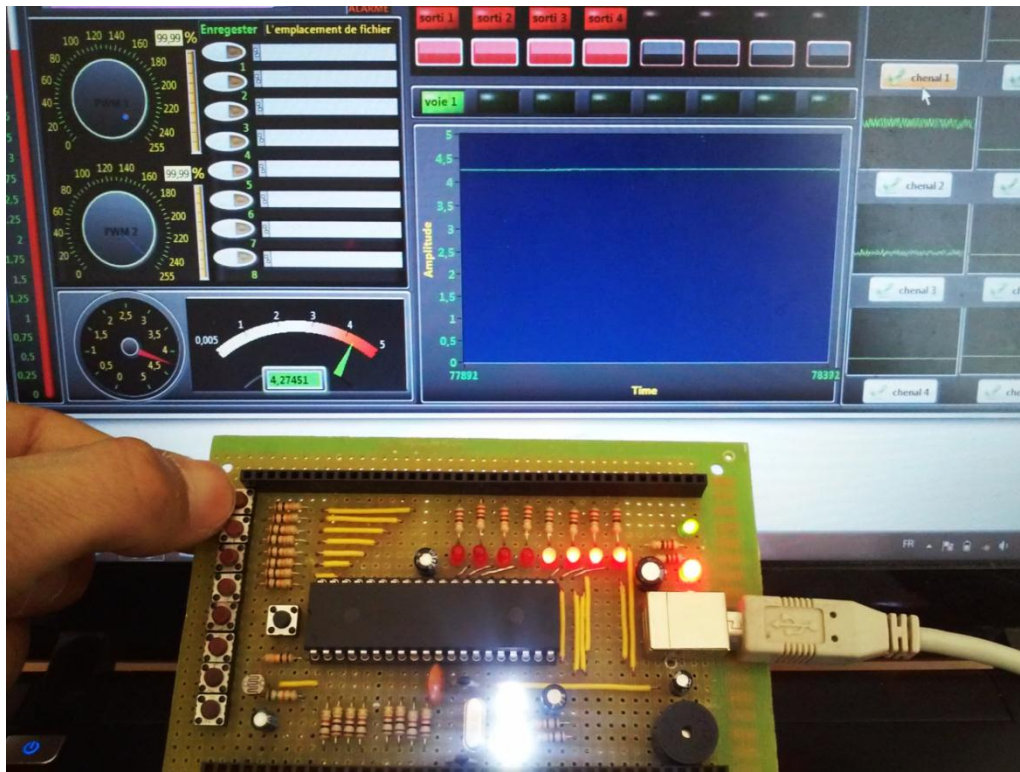


Figure IV.35 : Commande de 4 sorties numériques et 1 entrée numérique et 2 sorties PWM1 ~ 99% ,PMW ~ 99%.

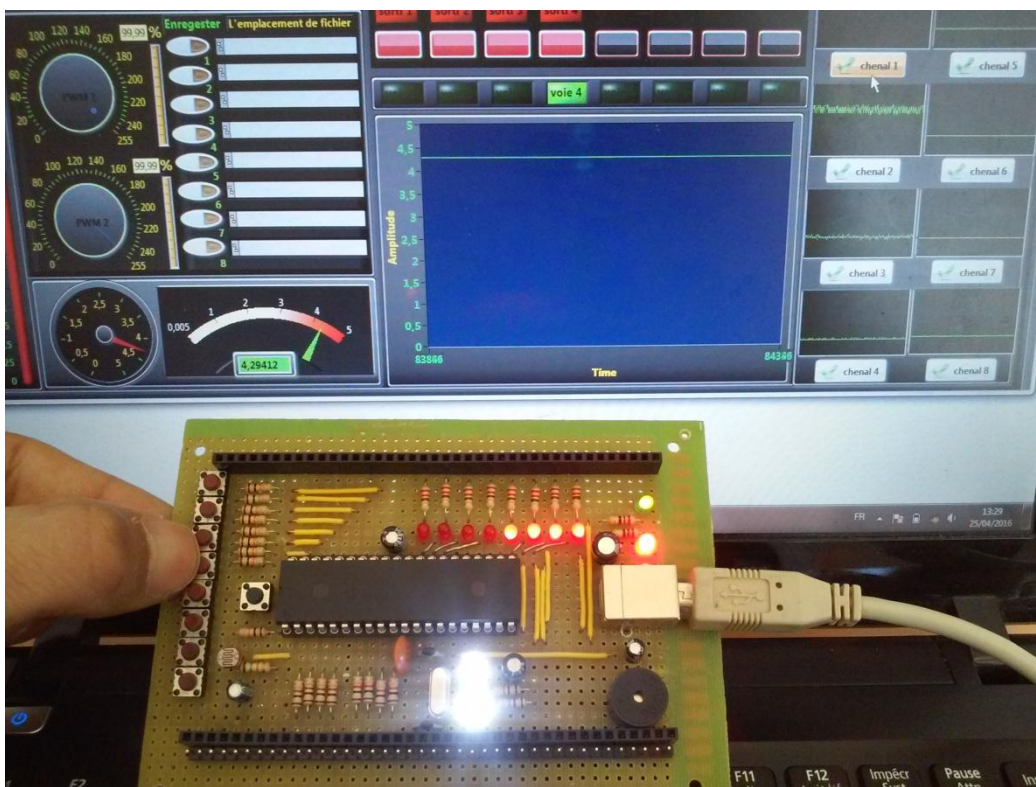


Figure IV.36 : Commande de entrée numérique voie 4.

IV.11. Conclusion

Dans ce chapitre nous avons abordé la partie logicielle qui est composée de deux parties, une partie servant à faire fonctionner la carte de commande et une autre servant à communiquer avec un ordinateur pour pouvoir envoyer des commande à la carte.

Ensuite après les résultats obtenues on peut dire que la carte donne des meilleures résultats puisque elle arrive à donner une ligne droite quand on applique une tension continue qui ne dépasse pas 5V à l'entrée de chaque chaine et même lorsque on applique des tensions des différentes natures (sinusoïdales ,rectangulaires ,triangulaires), à partir du GBF au niveaux de différents chaine on obtient des signaux similaires à l'entrée .

Conclusion générale

Conclusion générale

Dans le cadre de ce projet, nous avons présenté les différentes étapes qui permit de réaliser une carte électronique à base de microcontrôleur PIC18F4550 pour faire la liaison entre le PC, avec une interface graphique exécutable réalisée par LABVIEW, et le milieu extérieur via le port USB, compatible avec l'USB 2.0, et plus précisément la norme USB Full Speed. Cette carte consiste de modules d'acquisition des signaux analogiques et numériques, numérisation, transmission vers l'ordinateur et de restitution des données et la visualisation sous forme de graphes, ainsi que pour commander les actionneurs à travers le PC. De plus, les domaines d'utilisations par rapport à sa importance sont le domaine de contrôle (surveillance) et de commande (on peut commander à travers le mode de commande PWM (MLI) ou la commande numérique où dans ce type de commande on peut commander jusque aux 8 actionneurs manuellement ou automatiquement selon le type du choix de l'interface de l'utilisateur et aussi surveiller jusqu'aux 8 capteurs analogiques et 8 capteurs numériques au même temps, avec la possibilité d'enregistrer les données de chaque entrée analogique sur un fichier Excel.

La réalisation de ce projet nous a énormément appris beaucoup de choses que ce soit sur le plan de la réalisation, que sur le plan de la programmation des microcontrôleurs (programmation embarquée), et la programmation software pour la commande et le contrôle de tous les options de microcontrôleur à travers le PC.

Ce travail reste, comme toute œuvre humaine, incomplet et perfectible, nous recommandons d'en améliorer la conception et pour cela nous proposons ci-dessous des améliorations pour les futurs développements :

- ✓ 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 Gbit/s,
- ✓ Etendre la gamme de la tension à une valeur plus importante,
- ✓ L'utilisation de réseau internet pour la communication avec l'interface de l'utilisateur.

Résumé

Il est difficile, de nos jours, de ne peut pas associer l'informatique à l'électronique. En effet, beaucoup de montages se programment par les ports parallèles et ports séries d'un PC.

La généralisation de l'utilisation des microcontrôleurs à changer profondément le concept de la mesure traditionnelle ce qui à influencer directement sur le coût et la forme des appareils. Les micro-ordinateurs sont des plates-formes privilégiées pour la réalisation des applications de commande et d'acquisition, de traitement et de restitution des données, avec la visualisation sur l'écran des résultats expérimentaux.

L'objectif de notre travail est de réaliser une carte d'acquisition des signaux analogiques (numériques) afin de les visualiser sur le PC par le port série (USB) qui comprend des modules d'acquisition, numérisation, transmission vers l'ordinateur et la visualisation sous forme de graphes, en exploitant l'extention.exe réaliser par le logiciel LABVIEW.

Mots clés : Conversion, Carte d'interface, Microcontrôleur, PIC 18F4550, LABVIEW.

Abstract

It is difficult nowadays to not able to associate computing with electronics. Indeed, many assemblies are programmed with parallel and serial ports of a PC.

The widespread use of microcontrollers profoundly changes the traditional concept of the measure to which direct influence on the cost and form of devices is done. Microcomputers are privileged platforms for the implementation of control applications and data acquisition, processing and retrieval of data, with the map on the screen of the experimental results.

The aim of our work is to achieve an acquisition card of analog signals (digital) to view on the PC via the serial port (USB), which includes acquisition modules, scanning, and transmission to the computer visualization as graphs with the use of extention.exe which is realized by the LABVIEW software.

Keywords: Conversion, Interface card, Microcontroller, PIC 18F4550, LABVIEW.

Bibliographie

- [1] Site officiel de National Instruments, "Introduction à l'acquisition de données", Date de publication: mars 15, 2016 <<http://www.ni.com/white-paper/3536/fr/>>.
- [2] Weisend J.G., " Handbook of cryogenic engineering ", Taylor & Francis, 1998.
- [3] Eric M., " Cours d'Informatique Industrielle ", livre, Université de Caen 2005-2006.
- [4] Olivier F., " la chaîne d'acquisition – restitution ", livre, Edition 2000.
- [5] Kaid N et Moulhi A., " exploration cardiovasculaire par étude corrélative des activités électrique et hémodynamique cardiaque et l'activité rhéologique pariétale ", mémoire de magister électronique biomédical université Abou BakrBelkaid -Tlemcen 2011-2012.
- [6] Magarotto E., "Support de cours de Transmission et Acquisition de Données", Licence EEA – IE Version provisoire du 15 décembre 2003.
- [7] Matthieu K., "l'USB et sa norme ", v1.0 - 21 septembre 2002 < <http://u.s.b.free.fr>>
- [8] Jeff K., " LabVIEW Programmation et applications ", 3eme édition ,Austin, Texas 4 février 2015 .
- [9] Heidari Y et Blondy G., "Banc de Puissance", Tuteur du projet: M. Mikolajczyk, Université Blaise Pascal, Projet 2007-2008.
- [10] Decourt R., " labview programmation avancé ", livre, Edition 2011.
- [11] National Instruments," Initiation à LabVIEW", Octobre 2005.
- [13] Emmanuel G.," Introduction à LabVIEW Premiers pas vers l'expérience", Novembre 2007.
- [12] National Instruments, " LabVIEW. Principes de base de LabVIEW" , Août 2007.
- [14] Site officiel de National Instruments <https://www.ni.com/academic/why_labview/f/>
- [15] David F., Pierre-Armand D., "exploitation de labview dans l'enseignement: un langage de programmation multiplateformes ", article, page de 1 à 6 , 05/12/2008 .
- [16] BOUANANI M.,KERZABI F., " étude et réalisation d'un carte d'acquisition pour des signaux lents ", Mémoire de master en électronique, Université Abou BekrBelkaïd de Tlemcen, Année 2012-2013.

- [17] Saadi R et Salhi N., " Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc ", Mémoire de master en électronique, Université Mohamed KhiderBiskr , Promotion 2009-2010.
- [18] Bekhouche S., " Réalisation d'un oscilloscope numérique portatif à affichage sur écran PC via le port USB ", Mémoire de master en électronique, Université Mohamed KhiderBiskr , Année 2012-2013.
- [19] Smail S., "Commande de l'éclairage public et mesure de la température à base de PIC18F4550 ", Mémoire de master en électronique, Université Mohamed Khider Biskr , Juin 2013.
- [20] Alibi E et Jawadi S., " Conception et réalisation d'un enregistreur de données ", Université Virtuelle de Tunis , Année 2010-2011.
- [21] <http://microchip.com/>, " PIC18F4550 Data Sheet ", Mai 2013.
- [22] Guessoumi F et Saadi A., " Commande de panneau solaire à l'aide d'une carte à pic ", Mémoire de master en électronique, Université Mohamed Khider Biskra, 2009-2010.
- [23] Pascal M., "Apprendre la programmation des PIC High Performance par l'expérimentation et la simulation", DUNOD Editions : ETSF, Paris, France, Juillet 2010.