République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Bejaïa

Faculté de Technologie

Département de Génie Electrique

Mémoire de Fin de Cycle

En vue de l'obtention du diplôme Master en Electrotechnique
Option : Automatisme Industriel.

Thème

Automatisation d'un système de pompage à l'aide d'une carte PICPLC16 v6

Présenté par :

Mr.ZENADI Djamel
Mr.SAHNOUNE Karim

Encadré par :

M^{elle}.MEZZAH Samia
Mr.LAOUCHICHE Louhab

Promotion: 2012-2013

Nous remercions DIEU tout puissant de nous avoir donné la force, la santé, le courage et la patience de pouvoir accomplir ce travail.

Un grand merci à toutes nos familles surtout nos parents pour leur encouragement et leur suivi avec patience du déroulement de notre projet.

Nous tenons à remercier vivement notre promotrice M^{elle} MEZZAH d'avoir accepté de nous guider tout au long du travail.

Nos remerciements vont également à tout le personnel de l'entreprise SPA Générale Emballage d'Akbou pour leur hospitalité surtout à Mr LOUCHICHE Louhab.

Nos sincères remerciements s'adressent aussi à tous ceux qui ont contribué de prés ou de loin à la réalisation de ce travail.

Enfin, nous tenons aussi à remercier également tous les membres de jury pour avoir accepter d'évaluer notre travail.

Merci à tous

Dédicaces

Je dédie ce modeste travail à mes très chers Parents, mes frères et sœurs.
A tous mes amis Farid, Toufik, Lamine, Mounir, Mourad, Adel, Brahim, Zahir, Toufik et tous leurs copains de chambre.

A mes copains de chambre Massi et L'vieu, Mouhand et à toute la clique master1 Informatique.
A mon binôme Djamel, ses copains de chambre et toute sa famille.
Aux enseignants Mr Laadjouzi samir et Mr Hammou Farid.
A toute la section Génie Electrique, surtout la promotion Automatisme Industriel.

Une grande partie de ce travail est dédié à une personne qui veut rester anonyme, qui a mit toute ses ressource à ma disposition et qui a contribué à la finalisation de ce travail.

A ceux qui cherchent leurs noms ici.

REEMKA

Je dédie ce modeste travail

A mon ***Père*** *et à ma très chère* ***Mère****, en témoignage et en gratitude de leurs dévouement, de leurs soutien permanent durant toutes mes années d'études, leurs sacrifices illimités, leurs réconfort moral, eux qui ont consenti tant d'effort pour mon éducation, mon instruction et pour me voir atteindre ce but, pour tout cela et pour ce qui ne peut être dit, mes affections sans limite.*

- *A mes frères Hamza et Mohaned.*
- *A mes sœurs Dalila et Atika.*
- *A mon oncle mustapha et sa famille.*
- *A mes amis Adel, Brahim, Sofiane,Rafik et a tout mes amis de la promotion automatisme industriel 2012/2013.*
- *A mon binôme* ***KARIM*** *et toute sa famille.*
- *A mes copains de chambre, Mokran, Djilali, Ghanou et Habib.*
- *Ceux qui m'ont soutenu pendant toute la durée de mes études.*

Djamel

Sommaire

Sommaire

Introduction générale 1
Présentation de l'entreprise 3

Chapitre I : Automatisation et microcontrôleur

I.1 Introduction 5
I.2 Les systèmes automatisés 5
I.3 Objectif de l'automatisation 5
I.4 Structure d'un système automatisé 5
I.4.1 La partie opérative 5
I.4.2 La partie commande 6
I.4.3 La partie dialogue 6
I.5 Les choix technologiques 7
I.6 Les microcontrôleurs 8
I.6.1 Les avantages du microcontrôleur 8
I.6.2 Structure d'un microcontrôleur 9
I.7 Le microcontrôleur PIC 9
I.7.1 Définition d'un PIC 9
I.7.2 Les différentes familles des PICs 10
I.7.3 Identification d'un PIC 10
I.7.4 Classification des microcontrôleurs 10
A. Selon l'organisation des mémoires 10
B. Selon l'unité centrale de traitement 11
I.7.5 Familles de PIC disponibles sur le marché 12
I.7.6 Choix d'un PIC 13
I.8 La programmation des PICs 13
I.9 MPLAB 13
I.9.1 Les outils de MPLAB 14
I.9.2 Compilation 14
I.10 Conclusion 14

Chapitre II: Le microcontrôleur PIC18f4520

II.1 Introduction 15
II.2 Présentation du PIC18f4520 15
II.3 Architecture interne du PIC18f4520 16

II.4 Caractéristiques du PIC18f4520 17
II.5 Classification des éléments 17
II.5.1 Le noyau 17
II.5.2 Les périphériques 17
II.5.3 Les fonctions spéciales 18
II.6 Organisation de la mémoire 18
II.6.1 Mémoire de programme (Flash) 18
II.6.2 Mémoire RAM 19
II.6.3 Mémoire EEPROM 19
II.7 Les ports d'entrées/sorties (I/O) 19
II.8 Le reset 20
II.9 Les Timers 20
II.9.1 Timer0 21
II.9.2 Timer1 22
II.9.3 Timer2 22
II.9.4 Timer3 23
II.10 Interruption 24
II.11 Watch dog (chien de garde) 26
II.12 Mode sommeil (mode SLEEP) 26
II.13 Les périphériques 26
II.13.1 Les module CCPM 26
II.13.1.1 Mode Capture 26
II.13.1.2 Mode Compare 26
II.13.1.3 Mode PMW 26
II.13.1.4 Les comparateurs 27
II.13.2 Module de conversion analogique/numérique 27
II.13.3 Liaison série RS232 28
II.13.3.1 Registre de contrôle et d'état 28
II.13.3.2 Fonctionnement de la liaison série en mode transmission 30
I.14 Conclusion 30

Chapitre III: La carte PICPLC16 v6

III.1 Introduction 31
III.2 Description de la carte PICPLC16 v6 31
III.3 Le microcontrôleur 32
III.4 Le programmateur PIC flash USB 2.0 avec mikroICD 33
III.5 Alimentation d'énergie 33
III.6 Interface de communication RS-232 34

III.7 Communication RS-485 35
III.8 Le module Ethernet 35
III.9 Le connecteur GSM 36
III.10 Horloge temps réel (RTC) 37
III.11 Les optocoupleurs 37
III.12 Les relais 38
III.13 Les ports d'entrées/sorties 38
III.14 Conclusion 39

Chapitre IV : Programmation et Test

IV.1 Introduction 40
IV.2 Description du système 40
IV.2.1 Les pompes 41
IV.2.2 Les capteurs 41
IV.3 Cahier des charges 41
IV.4 Programmation du PIC 18F4520 41
IV.5 Structure du programme à implanter dans le microcontrôleur 42
IV.5.1 Programme principal 42
IV.5.2 Routine principale 44
IV.5.3 Routine d'interruption 45
IV.6 Programmation et simulation 46
IV.6.1 La programmation dans le MPLAB 46
IV.6.2 La simulation dans l'ISIS PROTEUS 47
IV.7 Le test de la carte 51
IV.7.1 Chargement du programme dans le PIC 52
IV.7.2 Test de fonctionnement de la carte 53
IV.8 Interface de communication homme/machine 54
IV.9 Conclusion 55

Conclusion générale 56

Liste des figures

Liste des Figures

Figure I.1 Structure détaillée d'un automatisme. 6
Figure I.2 Structure interne d'un microcontrôleur. 9
Figure I .3 Structure de Von Neumann. 11
Figure I.4 Architecture de Harvard. 11

Figure II.1 Architecture externe du PIC18f4520.....15
Figure II.2 Architecture interne du PIC18f4520. 16
Figure II.3 Organisation de la mémoire et de la pile. 18
Figure II.4 Bloc diagramme du Timer0 (mode 8 bits). 21
Figure II.5 Bloc diagramme du Timer1 22
Figure II.6 Bloc diagramme du Timer2. 23
Figure II. 7 Bloc diagramme du Timer3.....23
Figure II.8 Déroulement d'un programme lors d'une interruption. 25
Figure II.9 Module de conversion analogique/numérique. 27
Figure II.10 Principe de transmission. 28
Figure II.11 Fonctionnement de l'EUSART en transmission 30

Figure III.1 La carte de développement PICPLC16 v6.....31
Figure III.2 Le PIC18F4520 en DIP40.....32
Figure III.3 Le programmateur PICflash.....33
Figure III.4 L'alimentation d'énergie.....34
Figure III.5 Module RS-232 34
Figure III.6 Module RS-485 35
Figure III.8 Le connecteur GSM 36
Figure III.9 Horloge temps réel 37
Figure III.10 Les optocoupleurs 38
Figure III.11 Les relais 38
Figure III.12 Ports d'entrées/sorties 39

Figure IV.1 Description de système 40
Figure IV.2 L'organigramme principal 44

Figure IV.3 L'organigramme de la routine d'interruption 45
Figure IV.4 Fenêtre de fichier .C 46
Figure IV.5 Compilation du programme 47
Figure IV.6 Le schéma de simulation dans l'ISIS 47
Figure IV.7 La simulation du test N°1 48
Figure IV.8 La simulation du test N°2 49
Figure IV.9 La simulation du test N°2 après 2 min 49
Figure IV.10 La simulation du test N°3 50
Figure IV.11 Réalisation du montage 51
Figure IV.12 Editeur du programmateur PICflash 52
Figure IV.13 Chargement du programme 53
Figure V.14 Interface homme/machine 54

Listes des Tableaux

Liste des Tableaux

Tableau I.1 les principales solutions technologiques en automatisme.....7
Tableau II.1 registre INTCON 24
Tableau II.2 registre TXSTA 28
Tableau II.3 registre RCSTA 29

Introduction générale

Introduction Générale

L'automatisation s'est généralisée à l'ensemble des activités de production, tant que l'industrie, que dans les activités de services. Quel que soit son domaine d'application et les techniques auxquelles elle fait appel, l'automatisation s'est constamment développée dans l'unique but de réduire la pénibilité du travail humain, d'augmenter la sécurité et d'améliorer la productivité du travail.

L'utilisation croissante de l'automatisation a influencé en profondeur la vie quotidienne et l'évolution générale de la société. Tout au long de l'histoire de l'industrie, cette automatisation a en effet permis une augmentation constante de la productivité (quantité et qualité), ce qui a permis de réduire considérablement le temps de travail nécessaire à la production.

Le développement de la technologie et la complexité des schémas électriques a permis de remplacer les blocs de fonctionnement par des microcontrôleurs, ils sont une révolution dans le monde des circuits intégrés car, ils sont adaptés à accomplir de nombreuses fonctions. Leur utilisation ne se limite plus pour des commandes d'entrées/sorties, mais aussi pour des applications beaucoup plus complexes.

Notre mémoire a pour objectif l'automatisation d'un système de pompage et l'étude de la carte de développement PICPLC16. Nous avons focalisé notre travail à établir le programme adéquat pour notre système.

Maintenant que nous avons présenté l'idée générale de notre projet, nous allons présenter la démarche que nous avons envisagée en indiquant le contenu des différentes parties qui constituent ce mémoire.

Ce mémoire est scindé en quatre chapitres. Le premier chapitre est dédié à la présentation des généralités sur les systèmes automatisés, des généralités sur les microcontrôleurs et l'outil de programmation MPLAB.

Dans le deuxième chapitre nous allons présenter le microcontrôleur PIC18f4520 tout en parlant sur son architecture interne et en décrivant ses constituants tel que les mémoires, les périphériques…etc.

Puisque le PIC ne fonctionne pas sans être implémenté sur une carte, dans ce troisième chapitre nous allons examiner la carte de développent PICPLC16 en la définissant et en présentant les éléments qui là constituent.

Dans le quatrième et dernier chapitre de ce manuscrit on présentera la programmation et la compilation dans MPLAB, la simulation du programme avec le logiciel de ISIS proteus, et vu l'importance de la supervision dans l'industrie moderne, nous allons proposer un modèle d'une interface réalisée avec le GUI de MATLAB pour la supervision du système, à la fin on fera le test de la carte et on terminera le travail par une conclusion générale et des perspectives.

Présentation d’entreprise

Présentation de l'entreprise

SARL Général Emballage est leader confirmé au niveau national de l'industrie du carton ondulé, fondée en Aout 2000 à la zone industrielle TAHARACHT Akbou, véritable carrefour économique de Bejaia.

Actuellement, elle entre en production sur trois sites industriels (Akbou, Oran et Sétif). Elle est entrée en exploitation en 2002, et a obtenu le Trophée de la Production (Euro-Développement PME) en 2007. En raison de son développement. Général Emballage débute son exportation vers Tunis en 2008.

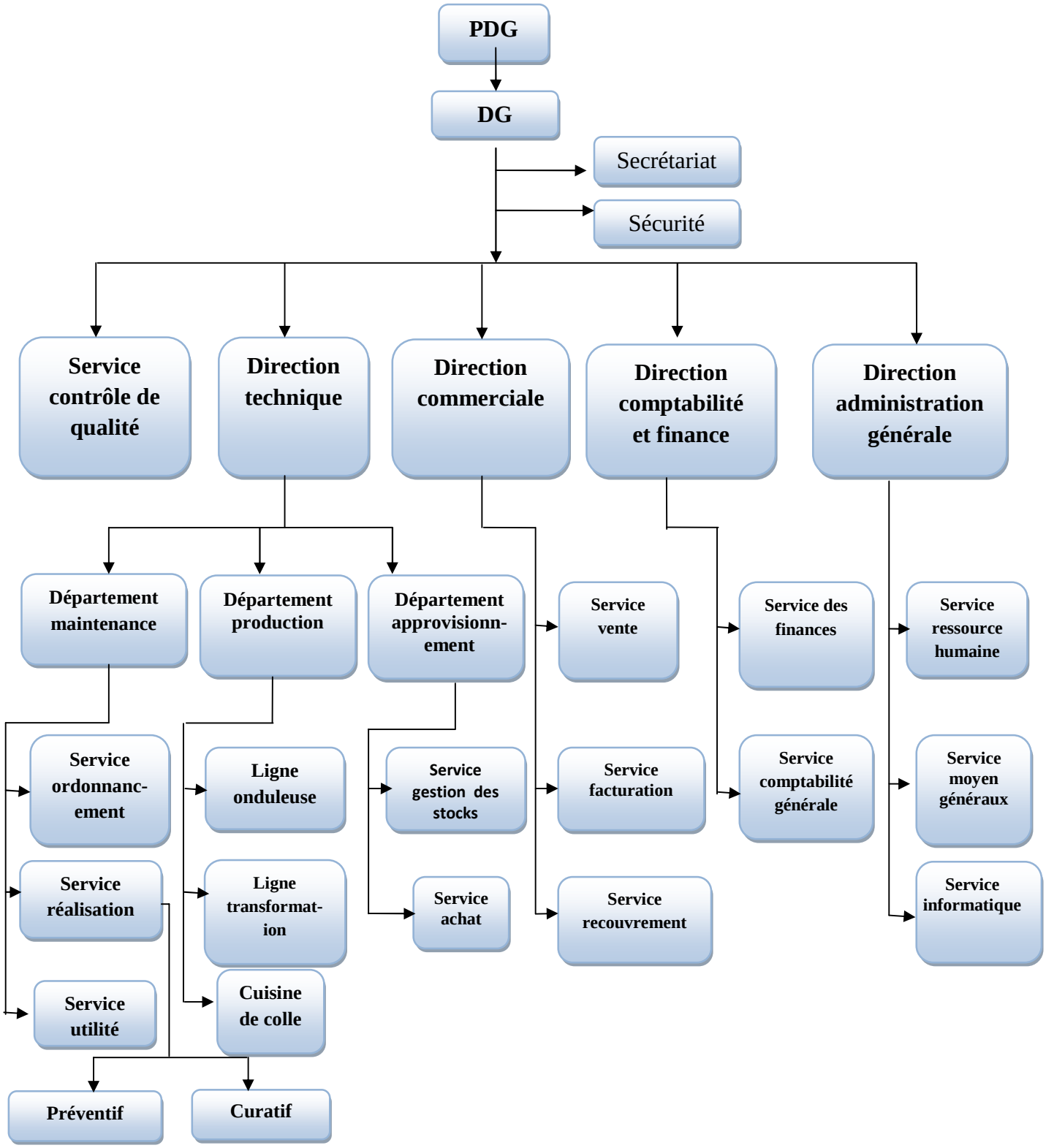
Général Emballage réalise depuis quelques années déjà une croissance à deux chiffres. Les indicateurs de performance de l'entreprise n'ont jamais été aussi mondiaux de certification en Algérie. Dans le cadre de partenariat, l'entreprise signe un accord professionnel avec l'Université de Bejaia en Juillet 2012.

Général Emballage a su au cours des années mettre son expérience au service du client grâce à une recherche permanente de l'efficience qui s'est traduite en 2013 par la certification ISO 9001 : 2008 de son système de management de la qualité.

Situation géographique de l'entreprise

L'entreprise GENERAL EMBALLAGE est implantée dans une zone industrielle TAHARACHT Akbou, véritable carrefour économique de Bejaia. Située à deux Kilomètres d'une grande agglomération (Akbou) et à quelque dizaine de mètres de la voie ferrée. L'entreprise est à 60 km de Bejaïa.

Organigramme de l'entreprise



Organisation général de Général Emballage

Chapitre I

Automatisme et microcontrôleur

I.1 Introduction

Aujourd'hui, dans un monde où la concurrence règne, automatisé son unité de production n'est plus un choix, mais une nécessité pour toute entreprise voulant rester en activité. L'automatisation améliore la qualité du produit, facilite la tâche de l'opérateur et du maintenancier.

L'évolution dans le domaine de l'électronique a pu élaborer des contrôleurs intelligents appelés microcontrôleurs, ces derniers sont caractérisés par leur taille et leur architecture interne qui leur confie souplesse et vitesse. Parmi ces microcontrôleurs on distingue les PICs présentés par l'entreprise américaine MICROCHIP.

I.2 Les systèmes automatisés

Un système de production est dit automatisé, lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences ou étapes. L'automatisation d'un procédé (un équipement industriel) consiste à en assurer la conduite par un dispositif technologique.

I.3 Objectif de l'automatisation

Les objectifs d'automatisé un système sont **[1]** :

- Accroître la productivité du système et améliorer la qualité du produit.
- Augmenter la sécurité du personnel.
- Contrôler et protéger les installations et les machines.
- Améliorer la flexibilité de production.
- Eliminer les tâches complexes, dangereuses, pénibles ou indésirables en les faisant exécuter par la machine.

I.4 Structure d'un système automatisé

Tout système automatisé est composé de trois parties principales :

I.4.1 La partie opérative

C'est la partie visible du système, elle effectue les opérations en exécutant les ordres qui lui sont donnés par la partie commande. Elle comporte les éléments suivants :

- Des préactionneurs, lesquels reçoivent des ordres de la partie commande.
- Des actionneurs qui ont pour rôle d'exécuter ces ordres, il transforme l'énergie pneumatique, hydraulique ou électrique en énergie mécanique.

- D'une détection (capteurs) qui informe la partie commande de l'exécution du travail.

I.4.2 La partie commande

C'est cette partie qui élabore les ordres nécessaires à l'exécution du processus**,** en fonction des informations qu'il reçoit de la partie opérative, il les restitue vers cette partie en direction des préactionneurs.

La partie commande se présente sous forme de :

- Carte électronique.
- Automates programmables industriels API.
- Calculateur (microordinateur).

I.4.3 La partie dialogue

Sa complexité et sa taille dépendent de l'importance du système. Cette partie regroupe les différentes commandes nécessaires au bon fonctionnement du procédé : marche-arrêt, arrêt d'urgence, marche automatique [1]. La figure suivante présente la structure détaillée d'un système automatisé.

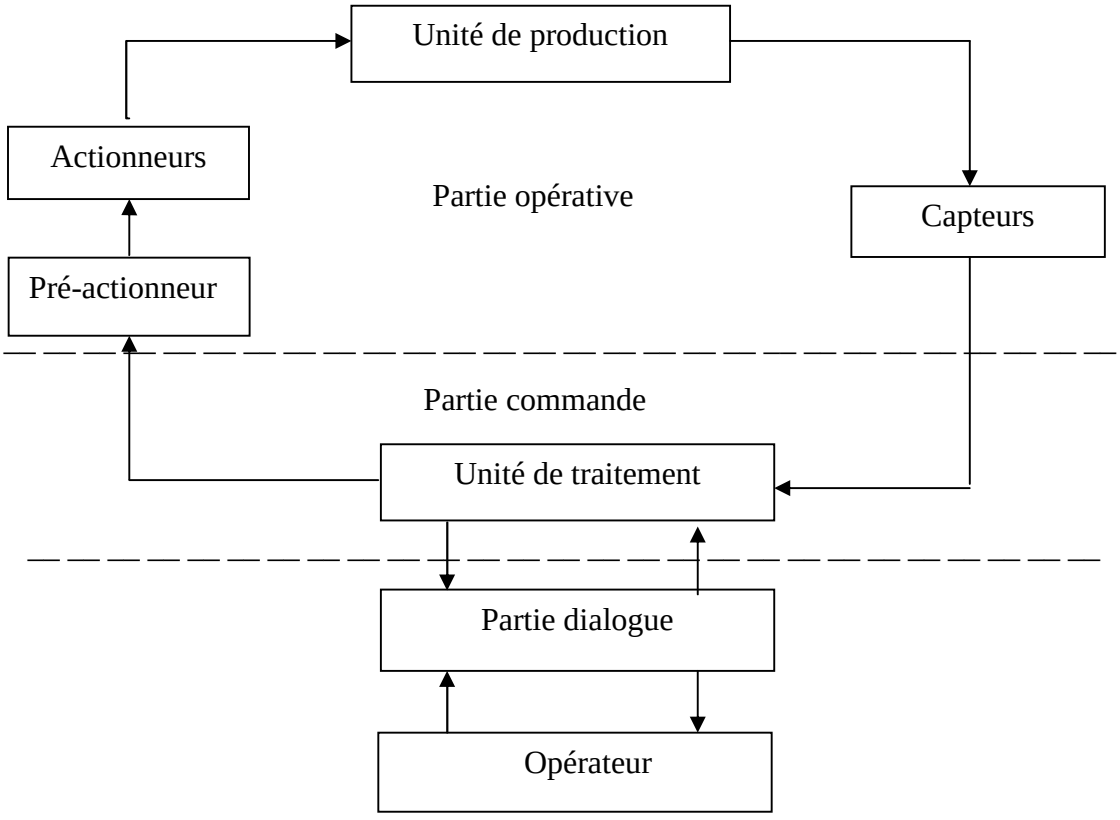


Figure I.1 Structure détaillée d'un système automatisé.

I.5 Les choix technologiques

Lorsque l'automaticien a décrit le fonctionnement de l'automatisme en utilisant le GRAFCET ou L'ORGANIGRAMME, il a le choix pour réaliser la partie commande de l'automatisme, entre diverses solutions technologiques qui sont récapitulées dans le tableau ci-dessous. En fait, trois grandes familles technologiques sont à la disposition des automaticiens : les constituants électromagnétiques, électroniques et pneumatiques. A l'intérieur de chacune des familles on doit prendre en considération la nature du traitement réalisé. Une troisième caractéristique concerne le type d'utilisation liée à la nature du système mis en œuvre : universel pour les systèmes unitaires ou semblables, spécifique pour les systèmes répétitifs [2]. Le tableau suivant présente les principales solutions technologiques en automatisme.

<table>
<tr><td colspan="4">Les principales solutions technologiques en automatisme</td></tr>
<tr><td colspan="2">Famille technologique de base</td><td>universel</td><td>spécifique</td></tr>
<tr><td colspan="2" rowspan="2">Electromagnétique</td><td>Relais</td><td rowspan="2"></td></tr>
<tr><td>Contacteurs auxiliaires</td></tr>
<tr><td rowspan="4">Electronique</td><td>Câblée</td><td>Blocs ou cartes</td><td>Cartes spécifiques</td></tr>
<tr><td rowspan="3">programmée</td><td>Mini ou
micro-ordinateur</td><td rowspan="3">Microsystèmes spécifiques</td></tr>
<tr><td>Automates programmables</td></tr>
<tr><td>Microsystème universels</td></tr>
<tr><td colspan="2">pneumatique</td><td>Logique à clapets</td><td></td></tr>
</table>

Tableau I.1 les principales solutions technologiques en automatisme [2].

I.6 Les microcontrôleurs

Un microcontrôleur se présente comme étant une unité de traitement de l'information de type microprocesseur contenant tous les composants d'un système informatique, à savoir : un microprocesseur, des mémoires et des périphériques (ports, timers, convertisseurs…). Chaque fabricant possède sa ou ses familles de microcontrôleurs.

Une famille se caractérise par un noyau commun (le microprocesseur, le jeu d'instructions…). Ainsi les fabricants peuvent présenter un grand nombre de pins qui s'adaptent plus au moins à certaines tâches. Mais un programmeur connaissant une famille n'a pas besoin d'apprendre à utiliser chaque membre, il lui faut connaître juste ces différences par rapport aux autres familles. Ces différences sont souvent, la taille des mémoires, la présence ou l'absence des périphériques et leurs nombres [3].

I.6.1 Les avantages du microcontrôleur

L'utilisation des microcontrôleurs pour les circuits programmables a plusieurs points forts et bien réels. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années [3].

- Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement du matériel et du circuit imprimé.

- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de données d'un composant à un autre.

- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux :
  - Moins cher que les autres composants qu'il remplace.
  - Diminuer les coûts de main d'œuvre.

I.6.2 Structure d'un microcontrôleur

Un circuit microcontrôleur doit contenir dans un seul boîtier tous les éléments de base, nous y retrouvons bien évidemment l'unité centrale, des périphériques (ports, timers, convertisseurs, et des mémoires…). Voir la Figure I.2

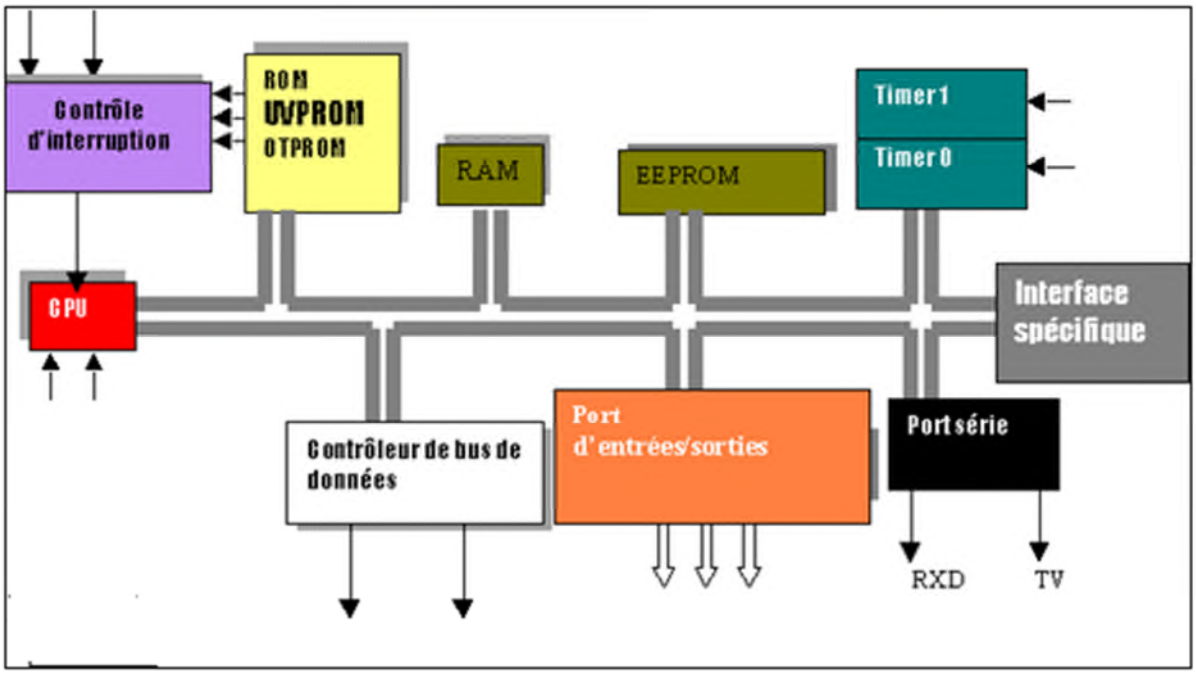


Figure I.2 Structure interne d'un microcontrôleur.

I.7 Le microcontrôleur PIC

I.7.1 Définition d'un PIC

Un PIC n'est rien d'autre qu'un microcontrôleur, c'est à dire une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants externes [3].

Les PICs sont des composants dits RISC (Reduce Instructions Set Computer), ou encore composant à jeu d'instructions réduit, sachant que plus on réduit le nombre d'instruction plus facile et plus rapide en est le décodage, et plus vite le composant fonctionne.

I.7.2 Les différentes familles des PICs [4]

MICROCHIP propose une très grande variété de PICs élargissant ainsi l'étendu des applications possibles, et offrant la possibilité d'adapter le PIC à l'application à réaliser. Les PICs sont classés en trois familles principales, chaque famille possède une architecture de base bien déterminée, et utilise un mot d'instruction qui est différent pour chaque famille.
Les trois familles sont :

- **les PICs base-line** : qui utilisent des mots d'instructions de 12bits.
- **les PICs mid-range** : qui utilisent des mots d'instructions de 14bits.
- **les PICs high-end** : qui utilisent des mots d'instructions de 16 bits et plus.

I.7.3 Identification d'un PIC

les PICs sont identifiés par une étiquette de ce genre : xx(L)XXyy-zz
xx : indique la famille.
L : indique qu'il fonctionne avec une plage de tension beaucoup plus tolérante.
XX : c'est le type de mémoire de programme.
exp : C : pour indiquer qu'il s'agit d'une mémoire EPROM ou EEPROM.
CR : pour dire qu'elle est de type ROM.
F : pour préciser le genre FLASH.
yy : identifie le PIC lui-même.
zz : la vitesse max du quartz.

I.7.4 Classification des microcontrôleurs [3]

On peut classer les microcontrôleurs selon deux critères importants :

- Selon l'organisation des mémoires.
- Selon l'unité centrale de traitement.

A. Selon l'organisation des mémoires

On distingue deux architectures souvent utilisées en conception des microcontrôleurs :

- **Architecture de Von Neumann**

Cette architecture est très utilisée dans la majorité des microprocesseurs et des microcontrôleurs, elle est commune à celles des ordinateurs, où la mémoire contient à la fois les instructions à exécuter et les données à manipuler, et elle ne dispose que d'un seul bus de donnée qui relie la mémoire à l'unité centrale. La figure suivante présente la structure de Von Neumann.

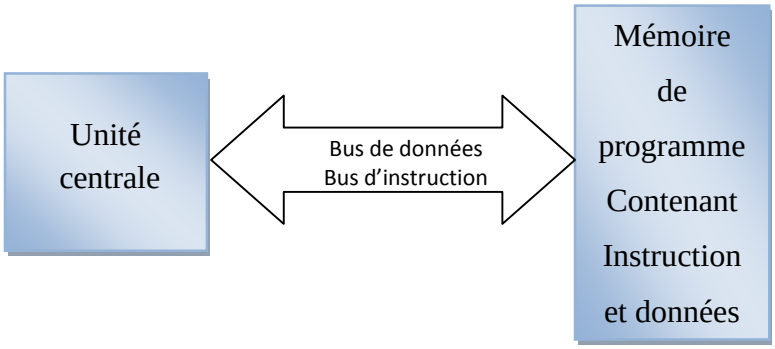


Figure I .3 Structure de Von Neumann.

- **Architecture Harvard**

Contrairement à l'architecture Von Neumann, ce type d'architecture contient des mémoires, l'une propre au programme et l'autre aux données. Elles sont séparées et câblées à l'unité centrale via des bus différents, ce qui offre une possibilité d'avoir simultanément la donnée et l'instruction. La figure suivante présente l'architecture Harvard.

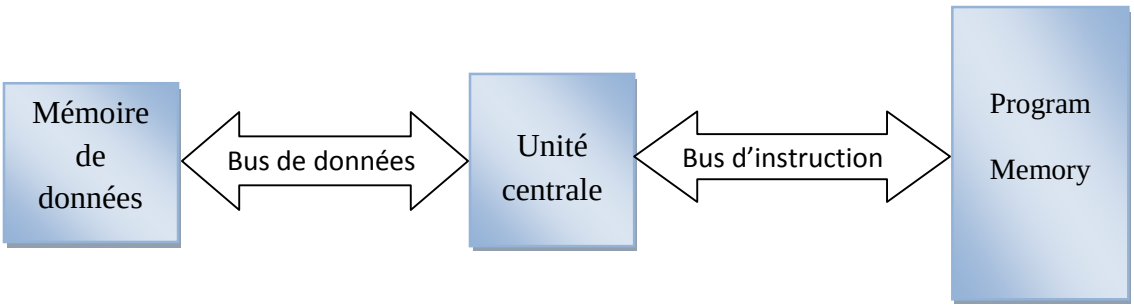


Figure I.4 Architecture de Harvard.

B. Selon l'unité centrale de traitement

Au niveau de l'unité centrale de traitement, on trouve deux types :

- **RISC** : qui veut dire microprocesseur à jeu réduit (reduced instruction-set computer).

Les PICs à unité RISC utilisent des instructions sur 12 à 14 bits ou sur mot ce qui se traduit par deux avantages : tous les emplacements de la mémoire contiennent une instruction, et un seul cycle machine suffit pour lire le code de l'instruction complet. En plus ils utilisent une structure de type Pipe-line.

- **CISC :** qui veut dire microprocesseur à jeu d'instruction complexe (complexe instruction-set computer). Comme son nom l'indique cette unité utilise un jeu d'instruction compliqué et encombré, difficile à maîtriser mais permet d'autre part de réaliser des traitements très précieux.

I.7.5 Familles de PIC disponibles sur le marché [3]

Les principales familles de PICs sont :

10Fxxx/12Fxxx : sont des composants récents. Ils ont comme particularités d'être extrêmement petits, simples et économiques.

16Cxxx/16Fxxx : est un composant de milieu de gamme. Il constitue la famille la plus fournie et la plus utilisée.

17Cxxx : est une gamme intermédiaire entre 16Cxxx et 18Fxxx.

18Cxxx/18Fxxx : est une famille qui a un jeu d'instructions plus complet puisqu'il comprend l'ordre de 75 instructions. Cette gamme d'instruction étendue lui permet de faire fonctionner du code en langage C compilé de manière nettement plus efficace que les familles précédentes. On peut les utiliser avec un quartz oscillant jusqu'à 48MHz.

24Fxxx : sortie en 2004. Elle est programmable en langage C comme tous les autres PICs.

PIC32 : sortie en novembre 2007, les PICs 32 sont des microcontrôleurs à 32 bits.

dsPIC30/dsPIC33 : le dsPIC (digital signal PIC) est le premier microcontrôleur qui a une architecture de 16bits(les autres étant à 8bits) de la société MICROCHIP. Il est adapté aux applications de traitement de signal.

I.7.6 Choix d'un PIC [5]

Le choix d'un PIC est lié à l'application envisagée, en se basant sur les directives suivantes :

- Il faut dans un premier temps déterminer le nombre d'entrées-sorties nécessaires pour l'application. Ce nombre nous donne la famille du PIC.
- Il faut déterminer si l'application nécessite un convertisseur analogique numérique. Ce qui va centrer un peu plus vers le choix d'un PIC.
- La rapidité d'exécution est un élément important, il faut consulter les databook pour vérifier la compatibilité entre la vitesse du PIC choisi et la vitesse maximale nécessaire au montage.
- La taille de la RAM et la présence ou non d'une EEPROM pour mémoriser les données sont également importantes pour l'application souhaitée.
- La longueur du programme de l'application détermine la taille de la mémoire du PIC recherchée.

Selon l'application, il appartient à l'utilisateur de choisir la famille qui répond au mieux à son cahier de charge.

I.8 La programmation des PICs

La programmation peut se faire par deus méthodes, soit en langage assembleur ou langage évolué tel que : pascal, Basic, C.

Le choix d'un langage dépend essentiellement de l'occupation en mémoire. Si on utilise un langage évolué on aura d'une part des mots puissants et directe, donc le programme comportera peu de mots et automatiquement peu d'erreurs, d'autre part il nécessite un logiciel précisément conçu pour la programmation des PICs, ce dernier est généralement coûteux et exige plus d'espace en mémoire, car une instruction de ce langage peut se traduire par plusieurs lignes en langage machine.

La plupart des programmateurs des PICs préfèrent le langage assembleur, qui se fait à l'aide d'un outil de programmation qui est le MPLAB [6].

I.9 MPLAB

MPLAB est un outil fourni gratuitement par la société MICROCHIP. A travers lequel il est possible :

- Ecrire un programme pour un PIC.
- D'assembler et compiler le programme.

- De simuler le fonctionnement du programme et de trouver d'éventuelles erreurs.
- De programmer le microcontrôleur.

I.9.1 Les outils de MPLAB

Le programme ayant été écrit et compilé, il faut vérifier qu'il rempli bien les fonctions attendues. Il est bien sûr possible de programmer le composant et de lancer l'exécution pour voir si tout fonctionne comme prévu ou non, mais en cas d'échec il n'est pas facile de trouver où est l'erreur. Il existe des émulateurs matériels qui permettent en faisant tourner le programme à sa vitesse normale de suivre l'évolution du contenu des mémoires internes et d'observer à quel moment le fonctionnement cesse d'être nominal. Ces émulateurs sont fragiles et coûteux et avant de faire appel à eux il est commode d'effectuer une exécution pas à pas du programme grâce à un simulateur logiciel, un tel outil qui exploite le fichier .OBJ Crée par l'assembleur est fourni par MICROCHIP dans la suite MPLAB.

I.9.2 Compilation

Le programmeur écrit son programme dans un ou plusieurs fichiers texte en utilisant le langage normalisé "C" (les fichiers "sources"). Ces fichiers ne peuvent pas être programmés dans le microcontrôleur cible tels quels : le CPU de celui-ci ne connaît que le code machine et ne comprend rien au texte des fichiers source. La tâche du compilateur est de "traduire" ces fichiers source en code machine (fichier .hex), en utilisant éventuellement des librairies de fonctions (mathématiques par exemple) fournies avec le compilateur.

I.10 Conclusion

Dans ce chapitre nous avons donné dans un premier temps une description générale sur les systèmes automatisés, leur structure et comment choisir une technologie d'automatisme. Après on a étudié les microcontrôleurs et spécifiquement les familles des PICs ainsi que leur programmation. Le prochain chapitre sera consacré à l'étude du PIC 18f4520 qui est le noyau de notre projet.

Chapitre II

Le microcontrôleur PIC18F4520

II.1 Introduction

Ce chapitre est consacré à la présentation du microcontrôleur PIC18f4520 tout en illustrant ces différentes caractéristiques afin de mieux l'exploiter.

On aura à définir l'unité de contrôle et tout ce qui l'englobe de part et d'autre de sa structure. La partie la plus importante sera celle qui traite les différents ports d'entrées/sorties, ainsi que les Timers qui vont êtres exploités dans notre projet.

II.2 Présentation du PIC18f4520

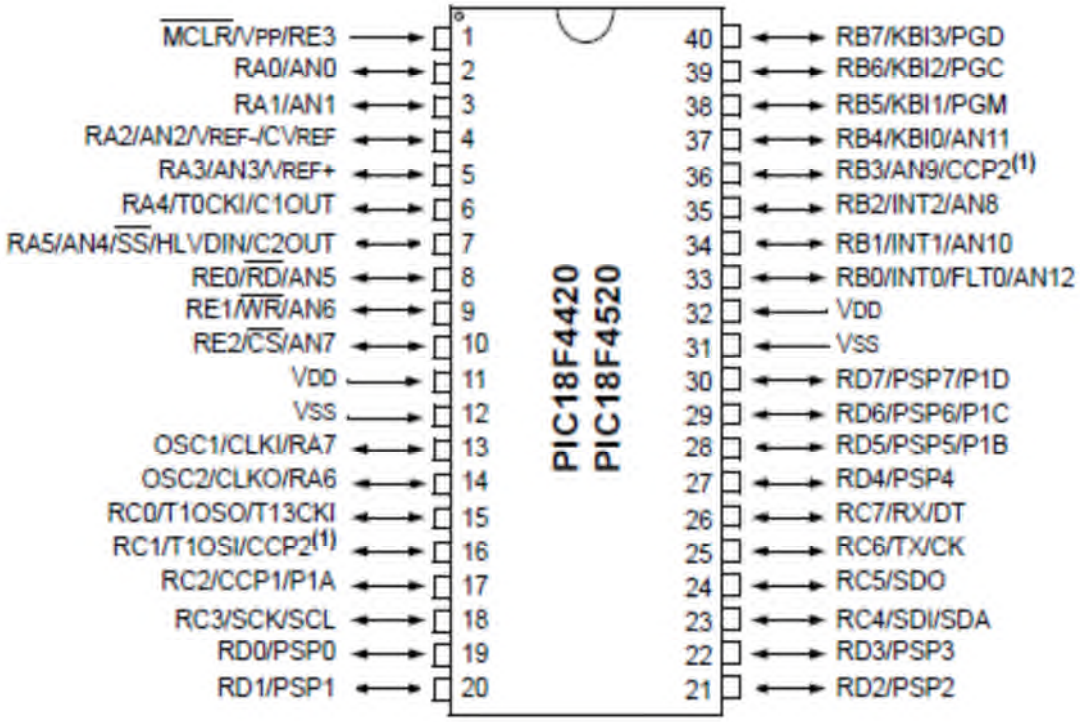


Figure II.1 Architecture externe du PIC18f4520.

On peut distinguer sur ce schéma :

- L'alimentation : V_{DD} (+5V) et Vss (0V).

-Les bornes du quartz (oscillateur a quartz) : OSC1 et OSC2.

- Entrée RESET (MCLR: Master CLeaR).

- Les différents ports d'Entrées/Sorties : PORTA, PORTB, PORTC, PORTD, PORTE.

II.3 Architecture interne du PIC18f4520

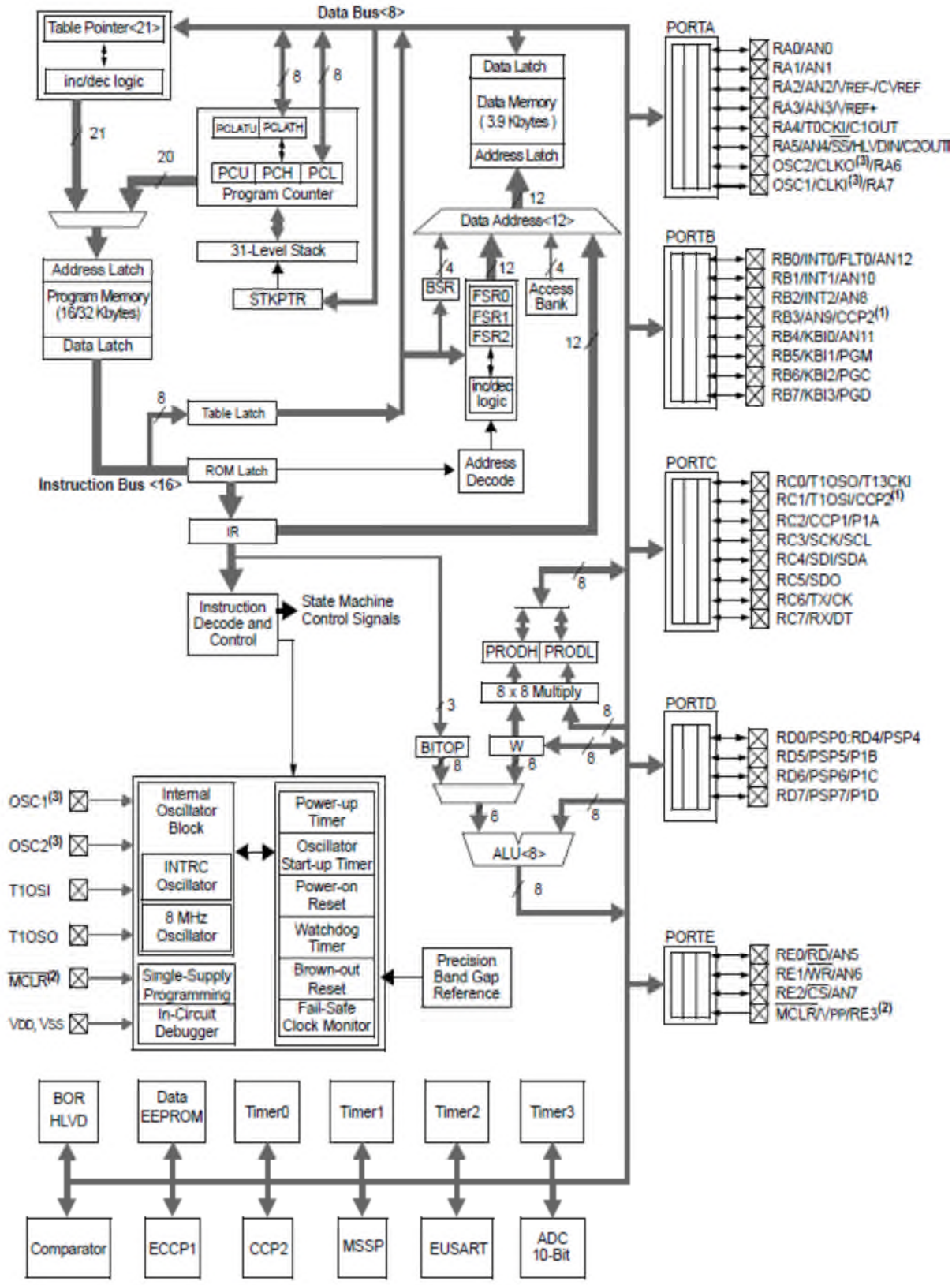


Figure II.2 Architecture interne du PIC 18f4520 [10].

II.4 Caractéristiques du PIC18F4520

Le PIC18f4520 est un microcontrôleur de chez l'entreprise Américaine MICROCHIP technologie, ses caractéristiques sont [10] :

- Vitesse maximal de 40 MHz.
- Mémoire de programme 32 KB.
- Mémoire RAM 1664B.
- 4 Timers (Timer0 sur 8- bits et 3 Timers sur 16 bits).
- Jusqu'à 20 sources d'interruption.
- EEPROM 256B.
- Un convertisseur analogique-numérique (CAN) 10 bits.
- Deux modules de génération d'impulsion à période réglable (PWM).
- Un module de communication série synchrone (MSSP).
- Une EUSART (Enhaced Universal Synchronous Asynchronous Receiver Transmitter).
- Un module de communication ≪ port parallèle ≫.

II.5 Calcification des éléments

On peut classer ces éléments en trois classes : le noyau, les périphériques et les fonctions spéciales.

II.5.1 Le noyau :

C'est la partie commune à tout les PICs, elle contient les éléments de base pour le fonctionnement du microcontrôleur, parmi eux on trouve :

- L'unité Arithmétique et Logique(UAL).
- Les mémoires : mémoire de programme, mémoire RAM et mémoire EEPROM.
- Les RESETs: MCLR, POR, BOR, PER.
- L'oscillateur.

II.5.2 Les périphériques :

Cette partie est propre au PIC, c'est elle qui le diffère des microcontrôleurs, elle permet de communiquer avec l'environnement extérieur, elle contient :

- Les ports d'entrées/sorties.
- Le convertisseur analogique/numérique(CAN).

- Le port parallèle esclave PSP.
- Les CCP1/2(compare/capture/PWM).
- L'EUSART.

II.5.3 Les fonctions spéciales :

Elles permettent d'améliorer la fiabilité du système et parmi ces fonctions on trouve :

- Les différents RESETs.
- Les Timers et chien de garde WDT.
- L'oscillateur interne RC.

II.6 Organisation de la mémoire

Il y a trois zones de mémoire dans l'architecture; mémoire de programme, mémoire de données et la mémoire EEPROM, la structure Harvard des PICs fournit un accès séparé à chacune.

II.6.1 Mémoire de programme (Flash)

C'est une mémoire réinscriptible qui conserve ses données lorsque le PIC n'est pas alimenté. Elle est utilisée pour stocker le programme [10].

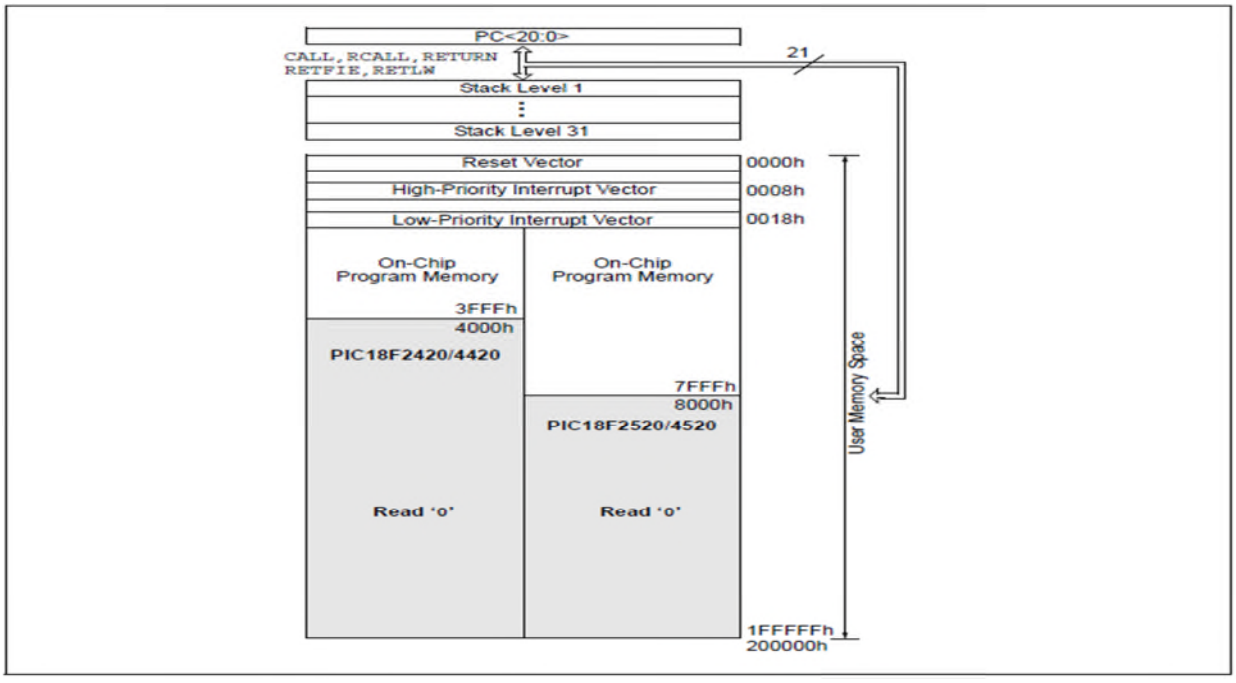


Figure II.3 Organisation de la mémoire et de la pile.

Comme la montre la figure ci-dessus, la mémoire s'étale linéairement de 0000h à 1FFFFFh. Le vecteur de RESET est placé à l'adresse 0000h, le vecteur d'interruption prioritaire élevé est à l'adresse 0008h et le vecteur faible priorité d'interruption est à 0018h, la pile est à 31 niveaux de 21 bits chacun, elle est utilisée lors d'un appel de sous programme ou de service d'interruption pour sauvegarder le contenue de compteur ordinale PC avant l'exécution de ce dernier afin de permettre au PIC de savoir qu'elle est l'instruction suivante à exécuter.

II.6.2 Mémoire RAM

C'est une mémoire volatile c'est- à-dire qu'elle s'efface quand le PIC n'est plus alimenté. Les variables utilisées dans le programme sont stockées à cet endroit.

II.6.3 EEPROM :

La particularité de ce genre de mémoire qui se comporte comme une mémoire vive est de conserver son contenue pendant une durée indéterminée, même en absence d'alimentation. Elle est d'une taille de 256bytes. Pour accéder à cette mémoire on utilise les registres suivants :

- EEDATA : contient la donnée à enregistrer ou celle issue de la mémoire.
- EEADR : contient l'adresse de la case mémoire dont on écrit ou on lit.
- EECON1 et EECON2 : ces registres permettent de définir le mode de fonctionnement de la mémoire ou de contrôler son accès.

II.7 Les ports d'entrées/sorties (I/O)

Les ports d'entrées/sorties font partie des ressources les plus utilisées sur les microcontrôleurs, ils sont très variable en nombre, ces derniers sont liés aux dimensions du boitier, donc aux nombres de pattes disponible.

Le PIC18f4520 contient cinq ports d'entrées/sorties: A, B, C, D et E qui permettent la communication entre le processeur et le milieu extérieur. Ils peuvent servir d'E/S numériques standards ou d'E/S de périphériques internes lorsqu'il s'agit d'une fonction spéciale telle que liaison série ou parallèle, convertisseur A/N.

Ces ports sont bidirectionnels, leurs configurations se fait par des registres spécifiques.

II.8 Le reset

Le reset est définit comme étant l'évènement qui redémarre le programme. Pour le PIC18F4520 il a plusieurs sources de reset [10] :

- POR: Power-on Reset.
- MCLR: Master Clear Reset Pin.
- WDT: Watchdog Timer.
- BOR: Brown-out Reset.
- Reset Instruction**.**
- Stack Full Reset.
- Stack Underflow Reset.

II.9 Timers

D'une façon générale la fonction Timer se charge de la gestion du temps. Le Timer peut être cadencé par une horloge externe ou interne.

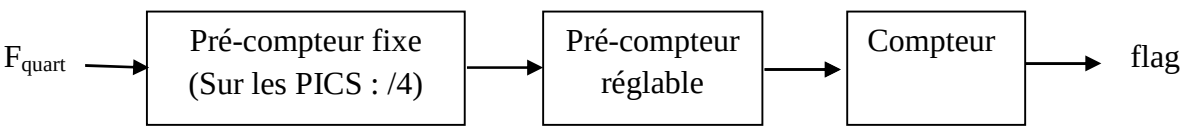
Tous les microcontrôleurs ont un ou plusieurs Timers pour gérer le temps. Le PIC 18f4520 possède 4 Timers (timer0….timer3).

Un Timer est un circuit périphérique (inclut dans le PIC) lancé par le programme et qui compte un nombre prédéfini d'impulsion du quartz. Ces impulsions peuvent provenir du :

- Quartz cadençant le microcontrôleur, on parle alors d'un fonctionnement en compteur de temps.
- D'une entrée TOR du PIC (exemple : RA4 pour timer0, RC0 pour timer1), on parle alors d'un fonctionnement en compteur d'évènement extérieur. Quand ce nombre d'impulsion est atteint, un drapeau (flag en anglais) se lève.une interruption peut être également générée.

Les Timers ont presque tous le même fonctionnement, ils ont tous un compteur, un pré-compteur (prescaler en anglais).

L'horloge du microcontrôleur est pré-divisée pour obtenir une fréquence plus lente. Dans le cas des PICs, l'horloge du quartz n'est pas directement connecté au pré-compteur mais passe par un diviseur par 4.



On règle le compteur à une valeur initiale et on attend qu'il atteigne une valeur finale. Lorsque le compteur atteint cette valeur finale, un indicateur (flag en anglais) devient actif.

Le temps mis par le compteur pour faire lever le flag est égal à :

$$T = \frac{1}{\frac{F}{\text{Valeur du précompteur fixe} \times \text{Valeur du précepteur réglable} \times \text{nombre d'impulsion}}}$$

II.9.1 Timer0

Le Timer0 peut être utilisé en mode 8 bits ou en mode 16 bits pour obtenir un temps plus long. Le Timer0 possède des sources d'horloges :

- Soit l'horloge du PIC (l'horloge à quartz) divisée par 4.
- Soit avec une horloge externe connectée sur la broche RA4/T0CKI.

Dans le cas de Timer0 en mode 8 bits, la formule du temps devient :

$$T = \frac{1}{\frac{F}{4 \times \text{Valeur du précompteur réglable} \times (256 - \text{valeur initiale})}}$$

Dans le cas de Timer1 et Timer0 en mode 16 bits, la formule du temps devient :

$$T = \frac{1}{\frac{F}{4 \times \text{Valeur du précompteur réglable} \times (65536 - \text{valeur initiale})}}$$

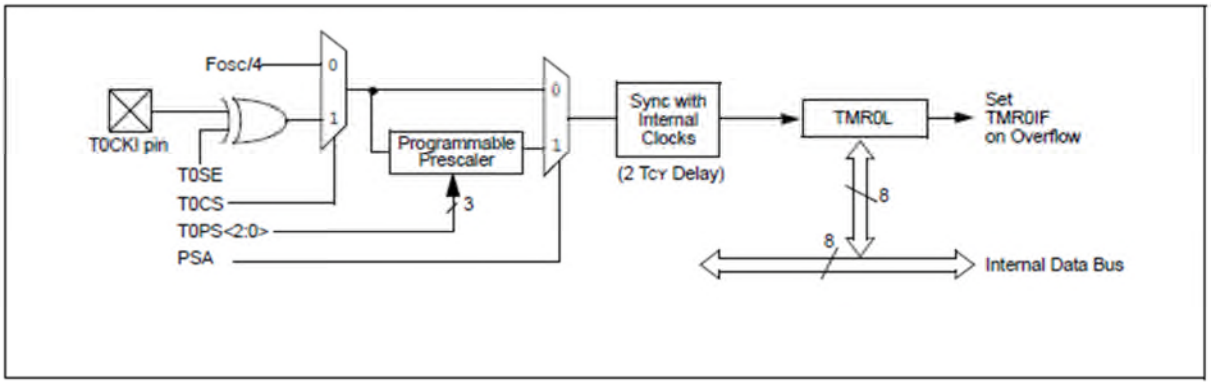


Figure II.4 Bloc diagramme du Timer0 (mode 8 bits) [10].

II.9.2 Timer1

Le timer1 fonctionne sur le même principe que le timer0 mais il est plus moderne dans sa conception. C'est un compteur à 16 bits divisé en 2 registres de 8 bits : TMR1L pour les poids faibles et TMR1H pour les poids forts. Son fonctionnement est résumé dans la figure ci-dessous.

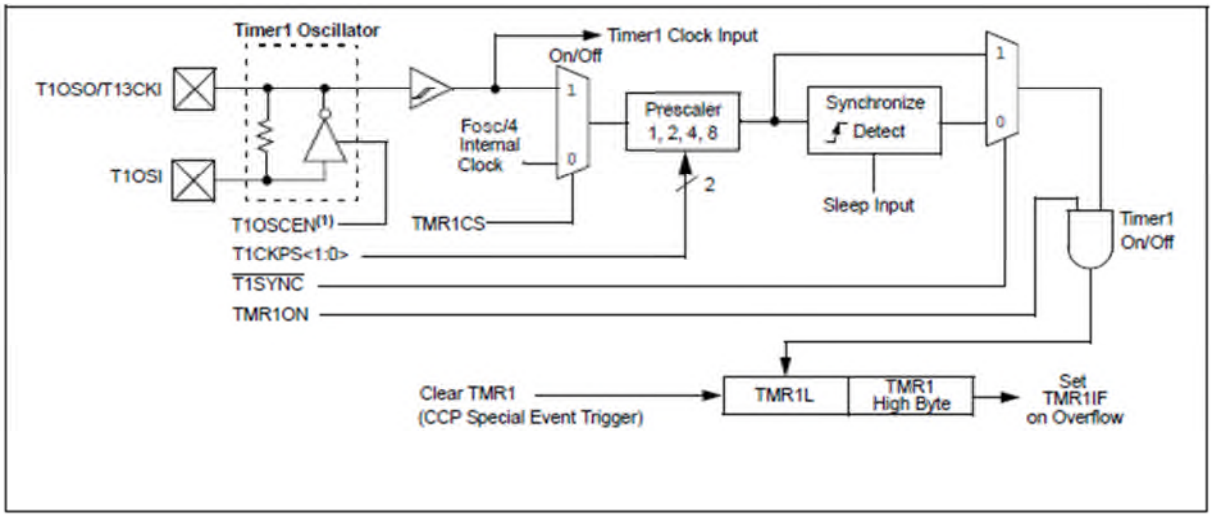


Figure II.5 Bloc diagramme du Timer1 [10].

Le timer1 possède deux sources d'horloge :

- Soit l'horloge du PIC(en général un quartz) divisée par 4.
- Soit avec une horloge externe connectée sur la broche RC0/T13CKI ou la broche RC1.

Le choix entre ces deux horloges se fait à l'aide du bit TMR1CS.

II.9.3 Timer2

Il est légèrement différent des Timer0, Timer1 et Timer3, puisque le début de comptage est obligatoirement 0x00 et que la fin de comptage est la valeur à entrer. C'est un compteur 8 bits, son horloge est divisée par 4(Fosc/4). Il est composé d'un registre de 8 bits TMR2, d'un pré-diviseur qui peut être paramétré avec l'une de ces trois valeurs : 1, 4 et 16, d'un post diviseur qui peut prendre des valeurs de 1 à 256, ainsi que d'un registre dit de période appelé PR2.

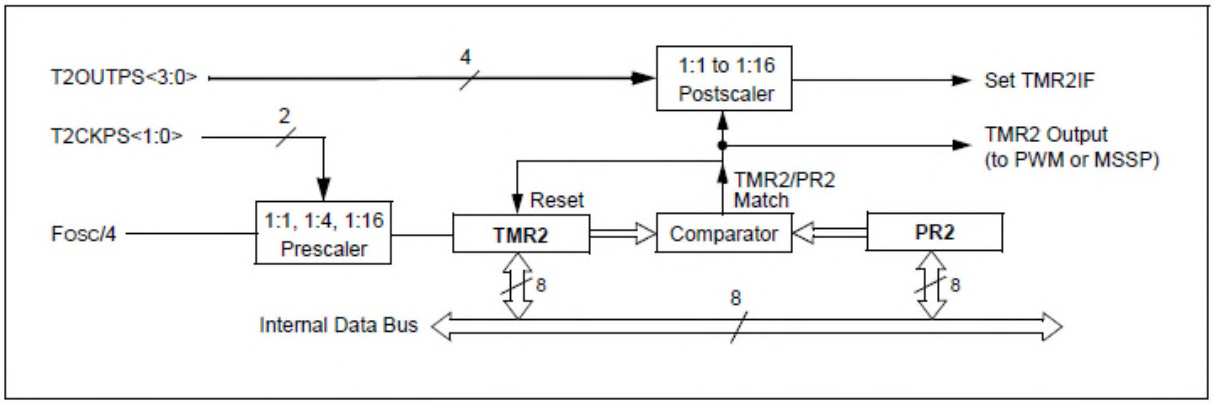


Figure II.6 Bloc diagramme du Timer2 [10].

II.9.4 Timer3

Il fonctionne comme le Timer0 et le Timer1.C'est un Timer de 16 bits, il est composé d'un registre de 8 bits TMR3, d'un pré-diviseur qui peut être paramétré avec l'une de ces valeurs : 1, 2, 4 et 8.

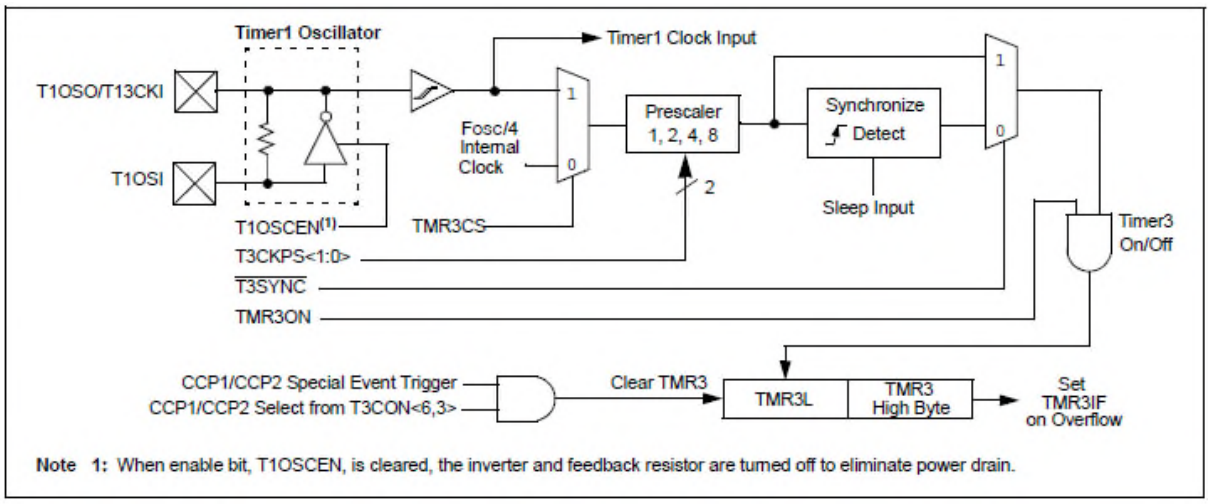


Figure II. 7 Bloc diagramme du Timer3 [10].

II.10 Interruption

Une interruption est un arrêt temporaire de l'exécution normale du programme principal par le microprocesseur afin d'exécuté un autre programme appelé routine d'interruption ou programme d'interruption. Ce programme d'interruption va permettre de gérer l'événement qui a provoqué cette interruption.

Les interruptions sont, en général, contrôlées par trois bits :

- Un bit de flag : Indique qu'une interruption a été déclenchée et indique la source.
- Un bit de validation : permet à l'utilisateur d'activer ou une interruption.
- Un bit de priorité : permet de sélectionner la priorité (haute/basse) de l'interruption.

Le PIC 18F4520 dispose de 20 sources d'interruptions externes et internes, parmi ces sources :

- Débordement des Timers.
- Fin de la conversion A/N.
- Interruption externe sur INT0/RB0.
- Ecriture dans la mémoire EEPROM.
- Réception ou fin d'émission d'une information sur la liaison série.

Les bits d'interruptions sont accessibles dans les registres : RCON, INTCON, INTCON2, INTCON3, PIR1, PIR2, PIE1, PIE2, IPR1, IPR2. Le deuxième registre est le contrôle global des interruptions, son contenu est précisé dans le tableau suivant :

GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF

Tableau II.1 Registre INTCON.

- RBIF : indique les changements d'état des lignes de RB4 à RB7 s'il est mis à 1.
- INT0IF : indique un front montant sur l'entrée INT/RB0 (interruption externe).
- TMR0IF : indique un débordement duTimer0.
- RBIE : autorise l'interruption provoquée pas les changements d'états des lignes RB4 à RB7.
- INT0IE : autorise les interruptions provoquées par la ligne externe INT/RB0.
- TMR0IE : autorise les interruptions dues au débordement du Timer0.

- PEIE : autorise les interruptions périphériques.
- GIE : permet de valider toute les interruptions ; c'est une validation générale.

Il existe encore d'autres registres qui interviennent dans le contrôle des interruptions, il s'agit : des registres d'autorisation d'interruptions périphériques PIE1 et PIE2 et des registres des indicateurs d'évènements PIR1 et PIR2.

- **La séquence de déroulement d'une interruption [7]**

Quand l'interruption se produit :

- Le microcontrôleur finit l'instruction en cours.
- Sauvegarde l'adresse suivante du programme principal.
- Sauvegarde le contexte.
- Exécuté le programme lié au périphérique demandant l'interruption.
- Restitue le contexte.
- Retourne au programme principal à l'adresse suivante.

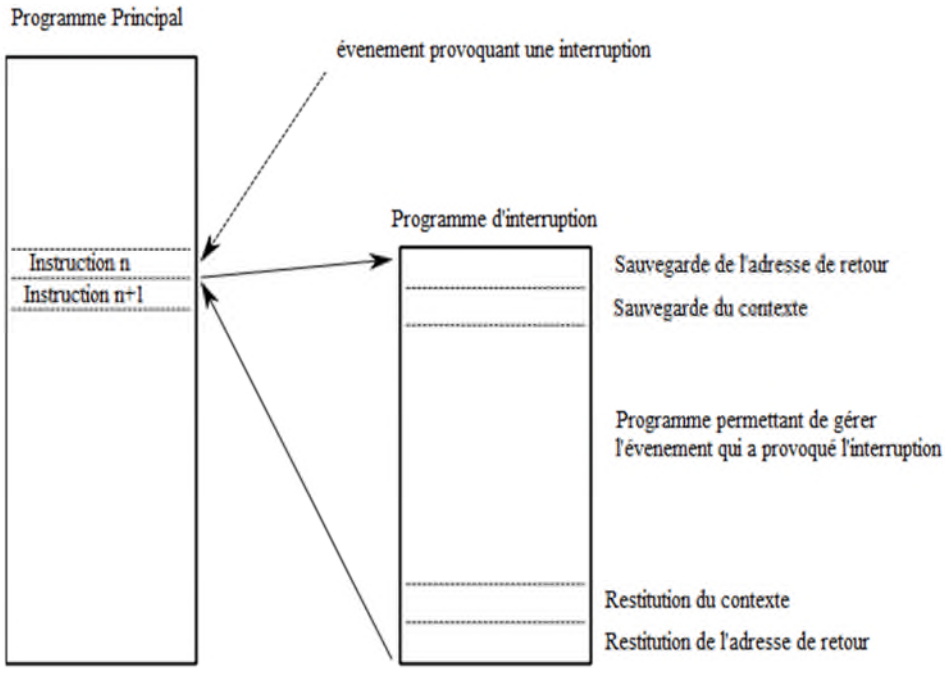


Figure II.8 Déroulement d'un programme lors d'une interruption.

II.11 Watch dog (chien de garde)

C'est un système de protection contre un blocage du programme. Par exemple, si le programme attend le résultat d'un système extérieur (conversion analogique numérique) et qu'il n'y a pas de réponse, il peut rester bloquer. Pour en sortir on utilise un chien de garde. Il s'agit d'un compteur qui, lorsqu'il arrive enfin de comptage, permet de redémarrer le programme. Il est lancé au début du programme. En fonctionnement normal, il est remis à zéro régulièrement dans une branche du programme qui s'exécute régulièrement. Si le programme est bloqué, il ne passe plus dans la branche de remise à zéro et le comptage va jusqu'au bout, déclenche le chien de garde qui relance le programme.

II.12 Mode sommeil (mode SLEEP)

La fonction Sleep arrête l'activité du système et réduit sa consommation en énergie en arrêtant l'activité da la PLL. Dans ce mode, la consommation est très basse, permettant l'alimentation par batterie. Le fonctionnement normal peut être repris (sortie du mode sleep) s'y il a une interruption, un reset ou expiration du temps du watchdog.

II.13 Les périphériques

II.13.1 Les module CCPM

II.13.1.1 Mode Capture

Le mode capture déclenche une action si un évènement prédéterminé apparait (ex : changement d'état sur une broche). Utilisé avec Timers, ce mode peut compter les temps d'arrivées.

II.13.1.2 Mode Compare

Le mode compare effectue une comparaison permanente entre le contenu d'un Timer et une valeur donnée pour déclencher une action si ces contenus sont égaux.

II.13.1.3 Mode PMW

Le mode PMW génère un signal rectangulaire de fréquence et de rapport cyclique choisis par l'utilisateur.

II.13.1.4 Les comparateurs

Les comparateurs permettent de comparer le signal analogique présent sur une broche du microcontrôleur à une valeur de référence.

1I.13.2 Module de conversion analogique/numérique

Le module de conversion analogique/numérique permet de convertir le signal analogique présent sur une broche du microcontrôleur en un signal numérique. Ce module dispose de 13 canaux d'entrées analogiques sur les pins AN0 À AN12 du port A. Il permet un échantillonnage ou bien une résolution sur 10 bits.

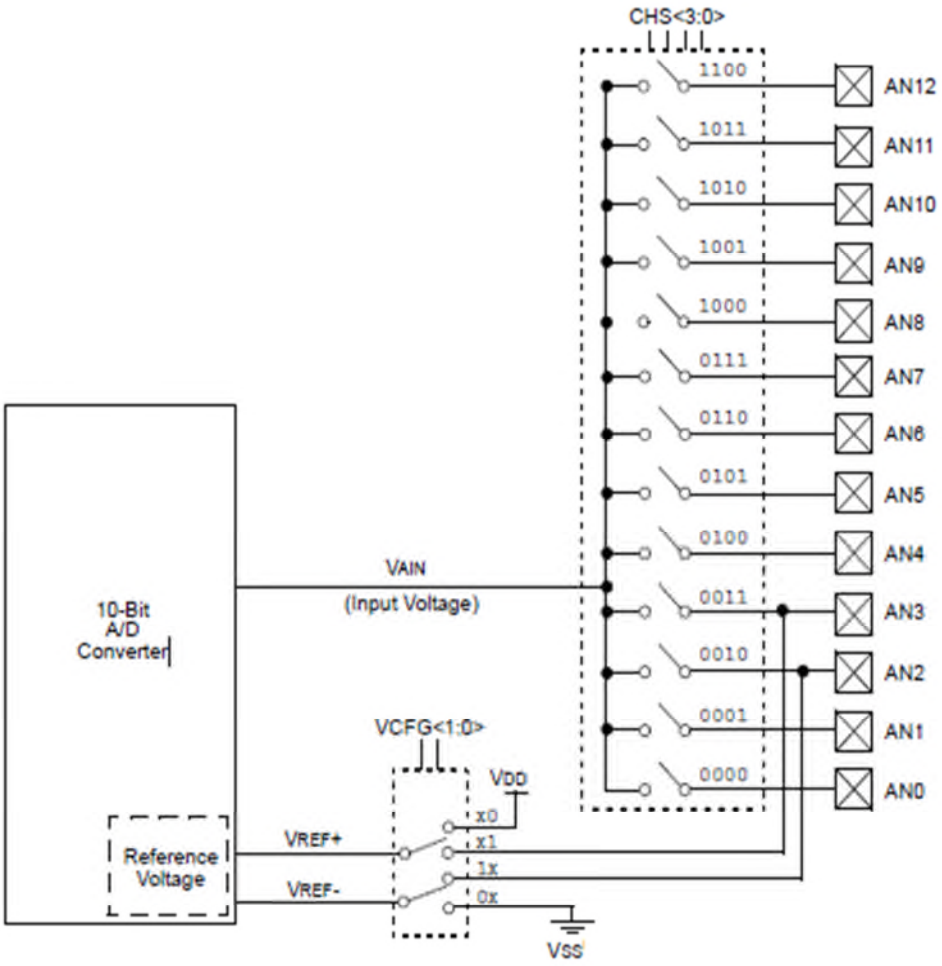


Figure II.9 Module de conversion analogique/numérique [10].

II.13.3 Liaison série RS232

La communication entre deux systèmes se fait de manière parallèle ou de manière série. La communication série est très importante dans le domaine de la télécommunication et plus généralement dans le transfert d'information.

Dans une communication série, les bits sont envoyés les uns à la suite des autres sur la ligne en commençant par le bit de poids faible. La transmission s'appuie sur le principe des registres à décalages.

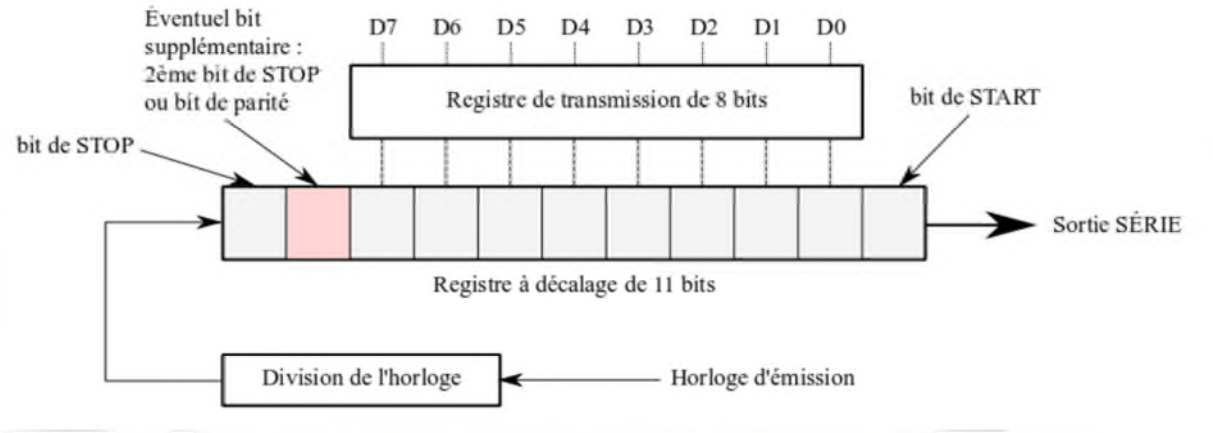


Figure II.10 Principe de transmission.

II.13.3.1 Registre de contrôle et d'état

Afin de configurer la liaison série et de vérifier que la transmission s'est bien effectuée, on dispose des registres de contrôle et d'état :

CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

Tableau II.2 Registre TXSTA.

CSRC : permet la sélection de l'horloge interne ou externe.

TX9: 1: permet de valider une transmission de 8 bits ou de 9 bits.

TXEN: permet l'activation de la transmission.

SYNC: permet de choisir le type de fonctionnement.

BRGH: permet de fixer la vitesse de transmission.

TRMT: état du registre à décalage.

TX9D: en mode 9 bits, il faut placer le 9éme bit dans TX9D du registre TXSTA.

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

Tableau II.3 Registre RCSTA.

SPEN : permet de valider le port série.

RX9 : permet de valider une réception de 8 bits ou 9 bits.

CREN : permet la validation de la réception.

ADDEN : validation de la détection d'adresse.

FERR : drapeau de détection d'erreur d'encadrement (Framing Error).

OERR : drapeau de détection d'Overrun Error.

RX9D : en mode 9 bits, il faut placer le 9éme bit dans RX9D du registre RCSTA.

Le registre de contrôle permet de :

- Fixer le format de transmission (7 ou 8 bits).
- Fixer le facteur de division de l'horloge.
- Fixer le test de parité.
- Fixer le nombre de bits STOP.
- Préciser le fonctionnement en interruption.

Le registre d'état permet de savoir :

- Si une transmission est en cours.
- Si une réception est en cours.
- L'état des lignes de contrôle.
- L'état des interruptions.

II.13.3.2 Fonctionnement de la liaison série en mode transmission

La figure suivante nous indique le fonctionnement de l'EUSART en transmission :

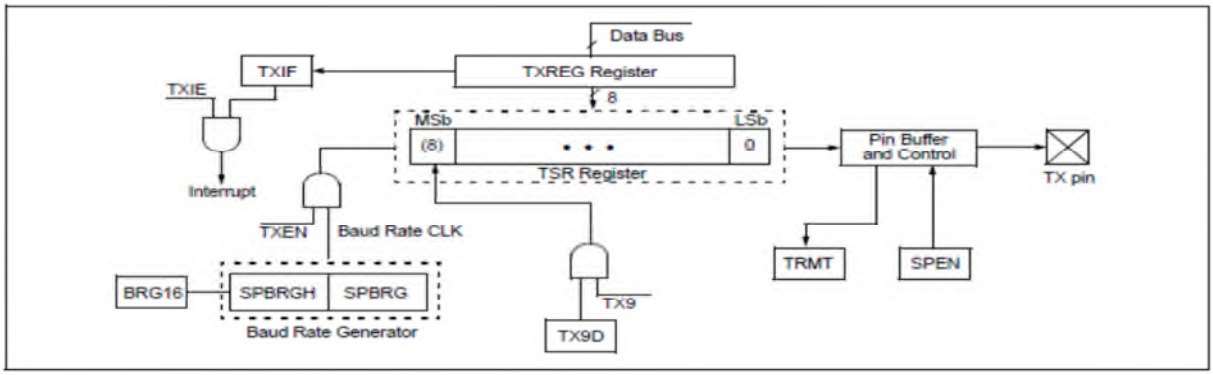


Figure II.11 Fonctionnements de l'EUSART en transmission.

Pour que deux équipements puissent échanger des données, il faut qu'ils soient configurés pour que :

- La vitesse de communication (baud rate) soit la même des deux côtés.
- Le nombre de bits de données soit identique.
- Le nombre de bits de STOP soit identique.
- Le type de contrôle de flux choisit soit le même.
- La parité soit la même.

I.14 Conclusion

En conclusion, dans ce chapitre nous pouvons dire que le microcontrôleur 18f4520 peut bien jouer le rôle d'une unité de contrôle pour notre système.

Sachant que l'évolution du niveau d'eau dans la bâche à eau est très importante nous pouvons dire que les Timers intégrés dans le PIC nous conviennent parfaitement.

Maintenant, nous pouvons passer à l'étude détaillée de la carte de développement PICPLC16.

Chapitre III

La carte PICPLC16 v6

III.1 Introduction

PICPLC16 V6 est une carte de développement industriel à base de PIC. Elle permet aux PICs d'être connectés avec des circuits externes. Le raccordement entre la carte et les circuits externes est établi à l'aide des relais. La carte contient aussi des modules de communication qui lui permettent de communiquer avec des dispositifs extérieurs comme le RS-232, le contrôleur d'Ethernet, et le module GSM.

III.2 Description de la carte PICPLC16 v6

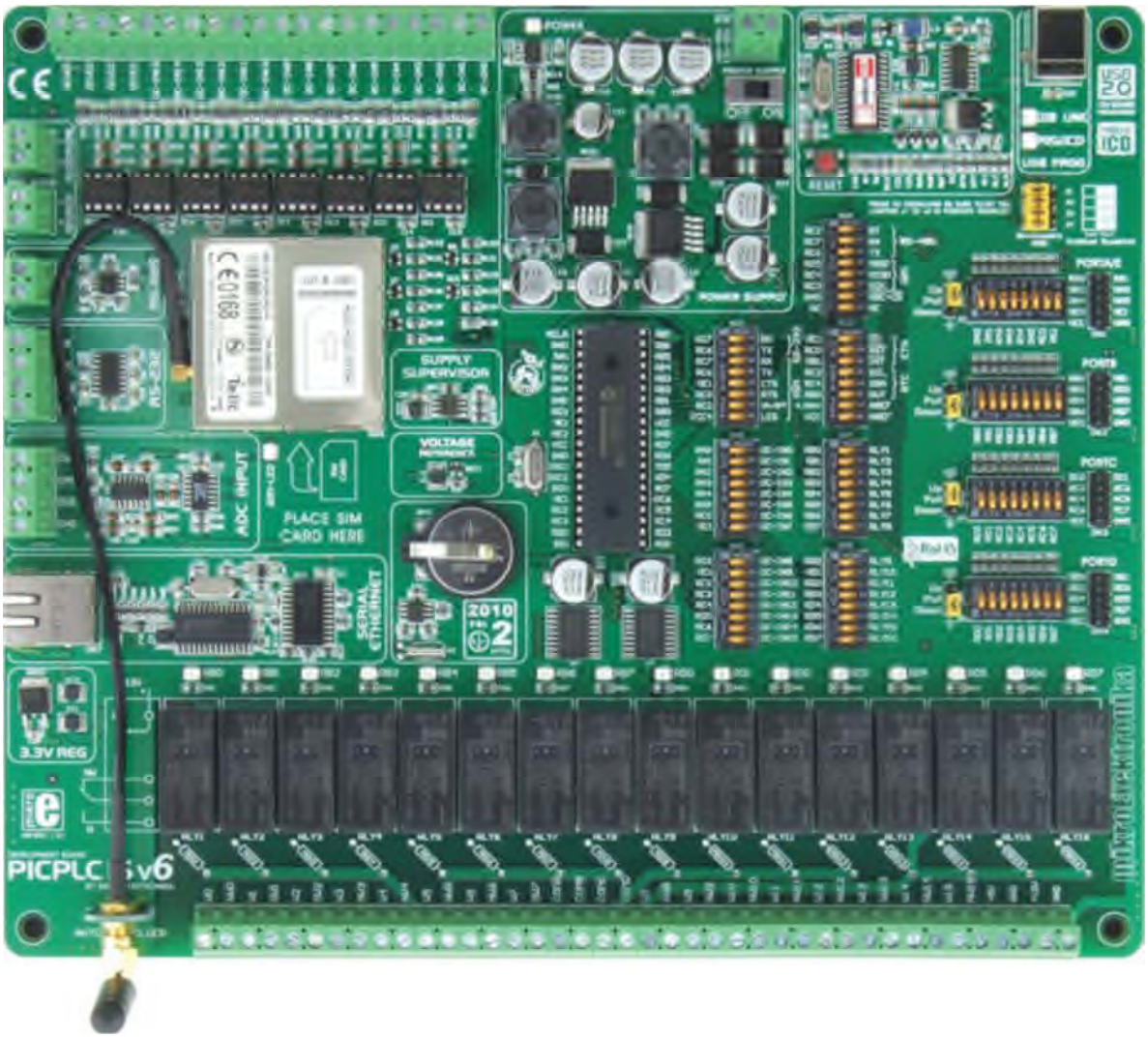


Figure III.1 La carte de développement PICPLC16 v6.

La carte est composée principalement [9]:

- Des optocoupleurs.
- Des relais.
- Un microcontrôleur pic18F4520.
- Une horloge temps réel (RTC).
- Un module Ethernet.
- Un module de communication RS-232.
- Un module de communication RS-485.
- Un régulateur de tension d'alimentation électrique.
- Un programmateur USB 2.0 avec mikroICD.

III.3 Le microcontrôleur

La carte de développement PICPLC16v6 comporte un PIC18f4520 de 40 pins dans le paquet DIP40 comme il est montré sur la figure suivante :

Figure III.2 Le PIC18F4520 en DIP40.

III.4 Le programmateur PICfalsh USB 2.0 avec mikroICD

Il n'y a aucun besoin d'utilisation d'équipement externe pendant la programmation du microcontrôleur, car le système de développement PICPLC16 a son propre programmeur qui est PICflash doté de la technologie mikroICD. Ce programmeur nous permet d'établir la connexion entre le microcontrôleur et le PC afin de charger un fichier (.HEX) dans le microcontrôleur. La figure III.3 Présente le lien entre le compilateur, le programmateur et le microcontrôleur.

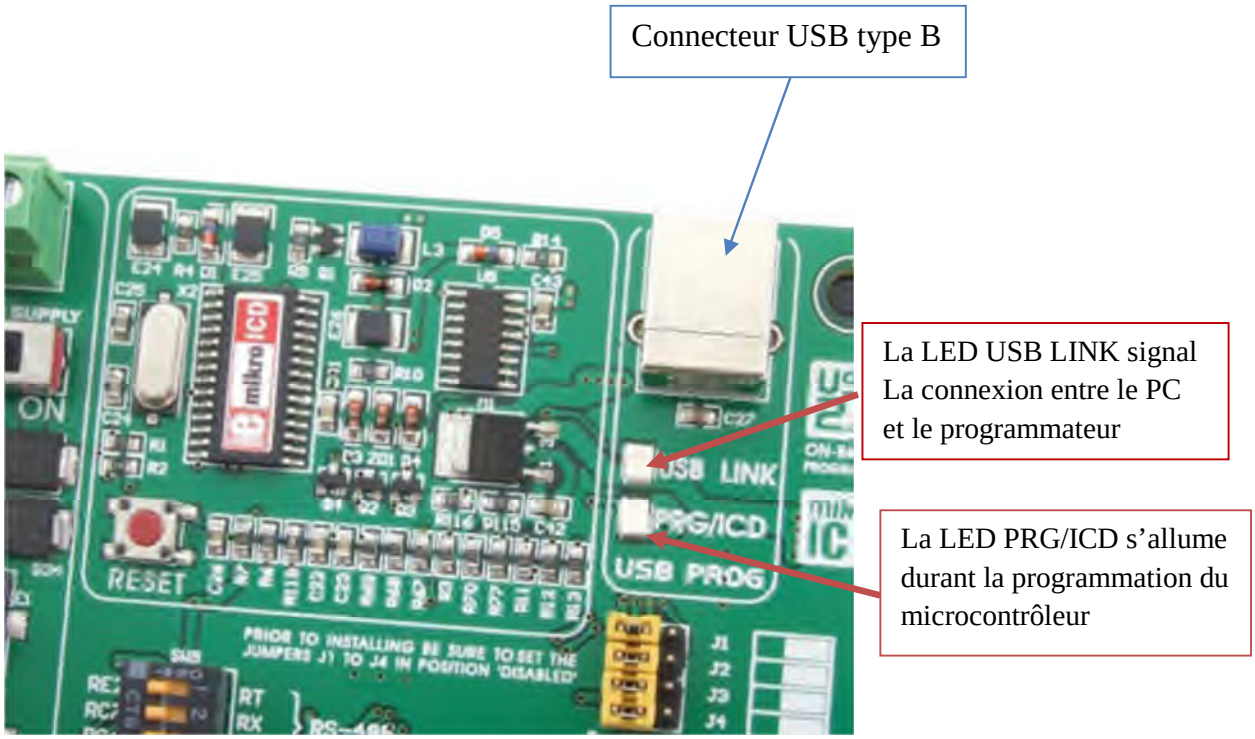


Figure III.3 Le programmateur PICflash.

III.5 Alimentation d'énergie

Le système de développement de PICPLC16 v6 est relié à la source d'alimentation d'énergie par l'intermédiaire du connecteur CN1. La tension d'alimentation électrique peut être C.C ou C.A. Une tension d'alimentation en courant continu peut être dans la gamme de 16V à 30V, tandis que la tension d'alimentation de courant alternatif peut s'étendre entre 12V et 22V. A savoir que le programmateur à bord ne peut pas actionner sans être relié à la source d'alimentation d'énergie bien qu'il soit relié à un PC par l'intermédiaire du câble d'USB.

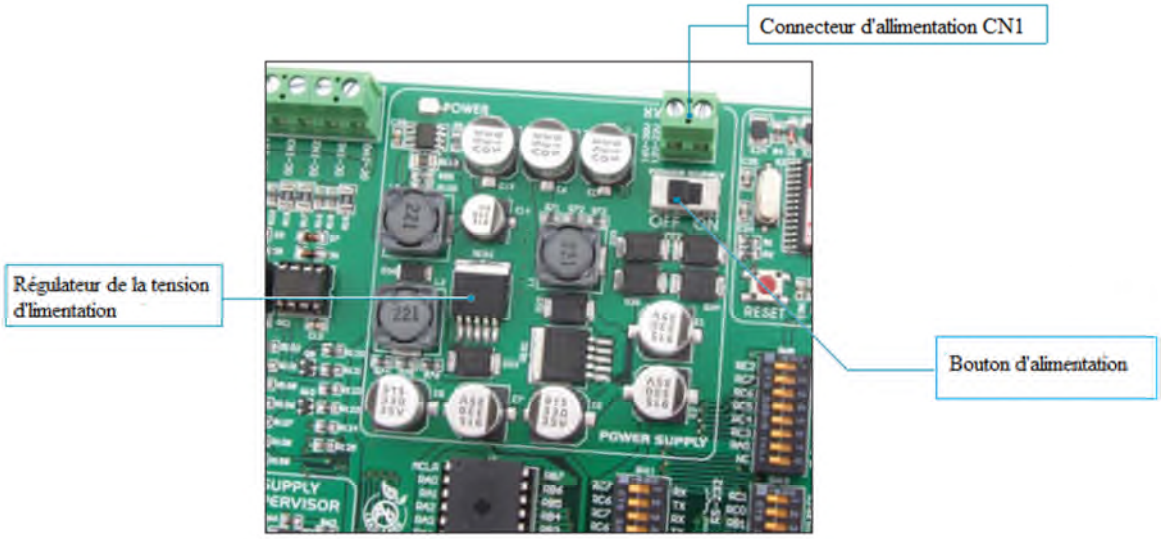


Figure III.4 L'alimentation d'énergie.

III.6 Interface de communication RS-232

La communication RS-232 permet le transfert des données point par point. Elle est utilisée généralement dans des demandes par acquisition de données, de transfert des données entre le microcontrôleur et une interface. Les niveaux de tension d'un microcontrôleur et d'un PC ne sont pas directement compatibles les uns avec les autres, un amortisseur de niveau de transition tel que le MAX232 doit être employé.

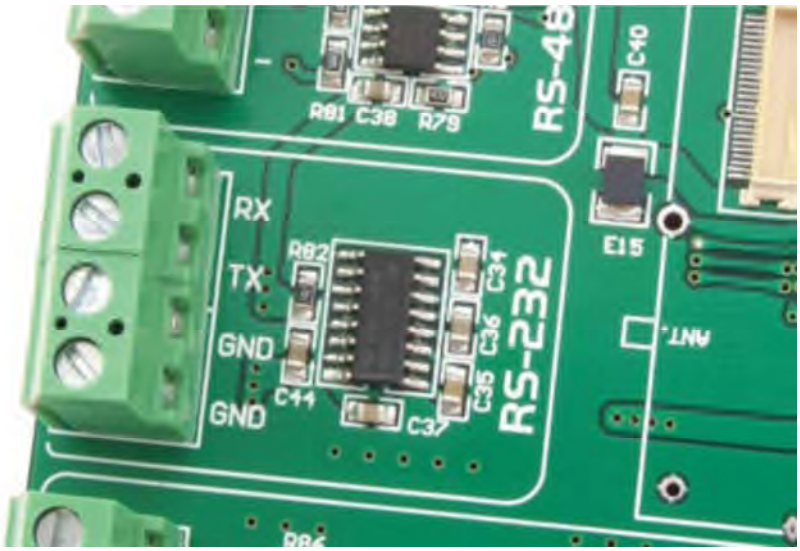


Figure III.5 Module RS-232.

III.7 Communication RS-485

La communication RS-485 permet le transfert de données point par point et point à-multipoint. Elle est utilisée généralement pour le transfert de données entre plusieurs microcontrôleurs.

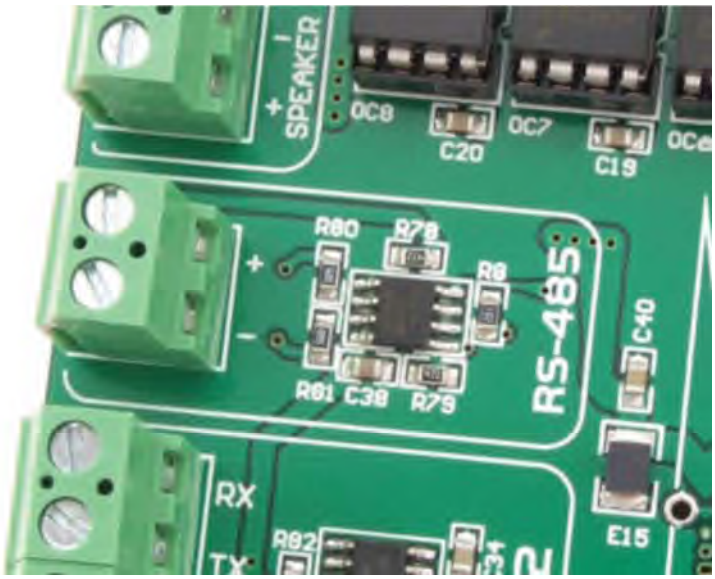


Figure III.6 Module RS-485

III.8 Le module Ethernet

La carte PICPLC16 V6 contient un module Ethernet qui lui permet de se connecter avec le réseau local (LAN). Le contrôleur ENC28J60 permet la communication d'Ethernet sur la carte. Il est employé pour transférer des données à partir de réseau LAN vers le microcontrôleur. La tension 3.3 V est exigée pour l'opération de ce contrôleur.

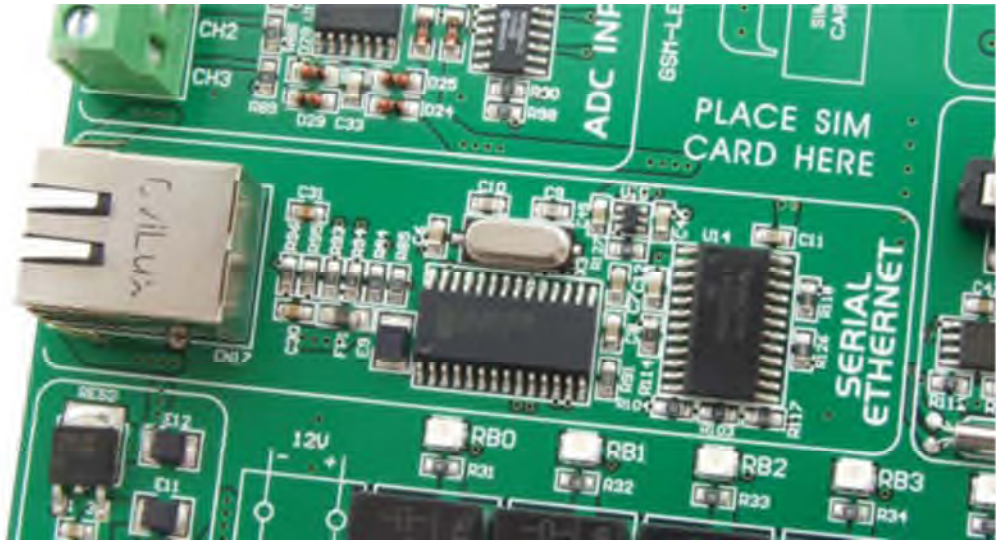


Figure III.7 Module Ethernet.

III.9 Le connecteur GSM

La carte PICPLC16 v6 est capable d'établir une communication avec le monde extérieur en utilisant le réseau GSM. Elle contient un connecteur intégré pour le module GSM, ce module comporte une fente pour placer une carte de SIM aussi bien qu'un connecteur pour l'antenne externe.

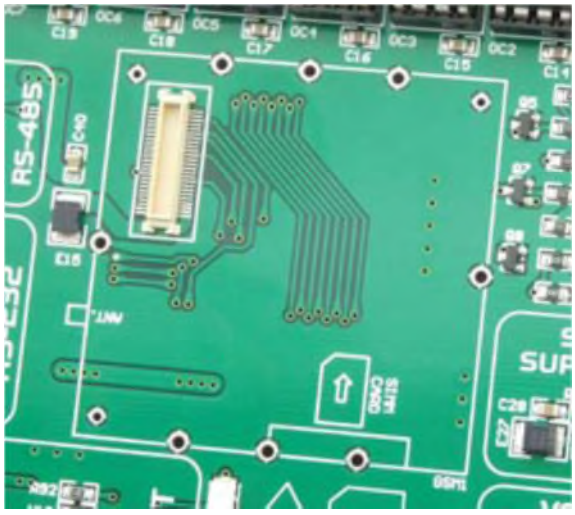


Figure III.8 Le connecteur GSM.

III.10 Horloge temps réel (RTC)

Le système de développement PICPLC16v6 peut conserver l'heure et la date exacte. Les caractéristiques de l'horloge temps réel sont les suivantes :

- Fournis les informations concernant les secondes, minutes, heures, jours, incluant les corrections en cas d'années bissextiles.
- Détection automatique des coupures d'alimentation.
- Consommation inférieure à 500nA.

L'horloge temps réel est très utilisée dans les systèmes d'alarme, contrôleur industriel et produits de grande consommation, etc. Celle fournie avec PICPLC16v6 est utilisée pour générer une interruption à une heure prédéfinie.

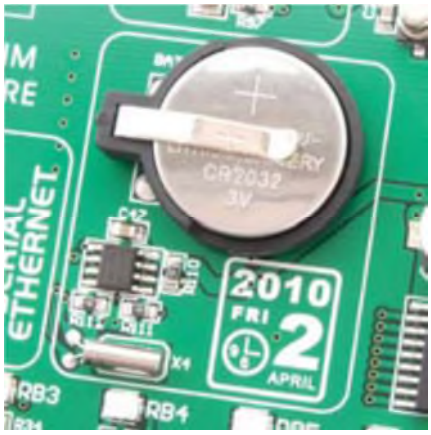


Figure III.9 Horloge temps réel.

III.11 Les optocoupleurs

PICPLC16v6 a 16 entrées optocoupleurs. Il est employé couramment dans des applications industrielles où des entrées qui doivent être électriquement isolé dans le reste de la carte de développement. Il sert à protéger le microcontrôleur contre les impulsions électriques qui pourraient se produire sur des lignes d'entrée.

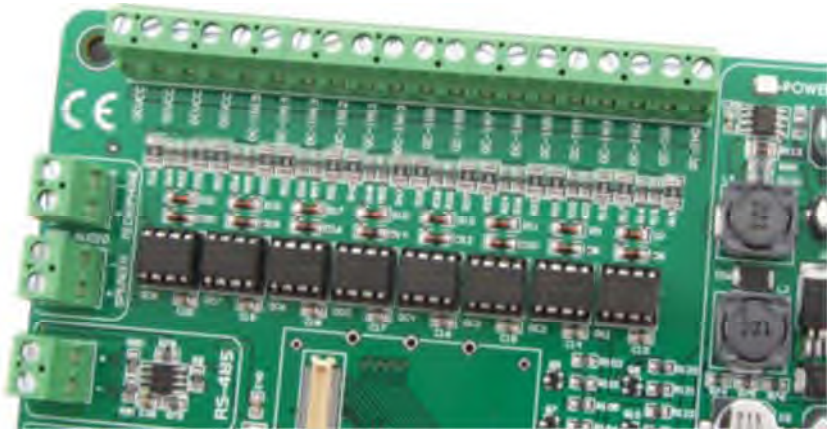


Figure III.10 Les optocoupleurs.

III.12 Les Relais

Les dispositifs industriels utilisent habituellement des puissances plus fortes que le microcontrôleur peut fournir par l'intermédiaire de ses ports d'entrée-sortie. Pour permettre au microcontrôleur d'être relié à des tels dispositifs, le système de développement est équipé de 16 relais à l'aide desquels il est possible de fournir jusqu'à 250V.

Figure III.11 Les relais.

III.13 Les ports d'entrées/sorties

Sur le côté droit de la carte de développement se trouvent 4 connecteurs mâles de 10 broches reliés aux ports d'E/S du microcontrôleur. Les broches du microcontrôleur ne sont pas directement reliées au connecteur, mais par l'intermédiaire des jumpers, les interrupteurs à position multiple SW1-SW4 servent à relier les broches aux résistances de tirage pull-up/pull-Down. La position des jumpers J5-J8 détermine si les ports utilisent une résistance de pull-up ou pull-down.

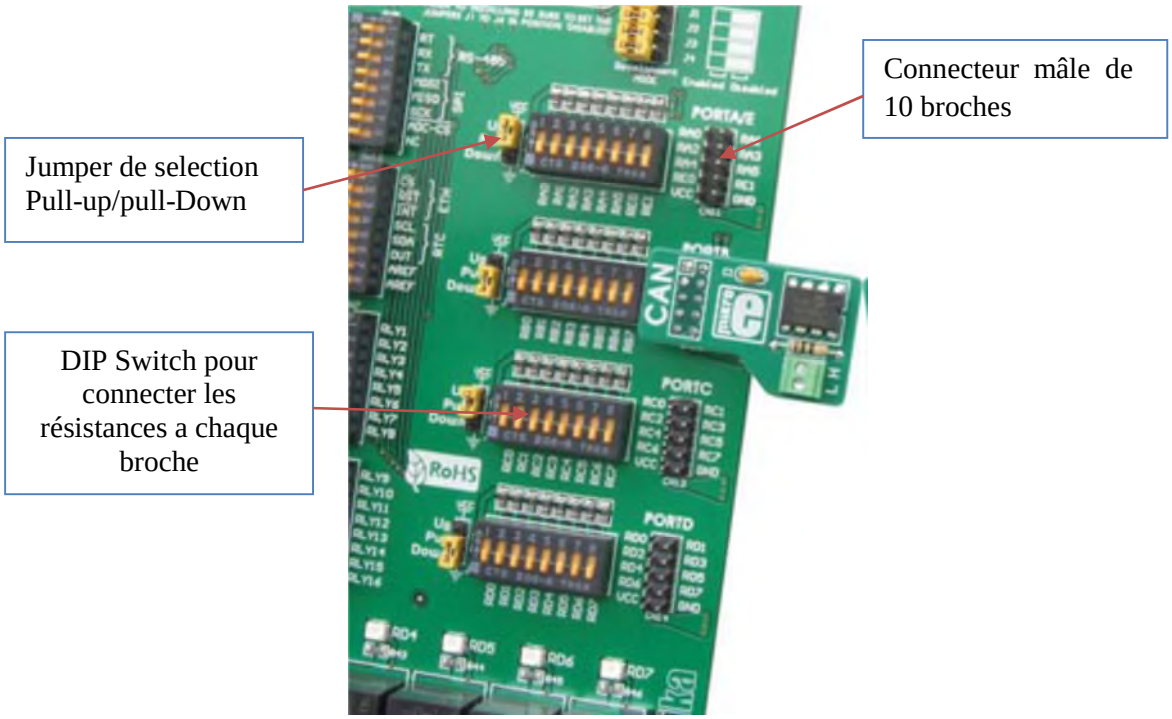


Figure III.12 Ports d'entrées/sorties.

III.14 Conclusion

Après que nous avons fait l'étude de la carte de développement, on peut dire que c'est un outil important pour le développement et la programmation des microcontrôleurs, vu le nombre important de modules intégrés et prêts à l'emploi et la capacité de la mémoire du microcontrôleur la rend très utile dans le domaine de l'industrie.

Chapitre IV

Programmation et test de la carte

IV.1 Introduction

Ce chapitre comporte deux parties. La première partie concerne les organigrammes de l'application et le programme à établir avec le langage **C** dans l'outil de développement **MPLAB** puis la simulation du programme dans **ISIS PROTEUS**. La seconde partie concerne le chargement du programme dans la carte de développement **PICPLC16 v6** et le test pratique de notre programme sur la carte.

IV.2 Description du système

Notre système est constitué de deux pompes, une bâche eau, et des capteurs de niveau comme la figure ci-dessous la montre :

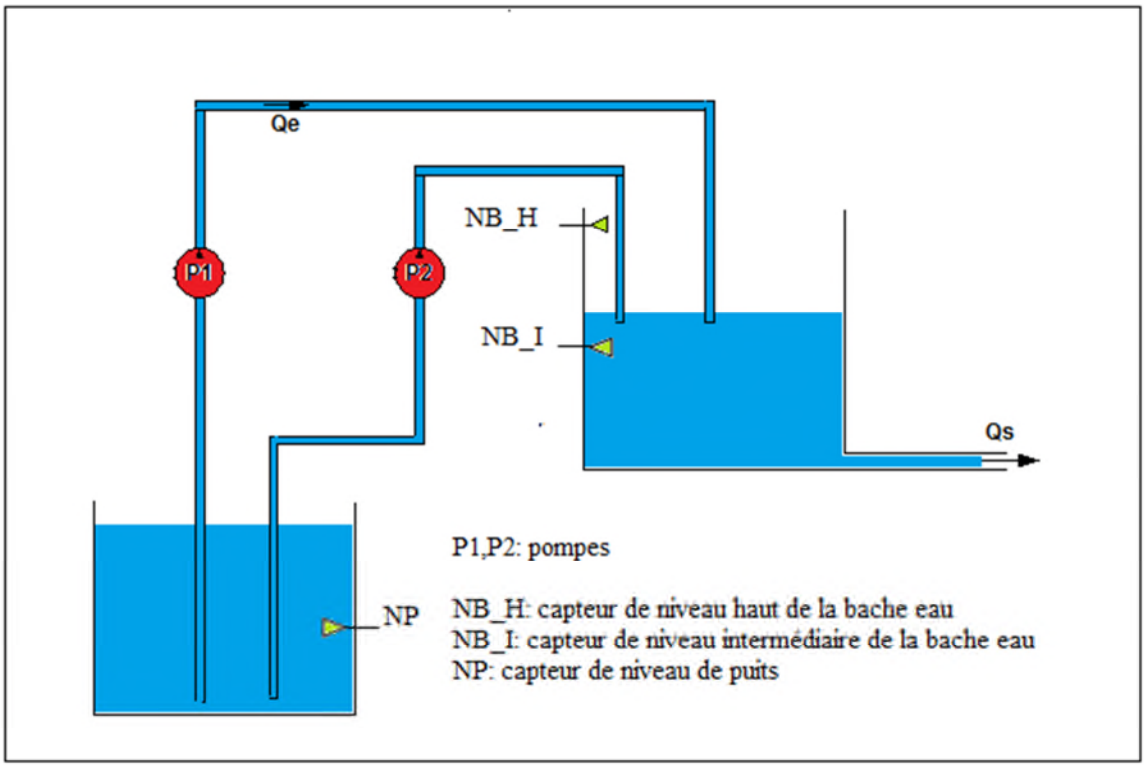


Figure IV.1 Description du système.

IV.2.1 Les pompes

Pour notre cas, on utilise les pompes centrifuges qui ont des performances variables ce qui leurs confèrent des utilisations diverses :

- Transfert de fluide (débit important).
- Transmissions de puissances (pression importante).

IV.2.2 Les capteurs

Un capteur est un organe charger de prélever une grandeur physique à mesurer (position, vitesse, température, …etc.) et de la transformer en une grandeur exploitable par la partie commande.

Les informations de sortie d'un capteur peuvent être de caractère analogique, logique ou numérique. Pour notre système on a utilisé des capteurs de niveau logique (TOR).

IV.3 Cahier des charges

Notre travail consiste à établir un programme afin de commander deux pompes en fonction des niveaux de la bâche à eau et le niveau du puits. Notre organigramme sera muni des conditions suivantes :

- Si NP=0 ou NB_H=1 alors P1=0 et P2=0 (les deux pompes sont éteintes).
- Si NB_I=0 alors P1=1 et P2=1(les deux pompes sont en marches).
- Si NB_I=1 alors P1 fonctionne durant un temps T et s'arrête puis P2 fonctionne durant un temps T (en alternance).

IV.4 Programmation du PIC 18F4520

La programmation consiste à définir une suite d'ordres qui font partie du jeu d'instructions du PIC utilisé pour répondre au cahier des charges.

Microchip propose gratuitement l'outil de développement MPLAB qui regroupe l'éditeur de texte, le compilateur C18, un outil de simulation et le logiciel de programmation.

Avant la construction d'un programme, il est recommandé de réaliser un organigramme qui représente le cheminement du programme à écrire. Cela va faciliter la programmation.

IV.5 Structure du programme à implanter dans le microcontrôleur

Le programme à implémenter dans la mémoire du PIC est composé de trois groupes de routine. Le premier est le programme principal, le deuxième est le groupe de routines appelé à partir du programme principal, le troisième groupe est la routine d'interruption. Cette routine doit être placée à l'adresse 0x008.

IV.5.1 Programme principal

Selon l'organigramme principale, notre programme commence par la partie de la configuration tel que :

- La configuration des ports :

Pour les entrées (NP, NB_I, NB_H) qui sont des entrées de type TOR pour notre cas, on a utilisé les pins RB4, RB5, RB6 du PORTB car il peut générer des interruptions qu'on exploitera dans notre programme. Pour les sorties (P1, P2) qui commande la pompe1 et pompe2, on utilisera les pins RD0 et RD1 de PORTD, mais si on utilise le PORTA il faut désactiver la fonction de conversion A/D en chargent les valeurs adéquates dans les registres ADCON0 et ADCON1.

- La configuration des interruptions

Tout d'abord il faut autoriser l'interruption global (GIE=1) et désactiver la priorité des interruptions (IPEN=0). Comme on a utilisé le PORTB en mode d'interruption, on autorise son interruption avec le bit RBIF du registre INTCON et on doit aussi activer l'interruption du timer0 avec le bit TMR0IF du même registre.

- La configuration de timer0

Pour la temporisation on a utilisé le timer0 en mode 16 bits, prédiviseur maximum (256), avec cette configuration on obtient une temporisation de 8 secondes avec une fréquence de 8 MHZ.

- La configuration du PORT RS232

Dans notre programme, il y a une partie ou on a configuré les paramètres pour la communication série pour relier le PIC à un PC via le PORT RS-232, pour pouvoir visualiser les états du système, on a donc configuré:

- La vitesse de communication 9600 baud (BRGH=0, BRG16=0 et SPBRG=12).
- Le nombre de bits des données (sur 8 bits).
- Le fonctionnement en mode de transmission (TXEN=1).
- Indiquer que les pattes R6 et R7 de PORTC sert à la liaison série (SPEN=1).

Après la configuration des ports et des interruptions on passe a la partie d'initialisation qui contient les différentes variables, leurs valeurs initiales et les sous programmes.

IV.5.2 Routine principale

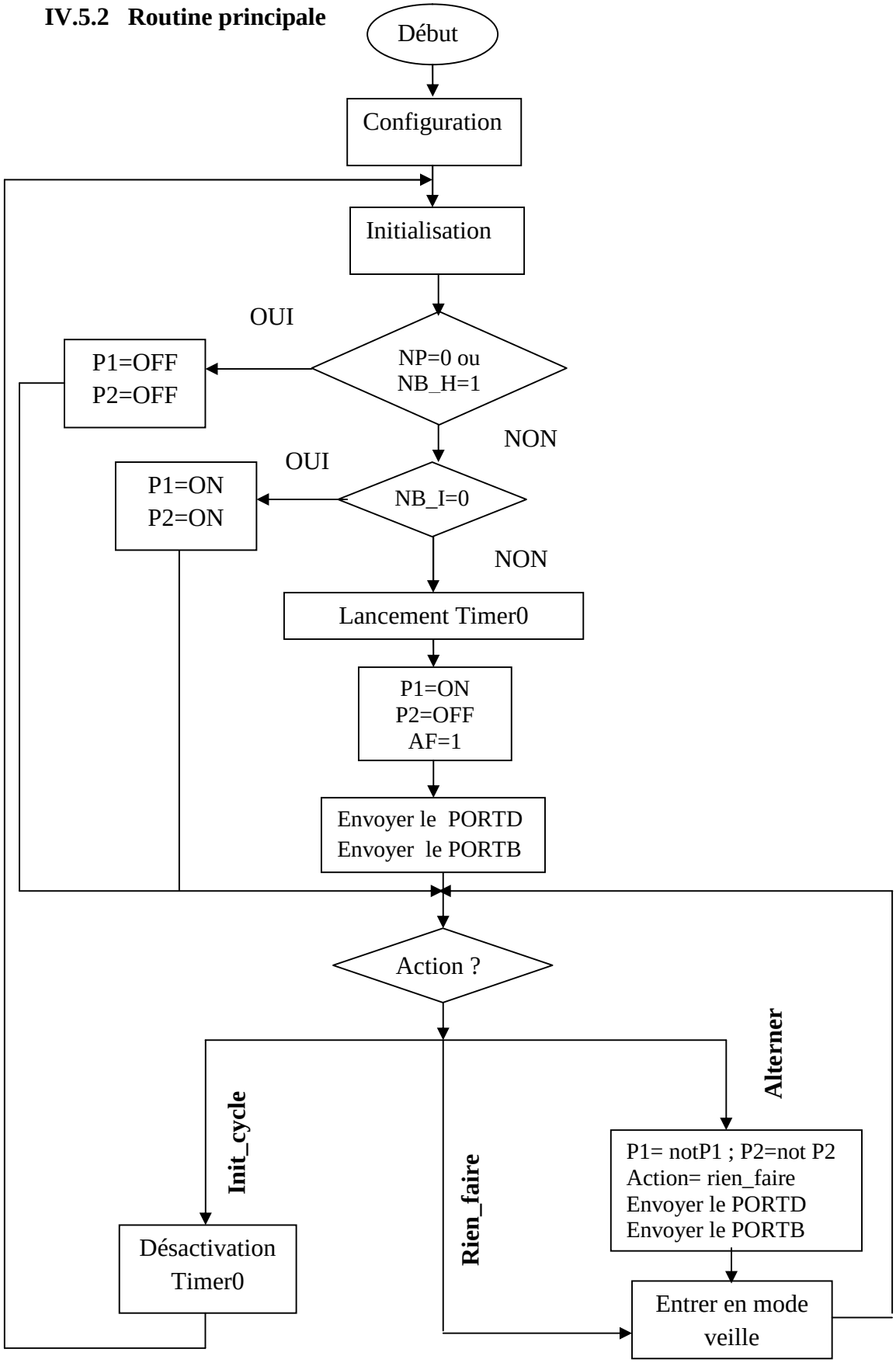


Figure IV.2 L'organigramme principal.

IV.5.3 Routine d'interruption

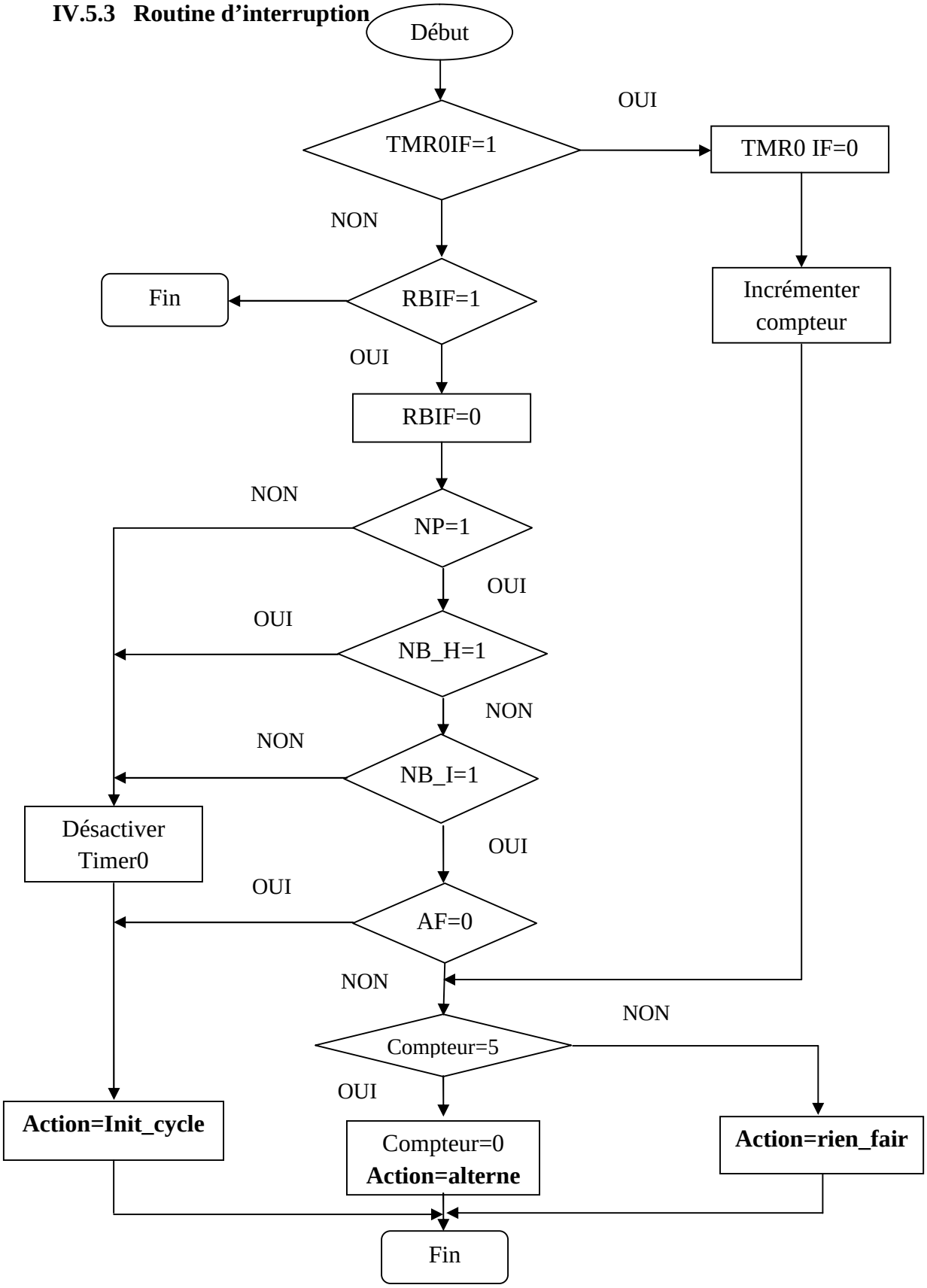


Figure IV.3 L'organigramme de la routine d'interruption.

IV.6 Programmation et simulation

IV.6.1 La programmation dans le MPLAB

Les étapes nécessaires pour avoir un programme qui s'exécute sur un PIC sont :

- L'écriture du programme en langage C et le sauvegarder avec l'extension .C

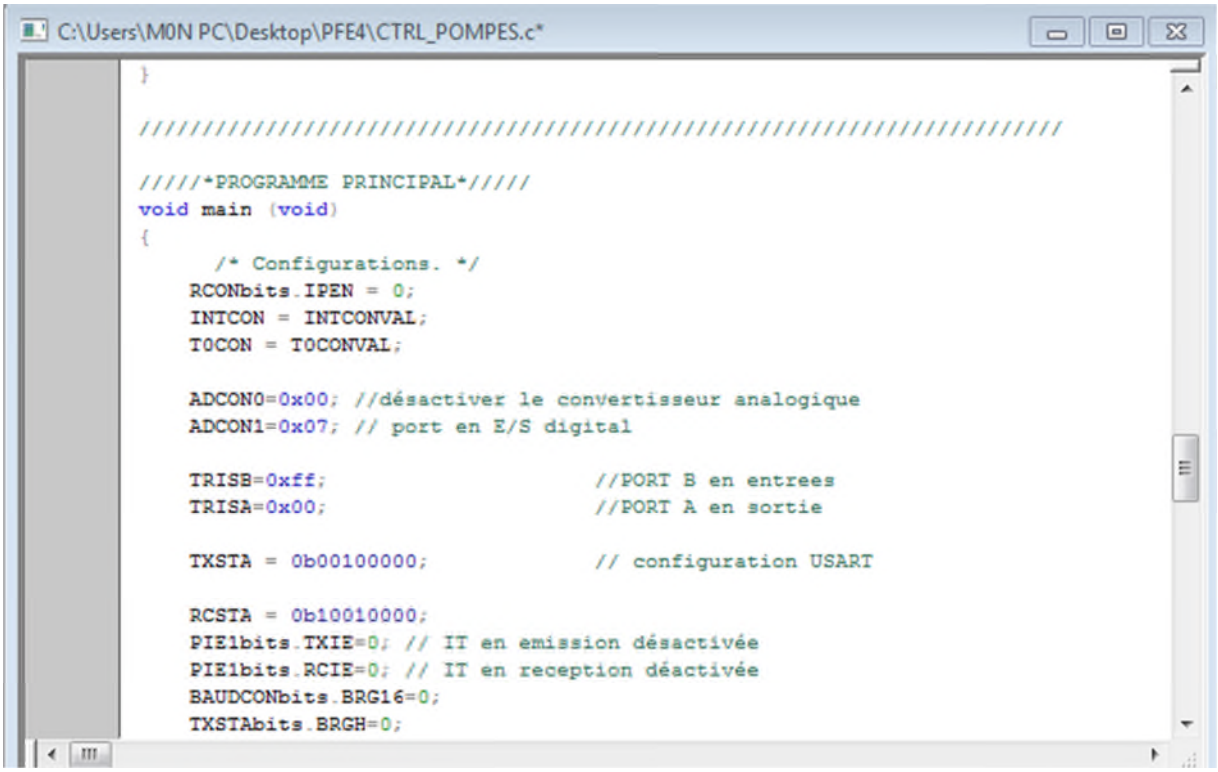


Figure IV.4 Fenêtre de fichier .C.

- Après l'écriture du programme on passe à la compilation. La figure IV.5 montre le succès de la compilation et un fichier d'extension (.hex) est généré.

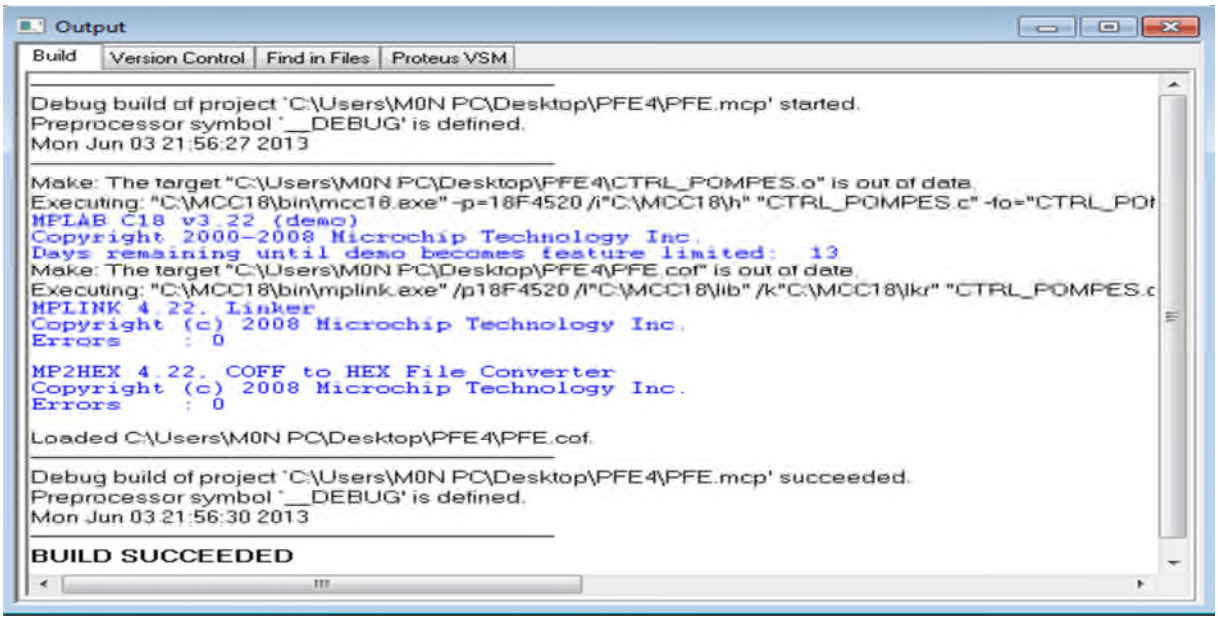


Figure IV.5 Compilation du programme.

IV.6.2 La simulation dans l'ISIS PROTEUS

Après la compilation du programme sans erreurs, on passe à la simulation dans l'ISIS proteus. La figure ci-dessous montre le schéma de simulation du programme sur l'ISIS.

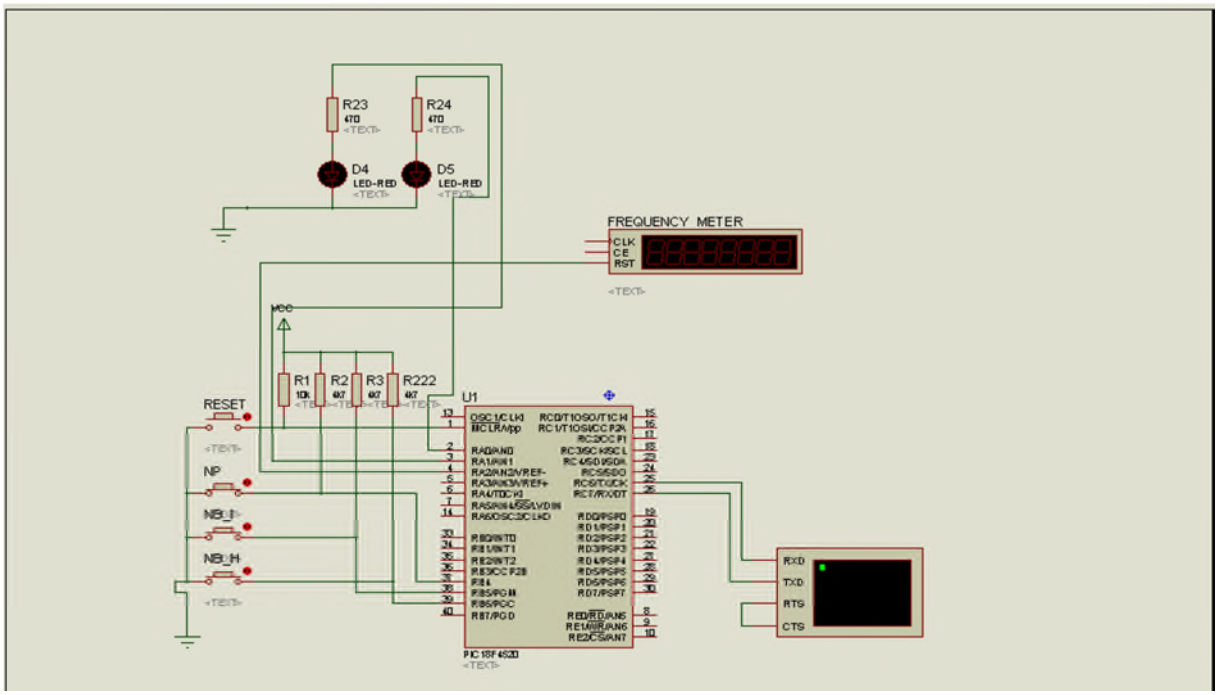


Figure IV.6 Le schéma de simulation dans l'ISIS.

Pour voir le bon fonctionnement du programme qu'on a réalisé sous MPLAB, on va faire quelque test sur l'ISIS avant de charger le programme dans le PIC.

Test n°1 : la mise en marche des deux pompes

On met les boutons NP=1, NB_H=0 et NB_I =0

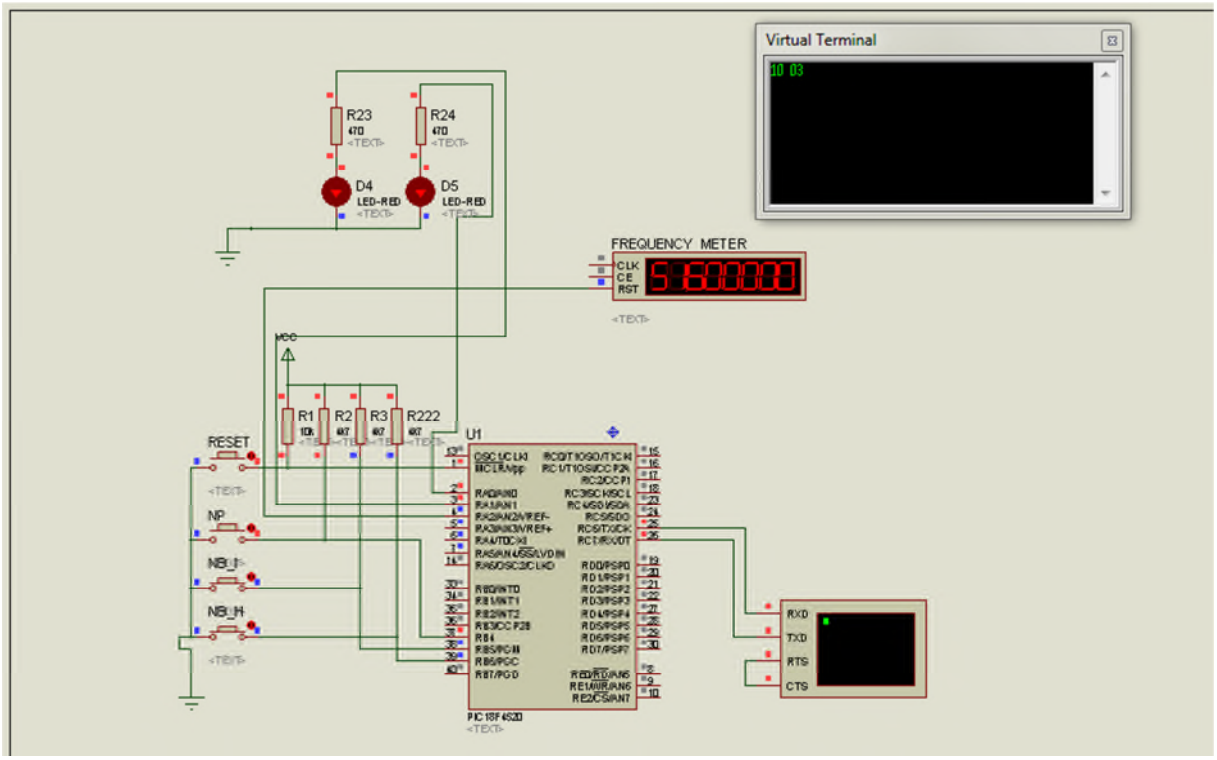


Figure IV.7 La simulation du test N°1.

La simulation donne le résultat de la figure ci-dessus, on voit bien que les deux leds représentant les deux pompes sont allumées et l'affichage de Virtual Terminal confirme l'émission des états des ports A et B vers l'interface graphique, ce qui valide ce premier test.

Test N°2 : le fonctionnement des pompes en alternance

NP=1 et NB_I=1 et NB_H=0.

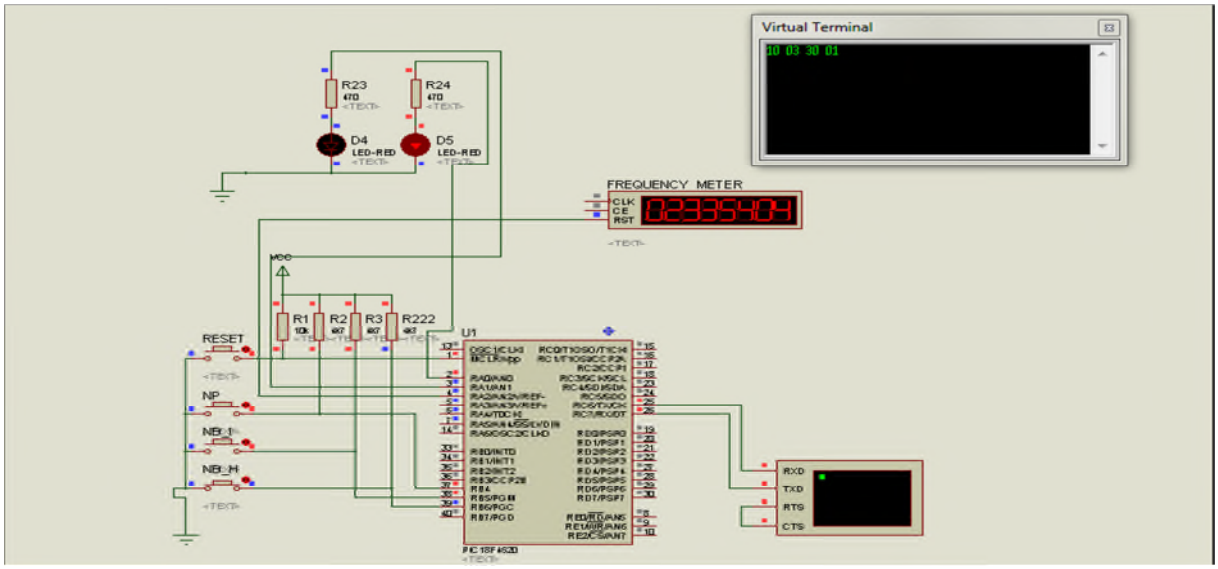


Figure IV.8 La simulation du test N°2.

Après une durée T=2min

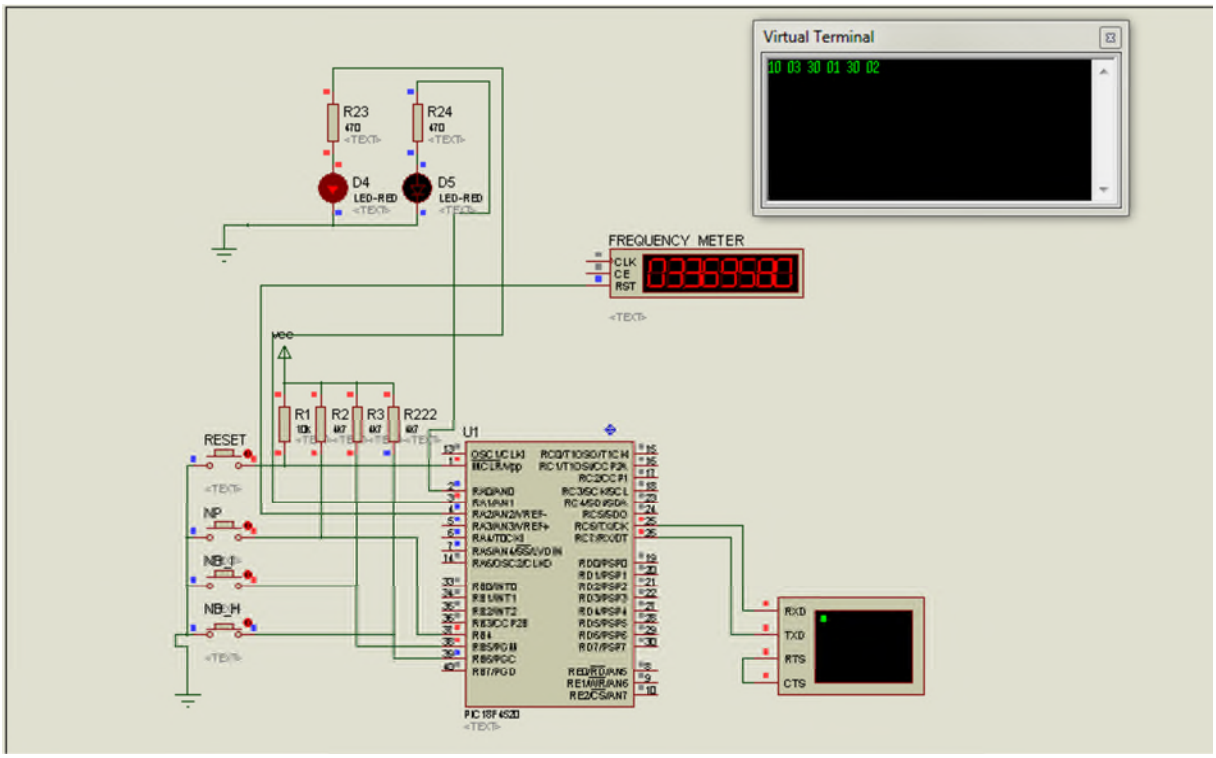


Figure IV.9 La simulation du test N°2 après 2 min.

Test N °3 : l'arrêt des pompes

Les leds qui représentes doivent être éteintes dans le cas ou la bâche eau (NB_H=1) est pleine ou il n'y a pas d'eau dans la source (NP=0).

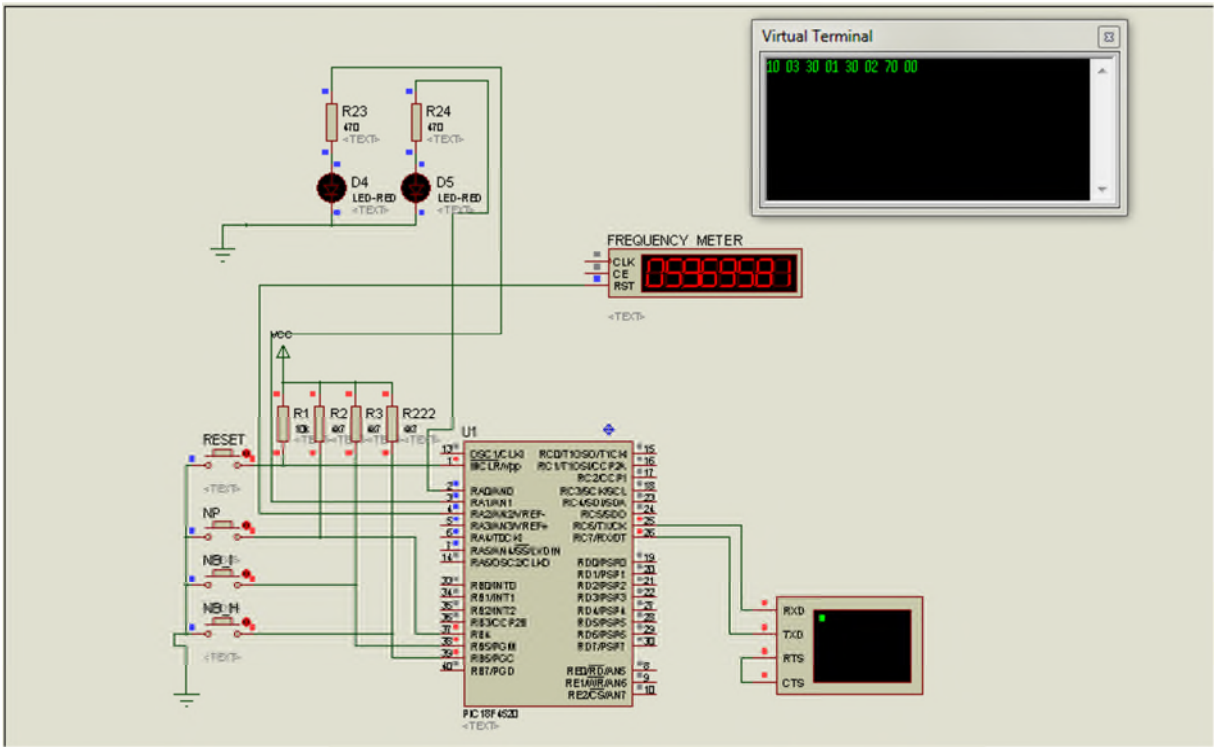


Figure IV.10 La simulation du test N°3.

Après les résultats de la simulation dans l'ISIS, le programme fonctionne selon l'organigramme établit, donc on peut passer au transfert du programme vers la mémoire du PIC et faire des tests sur la carte PIC plc16.

IV.7 Le test de la carte

La figure IV.11 montre le montage qu'on a réalisé afin de tester le fonctionnement de la carte. Dans cette partie on chargera d'abord le programme dans la mémoire du PIC puis on a effectuera les défirent test sur la carte.

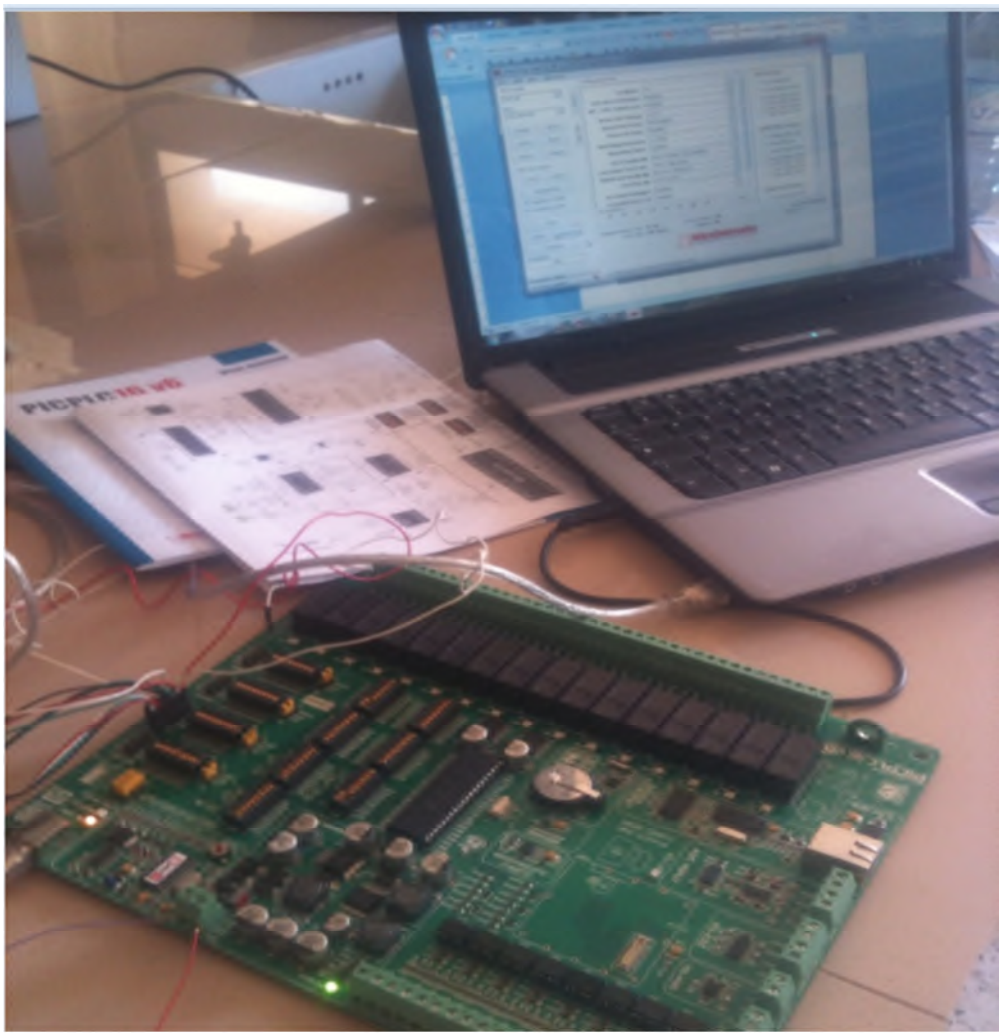


Figure IV.11 Réalisation du montage.

IV.7.1 Chargement de programme dans le PIC

Comme la carte PICPLC16 v6 a son propre programmateur qui est PICflash, alors on n'a pas besoin d'utiliser un programmateur externe. Pour charger le programme dans le PIC, on doit passer par les étapes suivantes :

Etape1 :

La Compilation du programme qu'on a écrit dans MPLAB pour générer le fichier (.hex).

Etape2 :

L'utilisation du programmateur PICflash pour sélectionner le microcontrôleur approprié, puis on clique sur load HEX pour charger le fichier (.hex) dans le programmateur et en cliquant sur le bouton write, on charge le code (.hex) dans le microcontrôleur.

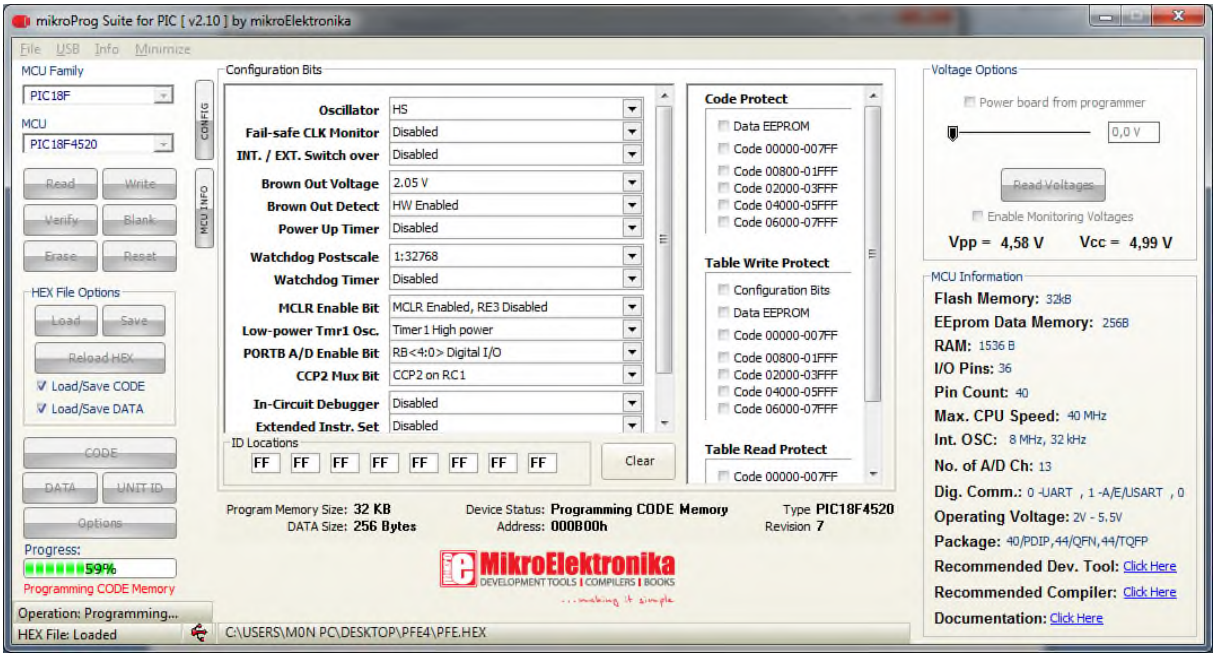


Figure IV.12 Editeur du programmateur PICflash.

Durant le chargement du programme, une led s'allume pour indiquer que le programme est entrain de se charger dans le PIC comme le montre la figure suivante :

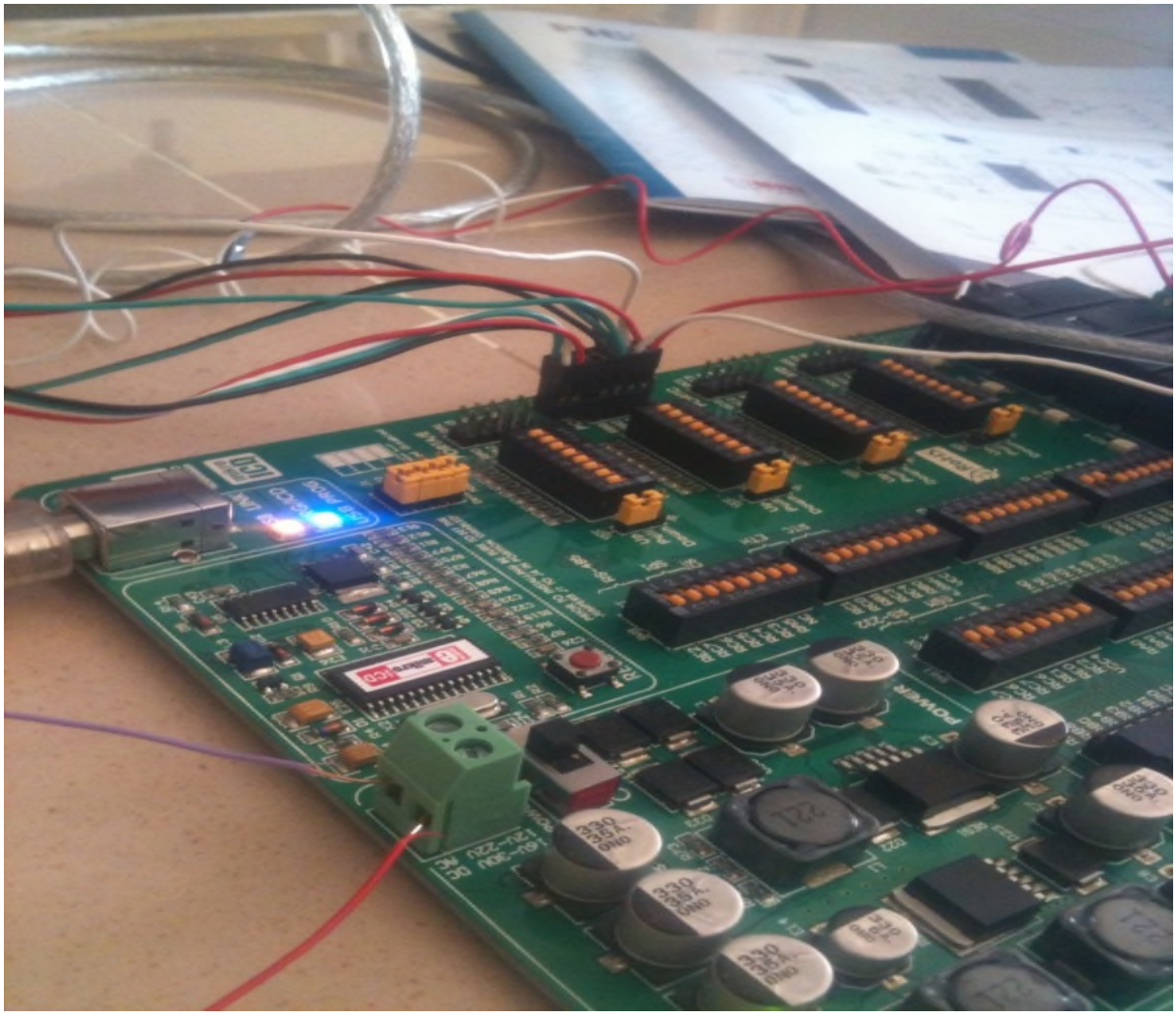


Figure IV.13 Chargement de programme.

IV.7.2 Test de fonctionnement de la carte

Après le succès du chargement de programme, nous avons fait des testes en mettant le port d'entrée (port B) à des niveaux 1 et 0 afin de voir si le résultat sur la carte est identique à celui de la simulation, en voyant l'allumage des leds reliées aux relais du port D (RD1 et RD2), on peut dire que les résultats sont satisfaisants.

IV.8 Interface de communication homme/machine

Chaque processus industriel comporte une interface de communication homme/machine permettant de piloter les modes de marche (marche, arrêt, auto, manuel, pas à pas...) et d'assurer la surveillance de l'état du processus.

Selon la nature du processus (manufacturier, batch, continu) et sa complexité, l'interface est plus ou moins sophistiquée et l'on peut considérer que les processus les plus exigeants en matière de communication homme-machine sont les procédés batch [8].

L'élément de base de l'interface de communication homme machine est une console opérateur comportant un écran graphique couleur, un clavier et une imprimante.

Dans cette partie, nous allons proposés un modèle d'interface homme/machine réalisée à l'aide de GUI de MATLAB pour la supervision du système.
L'interface réalisée est sur la figure suivante :

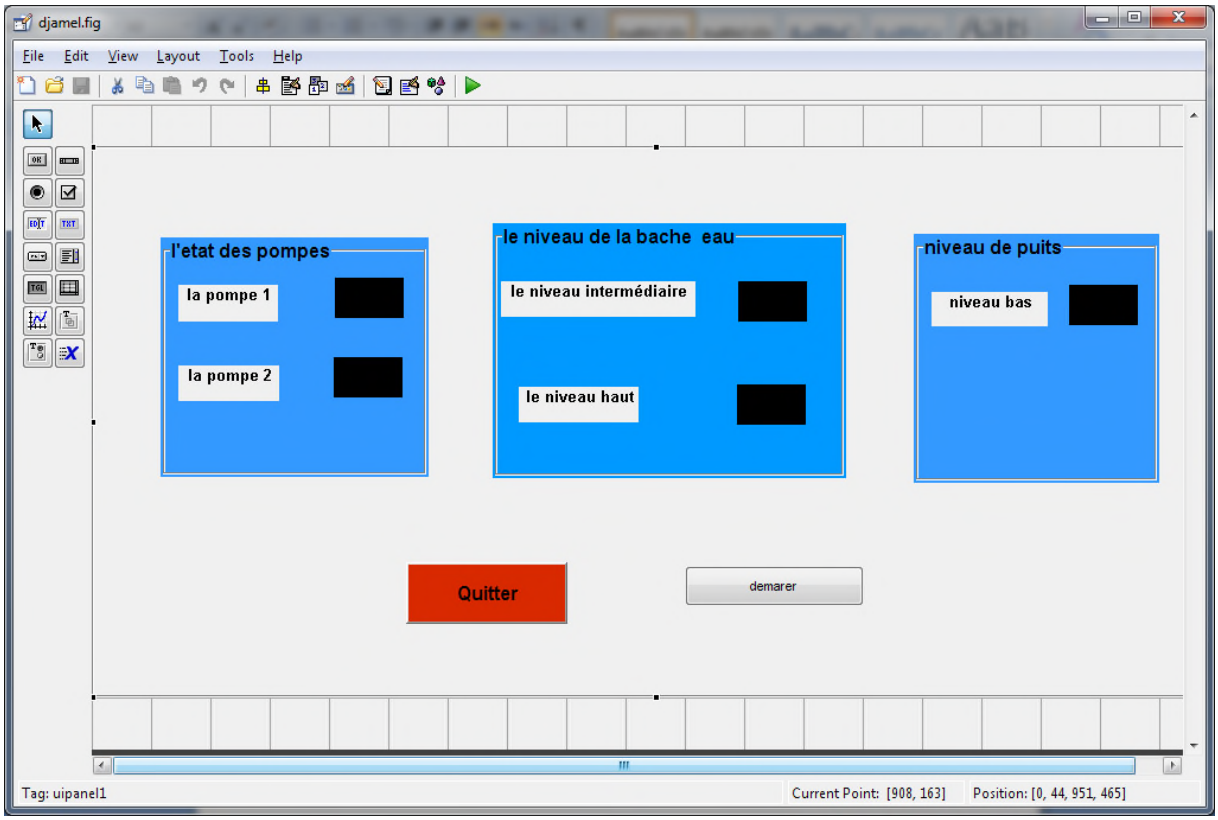


Figure V.14 Interface homme/machine.

IV.9 Conclusion

Dans ce dernier chapitre, nous avons fait une description de notre système, puis on a établis un programme selon le cahier des charges après avoir dresser l'organigramme général de fonctionnement de notre système. Le transfert du programme et le test sur la carte étaient un succès et ce résultat est le même que celui de la simulation sur l'ISIS. Finalement, on peut dire que le fonctionnement de notre programme est satisfaisant.

Conclusion générale

Conclusion Générale

Dans tout système de production automatisé, le bon choix de l'unité de traitement est fondamentale vue son rôle très important dans la gestion des entrées sorties du processus industriel.

L'objectif de notre travail était d'établir un programme pour commander un système de pompage et l'étude de la carte de développement PICPLC16v6.

Afin de mettre en clair le fonctionnement de notre système, nous avons dressé des organigrammes qui représentent le déroulement de toutes les étapes du processus. Le programme est réalisé à l'aide du logiciel MPLAB avec le langage C et après la simulation, les résultats obtenus sont satisfaisants.

L'étude de la carte nous a permis de mieux connaitre ses composants afin de bien exploiter les éléments adéquats pour notre application afin de tester notre application en utilisant le programmateur intégré.

Des améliorations peuvent être apportées pour rendre le processus de supervision efficace, on parlera alors de la télé-supervision qui facilitera la supervision et le contrôle des systèmes à distance et permettra aux utilisateurs la consultation du fonctionnement du système en ayant l'accès à l'interface à distance. Actuellement, le développement de la télécommunication a pris part dans la communication entre les humains, mais dans le domaine de l'industrie c'est une nouvelle méthode de contrôle et de commande des systèmes. La carte que nous avons utilisée dispose d'un connecteur pour le module GSM qui rendra la supervision plus rapide et efficace à chaque fois qu'une anomalie du fonctionnement du système se présente.

Cette expérience nous a été bénéfique sur plusieurs plans, sa nous a permit de nous familiariser avec les PICs et nous initié au logiciel de programmation et l'utilisation d'une carte nous a permis d'avoir un aperçu sur ses composants et comment tester le programme.

En fin, nous souhaitons que se travail soit une référence pour les futur ingénieurs.

Référence bibliographique

Bibliographie

[1] D.MERAT et J.C.BOSSY, «AUTOMATISME APPLIQUE **»,** Edition ANDRE CASTEILLA, Paris 1985.

[2] A.SIMON, «AUTOMATES PROGRAMMABLES, Programmation Automatisme & Logique programmée**»**, Edition L'ELAN 1983.

[3] C.TRAVENIER «les microcontrôleurs PIC, Description et mise en œuvre**»,** Edition DUNOD, Paris 2002.

[4] BIGONOFF, **«** LA PROGRAMMATION DES PICS**»**, PREMIERE PARTIE, PIC16F84, Révision 6.

[5] M.KACIMI et K. KACI « Le microcontrôleur 16F877 Etude et application », mémoire de DEUA en électronique, université de Bejaia, promotion2008.

[6] C.TAVERNIER, **«** les microcontrôleurs PIC recueil d'application**»**, édition DUNOD 3ème édition 2005.

[7] JL.AMALBERTI, PA.DEGRYSE, O.DELEAGE, D.FREY et JP.GUIRAMAND, « Informatique industriel: le PIC18F4520 et sa programmation en C**»,** Institut universitaire de technologie 1 de Grenoble.

[8] J. LE GALLAIS, « Technique de l'ingénieur, Traité informatique industriel**»,** R 7505.

[9] www.picplc16_v6_manual_v100.com

[10] www.DatasheetPIC18F4520.com

Résumé

L’objectif principal de ce travail est d’automatisé un système de pompage au sein de l’entreprise Général Emballage. La solution technologique qu’on a opté pour atteindre cet objectif, est d’utilisé une carte de développement PICPLC16 v6 à base du PIC18F4520. Dans ce contexte, nous avons dressé des organigrammes qui représentent le déroulement de toutes les étapes du processus selon le cahier des charges définit auparavant. Le programme est réalisé à l’aide du logiciel MPLAB avec le langage C, la simulation dans l’ISIS PROTEUS confirme le bon fonctionnement du programme. Enfin, le transfert du programme vers le PIC et le test de la carte valident le travail réalisé.