



Mémoire de Projet de fin d'étude

Préparé par

EL YDRISSI MASSAAB

Pour l'obtention du diplôme

Master Sciences et Techniques

Électronique, Signaux et Systèmes Automatisés

(E.S.S.A)

Intitulé

**Conception et réalisation d'un enregistreur de
données multi entrées/sorties**

Encadré par :

Pr. F.ABDI

Mr .SERBOUTTI MOHAMED (RAYNOV)

Soutenu le 13 Juin 2016, devant le jury composé de :

Pr. F.ABDI

: Encadrant

Mr. M.SERBOUTI

: Encadrant (RAYNOV)

Pr. ALI AHAILOUF

: Examineur

Pr. N.SAID ECHATOUI

: Examineur

À ma mère et mon père

En témoignage de leur affection

*Leurs sacrifices et leurs précieux conseils qui
m'ont conduit à la réussite dans tous ce que je fais ;*

À ma sœur et mes frères

En leur souhaitant la réussite dans leur vie,

À tous mes proches

*À tous ceux qui m'ont aidé afin de réaliser ce
travail, et à tous ceux que j'aime et qui m'aiment.*

Remerciements

Premièrement nous remercions Dieu source de toute connaissance.

Au terme de ce travail, je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mes tuteurs de stage au RAYNOV M. Mohamed SERBOUTI et M. Jamal EL MERABT pour tout le temps qu'ils m'ont consacré, leur directives précieuses, et pour la qualité de leur suivi durant toute la période de mon stage.

Je tiens aussi à remercier vivement le directeur du RAYNOV M. Mohamed SERBOUTI qui a accepté de m'accueillir en stage au sein de son organisme.

Je voudrai remercier également tout le personnel du RAYNOV pour sa gentillesse et son soutien notamment M. Abdrahim MAAKOUL.

Mes profonds remerciements vont à mon encadrant à la FST M. Farid ABDI qui a accepté d'encadrer mes travaux durant ces 4 mois de stage.

Mes plus vifs remerciements s'adressent aussi à tout le cadre professoral et administratif de La FST de Fès.

Mes remerciements vont enfin à toute personne qui a contribué de près ou de loin à l'élaboration de ce travail.

Résumé

Le présent document constitue le fruit de notre travail accompli dans le cadre de projet de fin d'études au sein de RAYNOV. L'objectif de ce projet est la conception d'un enregistreur de données multi inputs/outputs et le contrôle des sorties.

Nous avons commencé par faire l'étude des différents enregistreurs de données existants appartenant à différentes sociétés afin d'avoir une idée claire sur les difficultés et les problèmes que nous pouvons rencontrer lors de notre projet. Ensuite, nous avons élaboré un plan de travail qui nous a permis, d'aboutir à notre objectif.

Nous avons développé, sous LabVIEW, un software de notre enregistreur, non autonome, de données et la simulation de différentes tâches, et le fonctionnement global de notre programme. Nous avons alors pensé à rendre notre système totalement autonome.

Nous avons, ainsi, pu concevoir et réaliser un système basé sur le circuit intégré PIC18F4550. interfacé sous LabVIEW., totalement autonome, Qui apporte plus de fonctionnalités et peut réagir tout seul en cas d'urgence par contrôle de ses sorties. Contrairement aux autres enregistreurs qui font seulement l'acquisition et l'enregistrement de données.

Abstract

This document is the fruit of our work in the draft framework graduation in Raynov. The objective of this project is designing a data logger multiple inputs / outputs and control outputs.

We began by studying the various existing data loggers from different companies to have a clear idea about the difficulties and problems we may encounter in our project, Then we developed a work plan that allowed us to reach our goal.

We have developed in LabVIEW, a software of our recorder, not autonomous, data and simulation of different spots, and the overall operation of our program. We thought to make our completely autonomous system.

We have been able to design and implement a system based on the integrated circuit PIC18F4550, Completely autonomous, Who brings more features and can respond alone in emergencies by monitoring outputs. Unlike other loggers that only the acquisition and recording of data.

Sommaire

Remerciements	2
Résumé	3
Abstract.....	4
Liste des figures.....	7
Liste des tableaux	8
Glossaire Acronymes	9
Organisme d'accueil :	10
Introduction Générale	11
Chapitre 1 : Généralités sur les enregistreurs de données.....	13
1. Introduction	14
2. Le fonctionnement d'enregistreur de données.....	14
3. Les différents types des enregistreurs de données.....	15
4. L'utilisation d'enregistreur de données.....	16
5. L'enregistrement de données sur ordinateur	16
1) Visualisation en temps réel.....	17
2) analyse en temps réel.....	17
3) fonctionnalité définie par l'utilisateur	18
4) Téraoctets de stockage de données.....	18
5) La connectivité réseau	18
6. Exemples des enregistreurs dans le marché.....	19
7. Conclusion.....	21
Chapitre 2 : Choix de la Technologie et l'environnement de développement	22
I. Choix de microcontrôleur.....	23
1. Introduction	23
2. Définition d'un PIC	24
3. Caractéristiques principale de PIC18F4550	25
4. Brochage de PIC18F4550.....	27
II. Les logiciels de développement.....	27
i. MicroC.....	27
ii. ISIS PROTEUS	29
iii. LabVIEW	30
III. Conclusion.....	34
Chapitre 3 : Développement software de l'enregistreur de données	35

I. Introduction	36
II. Communication entre le hardware et le software	37
A. Introduction à la communication USB	37
B. Établir la communication entre le microcontrôleur et LabVIEW	38
III. Bus CAN	42
1. Introduction	42
2. Réalisation :	43
IV. Développement d'enregistreur de données avec ordinateur	44
1. Développement de l'interface.....	44
2. Programmation de microcontrôleur.....	49
3. Conclusion.....	51
V. Développement d'enregistreur de données autonome	51
1. Développement de l'interface.....	51
2. Les trames et adressage	56
3. L'envoi des trames	59
4. Réception des trames	64
VI. Description de fonctionnement et simulation	68
VII. Conclusion	76
Chapitre 4 : Routage et la Conception de circuit imprimé	35
1. Introduction	78
2. Réalisation de circuit imprimé	78
3. Génération de l'application exécutable	80
4. Création de l'installeur	81
5. Conclusion.....	83
Conclusion Générale	84
Bibliographie	85

Liste des figures

Figure 1 : les différents types des enregistreurs de données	15
Figure 2: domaines d'utilisation des enregistreurs de données	16
Figure 3: data Taker Universal.....	20
Figure 4: enregistreur de données DT80.....	21
Figure 5 : le microcontrôleur Atmega	24
Figure 6 : Brochage du PIC18F4550	27
Figure 7 : les étapes de programmation d'un microcontrôleur.....	28
Figure 8 : Interface du logiciel MikroC.....	29
Figure 9 : Interface du logiciel ISIS.....	30
Figure 10 : fenêtre d'édition	30
Figure 11 : programmation graphique.....	31
Figure 12: acquisition, analyse et présentation	31
Figure 13 : Logo officiel USB	37
Figure 16 : HID Terminal	38
Figure 17 : fichier USBdsc.c	39
Figure 18 : l'utilisation de NI-VISA	40
Figure 19 : Écran de sélection du bus matériel de l'Assistant VISA DDW	41
Figure 20 : Écran des informations de base sur le matériel de VISA DDW	41
Figure 21 : l'installation de driver INF.....	42
Figure 49 : le bus CAN.....	43
Figure 51 : MCP2551	43
Figure 52 : MCP2510.....	43
Figure 53 : réalisation de bus CAN	44
Figure 22 : les onglets de l'interface d'enregistreur avec ordinateur.....	45
Figure 23 : entrées numériques	46
Figure 24 : entrées analogiques.....	46
Figure 25 : programmation.....	47
Figure 26 : les cas possible par une condition	47
Figure 27 : états des entrées/sorties	48
Figure 28 : affichage	48
Figure 29 : affichage sous forme d'un graph.....	49
Figure 30: processus de fonctionnement au niveau de microcontrôleur	50
Figure 31 : les onglets de l'interface	52
Figure 32 : onglet Display.....	52
Figure 33 : Digital inputs	53
Figure 34 : Analog inputs.....	54
Figure 35 : Programming	54
Figure 36 : les cas d'une condition	55
Figure 37 : bus CAN	55
Figure 38 : onglet Graph.....	56
Figure 39 : présentation de la trame.....	58
Figure 40 : VISA Open.....	60
Figure 41 : VISA Write	60
Figure 42 : VISA USB control.....	61
Figure 43: VISA Close	61
Figure 44 : les cas d'une condition	63
Figure 45 : onglet de programmation.....	63
Figure 46 : l'envoi d'une condition	64
Figure 47: la mémoire RAM USB PIC18F4550.....	65
Figure 48: processus de la réception de données	67
Figure 54 : schéma synoptique de système de fermentation de la levure.....	70
Figure 55 : onglet Display.....	72
Figure 56 : configuration des entrées numériques d'enregistreur de données	73
Figure 57 : configuration des entrées analogiques	73

Figure 58 : configuration d'onglet programmation.....	74
Figure 59 : l'affichage des résultats dans le graph.....	74
Figure 60 : affichage des résultats dans des jauges.....	74
Figure 61 : simulation d'un cas de fonctionnement.....	75
Figure 62 : le fichier d'enregistrement des données.....	75
Figure 63 : PCB de la carte d'enregistreur.....	78
Figure 64 : vue réel.....	79
Figure 65 : la carte après la conception.....	79
Figure 66: création d'application (EXE).....	80
Figure 67 : les propriétés de l'application.....	81
Figure 68 : application .exe.....	81
Figure 69 : création de l'installateur.....	82
Figure 70 : les propriétés d'installateur.....	82
Figure 71: l'installateur.....	82
Figure 72: installation de l'application.....	83

Liste des tableaux

Tableau 1 : les caractéristiques générales de PIC18F4550.....	26
Tableau 2 : types de fils.....	33
Tableau 3 : octet d'adressage.....	61
Tableau 4: les adresses des types des données.....	62
Tableau 5 : adresses des entrées.....	62
Tableau 6: adresse de minimum/maximum.....	62
Tableau 7 : les fonctions de bus CAN.....	44
Tableau 8: la configuration des entrées d'enregistreur de données.....	72
Tableau 9: la configuration des Sorties d'enregistreur de données.....	72

Glossaire Acronymes

A

Ain : Analog input

AVR : est le terme utilisé par Atmel pour désigner la famille de microcontrôleurs.

C

CAN : Control Area Network

CDC : Convertisseur USB-série

E

EEPROM : Electrically Erasable Programmable Read Only Memory

EXE : Exécutable

H

HID : human interface device

L

LabView : Laboratory Virtual Instrument Engineering Workbench

P

PCB : printed circuit board

PIC : Programmable Intelligent Computer

PID : Product IDentification

R

RAM : Random Access Memory

ROM : Read Only Memory

V

VID : Vendor IDentification

VISA : Virtual Instrument Software Architecture

U

USB : Universal Serial Bus

Organisme d'accueil :

RAYNOV est une société Marocaine d'ingénierie, leur cœur de métier est la conception et la réalisation des systèmes de haute technologie, depuis l'étude de faisabilité du projet jusqu'à la réalisation et la mise en service des installations.

- Conception électronique
- Développement logiciel
- Système Robotique
- Banc de test
- Conception système-Produit.
- Spécialisations : Industries technologiques
- Secteur : Technologies
- Type : Société à responsabilité limitée(SARL)
- Taille de l'entreprise : 5 employés
- Fondée en : 2012

Introduction Générale

Ce projet de fin d'études marque l'achèvement et l'aboutissement d'un parcours universitaire enrichissant tant sur le plan théorique, pratique qu'humain. Il présente une opportunité instructive servant à nous garantir, une vision plus réaliste de l'univers du travail dans lequel nous serons amenés à être intégrés.

Cet humble rapport présente le résultat d'un travail soutenu et assidu qui s'inscrit dans le cadre de la réalisation d'un projet de fin d'études visant l'obtention du diplôme de master science et technique en << Electronique Signaux et Systèmes automatisés>> à la faculté des sciences et techniques de Fès.

De la même façon que les machines à écrire ont cédé la place aux logiciels de traitement de texte, que le vinyle est passé aux CD et MP3. Ainsi les systèmes de mesure traditionnels et les thermomètres... sont remplacés par des enregistreurs de données, alimentées par batterie et équipés d'une mémoire embarquée. La technologie des enregistreurs de données actuels a créé une meilleure alternative et une meilleure solution de surveillance.

Dans ce cadre, RAYNOV a pris l'initiative de réaliser un enregistreur de données capable de faire l'acquisition, l'enregistrement et l'affichage en temps réel de données, de plus, l'enregistreur de données capable de contrôler ses sorties afin d'améliorer la surveillance et de réagir en urgence. C'est dans ce cadre que s'inscrit mon projet, qui consiste à concevoir un système qui répond à cette problématique.

Ce document comporte quatre chapitres principaux :

- ✓ Généralités sur les enregistreurs de données : dans ce chapitre on parlera d'une manière générale sur les enregistreurs de données, leurs fonctionnements et ses domaines d'utilisations. Ainsi les avantages d'enregistrement de données sur ordinateur, et on terminera par quelques exemples des enregistreurs de données.

- ✓ Le choix de la Technologie et l'environnement de développement : ce chapitre sera consacré à une étude technique pour choisir la technologie qui répond à nos besoins pour la conception de notre projet. Ainsi de définir les logiciels de développement et de simulation.

- ✓ Le développement software d'enregistreur de données : Dans ce chapitre on va aborder les démarches qu'on a suivies pour le développement de notre projet. La première partie décrira le développement de l'interface, la deuxième sera consacrée à la programmation de microcontrôleur et on terminera par une simulation de fonctionnement de notre projet.

- ✓ Conception de circuit intégré : ce chapitre a comme intérêt de présenter le projet que nous avons réalisé. Afin de comprendre son fonctionnement ainsi que son mode d'emploi.

Chapitre 1 : Généralités sur les enregistreurs de données

1. Introduction
2. Le fonctionnement d'enregistreur de données
3. Les différents types des enregistreurs de données
4. L'utilisation d'enregistreur de données
5. L'enregistrement de données sur ordinateur
6. Exemples des enregistreurs dans le marché
7. Conclusion

1. Introduction

Un enregistreur de données (data logger) est un instrument électronique qui enregistre les mesures à des intervalles déterminés sur une période de temps. Les mesures peuvent inclure : température de l'air , l'humidité relative , le courant AC / DC et de la tension , la pression différentielle , temps d'utilisation (éclairage, moteurs , etc.) , l'intensité lumineuse , la température de l'eau , le niveau de l'eau , l'oxygène dissous , l'humidité du sol , les précipitations , la vitesse et la direction du vent , humidité des feuilles , des signaux d'impulsion , occupation de la chambre , prise de charge , et beaucoup plus.



Les enregistreurs de données sont des dispositifs généralement compacts, équipés d'un microprocesseur interne ou microcontrôleur, stockage de données, et un ou plusieurs capteurs. Ils peuvent être déployés à l'intérieur, à l'extérieur, et sous l'eau, et peuvent enregistrer des données pour un maximum de mois, sans surveillance. Un enregistreur de données peut être une seule unité, dispositif autonome avec des capteurs internes, qui tient dans la paume d'une main, ou il peut être un instrument de collecte de données multicanale équipé d'un ou plusieurs capteurs externes [1].

2. Le fonctionnement d'enregistreur de données

Tout d'abord, l'enregistreur de données est relié à un ordinateur par l'intermédiaire d'une interface USB. Ensuite, le logiciel d'accompagnement de l'enregistreur de données est utilisé pour sélectionner les paramètres de journalisation (intervalles d'échantillonnage, l'heure de début, etc.) et d'activer l'enregistreur. L'enregistreur est alors déconnecté et déployé à l'endroit désiré, où il enregistre chaque mesure et la stocke dans la mémoire ainsi que l'heure et la date.

Après la période de surveillance souhaitée, l'enregistreur de données est ensuite reconnecté à l'ordinateur et le logiciel est utilisé à nouveau pour lire les données et afficher les mesures sous forme de graphiques qui montrent des profils au fil du temps. Les données

tabulaires peuvent être consultées aussi bien, ou exportés vers un tableur pour une autre manipulation.

3. Les différents types des enregistreurs de données



Figure 1 : les différents types des enregistreurs de données

Les quatre principaux types d'enregistreurs de données sont :

1. Les enregistreurs de données autonomes
2. Des systèmes d'enregistrement de données sur le Web
3. Les nœuds de données sans fil
4. Bluetooth Low Energy (BLE)

1) Les enregistreurs de données autonomes sont compacts, réutilisables et portables, et offrent un faible coût et une installation facile. Des enregistreurs de données avec capteurs internes sont utilisés pour la surveillance à l'emplacement de l'enregistreur, tandis que les enregistreurs avec capteurs externes peuvent être utilisés pour la surveillance à distance. La plupart des enregistreurs autonomes communiquent avec un ordinateur via une interface USB.

2) Les systèmes d'enregistrement de données sur le Web permettent un accès par Internet à distance, autour de l'horloge aux données à partir WI-FI, ou Ethernet. Ces systèmes peuvent être configurés avec une variété de capteurs enfichables externes et de transmettre les données collectées à un serveur Web sécurisé pour accéder aux données.

3) Les nœuds de données sans fil transmettent des données en temps réel de dizaines de points à un ordinateur central, ce qui élimine la nécessité de récupérer

et de décharger les données provenant de différents enregistreurs de données manuellement.

4) L'enregistreurs de données Bluetooth Low Energy mesurent et transmettent des données sans fil à des appareils mobiles sur une plage de 100 mètres

4. L'utilisation d'enregistreur de données



Figure 2: domaines d'utilisation des enregistreurs de données

L'enregistrement de données correspond à la mesure et à l'archivage de paramètres physiques ou électriques dans le temps. Les enregistreurs de données sont utilisés dans une large gamme d'applications telles que [2] :

- L'enregistrement de données à bord de véhicules
- La surveillance environnementale
- La surveillance de l'intégrité des structures
- La surveillance de l'état des machines.
- Les mesures courantes incluent celles de la température, de la déformation, de la tension, du courant, de la pression, de la force et de l'accélération...
- Essentiellement partout.

5. L'enregistrement de données sur ordinateur

L'enregistrement de données sur ordinateur va bien au-delà des enregistreurs de données autonomes traditionnels connectés au ordinateur, ce dernier fait désormais partie intégrante du système et l'enregistreur de données profite des améliorations constantes des performances du processeur, du disque dur, de l'affichage et du bus de périphériques



incorporés au ordinateur, ce qui offre des avantages supplémentaires par rapport aux méthodes traditionnelles d'enregistrement des données.

Aujourd'hui, la méthode la plus populaire de l'enregistrement des données est un enregistreur de données (ou enregistreur sans papier). Les enregistreurs de données sont des instruments de boîte autonome qui mesurent les signaux, les convertissent en données numériques, et stockent les données en interne. De nombreux enregistreurs de données comprennent des écrans intégrés et la capacité de transférer les données vers un ordinateur pour l'analyse en ligne, le stockage permanent, ou la génération de rapports [3].

Les Cinq avantages d'enregistrement sur ordinateur :

1. Visualisation en temps réel
2. analyse en temps réel
3. fonctionnalité définie par l'utilisateur
4. Téraoctets de stockage de données
5. La connectivité réseau

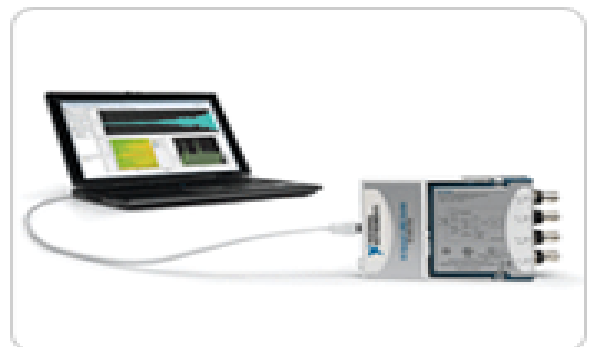
1) Visualisation en temps réel

Avec un enregistreur de donnée autonome traditionnelle, vous devez d'abord enregistrer les données, puis manuellement le transférer sur un PC avant de pouvoir l'importer dans un tableur ou un autre outil pour tracer et visualiser les données.

Parce qu'un enregistreur de données sur PC est toujours connecté à l'ordinateur, les mesures en direct sont affichées sur l'écran du PC en temps réel et vous pouvez voir les résultats instantanément. Vous pouvez utiliser des environnements de programmation, tels que NI LabVIEW, pour construire des interfaces utilisateur personnalisées pour contrôler votre appareil de mesure et l'affichage des résultats. La Création d'une interface utilisateur avec LabVIEW est aussi simple que de glisser-déposer des contrôles et des graphiques sur l'interface utilisateur [3].

2) analyse en temps réel

L'analyse des données avec un enregistreur de données autonome traditionnelle est généralement

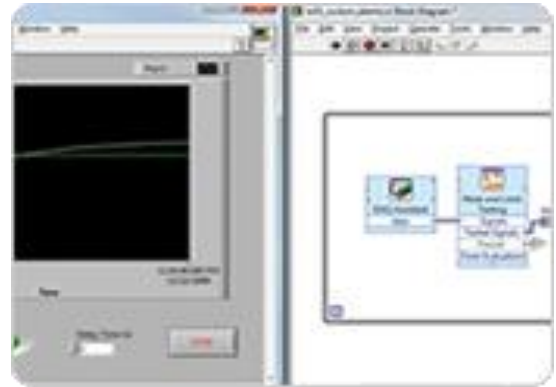


effectuée hors ligne uniquement après que les données ont été transférées vers le PC.

L'utilisation d'un enregistreur de données sur PC, vous pouvez tirer parti des processeurs multi-cœurs et de plus en plus de RAM disponible dans le PC pour effectuer le traitement et l'analyse de signaux sur vos données que vous l'acquérir. LabVIEW inclut de nombreuses fonctions mathématiques et de traitement de signaux communs qui utilisent des assistants de configuration et peut faciliter l'analyse à vos mesures.

3)fonctionnalité définie par l'utilisateur

Avec un enregistreur de données autonome traditionnelle, vous êtes généralement limité à la fonctionnalité matérielle et logicielle définie par le vendeur. Ces fonctions sont bonnes pour l'accomplissement de tâches à usage général, mais ils ne peuvent pas vous aider à répondre à vos besoins d'application uniques. Par exemple, vous voudrez peut-être enregistrer les données uniquement sous certaines conditions ou de générer des alarmes personnalisées qui ne sont pas intégrés dans l'enregistreur de données.



Les enregistreurs de données sur PC sont des instruments définis par logiciel. Cela signifie que la fonctionnalité de l'appareil est définie par le logiciel, et vous pouvez personnaliser le logiciel pour répondre à vos besoins spécifiques de l'application. En utilisant LabVIEW, vous pouvez facilement construire des fonctionnalités pour les alarmes personnalisées, les conditions d'exploitation, la génération de rapports et l'analyse des signaux. Vous pouvez enregistrer des données dans pratiquement tous les formats de fichier pour importer dans d'autres outils et le partage des données avec les autres [3].

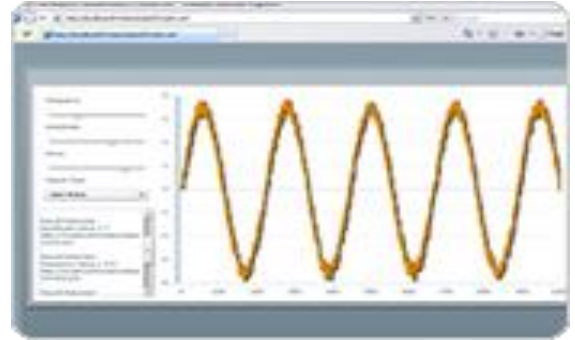
4)Téraoctets de stockage de données

Les traditionnels enregistreurs de données autonomes sont limités par la quantité de mémoire intégrée dans le dispositif.

Parce que le PC est en fait une partie d'un enregistreur de données sur PC, vous n'êtes limité que par la quantité d'espace disque dur sur le PC. Aujourd'hui, il n'est pas rare de trouver un disque dur de PC avec une capacité de téraoctet qui offre amplement d'espace pour vos mesures de courant ainsi que l'espace de stockage permanent.

5)La connectivité réseau

Pour les applications qui nécessitent une surveillance à long terme au cours des jours ou des semaines, vous pouvez avoir de la difficulté de vérifier continuellement les résultats. La surveillance à distance est utile parce que vous pouvez voir les résultats à partir d'un emplacement distant. L'utilisation d'un enregistreur de données sur PC, vous pouvez profiter de la connectivité réseau du PC pour transmettre les résultats sur un réseau pour la visualisation à distance [3].



6. Exemples des enregistreurs dans le marché

1. Enregistreur de données PCE-HT 71N

L'enregistreur de données détecte l'humidité et la température de l'air et la mémorise dans sa mémoire interne. Cet enregistreur de données en format réduit et avec une grande mémoire. L'appareil travaille comme enregistreur de données autonome qui enregistre les valeurs climatologiques dans des espaces de temps souhaités. Il peut transmettre toutes les valeurs à un ordinateur ou à un portable [4].



- Température : -40 ... +105 °C
- Humidité: 0 ... 100%
- Température du point de rosée se calcule avec le logiciel (précision ± 2 °C)

2. Enregistreur de données PCE-KD5

L'enregistreur de données PCE-KD5 se base sur le système opérationnel Linux ce qui lui donne une grande versatilité. Les possibles applications de cet enregistreur de données sont: la détection, visualisation et contrôle de valeurs de mesure et développements du processus. La communication de l'enregistreur de données avec des systèmes de contrôle de processus maîtres ou capteurs ou transducteurs esclaves peut être effectuée grâce aux différentes interfaces numériques [5].



- 48 entrées analogiques ou binaires
- 24 entrées de thermocouples
- 12 entrées RTD
- Dimensions: 96 x 96 mm

3. Exemple d'enregistreurs industriels

- dataTaker Universal Data Loggers



Figure 3: data Taker Universal

Parmi les Caractéristiques des canaux d'entrée universels elles peuvent être utilisés avec une large gamme de types d'entrée de signal et registres : température, tension, courant, 4-20 mA, Résistance, ponts, jauges de contrainte, fréquence, numérique...

Ils offrent également de puissantes fonctions d'alarme et de programmation pour permettre aux enregistreurs de traiter les mesures. Les systèmes dataTaker peuvent être utilisés pour l'acquisition de données locales pour la collecte de données sans surveillance à distance connecté à un ordinateur centralisé via un modem ou Internet; ou en tant que dispositif autonome [6].

dataTaker

DT80 Intelligent Universal Input Data Logger

DT80



The dataTaker DT80 is an intelligent data logger that provides an extensive array of features that allow it to be used across a wide variety of applications.

- 5 to 15 Universal Analog Sensor Channels
- Expandable up to 300 Analog Inputs
- 8 Bi-Directional Digital Channels
- 4 High Speed Counter Channels
- 4 SDI-12 Channels
- Advanced Alarming & Mathematical Capabilities
- Ethernet, USB, or RS-232 Communication with PC
- SDI-12 and Modbus Sensor Support
- Built-in Web Server, FTP, and an Enhanced Web Interface

Figure 4: enregistreur de données DT80

7. Conclusion

Dans ce chapitre j'ai présenté un panorama d'enregistreur de données et leur fonctionnement, premièrement j'ai exposé les enregistreurs de données traditionnelles qui correspondent seulement à la mesure et à l'archivage des données, ainsi les domaines d'utilisation de ces enregistreurs. À la suite j'ai présenté les enregistreurs de données sur ordinateur et leurs avantages. L'ordinateur fait désormais partie intégrante du système et l'enregistreur de données profite des améliorations constantes des performances du processeur, du disque dur, de l'affichage et du bus de périphériques incorporés au PC, ce qui offre des avantages supplémentaires par rapport aux méthodes traditionnelles d'enregistrement de données.

J'ai cité au début que notre système rassemble les deux points précédents, on veut réaliser un enregistreur de données capable de fonctionner sur un ordinateur et il peut fonctionner indépendamment d'ordinateur.

Ainsi notre système n'est pas conçu seulement au sauvegarde et l'acquisition de données, il est capable de contrôler ses sorties.

Dans le chapitre suivant, je détaille le choix de la technologie pour la conception de notre système, et je vais présenter les logiciels de développement.

Chapitre 2 : Choix de la Technologie et l'environnement de développement

- I. Choix de microcontrôleur
 - 1. Introduction
 - 2. Définition d'un microcontrôleur
 - 3. Définition d'un PIC
 - 4. Le choix de microcontrôleur
 - 5. Identification des PIC
 - 6. Caractéristiques principale de PIC18F4550
 - 7. Brochage de PIC18F4550
- II. Les logiciels de développement
 - 1. MicroC
 - 2. ISIS proteus
 - 3. LabVIEW
- III. Conclusion

I. Choix de microcontrôleur

1. Introduction

Les microcontrôleurs sont très utilisés dans le monde de l'industrie, notamment dans les systèmes embarqués. On pourra donc les retrouver dans l'aéronautique, l'aérospatial, l'automobile, l'électronique grand public. Leur polyvalence, et leur taille les rendent intéressants pour des modules de traitement de données numériques et aussi analogiques. Ils sont certes peu puissants comparés à des processeurs dédiés, mais ils les compensent par leur prix mais surtout leur taille : un microcontrôleur peut être comparé à une carte mère d'ordinateur.

Le choix d'un microcontrôleur 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 microcontrôleur.
- Il faut ensuite déterminer si l'application nécessite un convertisseur Analogique/ Numérique ce qui va centrer un peu plus vers le choix.
- La rapidité d'exécution est un élément important, il faut consulter les DATA-BOOK pour vérifier la compatibilité entre la vitesse maximale choisie et la vitesse max nécessaire au montage.
- La taille de la RAM interne et la présence ou non d'une EEPROM pour mémoriser des données est également important pour l'application souhaitée.
- La longueur de programme de l'application détermine la taille de la mémoire programme du microcontrôleur recherché [7].

Après l'étude des enregistreurs de données disponibles au marché, nous avons spécifié les caractéristiques de base de notre système :

- Huit entrées numériques présentes les données des capteurs (état logique : haut/bas).
- Huit entrées analogiques pour l'utilisation des capteurs analogiques (température, tension, courant...)
- Huit sorties pour contrôler des relais.
- Le système nécessite un convertisseur analogique/numérique.

- des tailles importantes des mémoires RAM, EEPROM et ROM.

2. Définition d'un PIC

Un PIC (Programmable Interface Contrôler) est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes. Les PICs sont des composants dits RISC (Reduced Instructions Set Computer), ou encore composants à jeu d'instruction réduit [7].

Afin de choisir un microcontrôleur adéquat à notre projet, nous avons pensé à deux familles de microcontrôleurs :

- **ATmega**

Les microcontrôleurs de la famille ATMEGA en technologie **CMOS** sont des modèles à 8 bits **AVR** basés sur l'architecture **RISC**. En exécutant des instructions dans un cycle d'horloge simple, l'ATMEGA réalise des opérations s'approchant de 1 MIPS par MHz permettant de réaliser des systèmes à faible consommation électrique et simple au niveau électronique [8].



Figure 5 : le microcontrôleur Atmega

- **PIC PIC18F4550**

PIC18F4550 appartient à PIC18F famille de microcontrôleurs. PIC18F4550 est un parmi les Microcontrôleurs avancées de la technologie de la puce. Ce microcontrôleur est très célèbre entre les amateurs et les apprenants en raison de ses fonctionnalités et des caractéristiques telles que l'ADC et de l'intégration USB. Un PIC18F4550 typique vient dans différents forfaits comme DIP, QFP et QFN. Ces forfaits peuvent être choisis en fonction de l'exigence du projet.

Après avoir analysé les data-book de chaque microcontrôleur et selon le nombre des entrées /sorties, et les autres caractéristiques envisager pour notre projet, nous avons extrait les caractéristiques suivantes [9] :

- Les mêmes configurations au niveau des nombres des entrées/sorties.
- Le même convertisseur analogique numérique pour les deux technologies.
- Une large version pour les AVR disponibles avec différentes capacités des mémoires.
- PIC18F4550 prend en charge à fonctionnalité USB qu'elle est livré avec un matériel intégré USB2.0 qui peut fonctionner en deux vitesses standard USB.
Speed USB pris en charge par PIC18F4550
 - Pleine vitesse 12 MBit par seconde
 - Faible vitesse de 1,2 Mbit par seconde
- Le microcontrôleur Atmega prend en charge seulement la fonctionnalité RS232.
- La vitesse d'échantillonnage pour PIC18F4550 est de 500 échantillons par seconde par contre pour les microcontrolleurs AVR est de 100 échantillons par seconde.

Conclusion :

Après les analyses précédents nous avons orienté vers le PIC18F4550 qui profite la fonctionnalité USB.

3. Caractéristiques principale de PIC18F4550

Les caractéristiques principales du 18F4550 sont résumées comme suit :

- 35 lignes d'entrées/sorties, répartis comme suit :
- 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 (32768 Octets) mots de 16 bits)

- 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

Le tableau suivant présente tous les caractéristiques générales de PIC18F4550

Valeur	caractéristique
48MHz	Mémoire programme FLASH
32Ko (32.	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 X1 6bit + 1X8bit)	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
1 CCP, 1 ECCP	Comparateurs
2	USB (canaux, vitesse, respect)
1, Full Speed, USB 2.0	Plage de tension de fonctionnement (V)
-40 À 85	Température (C)

Tableau 1 : les caractéristiques générales de PIC18F4550

4. Brochage de PIC18F4550

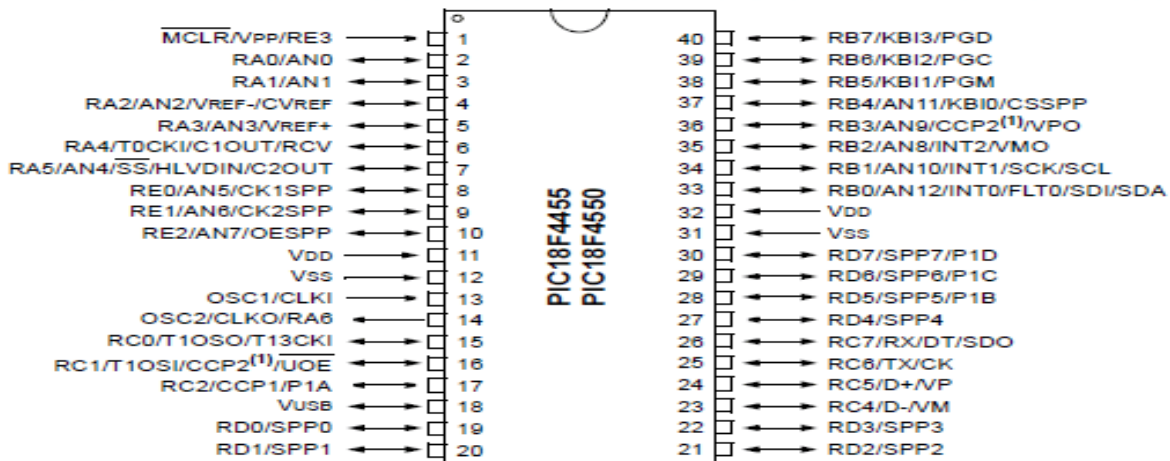


Figure 6 : Brochage du PIC18F4550

On peut distinguer sur ce schéma [9] :

- L'alimentation : VDD (+5V) et Vss (0V)
- Les bornes du quartz (oscillateur a quartz) : OSC1 et OS
- L'entrée RESET (MCLR : Master CLEAR)
- Les différents ports d'Entrées/Sorties : RAx, RBx, RCx, RDx, REx.

II. Les logiciels de développement

i. MicroC

1. Introduction

La programmation des microcontrôleurs PIC est supportée par plusieurs langages de programmation tel que : MikroC for PIC, MPLAB, MikroBasic PRO for PIC, HI-TECH C for PIC, flowcode...etc. Dans notre projet nous avons opté pour le compilateur de MIKRO C qui est un compilateur en langage C (langage évolué) et qui permet d'intégrer certaines routines en assembleur. Ce choix est à la fois un choix personnel et un choix technologique parce que il est basé sur le langage C.

2. Étapes de développement de programme

✚ L'élaboration d'un programme est un travail qui se fait en plusieurs étapes :

✚ La première étape : L'algorithme

✚ La deuxième étape : Écriture du programme

- ✚ La troisième étape : Simulation du programme
- ✚ La quatrième étape : Transfert du programme vers PIC

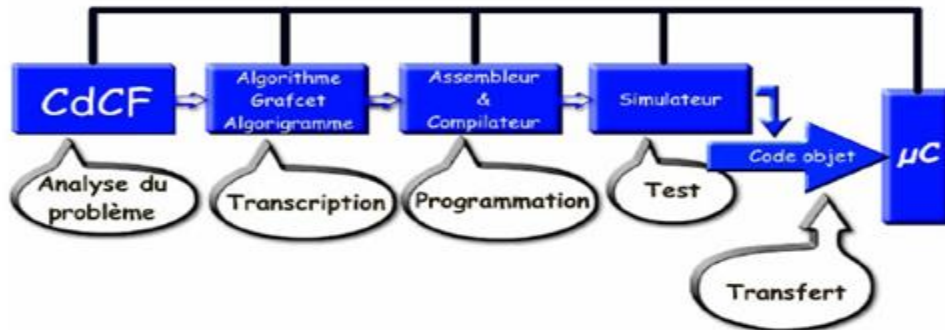


Figure 7 : les étapes de programmation d'un microcontrôleur

3. Présentation du MikroC

Le « MikroC » est un compilateur pour PIC Conçu par la société « Mikroelektronika », le compilateur C nouvelle génération "MikroC" pour microcontrôleurs PIC bénéficie d'une prise en main très facile. Il comporte plusieurs outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code...) ; Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, 1Wire, SPI, RS485, Bus CAN, cartes compact Flash, signaux PWM, afficheurs LCD et 7 segments...) de ce fait il est un des outils de développement incontournable et puissant.

Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète [10].

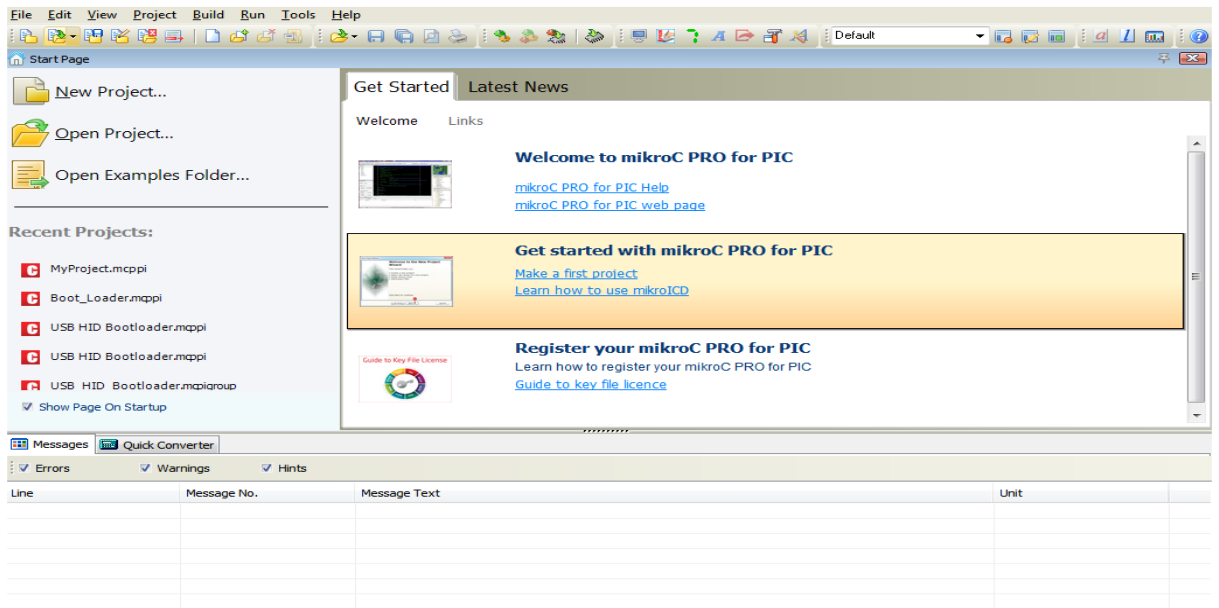


Figure 8 : Interface du logiciel MikroC

La saisie d'un programme en `C` répond pratiquement toujours à la même structure.

On peut noter que :

- Le symbole «#» est suivi d'une directive.
- Le symbole «//» est suivi d'un commentaire, ou bien «/*» indiquant le début d'un commentaire et «*/» indiquant sa fin.
- Chaque ligne d'instruction se termine par un «;».
- Le début d'une séquence est précédé du symbole « { ».
- La fin d'une séquence est suivie du symbole « } ».
- La notation des nombres peut se faire en décimal de façon normale ou en hexadécimal avec le préfixe «0x» ou encore en binaire avec le préfixe«0b» [10].

ii. ISIS PROTEUS

1. Présentation de logiciel de simulation isis

Isis est un éditeur de schémas qui intègre un simulateur analogique / logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes.

Dans cette section nous allons commencer par la présentation de la fenêtre du logiciel ISIS.

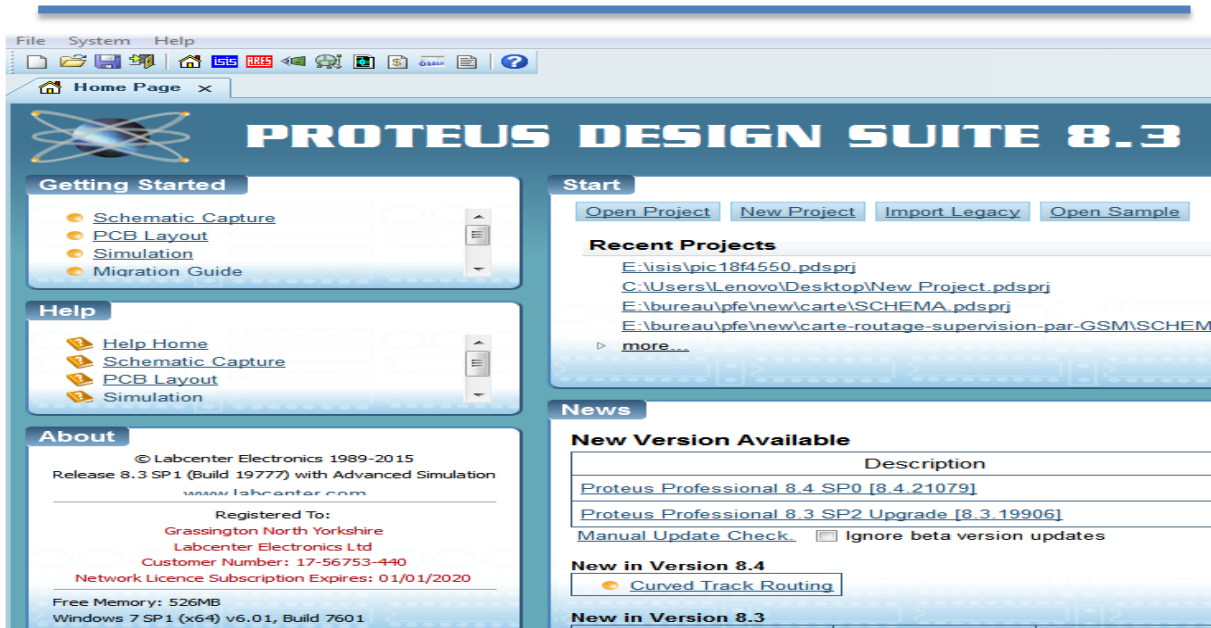


Figure9: Interface du logiciel ISIS

Pour ouvrir un nouveau projet il suffit de sélectionner « file » puis « new project ». Suivre les étapes et remplir les cases vides par les paramètres nécessaires. Une nouvelle fenêtre apparaîtra.

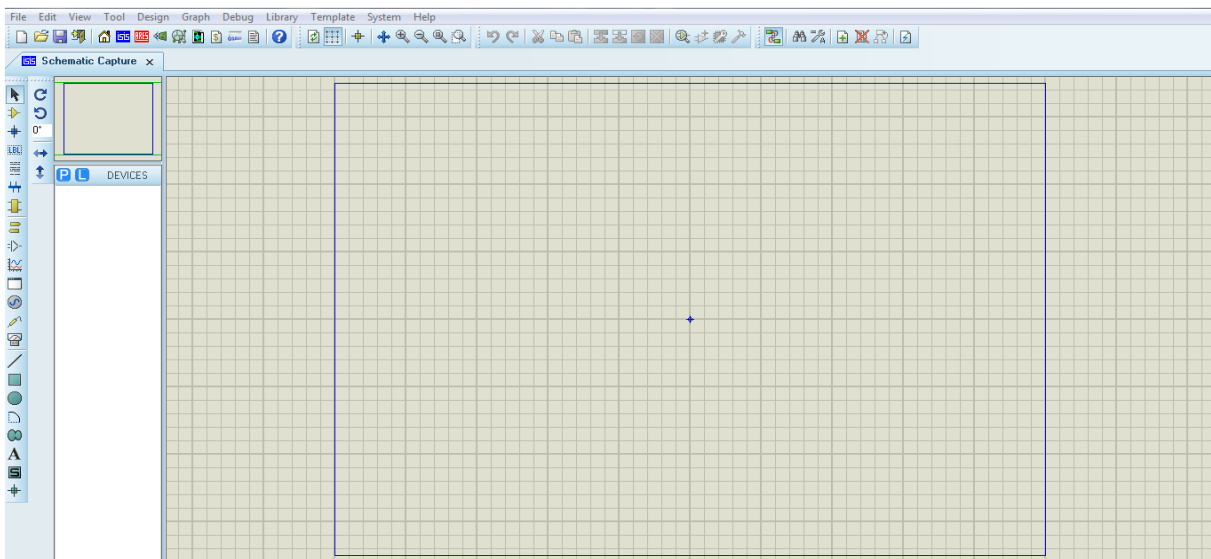


Figure 10 : fenêtre d'édition

Selon le cahier de charge, on ajoute à la fenêtre les composants nécessaires pour simuler notre projet.

iii. LabVIEW

1. Introduction :

L'utilisation de logiciel LabVIEW est conçue pour développer l'interface et établir la communication entre le microcontrôleur, a partir cette interface on visualise les entrées et les sorties ainsi l'acquisition, traitement et le contrôle.

2. Généralités



Figure 11 : programmation graphique

LabView (Laboratory Virtual Instrument Engineering Workbench) est un langage de programmation dédié au contrôle d'instruments et l'analyse de données. Contrairement à la nature séquentielle des langages textuels, LabView est basé sur un environnement de programmation graphique utilisant la notion de flot de données pour ordonnancer les opérations.

LabView intègre l'acquisition, l'analyse, le traitement et la présentation de données.

Pour l'acquisition de données et le contrôle d'instruments, LabView supporte les standards RS-232/422, USB, IEEE 488 (GPIB) et VXI/PXI, ainsi que les cartes d'acquisition de données [11].

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...

LabVIEW est un outil d'acquisition, d'analyse et de présentation de données

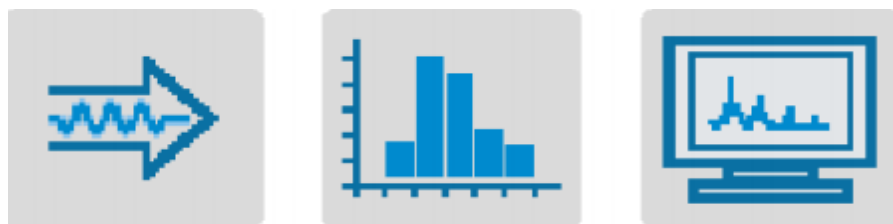


Figure 12: acquisition, analyse et présentation

Acquisition avec LabVIEW



LabVIEW permet l'acquisition de données par l'intermédiaire de diverses connectiques :

- PCI (Peripheral Component Interconnect)
- LAN (Local Area Network)
- CompactFlash
- PXI (PCI eXtensions for Instrumentation)
- PCMCIA (Personal Computer Memory Card International Association)
- Wi-Fi (IEEE 802.11 b/g/n) (Wireless Fidelity)
- IrDA (Infrared Data Association)
- Bluetooth
- USB (Universal Serial Bus)
- GPIB (IEEE 488) (General Purpose Interface Bus)
- Ethernet
- Firewire (IEEE 1394)
- Série (RS 232, RS 449, RS 422, RS 423, RS 485)

Analyse avec LabVIEW



LabVIEW inclut des outils pour l'analyse des données :

- Traitement du signal : Convolution, analyse spectrale, transformées de Fourier,...
- Traitement d'images : Masque, détection de contours, profils, manipulations de pixels,...
- Mathématiques : Interpolation, statistiques (moyennes, écart-type,...), équations différentielles,...

Présentation avec LabVIEW



LabVIEW inclut des outils d'aide à la présentation (communication) des données :

- Graphiques, tableaux, images, génération de rapport,...

- Par l'intermédiaire d'Internet : outils de publication web, serveur datasocket, TCP/IP, envoi d'alertes par email,...

3. Astuces pour travailler dans LabVIEW : Quelques raccourcis clavier

- <Ctrl+H> – Activer/désactiver la fenêtre d'aide contextuelle.
- <Ctrl+B> – Supprimer les connexions erronées du diagramme.
- <Ctrl+E> – Basculer entre la face avant et le diagramme.
- <Ctrl+T> –Mosaïque verticale des fenêtres.
- <Ctrl+Z> – Annuler (aussi dans le menu Édition).
- <Ctrl+C> – Copier un objet.
- <Ctrl+V> – Coller un objet.

Pour balayer les objets de la palette d'outils, presser la touche TAB.

Pour passer de la flèche à la bobine, appuyer sur la barre d'espace.

Pour incrémenter ou décrémenter rapidement des contrôles, appuyer simultanément sur la touche<majuscules> et sur les flèches du clavier numérique.

La police et le couleur des fils indiquent leurs types.








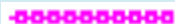

Type de la donnée	Scalaire	Tableau 1D	Tableau 2D	Couleur
Numérique				Orange
Booléen				Vert
Chaîne				Rose

Tableau2 : types de fils

III. Conclusion

Dans ce chapitre j'ai présenté la technologie de travail choisie pour la conception du projet, le choix de microcontrôleur est fixé sur l'utilisation de PIC18F4550. L'avantage majeur de cette dernière par rapport les autres technologies c'est qu'elle intègre le protocole USB 2.0. Le débit maximal peut atteint 480 Mbit/s. Or la communication est très puissante par rapport les autres technologies qui utilisent la communication série classique RS-232.

J'ai présenté aussi l'environnement de développement, il s'agit d'utiliser le logiciel de National Instrument LabVIEW pour le développement de l'interface, le microC pour programmer le PIC18F4550 et le logiciel isis proteus pour la simulation.

Dans le chapitre suivant je vais présenter la partie technique et mes solutions pour le développement software de data logger.

Chapitre 3 : Développement software de l'enregistreur de données

- I. Introduction
- II. Communication entre le hardware et le software
- III. Bus CAN
- IV. Développement d'enregistreur de données avec ordinateur
 - 1. Développement de l'interface
 - 2. Programmation hardware
 - 3. Conclusion
- V. Développement d'enregistreur de données autonome
 - 1. Développement de l'interface
 - 2. Les trames et adressage
 - 3. L'envoi des trames
 - 4. Réception des trames
- VI. Description de fonctionnement et simulations
- VII. Conclusion

I. Introduction

L'enregistrement des données implique la collecte des données avec le stockage pour le traitement de données plus tard. Un système d'enregistrement de données comporte trois composantes principales : une interface pour relier à un ordinateur, des capteurs et des logiciels.

Parmi les avantages de notre enregistreur de données ils permettent de faire le traitement et la visualisation en temps réel contrairement aux autres enregistreurs qui réalisent les deux tâches séparément, ils stockent les données dans des mémoires internes ou externes, ces derniers sont connectés par un ordinateur pour faire le traitement et la visualisation des graphes et des tableaux. Or la visualisation en temps réel facilite l'analyse des résultats et évite à chaque fois de connecter ou déconnecter l'enregistreur avec l'interface.

Pour cela nous avons développé un enregistreur de données qui est capable de faire l'acquisition et l'enregistrement en temps réel ce qui facilite la visualisation et l'analyse des résultats. Ainsi l'enregistreur n'est pas limité seulement pour l'acquisition et l'enregistrement, il est capable de contrôler ses sorties. Le contrôle des sorties est destiné pour augmenter la surveillance des systèmes. À partir de l'interface développée par Labview on visualise les états des entrées numériques et les valeurs des entrées analogiques. Ainsi trois onglets de l'interface permettent le traitement des données des entrées numériques et analogiques et le contrôle des sorties et l'enregistrement de toutes les données dans des fichiers Excel dans la mémoire d'ordinateur.

Dans ce cas le fonctionnement de notre enregistreur de données dépend fréquemment de l'ordinateur si on débranche le système de l'interface le fonctionnement et la communication sont désactivées. En outre, nous avons pensé à l'autonomie de l'enregistreur de données pour les utilisations qui ne dépendent pas fréquemment de connecter l'enregistreur avec l'ordinateur, dans ce cas l'utilisateur peut facilement connecter l'enregistreur de données avec l'interface et configurer les entrées et les sorties, selon leur besoin et selon son cahier de charge, après il débranche le périphérique et l'enregistreur continue le fonctionnement normal même s'il est déconnecté à l'interface. Cette version d'enregistreur combine les deux modes de fonctionnement, l'enregistreur peut fonctionner avec l'interface quand il est branché à l'ordinateur et il continue le fonctionnement normal quand il est déconnecté à l'interface. Puisque toutes les variables et le traitement seront sauvegardés dans la mémoire de microcontrôleur.

Dans ce chapitre je vais montrer quelles sont les étapes d'abord pour établir la communication entre le microcontrôleur et le logiciel. La première partie sera consacrée pour le développement de l'enregistreur de données avec un ordinateur et la description du fonctionnement. La deuxième traitera le développement de l'enregistreur de données autonome.

II. Communication entre le hardware et le software

A. Introduction à la communication USB

La communication entre le hardware qui est le microcontrôleur (PIC18F4550) et LabVIEW qui représente la partie logiciel établie à l'aide du protocole USB.



Figure 13 : Logo officiel USB

Trop souvent les ingénieurs de développement et responsables de projets considèrent l'USB comme seulement un moyen de connecter un équipement à un ordinateur en remplacement des classiques connexions RS232. Le bus USB est bien plus que ça par rapport à une communication série RS232. L'USB est un bus extrêmement complexe et sophistiqué, il représente un moyen de connexion aux possibilités tendues qui va bien au-delà de l'univers de périphériques informatique et électronique, et trouve tout naturellement sa place dans le domaine de l'instrument [12].

Vitesses de transmission :

La norme définit 3 vitesses de transmission, la vitesse étant fixée par les caractéristiques électrique de la connexion.

- Low speed : 1.5Mbits/s (résistance de pull-up de 1.5 KOhms sur D-)
- Full speed : 12Mbits/s (résistance de pull-up de 1.5 KOhms sur D+)
- High speed : 480 Mbits/s (séquence automatique de signaux électriques)

La norme décrit plusieurs types de connecteurs spécifiques. Chacun pour un usage bien particulier de telle sorte qu'en utilisant du matériel conforme à la norme, il soit absolument impossible de réaliser une connexion non conforme.

B. Établir la communication entre le microcontrôleur et LabVIEW

La communication USB est établie par créer des fichiers driver au niveau de Labview et au niveau de microcontrôleur. Cette partie est importante avant de commencer la programmation et l'analyse de programme il faut d'abord assurer que le hardware est bien connecté avec l'interface et que le transfert de données entre eux est bien établi.

1. Au niveau de microcontrôleur (PIC18F4550)

L'avantage majeur de PIC18F4550 est l'intégration de protocole USB par rapport aux autres microcontrôleurs qui sont caractérisés par la communication série classique moins rapide et classique que l'USB. Or pour établir la communication USB il faut créer un fichier driver au niveau de PIC18F4550 dont on définit un numéro d'identification de fournisseur VID (Vendor Identification) et le numéro d'identification du produit PID (Product Identification), Ces nombres identifient le matériel USB pour son installation et son ciblage.

Chaque logiciel de programmation des microcontrôleurs à une propre méthode de la création de ce fichier, si ce n'est pas le cas il est obligatoire de créer ce fichier manuellement. Concernant le microC on ouvre le logiciel dans l'icône **Tools** on choisit **HID Terminal** une fenêtre apparaîtra.

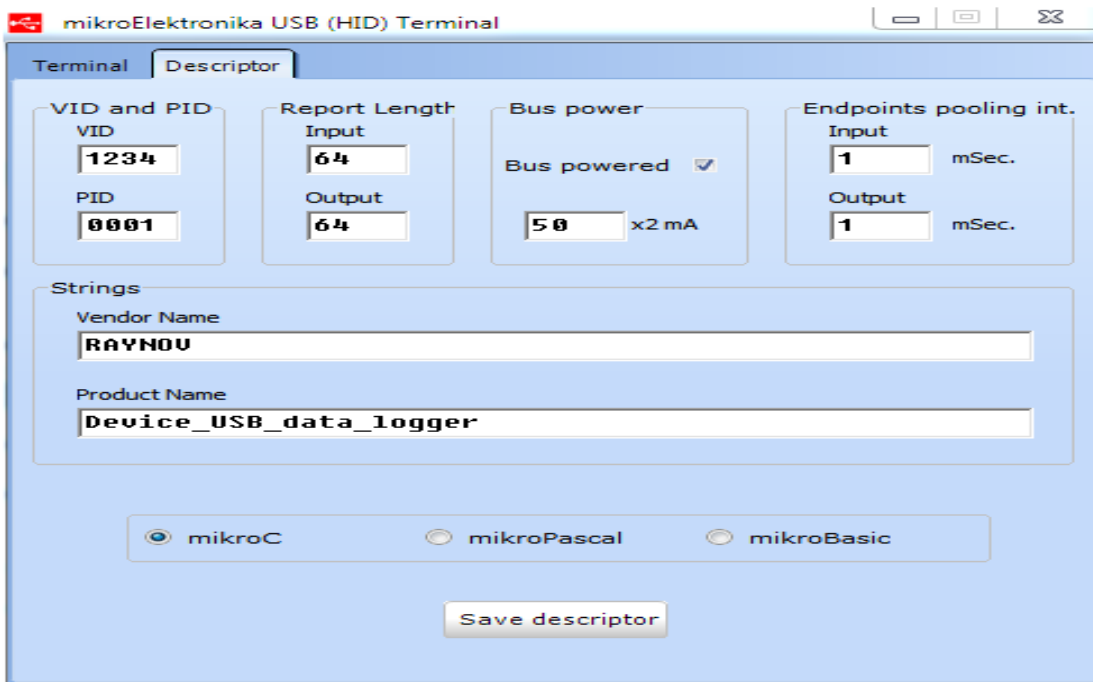
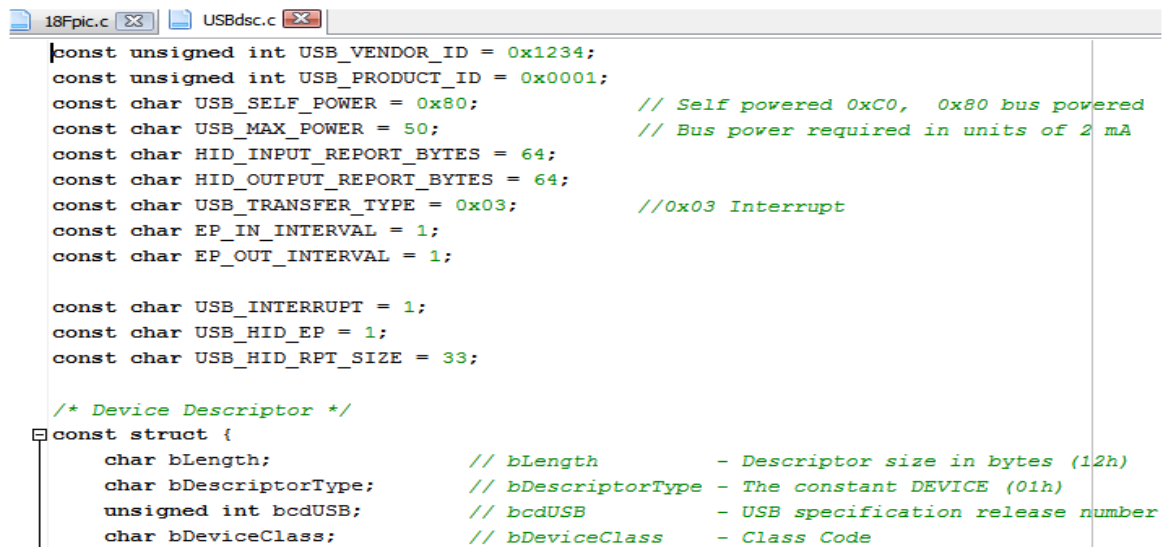


Figure 14 : HID Terminal

À partir de la figure, la première colonne permettra la définition de numéro d'identification de fournisseur VID et le numéro d'identification de produit PID sont les deux paramètres principaux car ces nombre identifiant le matériel USB.

La deuxième colonne permet la définition de la longueur des données transférées entre ce matériel et les autres périphériques.

Ainsi on choisit le nom de fournisseur et le nom de produit puis on choisit le logiciel microC et enregistre la description. Un fichier apparaîtra continuellement sous le nom **USBdsc**, il faut enregistrer ce fichier dans le même espace de travail et l'ajouter dans le projet.



```

const unsigned int USB_VENDOR_ID = 0x1234;
const unsigned int USB_PRODUCT_ID = 0x0001;
const char USB_SELF_POWER = 0x80;           // Self powered 0xC0, 0x80 bus powered
const char USB_MAX_POWER = 50;             // Bus power required in units of 2 mA
const char HID_INPUT_REPORT_BYTES = 64;
const char HID_OUTPUT_REPORT_BYTES = 64;
const char USB_TRANSFER_TYPE = 0x03;       //0x03 Interrupt
const char EP_IN_INTERVAL = 1;
const char EP_OUT_INTERVAL = 1;

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

/* Device Descriptor */
const struct {
    char bLength;           // bLength - Descriptor size in bytes (12h)
    char bDescriptorType;  // bDescriptorType - The constant DEVICE (01h)
    unsigned int bcdUSB;   // bcdUSB - USB specification release number
    char bDeviceClass;     // bDeviceClass - Class Code
}
    
```

Figure 15 : fichier USBdsc.c

Le fichier contient tous les paramètres définissent notre matériel USB et des fonctions générées par MicroC, permettront d'établir la communication avec ce matériel.

Le fichier crée en langage de programmation C et contient 163 lignes.

Conclusion :

Les étapes précédentes permettent de définir les paramètres de matériel USB et générer le fichier USBdsc.c. Il faut retenir le VID et le PID pour concevoir le driver au niveau de logiciel Labview pour respecter les mêmes configurations et les mêmes numéros, sinon la communication ne sera pas réalisée.

2. Au niveau de software (LabVIEW)

Le logiciel Labview intègre une riche bibliothèque qui contient plusieurs protocoles de communication pour permettre le connecté avec différents matériels parmi ces protocoles on trouve la communication USB.

Quand on parle d'une communication utilisant Labview, habituellement on pense à l'utilisation de NI-VISA.

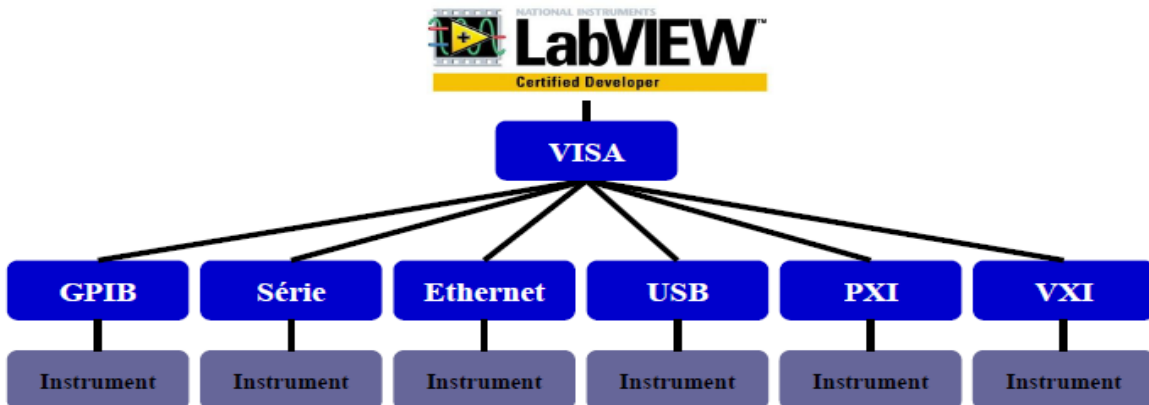


Figure 16 : l'utilisation de NI-VISA

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

NI-VISA doit être déjà installé sur l'ordinateur et le matériel USB ne doit pas être connecté. De plus, le driver du matériel USB ne doit pas être installé.

Trois étapes sont nécessaires pour configurer un matériel USB pour utiliser NI-VISA :

- Création d'un fichier INF grâce à l'Assistant de développement de drivers (DDW).
- Installation du fichier INF et du matériel USB grâce au fichier INF.
- Test du matériel avec NI-VISA Interactive Control.

La création de fichier INF faite par le driver Wizard pour cela sélectionnez Démarrage»Tous les programmes»National Instruments»VISA»VISA Driver Developer Wizard. La Figure suivante montre l'écran d'ouverture.

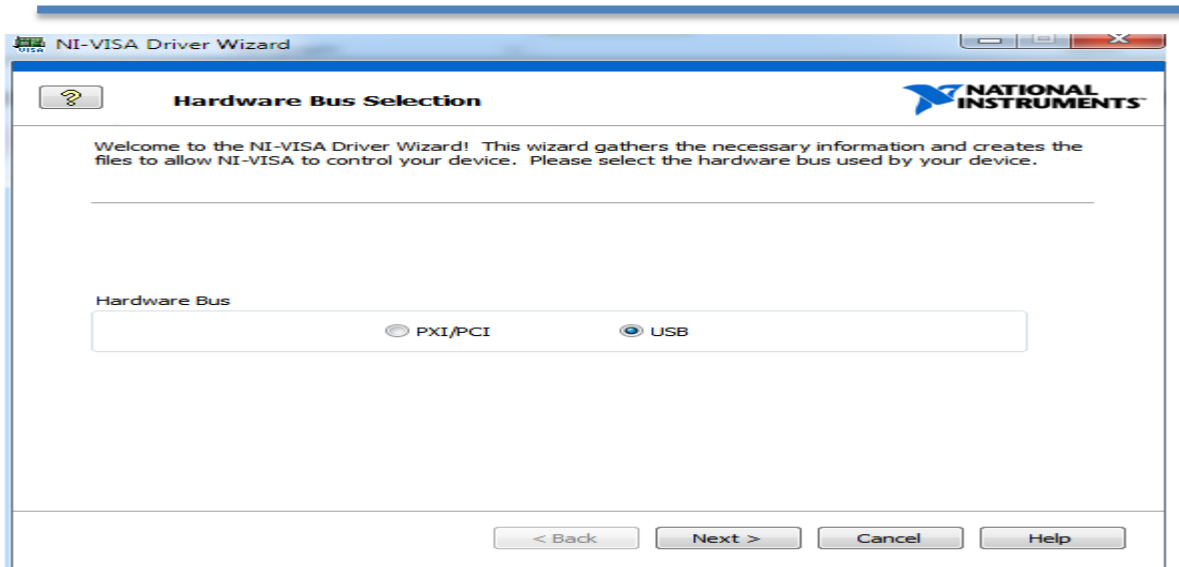


Figure 17 : Écran de sélection du bus matériel de l'Assistant VISA DDW

Pour la création d'un driver, cliquez sur USB puis sur Next. La fenêtre des informations de base du matériel de VISA DDW s'ouvre, comme indiqué à la Figure suivante.

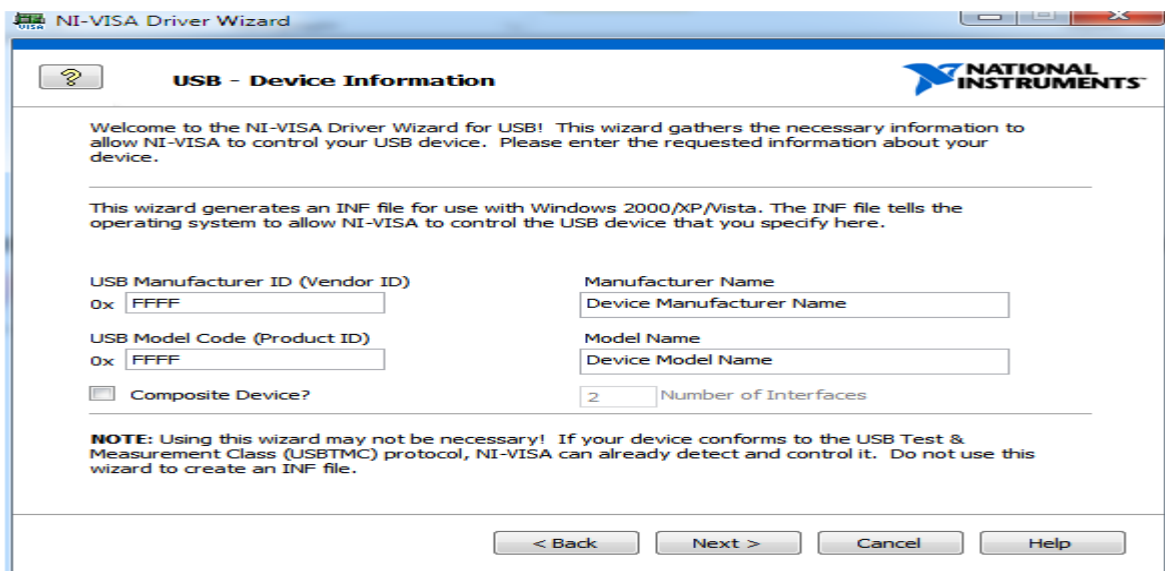


Figure 18 : Écran des informations de base sur le matériel de VISA DDW

À cette étape, on choisit les mêmes identifiant (ID) du fournisseur USB et l'ID du produit, qu'on a choisi lors de la création de fichier en microC pour la description de matériel de microcontrôleur. Ces nombres identifient le matériel USB pour son installation et son ciblage. Selon la spécification USB, ces deux identifiants sont des nombres hexadécimaux 16 bits. On rappelle qu'on a choisi le VID 0x1234 et le PID est 0x0001.

À ce niveau on choisit les paramètres adéquats et on continue l'installation jusqu'à une interface demande le chemin d'enregistrement des fichiers INF, puis on clique sur finish.

Une fenêtre apparaîtra permet l'installation de ce driver.

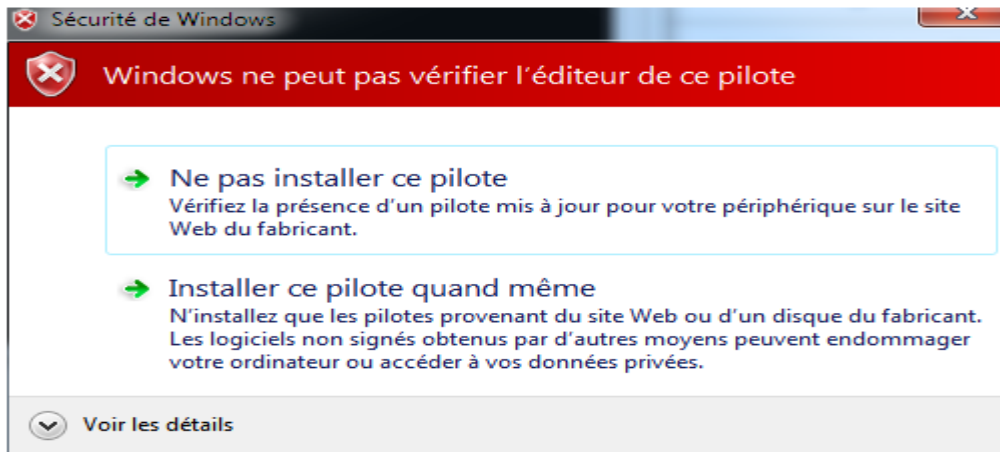


Figure 19 : l'installation de driver INF

Conclusion :

Après avoir installé les fichiers INF, nous devons faire un petit test pour assurer que le transfert des données est bien réalisé entre LabVIEW et le PIC18F4550. Pour cela il suffit juste d'ouvrir le NI-MAX, c'est une application intégrée dans la bibliothèque LabVIEW, elle permet d'envoyer des messages de l'interface vers le microcontrôleur, si la transmission est bien faite, alors on doit commencer le travail et la programmation de microcontrôleur et le développement de l'interface.

III. Bus CAN

1. Introduction

Le protocole CAN (Control Area Network) est un protocole de communication série qui supporte des systèmes temps réel avec un haut niveau de fiabilité. Il est avant tout à classer dans la catégorie des réseaux de terrain utilisé dans l'industrie pour remplacer la boucle analogique 20mA [13].

La structure du protocole du bus CAN possède implicitement les principales propriétés suivantes :

- hiérarchisation des messages.
- garantie des temps de latence.
- souplesse de configuration.
- réception de multiples sources avec synchronisation temporelle.
- fonctionnement multimaître.
- détections et signalisations d'erreurs.
- retransmission automatique des messages altérés dès que le bus est de nouveau au repos

- distinction d'erreurs : d'ordre temporaire ou de non-fonctionnalité permanente au niveau d'un nœud.
- déconnexion automatique des nœuds défectueux.

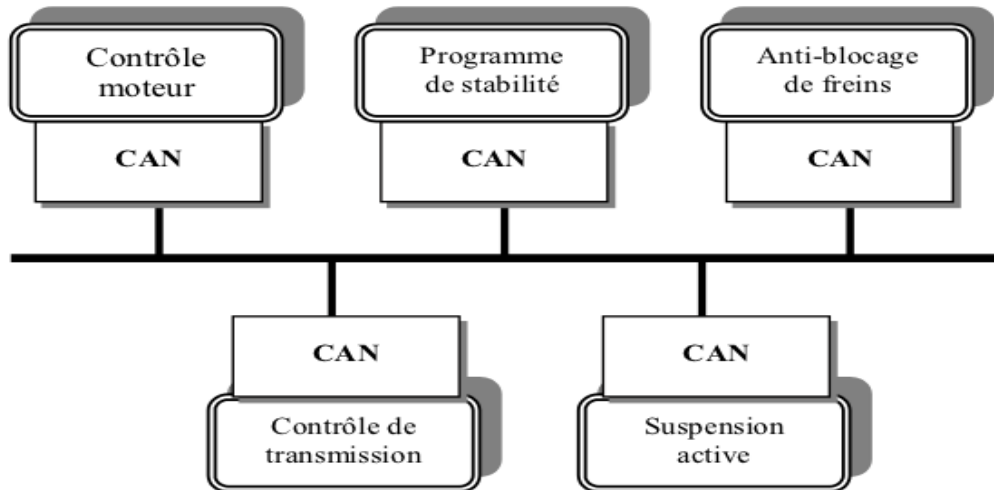


Figure 20 : le bus CAN

2. Réalisation :

Lors de la réalisation de notre application de bus CAN nous avons utilisé le controller MCP2510 et le transceiver MCP2551 de microchip



Figure 21 : MCP2551



Figure 22 : MCP2510

Le MCP2510 permet la connexion avec le pic , et permet de gérer la connexion hardware de CAN Le MCP2551 permet d'adapter et de générer des signaux électriques compatibles avec les spécifications du Bus CAN.

Puisque nous avons déjà programmé notre PIC à l'aide de logiciel MicroC de MikroElektronika aussi il satisfait notre besoin pour l'implémentation de bus CAN à l'aide de la bibliothèque CAN-SPI. D'abord il faut déclarer ses variable pour n'importe qu'il projet utilise la bibliothèque CANSPI.

LES VARIABLES qu'ils faut déclarer	Description :	Exemple :
extern sfr sbit CanSpi_CS;	Chip Select line.	sbit CanSpi_CS at RC0_bit;
extern sfr sbit CanSpi_Rst;	Reset line.	sbit CanSpi_Rst at RC2_bit;
extern sfr sbit CanSpi_CS_Direction;	Direction of the Chip Select pin.	sbit CanSpi_CS_Direction at TRISC0_bit;
extern sfr sbit CanSpi_Rst_Direction;	Direction of the Reset pin.	sbit CanSpi_Rst_Direction at TRISC2_bit;

Tableau3 : les fonctions de bus CAN

La bibliothèque CAN SPI :

- CANSPISetOperationMode
- CANSPIGetOperationMode
- CANSPIInitialize
- CANSPISetBaudRate
- CANSPISetMask
- CANSPISetFilter
- CANSPIRead
- CANSPIWrite

Chaque fonction à des variables bien définie et des paramètres a initialisé, pour plus de détails visité le Help de MicroC.

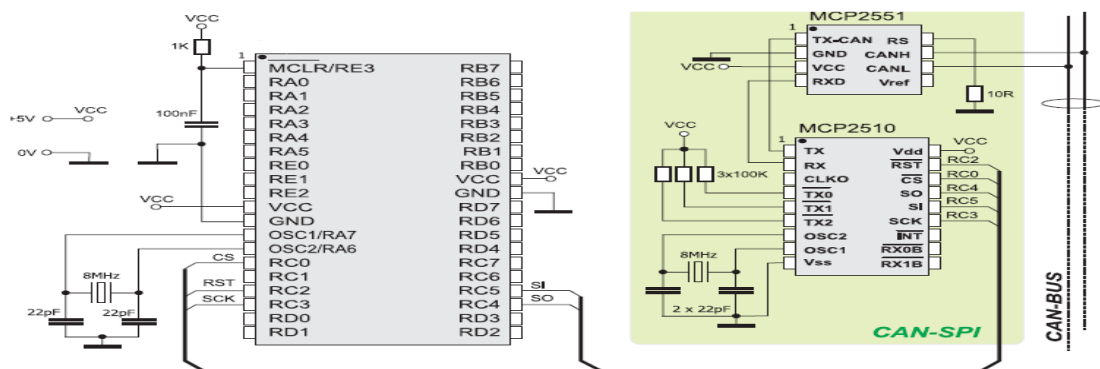


Figure 23 : réalisation de bus CAN

IV. Développement d'enregistreur de données avec ordinateur

1. Développement de l'interface

Le choix de LabVIEW pour le développement de notre interface pour l'enregistreur de données n'est pas par hasard, LabVIEW se différencie de la plupart des langages de programmation courants par deux caractéristiques essentielles :

- Premièrement, la programmation en G s'effectue en câblant des icônes graphiques sur un diagramme, lui-même ensuite compilé directement en code machine pour permettre aux processeurs de l'ordinateur de l'exécuter. Bien qu'il soit représenté graphiquement et non textuellement, le G obéit aux mêmes concepts de programmation que la plupart des langages traditionnels. Par exemple, le G inclut tous les éléments de construction standards, tels que types de données, boucles, gestion d'événements, variables, récursivité et programmation orientée objet.
- La seconde différence réside dans le fait que le code G développé avec LabVIEW s'exécute en fonction des règles de flux de données au lieu d'une approche basée sur des procédures plus classiques (en d'autres termes, une série de commandes séquentielles à exécuter), comme c'est souvent le cas avec la plupart des langages de programmation textuels tels que le C et le C++. Un langage de programmation par flux de données comme le G (ou Agilent VEE, Microsoft Visual et Apple Quartz Composer) met les données en valeur en tant que concept clé derrière chaque programme. Ce sont les données qui déterminent l'ordre d'exécution. C'est le flux de données entre les nœuds du programme, et non pas les lignes séquentielles de texte, qui détermine l'ordre d'exécution.

On veut développer une interface avec des onglets. Chaque onglet fait un tel traitement. L'intérêt est de faire l'acquisition des données numériques et analogiques afin de les traiter et les afficher sous forme des graphes et des jauges... Et enregistrer les données dans des fichiers. Le fonctionnement de cet enregistreur exige de le connecter avec l'interface, il dépend de l'ordinateur. Sinon la communication est désactivée.

L'interface développée comporte six onglets principaux :

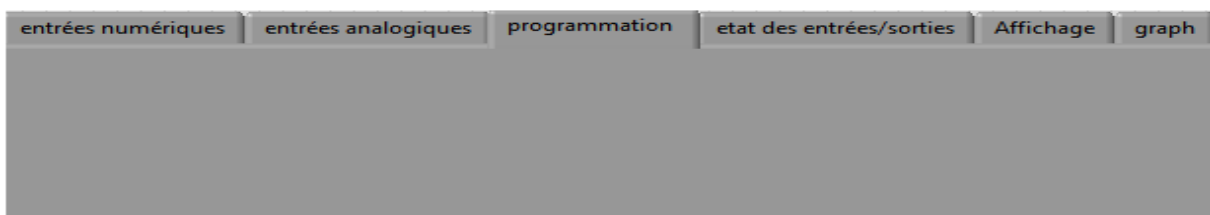


Figure 24 : les onglets de l'interface d'enregistreur avec ordinateur

1.1 Entrées numériques

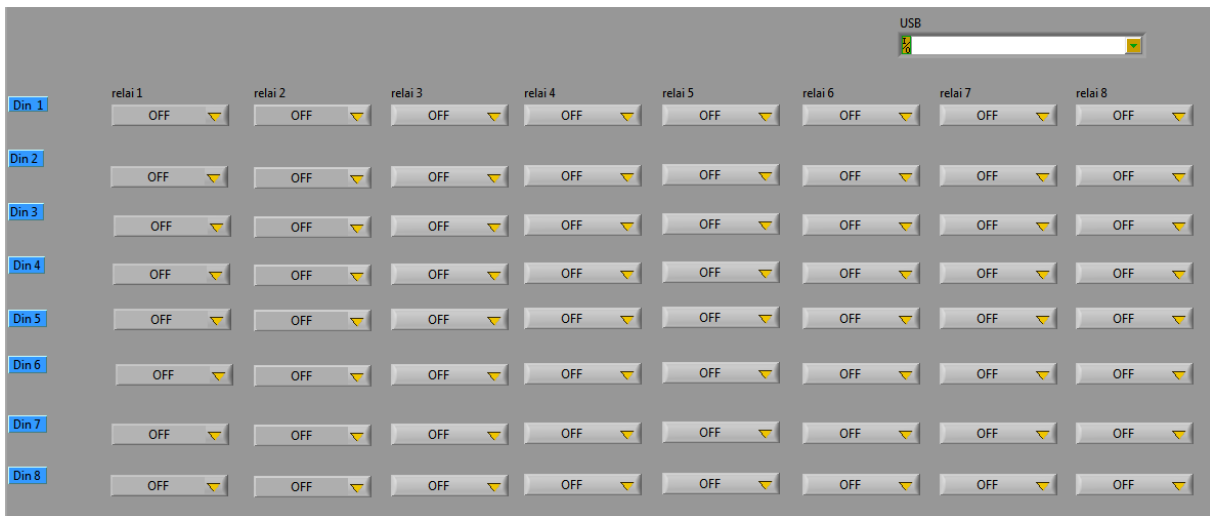


Figure 25 : entrées numériques

Cet onglet permet le traitement des entrées numériques, pour chaque entrée on définit les actions à activer si elle est à '1' logique. La figure montre clairement qu'on a huit entrées numériques et huit sorties. Les données d'entrées numériques correspondents a des données des capteurs si un capteur à l'état haut alors les sorties définies seront activées. Ce traitement permet l'enregistreur de réagir en cas d'urgence ainsi d'augmenter la surveillance des systèmes. L'icône en haut d'onglet permet le choix de port branché avec l'ordinateur.

1.2 Entrées analogiques

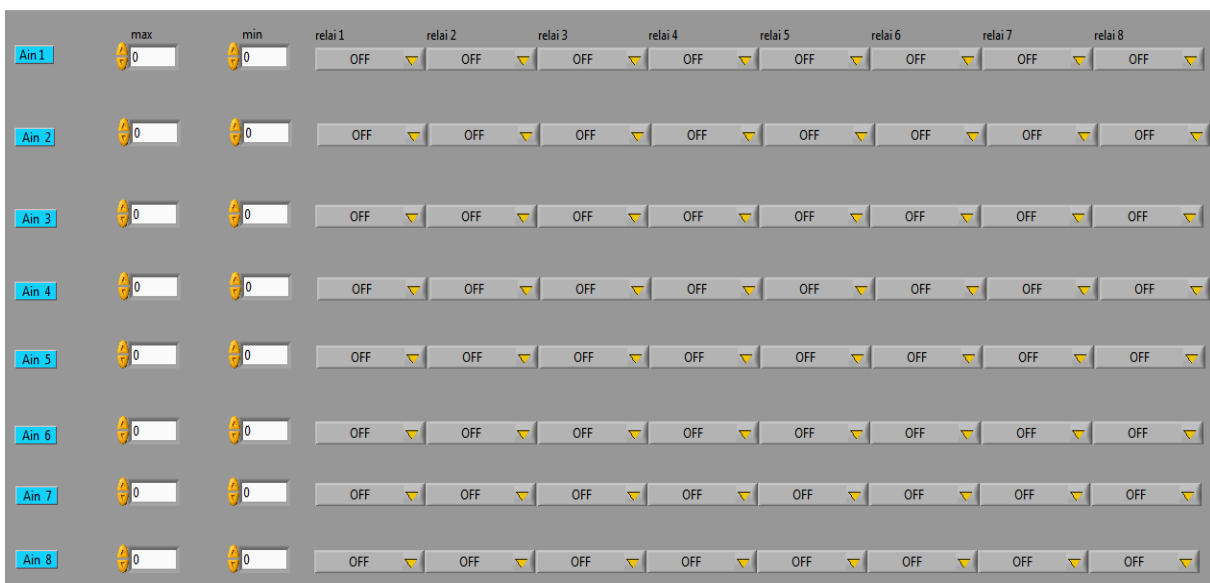


Figure 26 : entrées analogiques

Le traitement des entrées analogiques nécessite de définir un maximum et un minimum pour chaque entrée analogique, puis on définit les actions, si l'entrée dépasse le maximum ou elle inférieur au minimum, ces actions seront activées. À partir de ce traitement on peut établir la régulation des systèmes par des capteurs (température, humidité, pression ...) et garder leurs valeurs dans des marges bien définies par l'utilisateur à l'aide d'interface.

1.3 Programmation



Figure 27 : programmation

La partie programmation permet de définir quatre conditions par l'utilisateur et les actions à faire, la relation entre ces conditions est le AND logique, c'est-à-dire, si les quatre conditions sont vérifiées à la fois les actions s'activent, si l'une des conditions n'est pas vérifiée les actions seront désactivées. D'après la figure on constate qu'on a quatre lignes cela signifie qu'on peut programmer quatre cas possible, chaque cas contient quatre conditions.

Chaque condition peut être une variable parmi 18 cas possible :

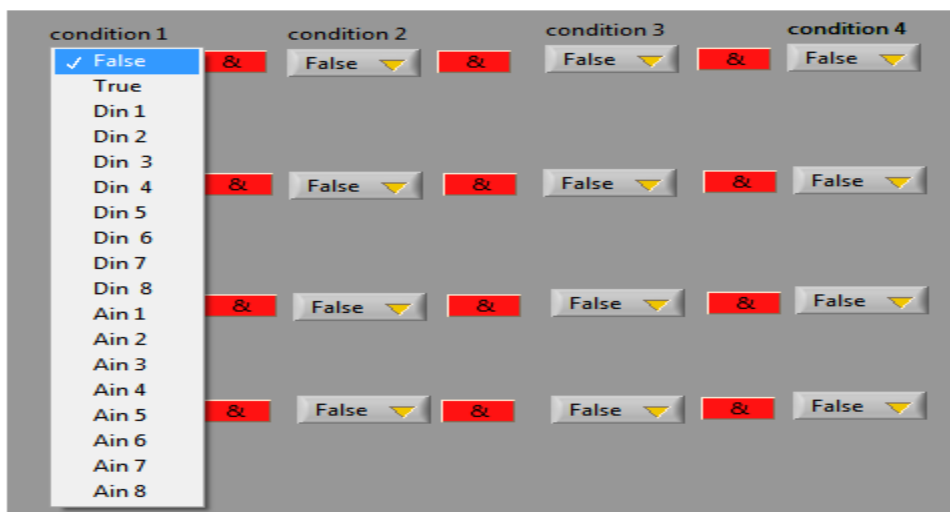


Figure 28 : les cas possible par une condition

Si l'une des conditions parmi les quatre est fausse alors les actions seront toujours désactivées quelque soit les autres conditions, par contre si elle est vraie alors l'exécution de l'action dépend seulement des autres trois conditions. Elle peut être aussi un cas parmi huit entrées numériques ou un parmi les huit entrées analogiques. Si la condition est à Din1 par exemple elle sera vraie lorsque l'entrée numérique 1 est à l'état haut. Par ailleurs si elle est à Ain 1 elle sera vérifiée quand l'entrée analogique dépasse le maximum ou elle est inférieure au minimum.

1.4 États des entrées/sorties



Figure 29 : états des entrées/sorties

Cet onglet consiste à afficher les états des entrées numériques s'elles sont à l'état haut ou bas, ainsi affiché quelles sont les relais activés, L'affichage est en temps réel. Cela permet l'utilisateur à suivre facilement les changements des états des entrées et des sorties de l'enregistreur de données à l'aide de l'interface sans avoir manipulé le matériel.

1.5 Affichage

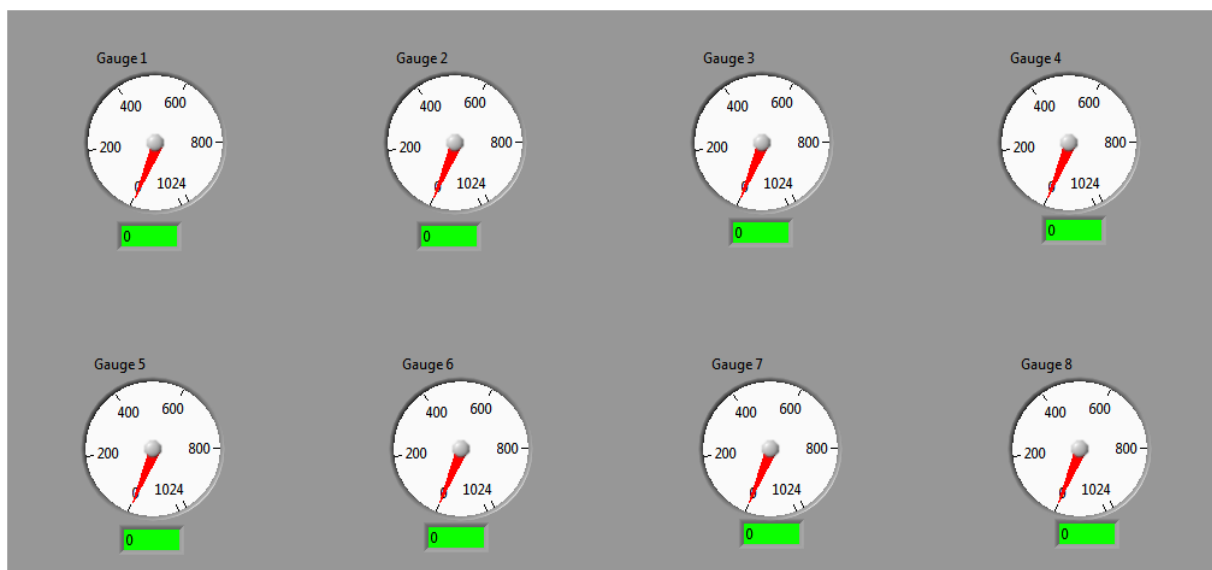


Figure 30 : affichage

Comme il est important de suivre le changement des états des entrées numériques et les états des sorties, il est important aussi de suivre l'évolution des entrées analogiques à partir des jauges, ce qui donne une vision claire sur les valeurs des capteurs branchés avec l'enregistreur de données.

1.6 Graphe

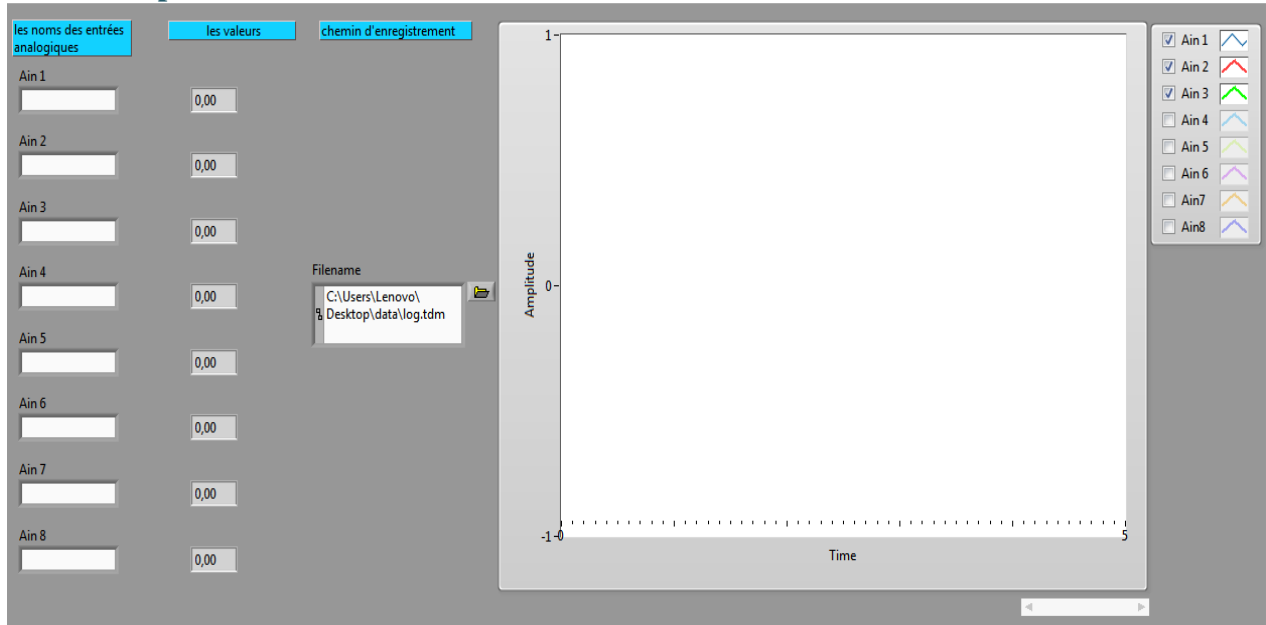


Figure 31 : affichage sous forme d'un graph

La majorité des enregistreurs de données réservent une partie pour l'affichage sous forme des graphes, il est intéressant de suivre les évolutions par des courbes notamment lorsqu'il s'agit de l'acquisition et l'affichage en temps réel. Pour notre projet nous avons décidé de tracer toutes les courbes dans un seul graphe afin de faciliter la comparaison et l'analyse des données. On peut également cocher ou décocher une courbe si on veut l'ajouter au graphe. A partir du même onglet on définit les noms des huit entrées analogiques pour l'enregistrer dans le fichier Excel, ainsi on choisit le chemin d'enregistrement.

2. Programmation de microcontrôleur

Le compilateur microC intègre une bibliothèque qui prennent en charge les dispositifs de classe HID quand un microcontrôleur PIC avec USB intégré est utilisé. La bibliothèque prend également en charge les routines génériques qui peuvent être utilisé avec la pilote de fournisseur spécifié. La description de la bibliothèque et les fonctions utilisées seront traitées à la deuxième partie de ce chapitre.

Le microcontrôleur dans ce cas permet l'échange des données avec l'interface, il envoie les données des capteurs (entrées numériques et analogiques) au logiciel labview afin de traiter ces dernières au code de diagramme Labview à partir les variables et les actions définies par

l'utilisateur à l'aide de l'interface. Puis il reçoit les actions et les résultats de traitement et les affectés aux sorties d'enregistreur de données.

La figure suivante montre le processus de fonctionnement au niveau de microcontrôleur :

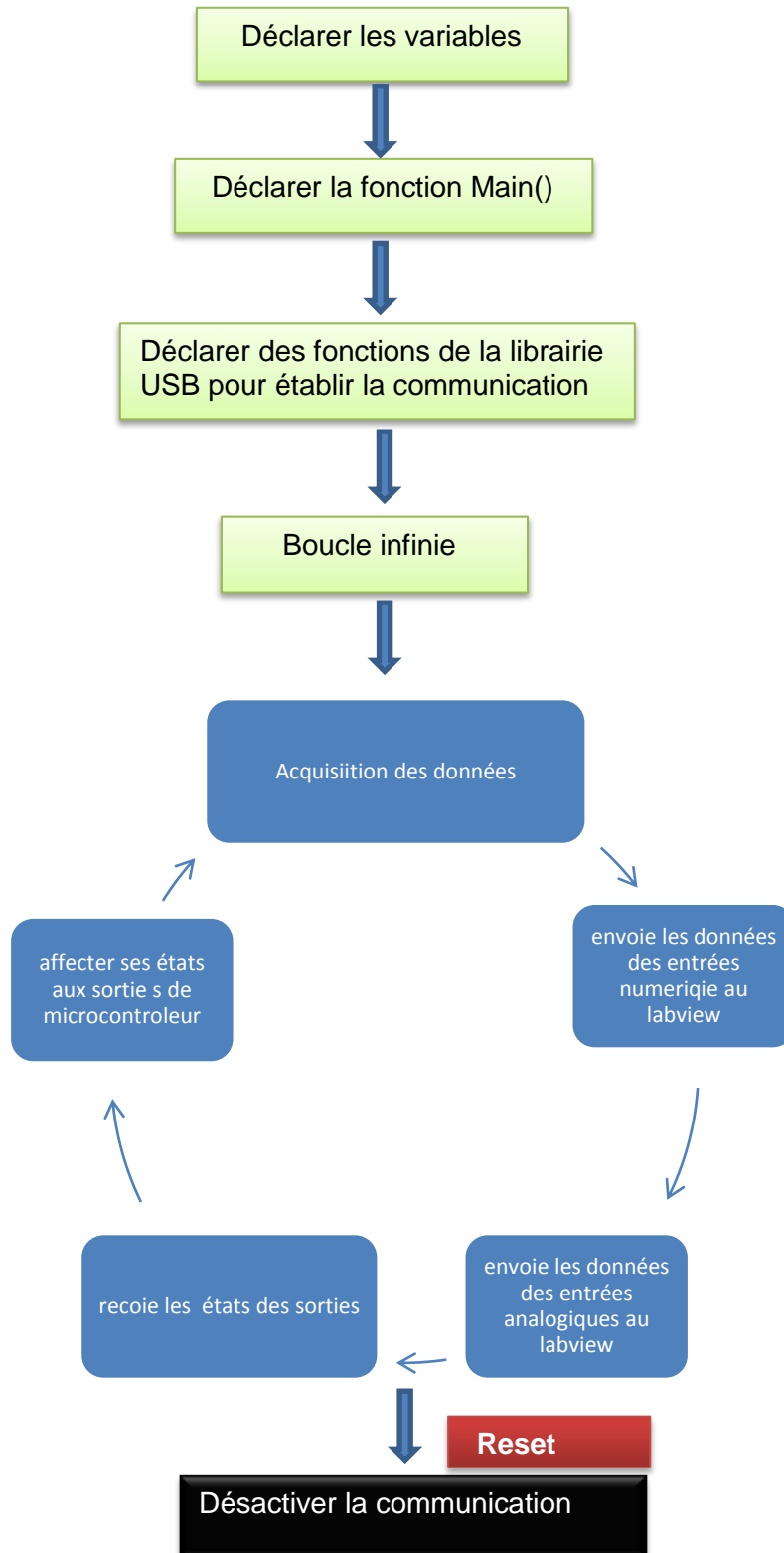


Figure 32: processus de fonctionnement au niveau de microcontrôleur

3. Conclusion

Nous avons développé un enregistreur de données puissant par rapport les classiques enregistreurs qui sont limités seulement pour l'acquisition et l'enregistrement de données. Notre projet bénéficie un nombre importants des entrées/sorties il peut supporter huit entrées numérique, huit entrées analogique et Controller huit sorties. Par ailleurs il est conçu pour exécuter des traitements sur ces entrées afin d'activer ou de Controller ses sorties.

D'un autre coté il profite d'une riche interface qui permettra de suivre les états des entrées et des sorties sous forme des jauges et des leds, ainsi de comparer et analyser les données sous forme des courbes. Toutes les données sont enregistrées dans un fichier Excel afin de garder une archive de l'évolution de chaque données. L'enregistrement sur PC bénéficie, pour l'enregistrement, de l'utilisation du disque dur d'ordinateur qui peut atteindre des téraoctets par rapport l'enregistrement sur des cartes externes avec des mémoires limitées.

Cette solution présente l'inconvénient d'être non autonome, le système doit être constamment connecté à l'ordinateur. Il fallait remédier à cela par un système totalement autonome indépendant du PC ou tout le programme d'acquisition et de contrôle et de conditions soit entièrement embarqué sur le PIC.

La partie suivante présente le développement d'enregistreur de données autonome qui peut fonctionner indépendamment au ordinateur et qui contient les même taches et les même actions de l'enregistreur avec ordinateur. Ce qui permet d'utiliser cet enregistreur dans des zones ou des taches qui n'ont pas besoin d'un ordinateur pour la visualisation en temps réel, ainsi il permet de fonctionner normalement quand il est connecté avec l'ordinateur.

V. Développement d'enregistreur de données autonome

1. Développement de l'interface

Au début nous avons planifié notre plan de travail pour la conception d'un enregistreur de données qui est capable de faire l'acquisition, l'enregistrement, le contrôle et le traitement des données. Pour cela le travail précédent exige toujours que l'enregistreur de données soit branché et connecté à l'interface si la communication est désactivée le fonctionnement s'arrête, car tout le traitement sur les variables définies à l'interface se fait au niveau de logiciel labview et le hardware (microcontrôleur) dans ce cas permet seulement l'envoi et la

réception des données avec l'interface. Le travail suivant permet le développement d'un enregistreur de données autonome qui peut fonctionner avec l'ordinateur on connectant le microcontrôleur avec l'interface, puis on définit les actions et les variables afin que le traitement se fasse au niveau de microcontrôleur et toutes les actions et les variables seront enregistrées dans la mémoire EEPROM de microcontrôleur. À ce niveau si on désactive la communication entre le hardware et l'interface le matériel continu le fonctionnement normal puisque toutes les variables sont enregistrées dans sa mémoire interne.

Pour cela l'interface est décomposée de six onglets principaux :

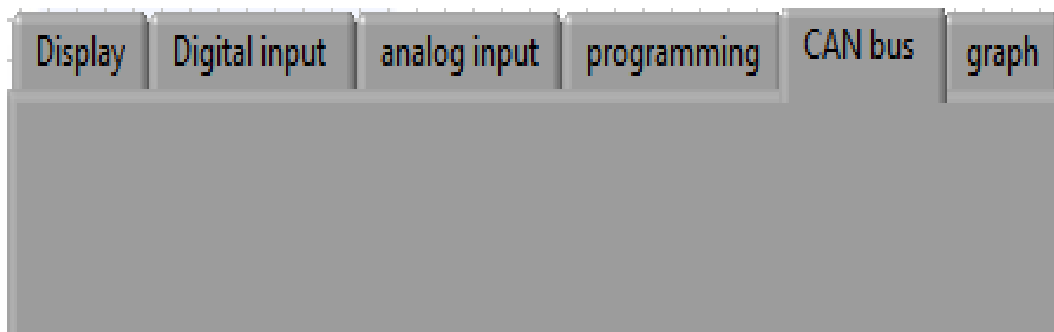


Figure 33 : les onglets de l'interface

Le premier onglet est réservé pour la configuration de port USB et afficher les valeurs et les états des entrées analogiques et numériques.

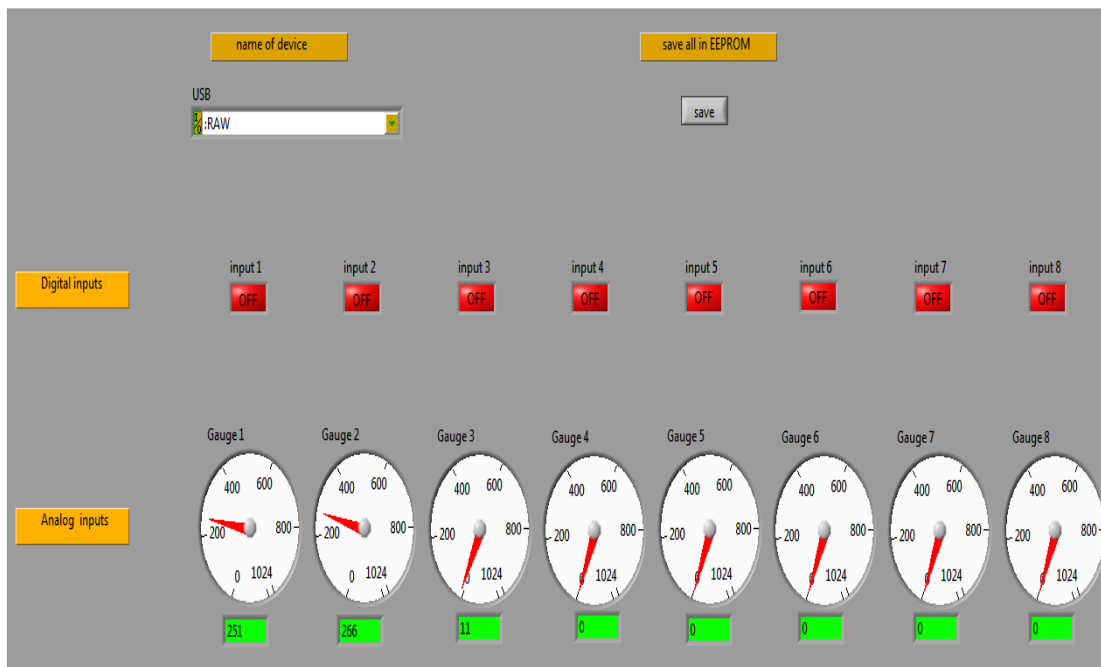


Figure 34 : onglet Display

Après la saisie de nom du port le quel le matériel est branché, couramment les valeurs et les états des entrées numériques et analogiques s'affichent. Cet onglet permet la visualisation réelle des entrées. Le bouton à droite permet le stockage de toutes les variables définies à l'interface a la mémoire EEPROM de microcontrôle, ce qui accède le microcontrôle de fonctionner indépendamment à l'interface et l'ordinateur.

➤ le deuxième onglet contient le traitement des données numériques et les sorties qu'on veut activer.

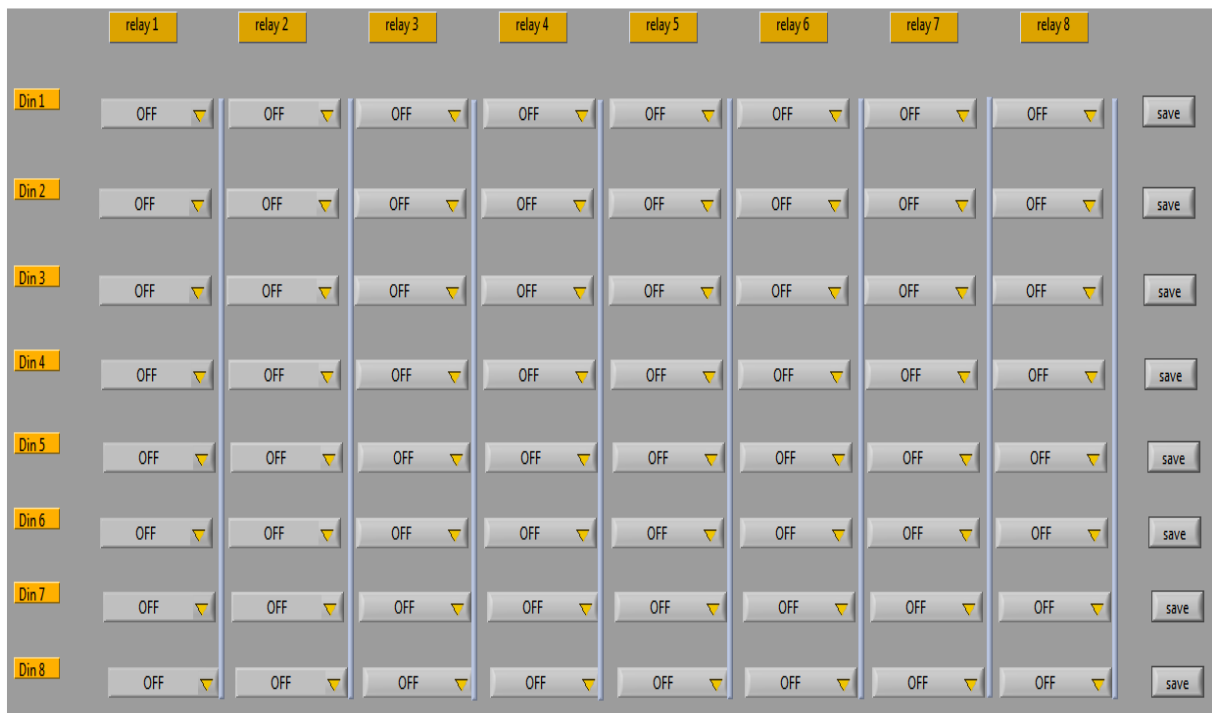


Figure 35 : Digital inputs

Après avoir défini à chaque entrée numérique les sorties que nous voulons activer on appuie sur les boutons « save » ce dernier permet d'envoyer une trame au microcontrôle et stocké les données définies dans des variables dans la mémoire RAM de microcontrôle.

Si une entrée est à l'état 1 alors les sorties définies sont activées. Les entrées numériques représentent les capteurs chaque fois qu'un capteur change d'état alors les sorties liées à ce capteur change d'état aussi.

➤ Le troisième onglet permet le traitement des données analogiques et définir à chaque entrée parmi 8, le maximum et le minimum afin d'activer les sorties souhaitées.

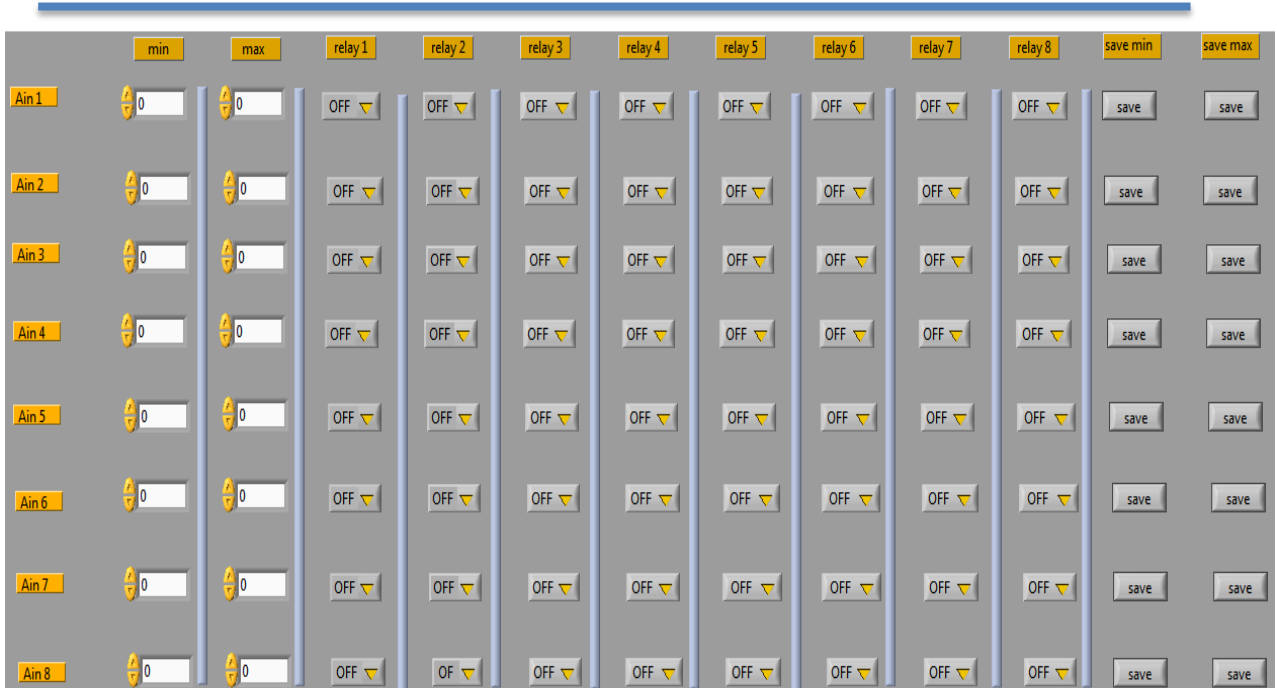


Figure 36 : Analog inputs

Pour chaque entrée on choisit le maximum et le minimum et les sorties à activer. On a deux boutons « save », le premier permet d'enregistrer le minimum et ses sorties. Et le deuxième permet d'enregistrer le maximum et ses sorties.

➤ Le quatrième onglet est consacré pour la partie programmation, à partir de cet onglet on peut choisir et définir des conditions et les actions. Cette partie permet l'automatisation d'un système et contrôler les sorties.



Figure 37 : Programming

Si les trois conditions définies sont vérifiées à la fois, les sorties définies sont activées. Sinon c'est l'une des conditions n'est pas vérifiée les relais seront désactivés. Les boutons « save » à droite permet d'envoyer les données au microcontrôleur pour le traitement.

Une condition peut prendre 18 cas possible, les huit entrées numériques et les huit analogiques et deux conditions vraie et fausse.

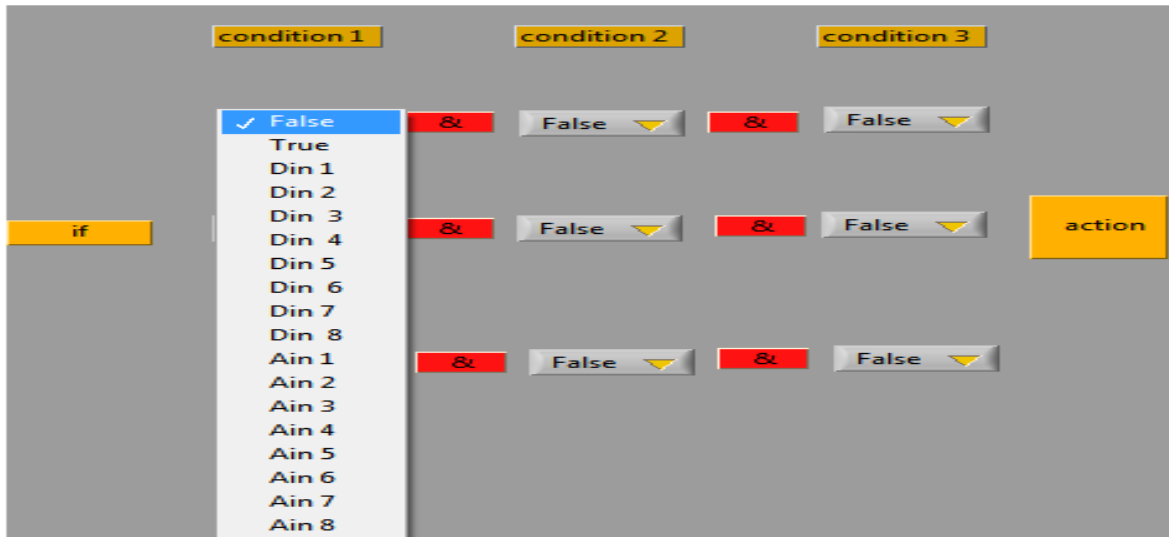


Figure 38 : les cas d'une condition

Si la condition choisit est Fausse les sorties ne seront jamais activées quel que soit les autres deux conditions, par contre si elle est vraie les sorties seront dépendants juste des deux autres conditions.

- Le cinquième onglet permet d'envoyer et recevoir des données à partir le bus CAN et changer les données et des messages entre les équipements connecté

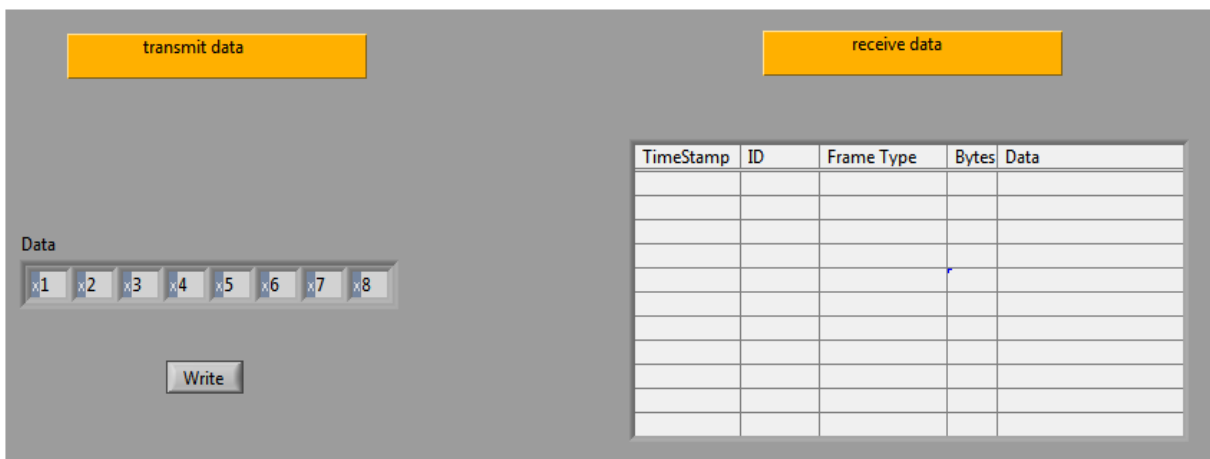


Figure 39 : bus CAN

- Le sixième onglet permet l'affichage des entrées analogiques sous forme des valeurs numériques, ainsi de tracer les courbes des huit entrées analogiques,

l'icône droite de graphe permet d'ajouter des courbes correspondes aux entrées analogiques dans le graphe. si l'un est décoché alors il n'apparaîtra pas avec les autres courbes.

À l'aide de Cet onglet on saisit les noms des entrées analogiques et le chemin d'enregistrement. Les données sont enregistrées dans un fichier Excel contient 9 colonnes, la première désigne la date de chaque échantillon et les autres colonnes correspondent aux huit entrées analogiques.

L'utilisation d'un seul graphe pour tracer les courbes est perspective, il permet de comparer les courbes entre elles et de visualiser facilement les variations. La visualisation est en temps réel.

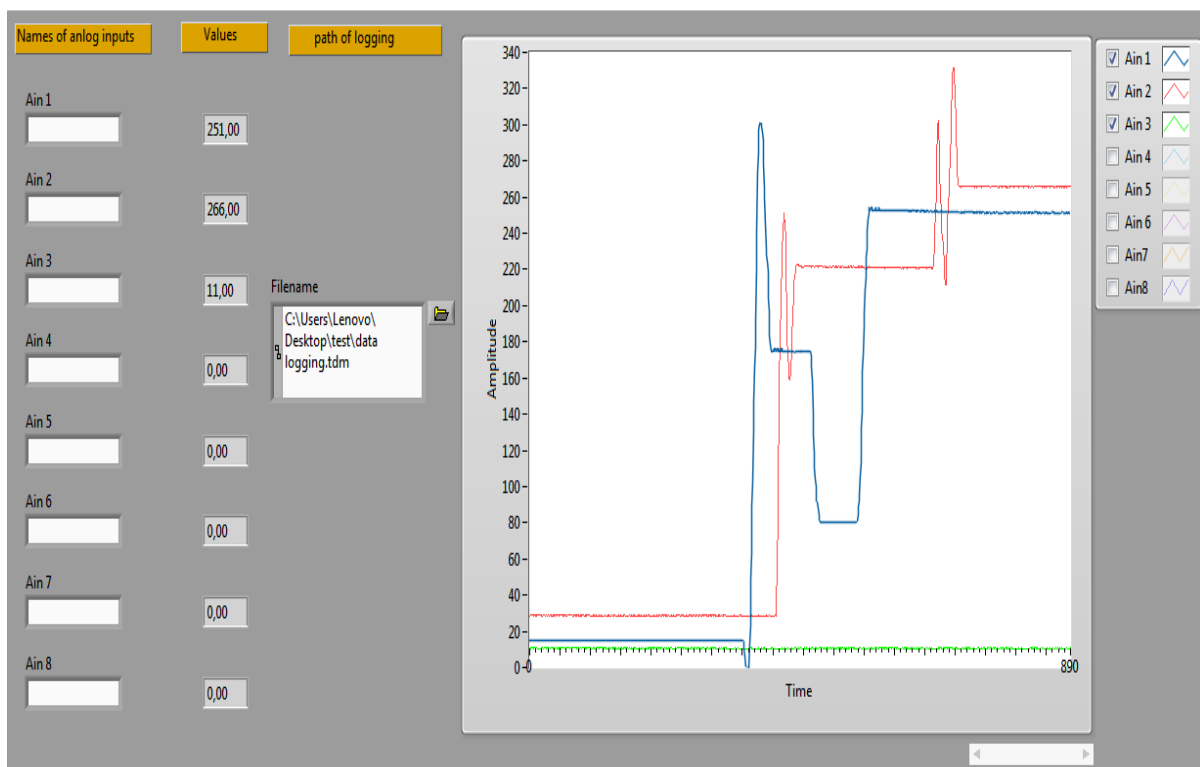


Figure 40 : onglet Graph

2. Les trames et adressage

Les enregistreurs de données classiques permettent de stocker seulement des données dans des fichiers afin de les exploiter quand les enregistreurs connecter à l'ordinateur et tracer des graphes manuellement, par exemple à l'aide d'Excel.

Tout à fait les enregistreurs de données modernes ils ont des fonctionnalités de plus, ils n'ont pas consacré seulement pour le stockage de données, mais ils peuvent faire le traitement et l'analyse des données en temps réel, ce qui fait on peut visualiser en temps réel les données de l'enregistreur.

Des fois l'utilisation et le choix des enregistreurs de données ne nécessite pas la visualisation en temps réel, c'est le cas des stations météo et des utilisations dans des zones difficiles de connecté un enregistreur de données avec l'ordinateur. Or notre but est de réaliser un enregistreur de données capable de répondre aux tous les besoins dans l'industrie et d'environnement. On cherche à généraliser notre enregistreur le plus maximum pour l'intégration dans différent domaines.

Pour cela nous avons pensé à la conception d'un enregistreur de données qui fait le stockage, la visualisation en temps réel et le traitement des données, en même temps, il peut fonctionner indépendamment de l'interface ou cas ou on ne veut pas faire la visualisation en temps réel. Or l'utilisateur ne trouve pas un problème de brancher et débrancher l'enregistreur à chaque fois qu'il veut visualiser les résultats comme le cas des classiques enregistreurs de données. Mais il peut utiliser le même enregistreur s'il veut la visualisation et le traitement en temps réel c'est le cas des enregistreurs modernes. Avec cette hybridation de fonctionnement et d'utilisation. L'enregistreur devient capable d'accompagner un ordinateur tout le temps pour l'acquisition et la visualisation en temps réel, aussi pour les fonctionnements et les utilisations indépendamment aux ordinateurs et aux interfaces.

La réalisation des taches précédentes au niveau de software nécessite que le traitement de données soit réalisé au niveau du microcontrôleur et non pas au niveau de logiciel Labview. car si on veut débrancher et employé le data logger indépendamment à l'interface il faut que le microcontrôleur reste capable de faire le traitement et mémorise les variables définies par l'utilisateur. Par ailleurs il est capable de fonctionner normalement quand il est branché à l'ordinateur.

Pour répondre à ces contraintes nous avons pensé a employer des trames à changer entre le microcontrôleur et l'interface, à ce niveau la définition des taches, des actions et des variables seront à l'aide de l'interface. Ainsi de suite les envoyés au microcontrôleur sous forme des trames, ce dernier enregistre ces variables dans la mémoire ROM et EEPROM et les employées au traitement, donc on peut débrancher l'enregistreur de l'interface et il continue le

fonctionnement normal, car tout ce qui est défini par l'utilisateur dans l'interface il est enregistré au niveau de microcontrôleur.

Chaque fois que l'utilisateur définit une telle action ou une valeur d'une variable il appuie sur le bouton « save » à l'instant le logiciel Labview envoie la trame qui contient ses actions et ses valeurs au microcontrôleur. Ce dernier reçoit cette trame et enregistre ses actions et ses variables dans des tableaux dans la mémoire EEPROM.

La taille de la trame est de 3 octets. Chaque octet contient des bits des adresses. Et Chaque bit désigne une donnée bien précise.

- Le premier octet contient la valeur de minimum ou de maximum, il présente 8 bits donc le maximum et le minimum peuvent atteindre la valeur $2^8 - 1 = 255$ valeurs .
- Le deuxième octet désigne les adresses, pour que le microcontrôleur soit capable de différencier entre une action d'une entrée numérique, analogique ou de la partie programmation.
- Le dernier octet contient les états des sorties. On a 8 sorties donc à chaque fois qu'on définit quelle sont les relais qu'on veut activer, leurs états enregistrés dans ce dernier octet et envoyés vers le microcontrôleur.

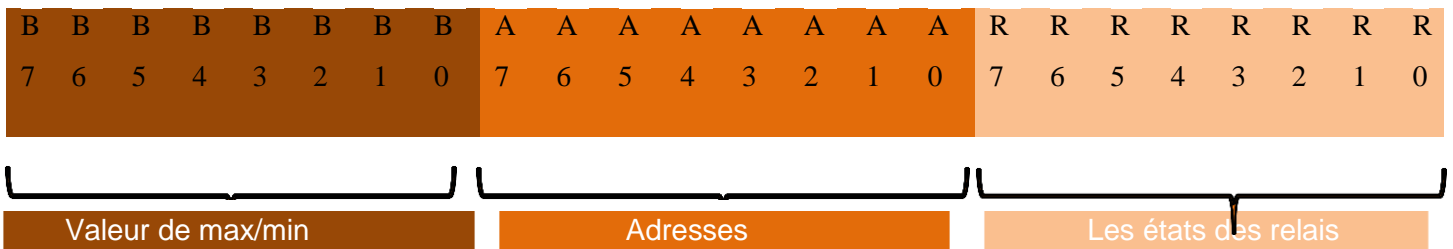


Figure 41 : présentation de la trame

À chaque clique sur un bouton « save » une trame est envoyée vers le microcontrôleur, on définit trois types des trames :

- ❖ Une trame désignée pour les entrées numériques, tous les bits du premier octet sont nuls, car on n'a pas un max/min pour les entrées numériques. Le deuxième octet contient une adresse signifie que cette trame est pour les entrées numériques. Et le troisième porte les états des relais.

❖ Une trame contienne le traitement des entrées analogiques, le premier octet comporte la valeur de maximum ou de minimum, selon le bouton appuyé. Le deuxième octet désigne que ses données sont pour les entrées analogiques, ainsi le dernier octet contient les états des sorties (relais).

❖ Une trame contienne les conditions de la partie programmation, chaque appuie sur le bouton « **save** » la trame envoyée porte la condition et les états des relais qu'on veut activer. Donc le premier octet contient la condition, le deuxième désigne que ses données sont de la partie programmation et le troisième contient les états des relais.

Lors de l'exécution, chaque fois les trames sont envoyées de l'interface vers le microcontrôleur.

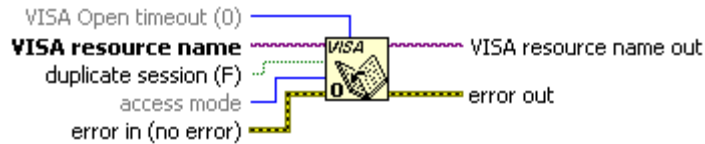
- si on appuie sur un bouton « save », les données appropriées sont envoyées et le microcontrôleur reçoit ses dernières et les affectant à des variables pour les exploiter dans le traitement.
- Sinon une trame est envoyée par défaut, avec une adresse inconnue vers le microcontrôleur qui ne la reçoit pas.

Après avoir fini la saisi dans l'interface on peut visualiser en temps réel les états de chaque entrée et les états des sorties, par contre on peut débrancher l'enregistreur de l'ordinateur et il continue le fonctionnement normal, car toutes les données saisies par l'utilisateur sont enregistrées dans la mémoire de microcontrôleur, alors il devient indépendamment de l'interface.

3. L'envoi des trames

Pour chaque entrée numérique ou analogique quand l'utilisateur définit des actions, il appuie sur le bouton « **save** ». A l'instant la trame appropriée à cette entrée est envoyée vers le microcontrôleur. L'envoi des trames s'effectue au niveau de l'interface (au niveau de logiciel labview) à l'aide de l'interface de programmation VISA.

VISA permet la lecture et l'écriture des données de port USB. Il faut d'abord ouvrir VISA avec les configurations appropriées au port connecté avec le hardware, la fonction **VISA Open** permet d'initialiser et ouvrir le port.



Inputs	
	VISA Open timeout
	VISA resource name
	duplicate session (F)
	access mode
	error in

Outputs	
	VISA resource name out
	error out

Figure 42 : VISA Open

Quand l'interface est connectée avec le microcontrôle via le port USB, la fonction **VISA Write** permet d'écrire les données au port.

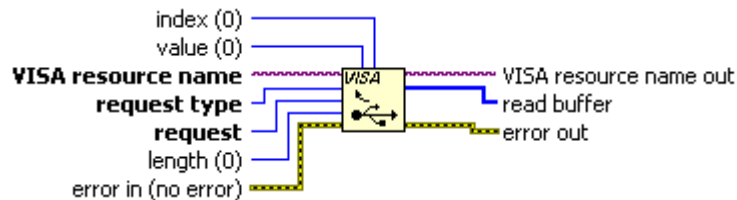


Inputs	
	VISA resource name
	write buffer
	error in

Outputs	
	VISA resource name out
	return count
	error out

Figure 43 : VISA Write

La fonction **VISA USB control** permet de lire les données de port USB afin de les exploiter dans le traitement labview.



Inputs	
	index

Outputs	
	VISA resource name out









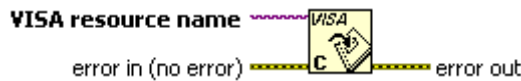
	value		read buffer
	VISA resource name		error out
	request type		
	request		
	length		
	error in		

Figure 44 : VISA USB control

Pour arrêter la transmission des données, on appelle la fonction **VISA Close**.






Inputs		Outputs	
	VISA resource name		error out
	error in		

Figure 45: VISA Close

Après établir la communication et la transmission des données entre l'interface et le PIC18F4550. Notre programme permet de traité et contrôlé les données, on a réservé les trois derniers bits d'octet adressage pour le type des données (numériques, analogique, programmation) :

A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----

Tableau 4 : octet d'adressage

A7	A6	A5	Types des données
0	0	0	Données numériques
0	0	1	Données analogiques
0	1	0	Données de partie programmation
0	1	1	Utilisation future
1	0	0	Utilisation future
1	0	1	Utilisation future

1	1	1	Écrire les données dans l'EEPROM
1	1	1	Utilisation future

Tableau 5: les adresses des types des données

Puisque on a trois types de trames, et nous avons huit entrées numériques et huit analogiques, donc à chaque type il faut distinguer entre ses entrées. On a réservé les trois premiers bits d'octet d'adressage pour différencier entre ces entrées. Le tableau suivant montre leurs adresses.

A2	A1	A0	entrée (numérique/analogique)	
0	0	0	Entrée	1
0	0	1	Entrée	2
0	1	0	Entrée	3
0	1	1	Entrée	4
1	0	0	Entrée	5
1	0	1	Entrée	6
1	1	0	Entrée	7
1	1	1	Entrée	8

Tableau 6 : adresses des entrées

Les deux bits qui restent A3 et A4 sont désignés pour faire la différence entre le maximum et le minimum.

A4	A3	Action
0	0	Désigne que L'octet 1 contient le maximum
0	1	Désigne que L'octet 1 contient le minimum
1	0	Utilisation future
1	1	Utilisation future

Tableau 7: adresse de minimum/maximum

On résume, l'octet 2 de la trame (octet d'adressage) c'est-lui qui permet de différencier entre les entrées et leurs types. Concernant les données de la partie programmation, il faut envoyer les trois conditions et les actions. Chaque condition peut prendre 18 cas. Donc il faut 5 bits pour adresser les 18 cas.

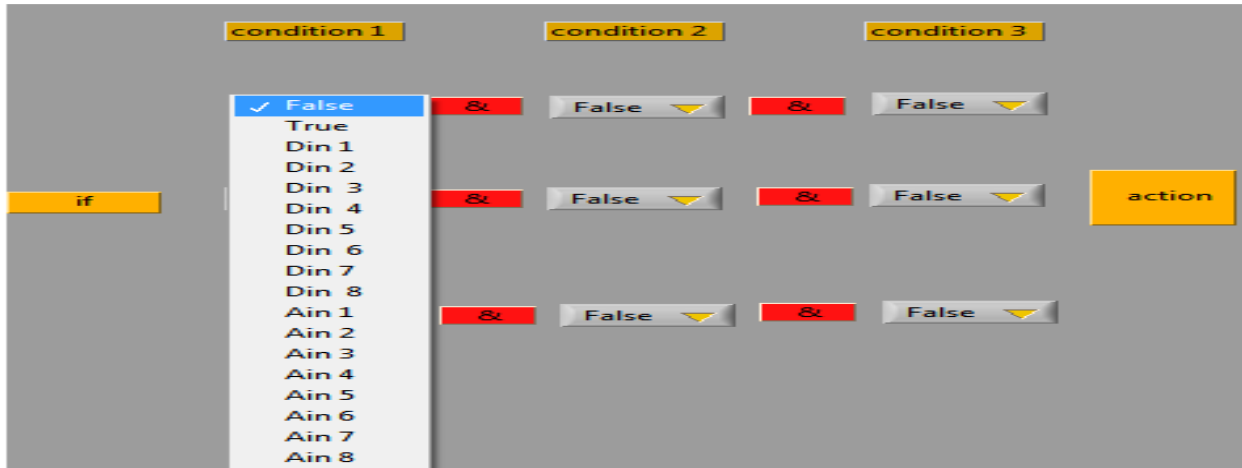


Figure 46 : les cas d'une condition

Or on ne peut pas envoyer les trois conditions et les actions dans la même trame. Ce qui fait-on a décidé d'envoyer chaque condition dans une seule trame. La figure ci-dessous montre l'onglet de la partie programmation chaque ligne contient 3 boutons permet de choisir les conditions. Et 8 boutons permet l'activation des sorties, et fini par 3 boutons de « save » le premier correspond à l'envoi de première condition, le deuxième onglet pour la deuxième condition, et le dernier pour la troisième condition.



Figure 47 : onglet de programmation

D'après l'onglet précédent on constate qu'on a 3 conditions pour chaque ligne, or 9 conditions en tout, nous avons réservé les 4 premiers bits (A3, A2, A1, A0) d'octet d'adressage pour distinguer entre eux.

La figure suivante présente La forme de la trame pour l'envoi d'une condition.

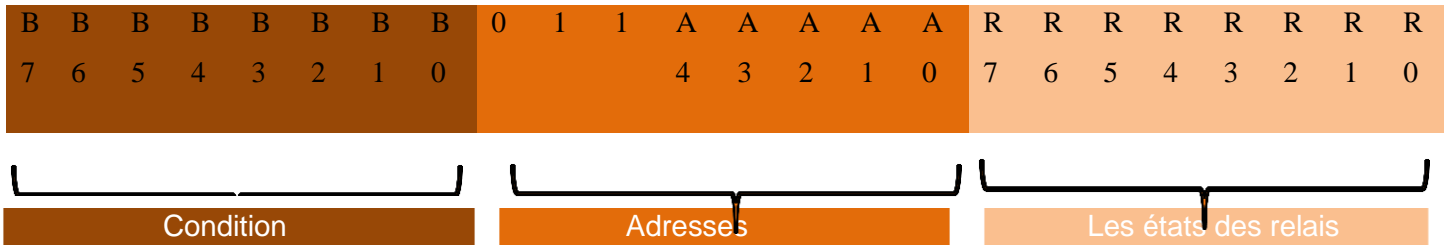


Figure 48 : l'envoi d'une condition

Conclusion

Chaque bouton « save » correspond à un type des données (numériques, analogiques, programmation), lors d'une appuie sur le bouton la trame approprié est envoyée de l'interface vers le microcontrôleur sous forme de trois octets, le premier contient les valeurs de minimum, maximum et la condition. Le deuxième désigne l'adresse correspondante et le troisième octet contient les états des sorties. Nous avons réservé certain bits pour l'utilisation future et l'amélioration de projet.

4. Réception des trames

Après avoir envoyé les données de l'interface sous forme des trames, le microcontrôleur reçoit ses trames et stock ses variables dans des tableaux afin de les exploitées dans le traitement.

Le compilateur microC intègre une bibliothèque qui prennent en charge les dispositifs de classe HID quand un microcontrôleur PIC avec USB intégré est utilisé. La bibliothèque prend également en charge les routines génériques qui peuvent être utilisés avec la pilote de fournisseur spécifié.

➤ **HID_Enable()** cette fonction permet la communication USB et nécessite deux arguments, l'adresse de la lecture tampon et l'adresse d'écriture tampon. Avant de pouvoir utiliser toutes les autres fonctions, cette fonction doit être appelée en premier. Cette fonction ne renvoie aucune donnée.

Exemple

```
unsigned char read_buff[64] absolute 0x500;
unsigned char write_buff[64] absolute 0x540;
```

```
HID_Enable(&read_buff,&write_buff); // activer la communication HID
```

Remarque :

Les tampons USB pour l'envoi et la réception de données doivent être placés dans l'USB RAM. Pour chaque microcontrôleur il faut consulter sa fiche technique pour avoir les adresses convenable d'USB RAM. Pour notre pic18f4550 USB mémoire RAM est de 0x500 à 0x7FF comme le montre la figure ci-dessous.

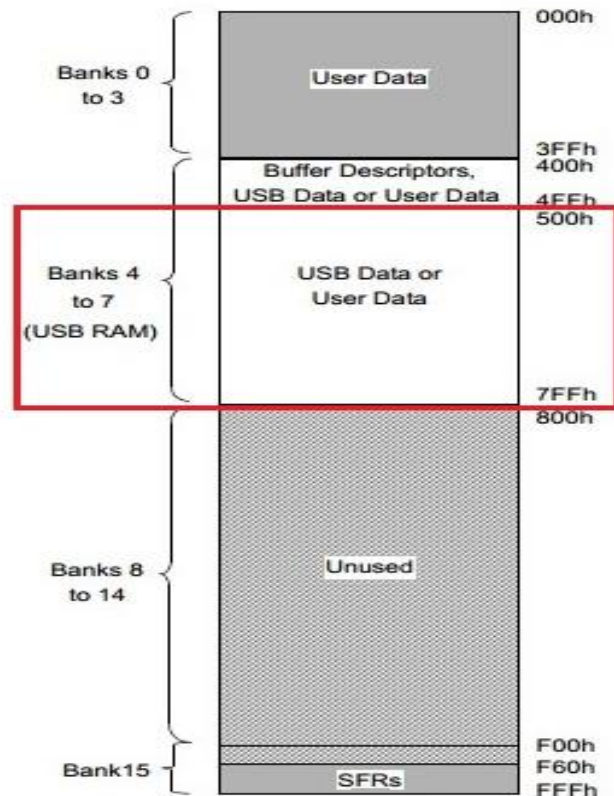


Figure 49: la mémoire RAM USB PIC18F4550

➤ **HID_Read ()** cette fonction reçoit des données à partir de bus USB et le stocke dans la mémoire tampon de la lecture. Elle n'a pas d'argument mais renvoie le nombre de caractères reçus.

➤ **HID_Write ()** cette fonction envoie des données à partir du tampon d'écriture sur le bus USB. Le nom du tampon (le même tampon que celui utilisé lors d'initialisation) et la longueur des données à transmettre doit être spécifiées comme argument de la fonction. Si la transmission de données à échouer, la fonction renvoie 0, sinon elle retourne le nombre d'octet transmis.

Exemple : `HID_Write (&write_buff, 17) ;`

➤ **USB_Interrupt_Proc ()** cette routine utilisée pour l'entretien des divers évènements de bus USB, elle doit être appelée à l'intérieur de la routine d'interruption USB.

➤ **HID_Disable** cette fonction désactive le transfert de données. Elle n'a pas d'argument et ne renvoie aucune donnée. Exemple : `HID_Disable ()` ;

Lors de programme nous avons créé des fonctions qui permettent l'acquisition de données émis par labview, ainsi pour envoyer des données à l'interface pour les afficher sous forme des graphes et des jauges.

- **GET_Digital_Data()**; cette fonction permette d'envoyée les valeurs des entrées numériques à l'interface.
- **GET_Analog_Data()**; cette fonction permette d'envoyée les valeurs des entrées analogiques à l'interface.
- Des fonctions pour le traitement et le control des sorties.

Conclusion

Les fonctions de la bibliothèque USB intégrées dans le logiciel microC permettent d'établir la communication entre l'interface et le pic18f4550. Des fonctions déclarées dans le code permettent de lire les données via l'interface et d'écrire (d'envoyer) des données vers l'interface pour les afficher et les sauvegarder dans des fichiers. La figure suivante montre le processus pour la réception de données au niveau de microcontrôleur.

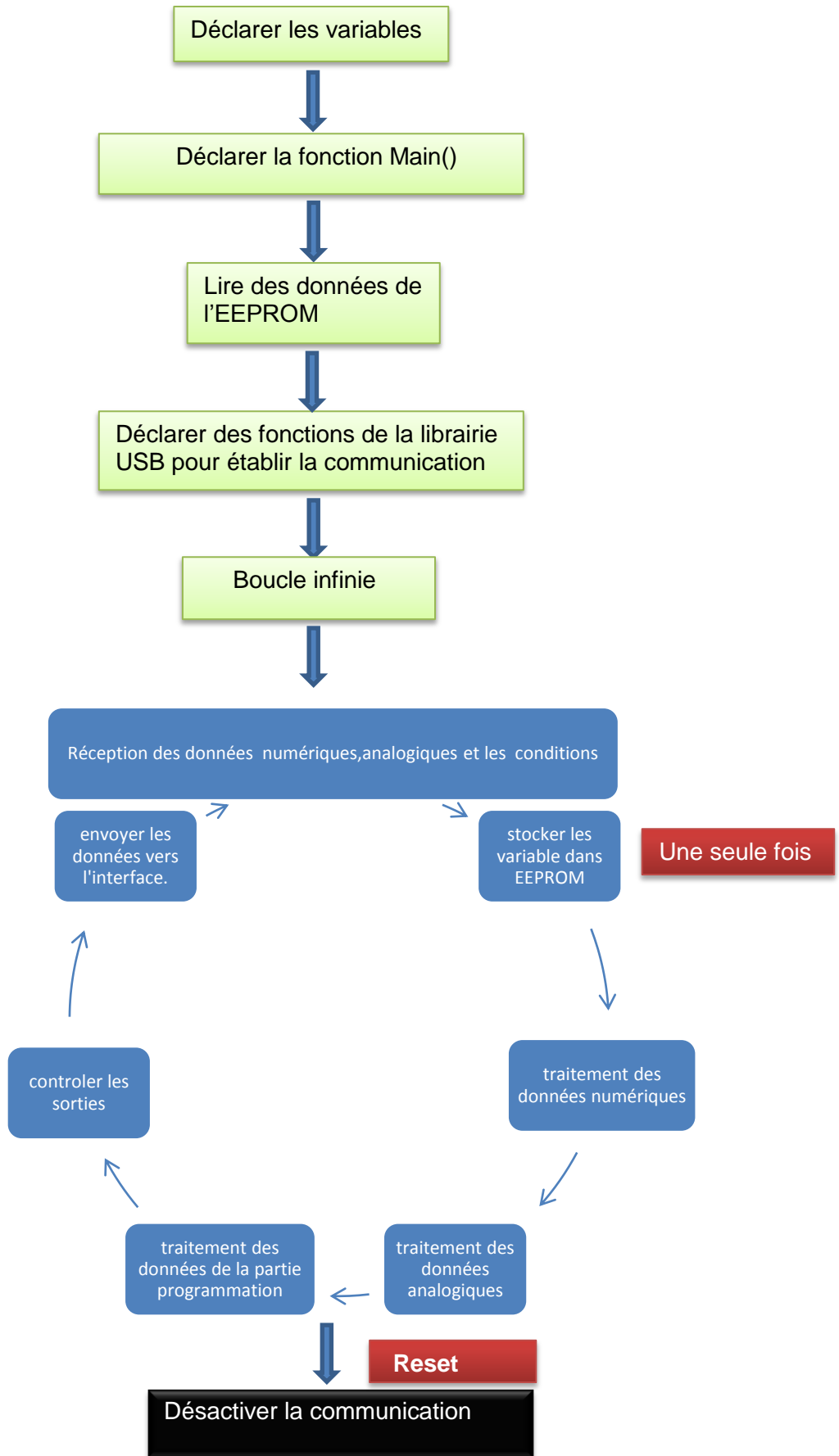


Figure 50: processus de la réception de données

VI. Description de fonctionnement et simulation

Depuis la définition de cahier de charge complet et lors de la programmation et d'analyse de fonctionnement nous avons décomposé la programmation de notre projet en cinq parties principales, chaque partie est accompagnée d'une simulation de fonctionnement, le passage de la partie suivante nécessite que les parties précédentes donnent des résultats souhaités. Durant ces cinq parties j'ai travaillé avec trois logiciels : labview pour le développement de l'interface le microC pour programmer le microcontrôleur et isis proteus pour la simulation.

➤ La première partie destinée seulement à l'acquisition des données des entrées numériques et analogiques et les afficher sous forme des jauges et des graphes. Cette partie est l'introduction aux logiciels de développement et de simulation, ainsi me familiariser avec des membres d'équipe et des conditions de travail.

➤ La deuxième partie pour réaliser le traitement et le contrôle des entrées numériques, lors de cette partie nous avons travaillé sur 8 entrées numériques, pour chaque entrée on peut définir des sorties à activer parmi 8 sorties. La simulation nécessite de développer un onglet conçus à cette partie et des fonctions au niveau de microcontrôleur et la simulation avec isis proteus.

➤ La troisième partie dédiée au traitement et le contrôle des entrées analogiques. À partir de l'interface on peut définir un minimum et un maximum pour chaque entrée parmi 8 et les relais (sorties) qu'on veut activer si une entrée dépasse le maximum ou elle est inférieure au minimum.

➤ La quatrième partie permet de définir une tâche importante, cette dernière permet de choisir trois conditions et les sorties qu'on veut activer si ces conditions sont vérifiées à la fois. Sinon, si l'une des conditions n'est pas vérifiée les sorties défini lors cette parties ne seront pas activées. Une condition peut prendre une entrée parmi les 8 entrées numériques, ainsi parmi les 8 entrées analogiques, ou peut prendre un cas vrai ou faux. A travers cette partie on peut automatiser un système et la possibilité de rassembler les résultats des plusieurs entrées pour faire une action.

➤ La dernière partie conçue pour l'enregistrement des données dans des fichiers, ainsi pour le design de l'interface et la documentation et l'optimisation des codes de programmation.

Durant ces parties nous avons obtenue de bons résultats et nous avons dépassé les contraintes et les obstacles rencontrés lors de la programmation. Pour chaque partie la simulation de fonctionnement étant obligatoire avant d'entamer la tâche suivante.

Le reste de ce chapitre traitera une simulation globale de fonctionnement d'enregistreur de données. Nous essayerons de réaliser le système suivant à l'aide de notre data logger.

Cahier de charge :

La fabrication et la fermentation de la levure nécessite plusieurs processus à respecter. Le principal processus c'est la fermentation de la levure dans des cuves supportés des dizaines de tonnes. Cette partie de fabrication n'est pas tout à fait simple c'est la partie qui nécessite plus de centralisation et toujours assuré les bonnes conditions à l'intérieur des cuves pour garantir l'accélération de la fermentation et obtenu la bonne qualité.

Les paramètres influençant l'activité levurienne sont :

❖ Concentration d'alcool

Passée une certaine concentration (de 25 à 37 %), il agit comme un poison pour les levures.

❖ La Température

L'augmentation de la température, jusque 40°C cependant, accélère la fermentation des sucres par la levure. À partir de 50°C (début de la cuisson du pain), l'inactivation des levures commence. On considère que la température Optimale de fermentation se situe aux environs de 27°C. Aux températures basses, proches de 10°C, l'activité est quasiment nulle.

❖ L'humidité :

L'augmentation de l'humidité conduit à l'inactivation de la levure et la déshydratation. L'humidité optimale est environ de 50%.

Pour cela il faut toujours assurer la régulation de la température et les autres caractéristiques qui influencent sur la fermentation de la levure et les garder environ des valeurs optimales.

Notre enregistreur de données n'est pas dédié pour l'automatisation des systèmes, même si il peut jouer le même rôle des systèmes de programmation. Habituellement il fonctionne comme les autres enregistreurs de données industriels il est responsable seulement sur l'enregistrement, l'acquisition et affichage des données. Comme il est déjà cité notre enregistreur de données permet l'acquisition,

enregistrement et le contrôle des sorties ou cas ou d'urgence ou de défaillance. Le contrôle des sorties rendre notre projet plus puissant et élargie ses domaines d'utilisation afin de trouver rapidement sa place dans l'industrie.

La figure suivante montre un schéma synoptique de notre solution pour la fermentation de la levure.

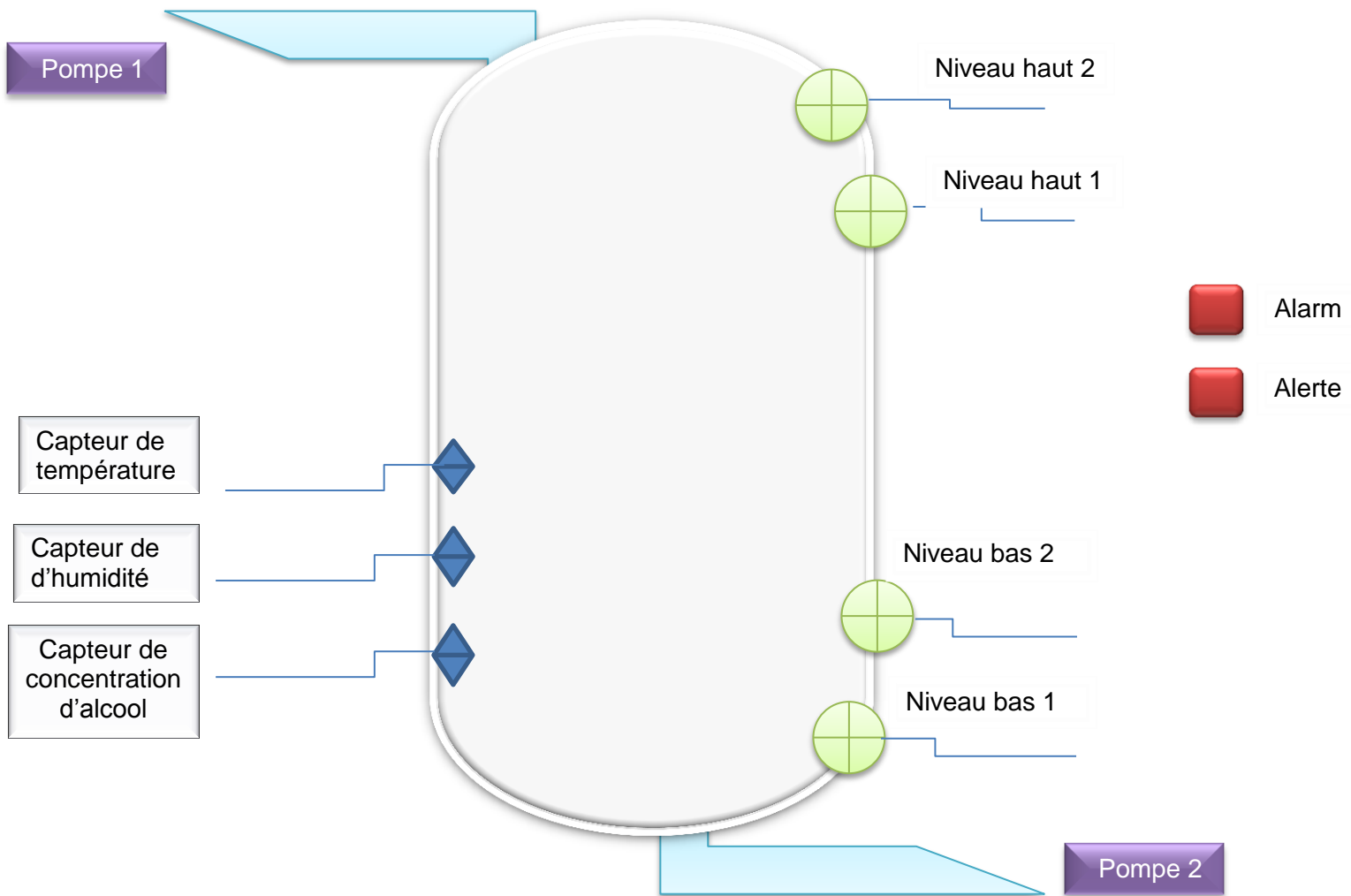


Figure 51 : schéma synoptique de système de fermentation de la levure

D'après la figure précédente et d'après le cahier de charge on définit les entrées et les sorties :

- ❖ Quatre capteurs de niveau (**entrées numériques**).
- ❖ Trois capteurs : température, d'humidité et concentration d'alcool (**entrées analogiques**).
- ❖ Alarme, Alerte, Pompe 2 et Pompe 1 (**sorties**).

Fonctionnement :

1. Entrées analogiques

- ✓ Si la température dedans le cuve dépasse 40 °C, ou elle est inferieure a 10°C lancer une alerte pour aviser les responsables.
- ✓ Si la concentration d'alcool dépasse 37% déclencher l'alarme et si l'humidité est supérieure à 50% déclencher l'alerte.

2. Entrées numériques

- ✓ Si le niveau de la levure dépasse le niveau haut 1 ou il est moindre au niveau bas 2 déclenché l'alerte.
- ✓ Si il est inférieur au niveau bas 1 déclenché l'alarme.
- ✓ Si le niveau de la levure dépasse le niveau haut 2 déclenché l'alarme et démarrer la pompe 2 pour vider le cuve.

3. Programmation

- ✓ Ou cas où la température dépasse le maximum et l'humidité aussi dépasse son maximum déclencher l'Alarme.

4. Affichage

Affichage des données des capteurs de température, pression et de la concentration d'alcool sous forme d'un graphe et des jauges pour permettre le responsable de suivi facilement le déroulement de fonctionnement du système. Ainsi afficher les états marche/arrêt des capteurs de niveau pour bien accompagnée la variation de niveau de cuve. L'acquisition et l'affichage de toutes les données sont en temps réel.

5. Enregistrement

Enregistrement des données des températures, pression et la concentration d'alcool dans un fichier Excel accompagné de la date exacte de chaque échantillon.

La simulation de ce système faite par le logiciel de simulation isis proteus, pour cela on affecte les entrées de data logger aux capteurs. Pour cela nous avons choisi la configuration suivante :

Capteurs	Entrées d'enregistreur
Capteur de Température	Entrée analogique 1 (Ain1)
Capteur d'humidité	Entrée analogique 2 (Ain2)
Capteur de concentration d'alcool	Entrée analogique 3 (Ain3)
Capteur de niveau bas 1	Entrée numérique 1 (Din1)
Capteur de niveau bas 2	Entrée numérique 2 (Din2)

Capteur de niveau haut 1	Entrée numérique 3 (Din3)
Capteur de niveau haut 2	Entrée numérique 4 (Din4)

Tableau 8: la configuration des entrées d'enregistreur de données

Sorties	Sorties d'enregistreur
Alerte	Relais 1
Alarme	Relais 2
Pompe 2	Relais 3

Tableau 9: la configuration des Sorties d'enregistreur de données

L'étape suivante est de brancher l'enregistreur de données avec l'ordinateur et on ouvre l'application pour configurer l'interface. Le premier onglet de l'interface pour définir le nom de Port de votre PC branché avec le data logger, chaque fois l'interface est connectée avec le matériel les états des capteurs (entrées numériques) et les valeurs des entrées analogiques s'affichent.

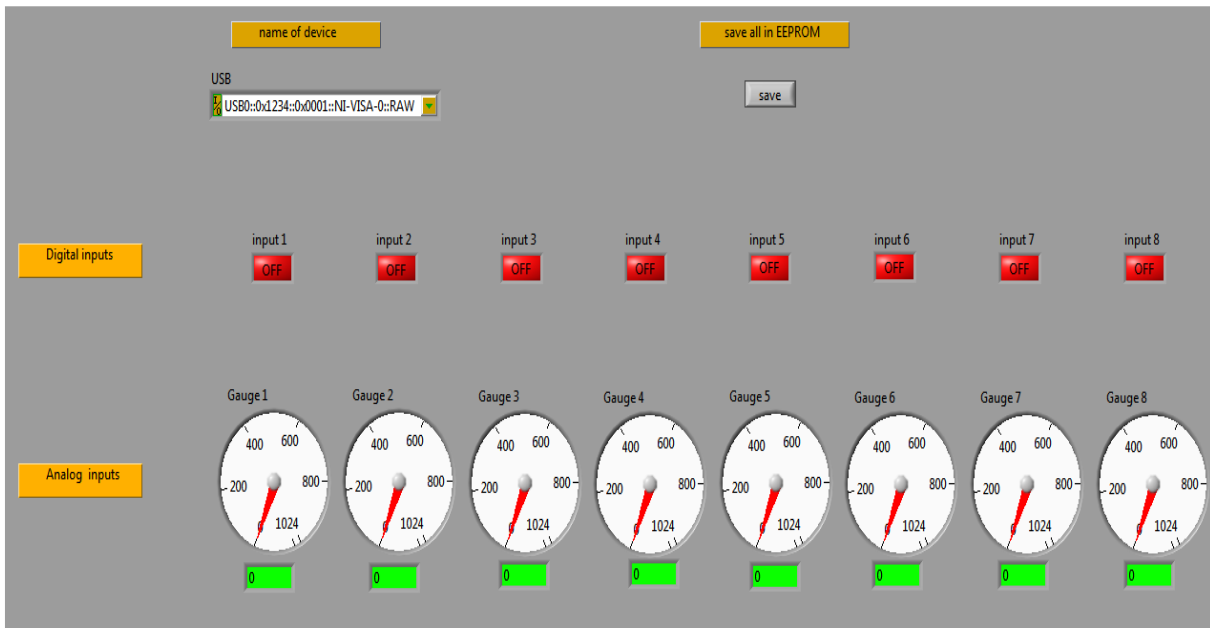


Figure 52 : onglet Display

Le deuxième onglet sert à configurer les entrées numériques qui sont les capteurs de niveau. Si le niveau de la levure est inférieur au niveau bas 1 déclencher l'alarme. L'alarme est la sortie 2(Relais 2).ainsi Din2 et Din3 (niveau bas 1 et niveau haut 1 déclenchent l'alerte). Din4 (niveau haut 2) déclenche l'alarme et démarre la pompe 2 (Relais 3).

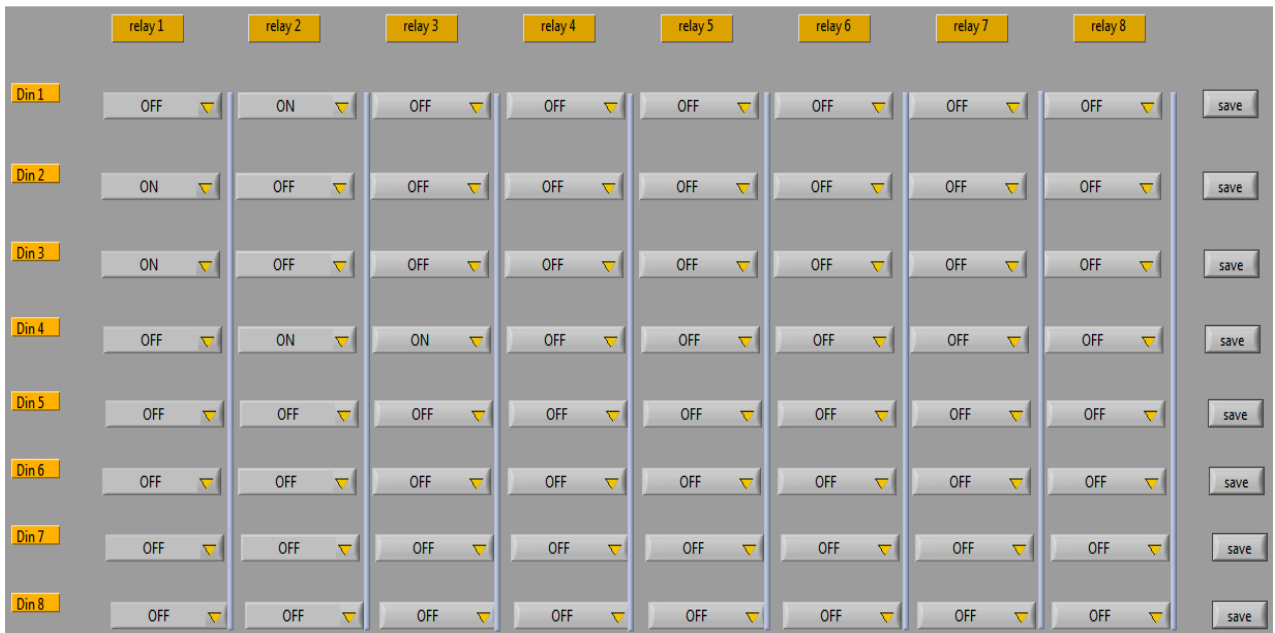


Figure 53 : configuration des entrées numériques d'enregistreur de données

L'étape suivante consacrée pour les entrées analogiques. Si Ain1 (température) dépasse 40°C ou inférieur à 10°C déclencher l'alerte (Relais 1).

Si l'humidité est supérieure à 50% déclenche l'alerte, et si la concentration d'alcool dépasse 37% signalée l'alarme.

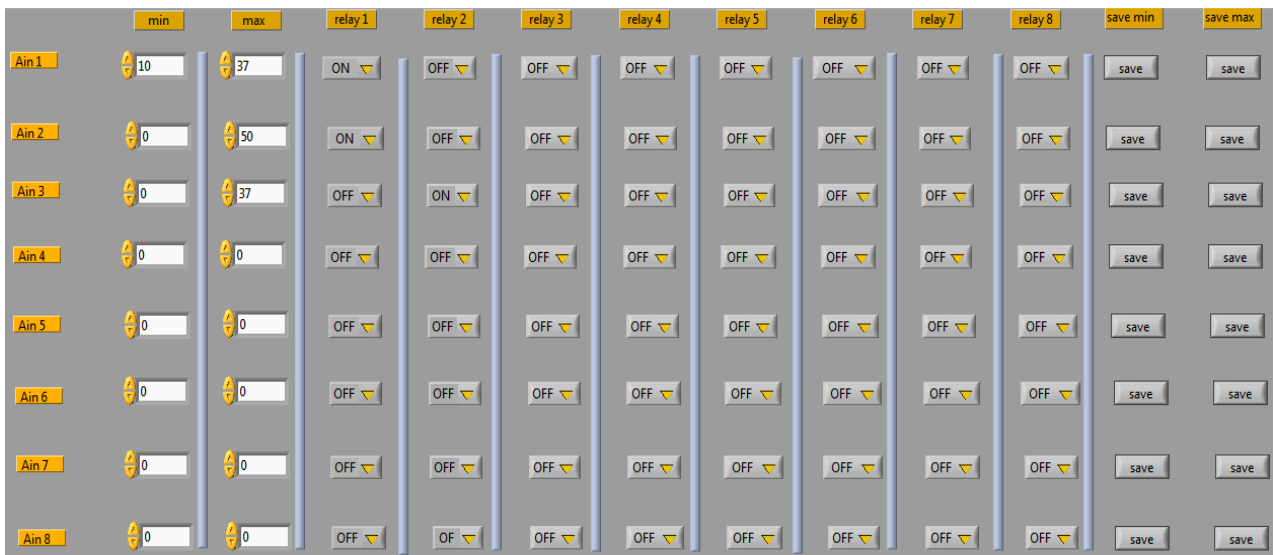


Figure 54 : configuration des entrées analogiques

La dernière partie et la partie programmation, si la température dépasse le maximum et aussi l'humidité déclencher l'alarme.

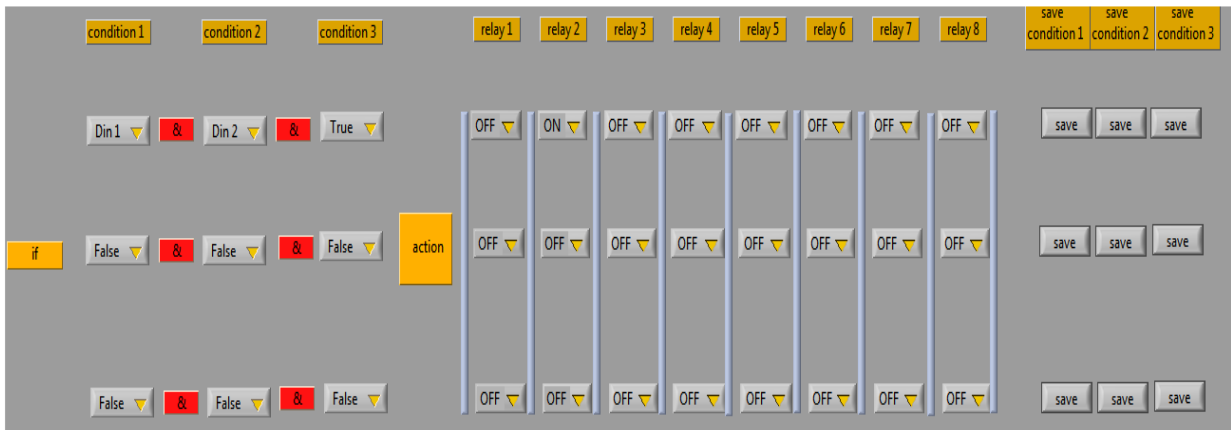


Figure 55 : configuration d'onglet programmation

L'affichage des résultats des capteurs sous forme d'un seul graphe permet la comparaison entre les autres entrées et facilite la visualisation.

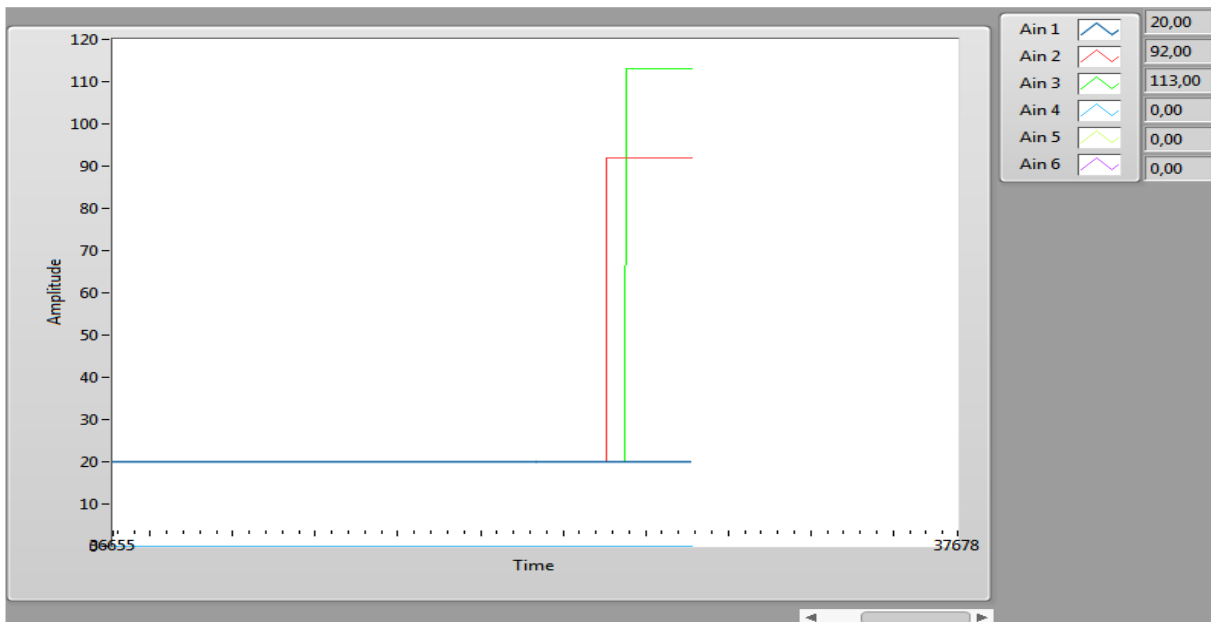


Figure 56 : l'affichage des résultats dans le graph

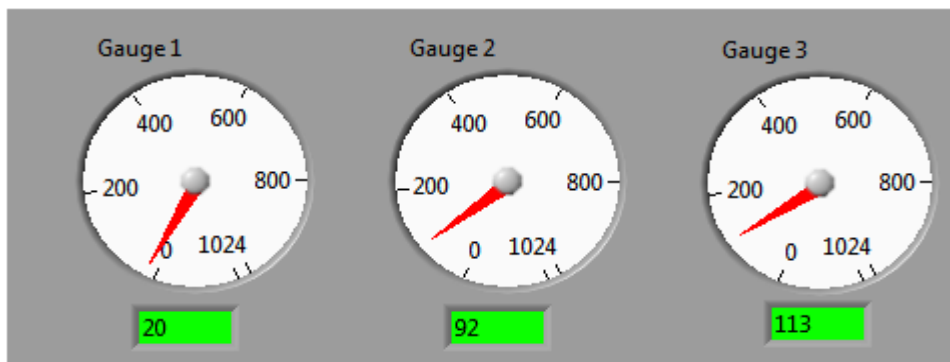


Figure 57 : affichage des résultats dans des jauges

Toutes les configurations faites fonctionnent bien, la figure suivante montre une simulation des entrées et ses contrôles sur les sorties. Si Din4 (niveau haut 2) est atteint déclencher l'alarme et démarrer la pompe 2. Si encore la température dépasse 50 degré déclencher l'alerte.

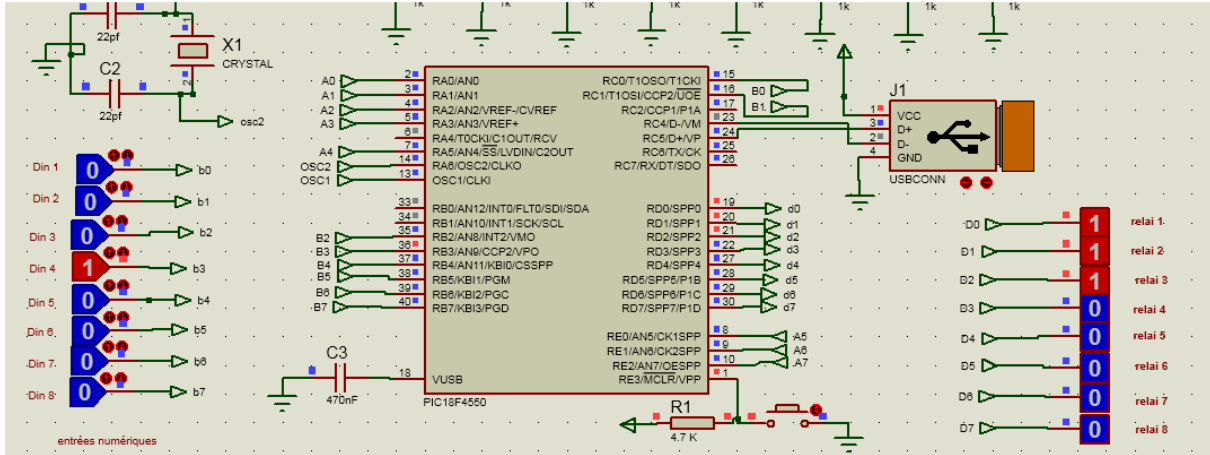


Figure 58 : simulation d'un cas de fonctionnement

Les données des capteurs température, humidité et pression sont bien enregistrées dans le fichier Excel. la figure suivante montre le fichier et la forme d'enregistrement.

	A	B	C	D	E	F	G	H	I
Time		Température °C	Humidité %	Alcool %	Ain4	Ain5	Ain6	Ain7	Ain8
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0
24/05/2016 11:44:00,000 AM		20	41	31	0	0	0	0	0

Figure 59 : le fichier d'enregistrement des données

Toutes les configurations misent sur les entrées numériques, analogiques et la programmation fonctionnent bien. Notre projet à répod bien à ce cahier de charge, il peut couvrir une large gamme des fonctionnalités dans différent domaines : industrie, environnement, surveillance, domotique...

VII. Conclusion

Dans ce chapitre j'ai présenté la partie software et les étapes suivies pour réaliser notre projet. J'ai expliqué les différents points pour établir d'abord la communication entre le hardware qui est le PIC18F4550 et le logiciel Labview qui représente la partie software. Ensuite, j'ai entamé la première partie qui consiste à développer l'enregistreur de données avec ordinateur ce dernier permet de faire l'acquisition, le contrôle, l'affichage, l'enregistrement et le traitement des données. Tout le traitement s'exécute au niveau de logiciel labview, dans ce cas le fonctionnement de microcontrôleur engendre seulement l'échange des états, des valeurs des entrées et les sorties avec l'interface.

La deuxième partie était conçue pour le développement d'enregistreur de données autonome. Il combine les deux modes de fonctionnement, quand il est connecté à l'interface le fonctionnement sera le même que celui d'enregistreur de données avec ordinateur. si le matériel est débranché de l'ordinateur il garde le fonctionnement normal puisque toutes les variables et toutes les actions définies par l'utilisateur sont enregistrées dans la mémoire EEPROM de microcontrôleur. J'ai présenté aussi l'interface réalisée et les différents onglets et son fonctionnement. En outre j'ai défini les trames et les adresses envoyées par l'interface et la réception par le microcontrôleur.

La dernière partie de ce chapitre est dédiée à une simulation globale du projet, pour montrer que nous avons répondu à un cahier de charge d'une société de fabrication de la levure.

Le chapitre suivant concerne la conception de la carte de la simulation, et la création de l'application .EXE ainsi l'application « **installer** » qui permet d'inclure tous les drivers lors de l'installation, elle est compatible avec toutes les versions des systèmes d'exploitation.

Chapitre 4 : Routage et La conception de circuit imprimé

1. Introduction
2. Réalisation de circuit imprimé
3. Génération de l'application exécutable
4. Génération de l'application d'installation
5. Conclusion

1. Introduction

Après le développement software d'enregistreur de données et la simulation de fonctionnement en utilisant le logiciel de simulation Isis proteus. La partie suivante sera consacrée à la réalisation de circuit imprimé on réalisant le PCB de notre carte, ainsi la création de l'application exécutable et le fichier complet d'installation.

La dernière partie de ce chapitre présente une simulation réelle de notre projet et la description de fonctionnement.

2. Réalisation de circuit imprimé

Le routage et la réalisation de la carte est la dernière étape après le développement et la simulation de fonctionnement d'enregistreur de données. Nous avons choisis le logiciel **real PCB** pour la réalisation de circuit imprimé.

Cette opération consiste à créer les liaisons entre les composants électroniques afin que chacun soit irrigué électriquement de manière optimale. Pour débiter, il faut savoir que le routage de l'alimentation de la carte électronique est primordial. De plus, plusieurs règles pour le routage sont très importantes, comme éviter autant que faire les angles droits (qui tendent à freiner le débit (théorème de Bernoulli)).

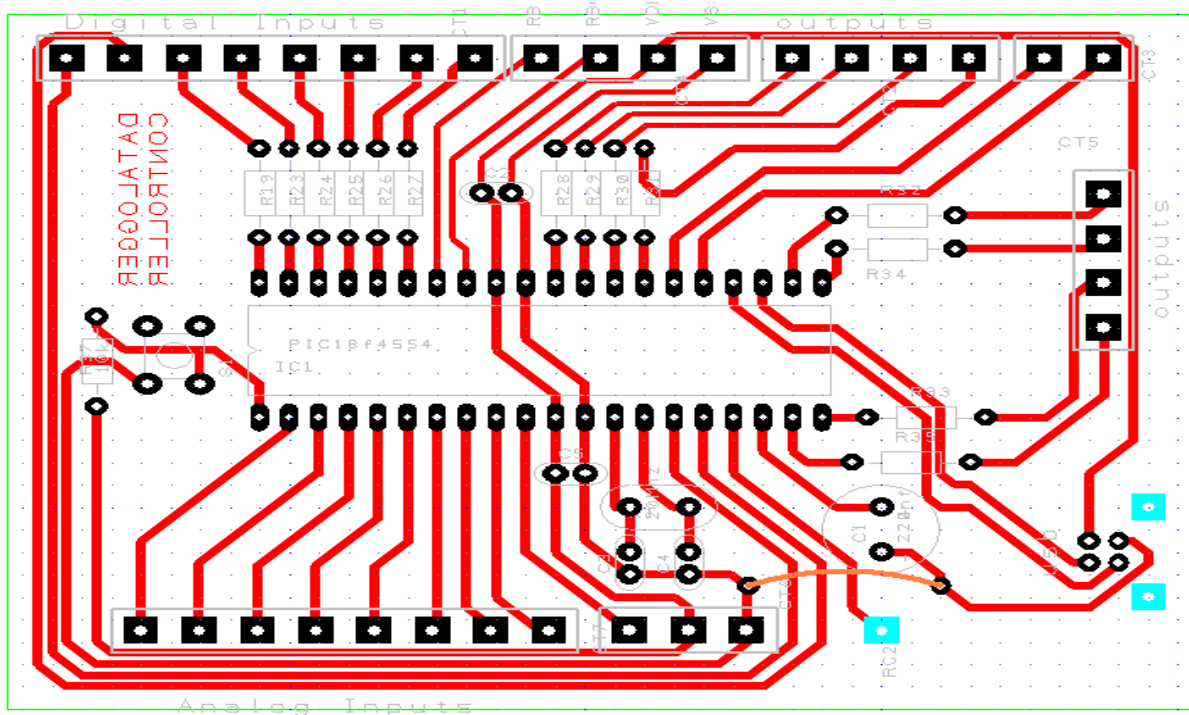


Figure 60 : PCB de la carte d'enregistreur

Le PIC 18F4550 est centré au milieu, à partir des borniers qui entourent la carte, on connecte les sorties de microcontrôleur avec les relais et les entrées avec les capteurs. Le bouton en tête de microcontrôleur présente le RESET, il permet de désactiver la communication avec l'interface.

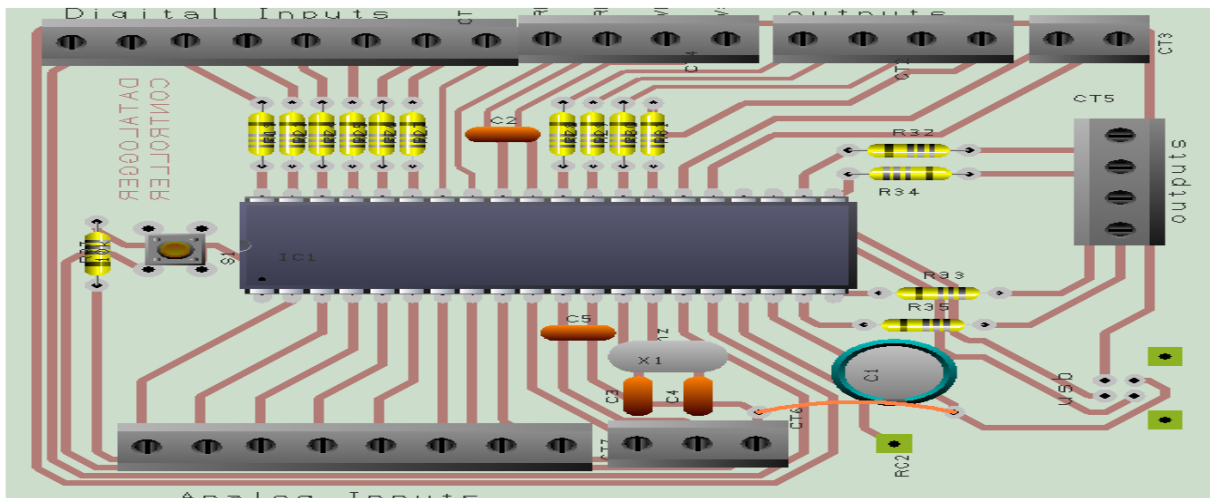


Figure 61 : vue réel

Après la réalisation de PCB de circuit on imprime le circuit sur la carte de cuivre, La fabrication nécessite ensuite de fixer les composants sur la plaque grâce au perçage opéré par une perceuse. Une fois les composants en place, il faut les solidariser à la plaque par soudure. La figure ci-jointe montre la carte après la conception et le soudage de tous les composants.

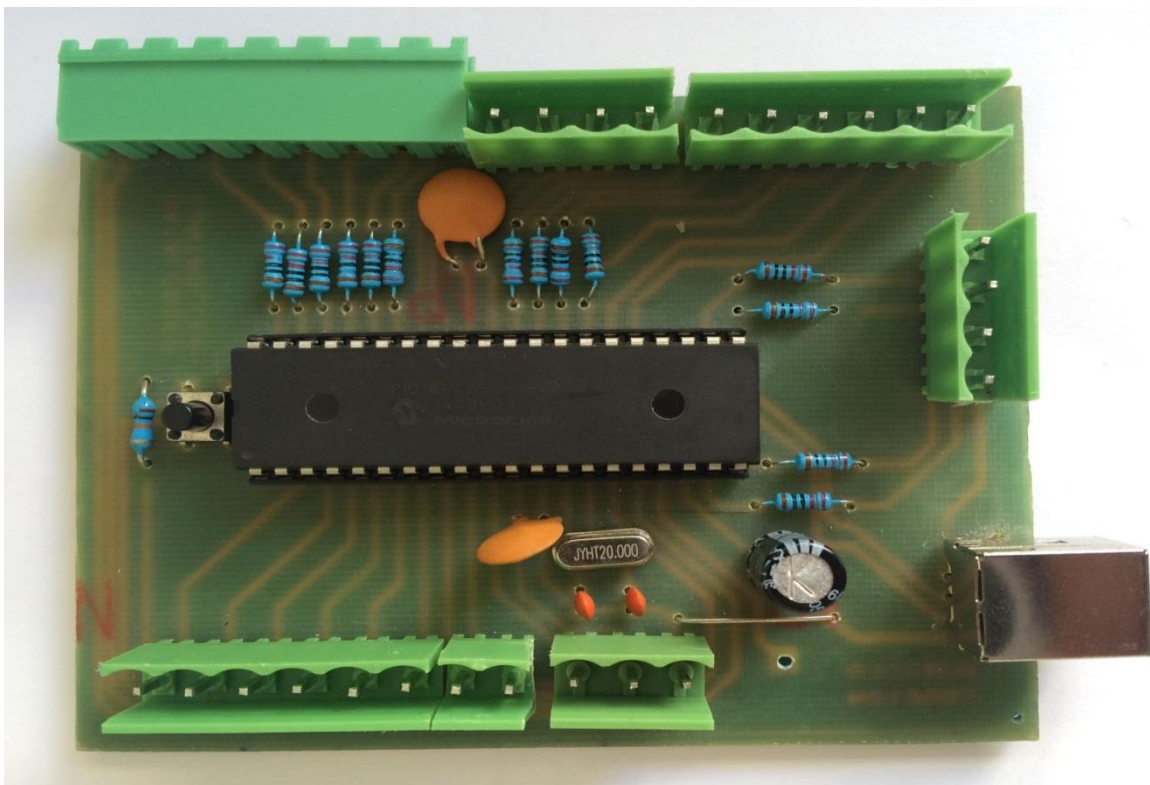


Figure 62 : la carte après la conception

3. Génération de l'application exécutable

La création de l'application exécutable permet de générer une application (.EXE) il est facile d'ouvrir l'application et accéder directement à l'interface que d'ouvrir le projet labview et ouvrir le fichier VI correspondant.

Les applications sont utiles lorsque vous voulez que les utilisateurs puissent exécuter des VIs sans installer le système de développement LabVIEW. Néanmoins les exécutables créés exigent la présence du moteur d'exécution LabVIEW qui sera distribué par l'installateur ou téléchargeable sur le site de National Instruments.

Sur Windows, les applications ont l'extension « .exe » et sur Mac OS « .app ». En contre partie, l'utilisateur de l'exécutable n'aura pas accès au code (diagramme) du programme. Il aura devant lui un instrument virtuel exécutant les tâches configurées auparavant, lors de l'écriture du VI. Un exécutable contient l'ensemble des sous-VIs constituant le programme initial.

Parmi les principales étapes de la création de l'application exécutable sont :

- Si vous avez créé votre dossier sans projet, ouvrez votre VI principal. Cliquez sur **Tools>> Build Application (EXE) from VI...**

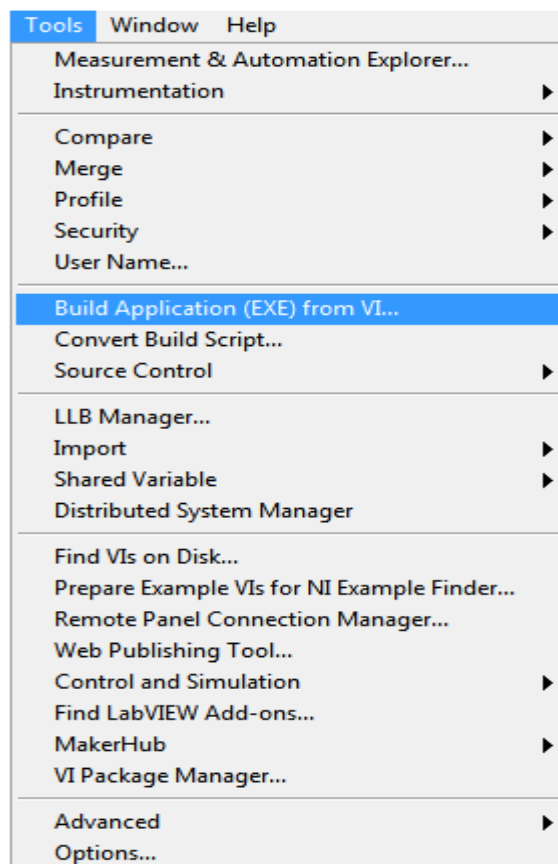


Figure 63: création d'application (EXE)

On choisit après un chemin pour enregistrer l'application. Puis on appuie sur continuer, Une boîte de dialogue apparaîtra :

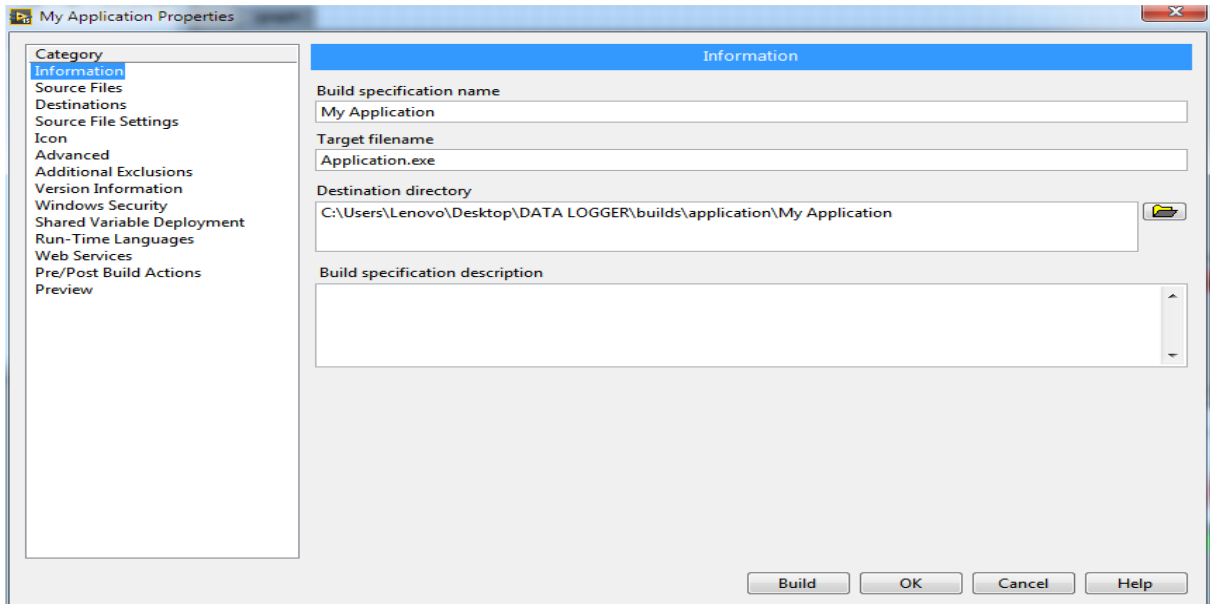


Figure 64 : les propriétés de l'application

Après saisir la configuration et les descriptions de l'application on appuie sur **Build**, l'application est donc créée dans le dossier approprié.

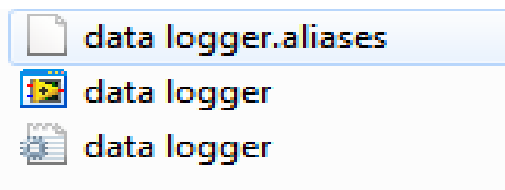


Figure 65 : application .exe

L'Application Builder permet de créer des applications autonomes pour fournir des versions exécutables de VIs à d'autres utilisateurs. Ces applications exécutables exigent la présence de moteur d'exécution LabVIEW. Elles permettent aux utilisateurs d'accéder à un instrument virtuel exécutant les tâches configurées auparavant sans accès au code du programme.

4. Création de l'installeur

La création de l'installeur permet d'importer tous les drivers nécessaires pour l'exécution de programme, ainsi il n'exige pas la présence d'un moteur d'exécution LabVIEW. Il suffit d'installer le fichier et exécuter l'application. L'installeur de Labview est compatible avec toutes les versions de Windows.

La création de l'application exécutable est obligatoire avant de commencer la génération de l'installeur. Clic droit sur **Build Specification** puis **NEW installer**

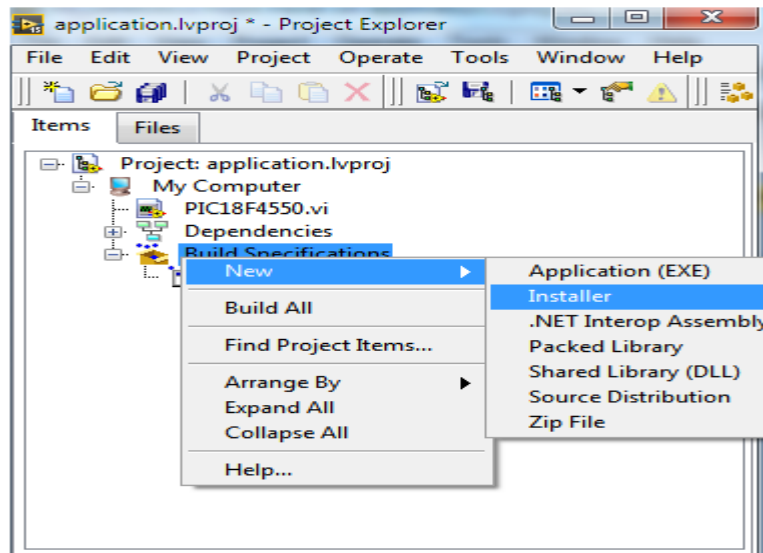


Figure 66 : création de l'installateur

Une fenêtre s'affichera :

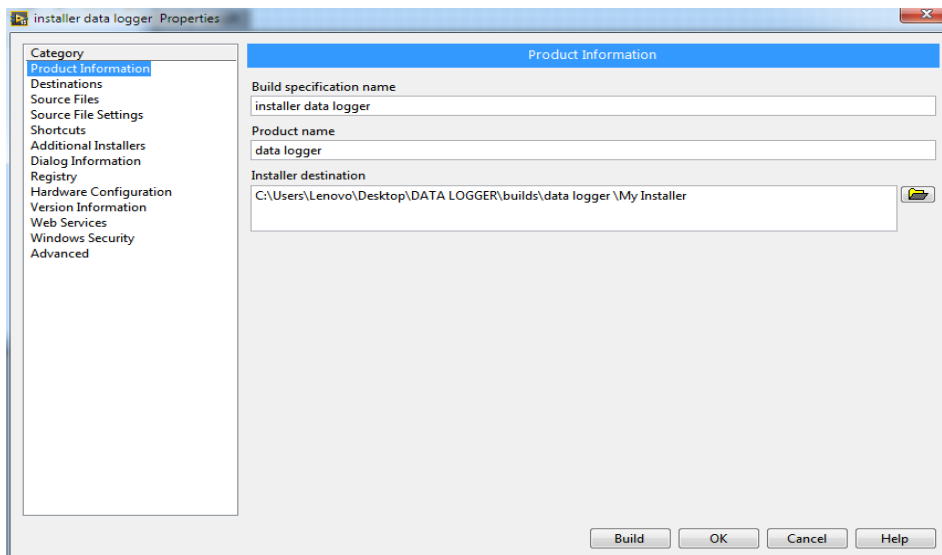


Figure 67 : les propriétés d'installateur

Après le remplissage de toutes les propriétés et les configurations de l'installateur, et d'importer les drivers nécessaire, on appuie sur **Build**. L'installateur est bien créé dans le dossier approprié.



Figure 68: l'installateur

Lancer l'installation et ouvrir l'application, qui permette d'accéder directement à l'interface de programme.

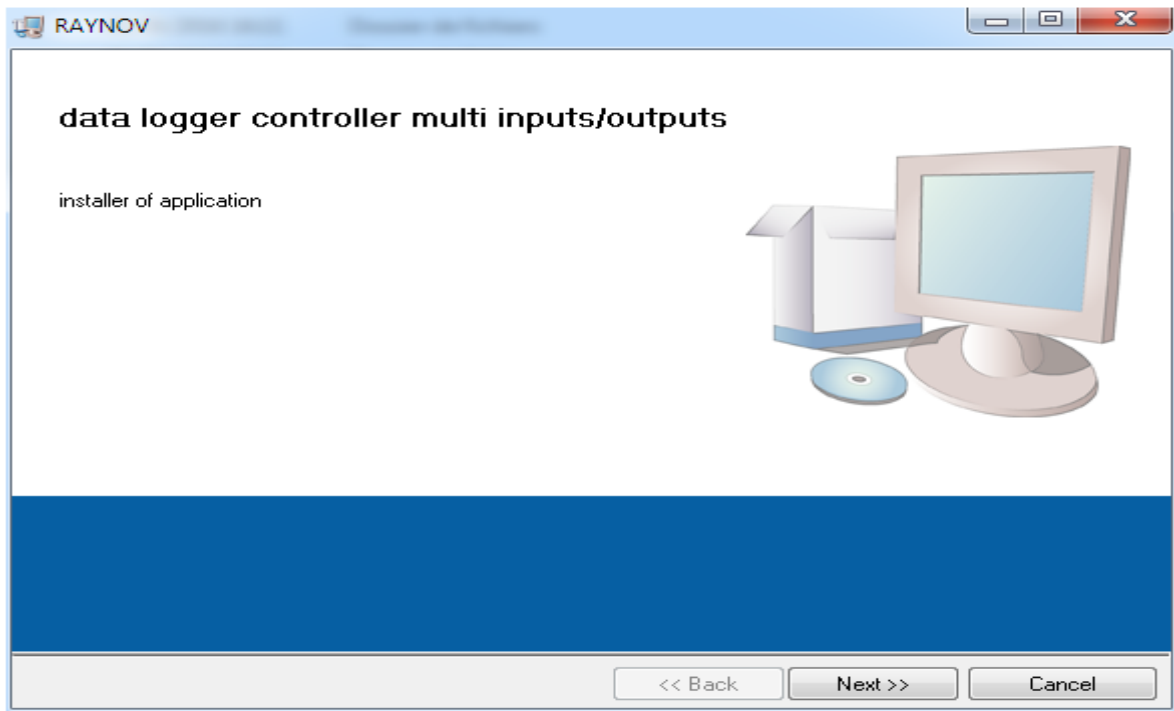


Figure 69: installation de l'application

L'installeur permet d'importer tous les drivers et les bibliothèques utilisées dans le programme. L'utilisateur peut utiliser l'application sans avoir installé Labview ou un moteur d'exécution labview, et il peut accéder seulement à l'interface sans avoir accéder au code de programme.

5. Conclusion

Dans ce chapitre j'ai présenté les étapes pour le routage et la conception de circuit imprimé, la première consiste à réaliser le PCB de la carte, Cette opération consiste a créé des liaisons entre les composants électroniques afin que chacun soit irrigué électriquement de manière optimale. La deuxième étape c'est de fixer et souder les composants sur la plaque.

La deuxième partie était réservée pour la création de l'application exécutable qui permette d'exécuter le programme facilement, et ne permette pas à l'utilisateur d'accéder au code de programme.

Au contraire à l'application exécutable qui exige la présence de logiciel labview ou le moteur d'exécution Labview, la création de l'installeur est la solution pour importer tous les drivers et les bibliothèques utilisés lors de l'exécution dans le dossier d'installation sans exiger a l'utilisateur d'installé Labview.

Conclusion Générale

Dans le cadre de ce projet de fin d'études nous avons réalisé un projet qui consiste en un système d'enregistrement de données, le projet est le fruit de quatre mois de travail continu.

Au début nous avons commencé par l'étude des différents enregistreurs de données existants appartenant à des sociétés de pays différentes afin d'avoir une idée claire sur les difficultés et les problèmes que nous pouvions rencontrer lors de la réalisation notre projet.

La deuxième partie était consacrée à la présentation de la technologie de travail choisie pour la conception du projet, le choix de microcontrôleur est fixé sur l'utilisation de PIC18F4550, L'avantage majeur de cette dernière par rapport les autres technologies c'est qu'elle intègre le protocole USB 2.0. Ainsi j'ai présenté l'environnement de développement, il s'agit d'utiliser le logiciel de National Instrument LabVIEW pour le développement de l'interface, le microC pour programmer le PIC18F4550.

En outre, dans la troisième partie nous avons développé deux versions d'enregistreur de données, le fonctionnement de la première dépend fréquemment de l'ordinateur. Si on déconnecte la communication entre l'interface et le microcontrôleur le fonctionnement s'arrête. La deuxième version est un enregistreur de données autonome qui présente une combinaison des deux modes de fonctionnement. Quand il est branché à l'ordinateur, et garde le fonctionnement normal quand il est débranché de l'ordinateur puisque tous les variables défini à l'interface sont enregistrés dans la mémoire EEPROM de microcontrôleur. Le traitement des données s'exécute au niveau de microcontrôleur au contraire de la première version tout le traitement était au niveau de software (labview).

La dernière partie de projet était celle de la réalisation et la conception de circuit imprimé, après la simulation de fonctionnement de projet nous avons créé l'application exécutable et le fichier complet de l'installation.

Finalement on peut constater que notre projet présente l'avantage, par rapport aux autres enregistreurs de données, d'effectuer le traitement des données des entrées, et le contrôle des sorties pour améliorer le niveau de surveillance et de régulation. Contrairement aux autres enregistreurs qui effectuent juste l'acquisition et l'enregistrement de données.

Ce projet marque l'aboutissement d'un parcours universitaire enrichissant tant sur le plan théorique, pratique qu'humain. Il présente une opportunité instructive servant à nous garantir, une vision plus réaliste de l'univers du travail.

Bibliographie

[1] : Omega Engineering. Data logger. **[En ligne]** Disponible sur :

<http://www.omega.fr/prodinfo/enregistreurs-de-donnees.html> (consulté le 02/03/2016).

[2] : datataker. Centrale d'acquisition intelligente. 2014, p 21-22 disponible sur :

http://www.dimelco.com/francais/mysearchengine/centrales_mesures/docs/doc_CEM20_1216.pdf (consulté le 02/03/2016).

[3] : Chris delvis. National instruments Corp . Advantages of PC-based Data logging. **[En**

ligne] (2010, mise à jour 2016) Disponible sur <http://machinedesign.com/archive/advantages-pc-based-data-logging> (consulté le 02/03/2016).

[4] : Pce Instruments. [Enregistreur de données PCE-HT 71N](#). **[En ligne]** Disponible sur :

https://www.pce-instruments.com/french/instruments-de-mesure/mesureur/enregistreur-donnees-pce-instruments-enregistreur-de-donn_es-pce-ht-71n-det_2208253.htm (consulté le 03/03/2016).

[5] : Pce Instruments. Enregistreur de données PCE-KD5. **[En ligne]** Disponible sur :

https://www.pce-instruments.com/french/instruments-de-mesure/mesureur/enregistreur-donnees-bd-simex-enregistreur-de-donn_es-pce-kd5-det_2208267.htm (consulté le 03/03/2016).

[6] : datataker. Centrale d'acquisition intelligente. 2014, p 8 disponible sur :

http://www.dimelco.com/francais/mysearchengine/centrales_mesures/docs/doc_CEM20_1216.pdf (consulté le 03/03/2016).

[7] : Alebi Elmehdi, Jawadi Sami. *Conception et réalisation d'un enregistreur de données*.

Projet de fin d'étude **[En ligne]**. Société micro 21 Gafsa, Université de Tunisie. 2010/1011, 66p. Disponible sur :

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjGrcWyh4rNAhWEJcAKHa5GB_sQFggdMAA&url=http%3A%2F%2Fpf-mh.uvt.rnu.tn%2F491%2F1%2FConception_et_r%25C3%25A9alisation_d%25E2%2580%2599un_enregistreur_de_donn%25C3%25A9es_.pdf&usg=AFQjCNG_t2uZ9P7RLf5025vFKHhMnbHfMA&sig2=9IajbmiqX62O3oPpTfxZrw (consulté le 20/03/2016).

- [8] :ATmel. Mega AVR controllers. **[En ligne]** Disponible sur <http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx> (consulté le 20/03/2016).
- [9] : Microchip. Data sheet PIC18F4550. 2009, 438p. disponible sur : https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0ahUKEwjxpp3riIrNAhUsIsAKHYJICaYQFgg0MAY&url=http%3A%2F%2Fwww1.microchip.com%2Fdownloads%2Fen%2FDeviceDoc%2F39632e.pdf&usg=AFQjCNFNi_QIXM89BfgZ0NeS_PdZ3xLaSQ&sig2=T2IRD4Ghl2HFyi6JqAawnw (consulté le 20/03/2016).
- [10] : abk. Électronique : Programmation en microC **[En ligne]**, FACULTE DES SCIENCES DE BOGARA ALGERIE. 123p Disponible sur : https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiVz4WdjIrNAhWMKcAKHU1wCwEQFggdMAA&url=http%3A%2F%2Fwww.univ-bouira.dz%2Ffr%2Findex.php%2Fcours-a-telecharger%2Fcategory%2F680-Mr%2520TOURCHINE%2520V%2520-%2520G%25C3%25A9nie%2520Electrique%3Fdownload%3D3915%3A03-%2520Programmation%2520en%2520mikroC&usg=AFQjCNHwvvcXWbxZFmMz9o0IVT-u_Ppm0A&sig2=w5R5Ruj_TrFckFKpjfOYPA (consulté le 21/03/2016).
- [11] : Fabien. Initialisation au labview. **[En ligne]** formation des enseignants.2014, 14p Disponible sur : https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiIij4rNAhWJAsAKHS_BA8sQFggdMAA&url=http%3A%2F%2Feducol.education.fr%2Fsti%2Fsites%2Feducol.education.fr.sti%2Ffiles%2Fressources%2Ftechniques%2F3437%2F3437-1-initiation-labview-le-vi.pdf&usg=AFQjCNEC93A17pfzubLy9gTuUzlsTmgqdw&sig2=ADTGlhgToJ34_87WBKnclw (consulté le 21/03/2016).
- [12] : Sylvain Chalmin, Bacem ElIdilbi. Informatique des Systèmes Embarqués : implémentation de protocole USB.2009, 25p (consulté le 15/04/2016).
- [13] : 7Robot. Electronique : À la découverte de bus CAN **[En ligne]** 2011 Disponible sur : <http://www.bde.enseiht.fr/clubs/robot/node/44> (consulté le 20/04/2016).