

11/621.741

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8Mai 1945 – Guelma
Faculté des Sciences et de la Technologie
Département d'Electronique et Télécommunications



**Mémoire de Fin d'Etude
pour l'obtention du Diplôme de Master Académique**

**Domaine : Sciences et Techniques
Filière : Electronique
Spécialité : Systèmes Electroniques**

Commande d'un ascenseur par une logique programmée

Présenté par :
**Dorghal abd el wahid
bouchlagheme houssem eddine**

Sous la direction de :
DR Younsi azize



JUIN 2012

Remerciements



En premier lieu, nous tenons à remercier et à exprimer nos gratitudee à notre encadreur Dr younsi qui nous a orienté et nous a soutenu

Durant notre projet de fin d'étudee.

Nous témoignons reconnaissance et gratitudee aux aimablee personnee quelque soit de prés ou de loin, qui nous ont soutenu, acceptent et orientant durant notre projet de fin d'étude au sein de l'université de

08 mai 1954 Guelma et spécialement dans le département électronique/télécommunication.

Enfin, nous remercions vivement les membree de jury.

Table de matières

Introduction

Cahier de charge

1. sujet
2. description

Chapitre 1 : les ascenseurs

1. Introduction
2. Définitions
3. Applications
4. Les composants principaux d'un ascenseur
 - 4.1. La gaine
 - 4.2. Les guides
 - 4.3. La cabine
 - 4.4. Les commandes d'appel
 - 4.5. Les commandes d'envoi
 - 4.6. Les portes palières
 - 4.7. Le contrepoids
 - 4.8. Les câbles de traction
 - 4.9. Le treuil
 - 4.10. Le moteur
 - 4.11. Parachute
 - 4.12. Amortissement
5. La sécurité
6. Différentes technologies des ascenseurs
 - 6.1. Les ascenseurs hydrauliques
 - 6.2. Les ascenseurs à traction à câbles

Chapitre 2 : Différentes techniques pour commander un ascenseur

1. Différentes techniques de commande d'un ascenseur
2. Commande d'un ascenseur par une logique câblée :
 - 2.1. Avantages :
 - 2.2. Inconvénients :
3. Commande d'ascenseur par un automate programmable :

- 3.1. Avantages :
- 3.2. Inconvénients :
- 4. Commande d'ascenseur par un microprocesseur :
 - 4.1. Avantages :
 - 4.2. Inconvénients :
- 5. Commande d'ascenseur par un microcontrôleur :
 - 5.1. Avantages :
 - 5.2. Inconvénient :

Chapitre 3 : étude d'un microcontrôleur PIC

- 1. Introduction :
 - 1.1 Généralités sur les microcontrôleurs PICs
 - 1.2. Les différentes familles du microcontrôleur PICs :
 - 1.3. Identification du microcontrôleur PIC :
- 2. Architecture générale d'un microcontrôleur PIC :
- 3. Choix du microcontrôleur :
 - 3.1. Pourquoi un microcontrôleur PIC 16F877 ?
 - 3.2. Description matérielle du microcontrôleur PIC 16F877

Chapitre 4 : Aspects pratiques : Commande d'un ascenseur a l'aide d'un PIC 16F877

- 1. Position du problème :
- 2. Le programme :

Conclusion

Bibliographies

Annexes

Introduction

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 le microprocesseur, des mémoires et des circuits d'entrées-sorties (ports parallèles, *timers*, convertisseurs...). Chaque fabricant a sa (ou ses) famille(s) de microcontrôleurs.

Une famille se caractérise par un noyau commun (le microprocesseur, le jeu d'instruction...). Ainsi les fabricants peuvent présenter un grand nombre de microcontrôleurs 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 de cette famille, il lui faut juste connaître ses différences par rapport au circuit principal de la famille. Ces différences sont souvent, la taille des mémoires, la présence ou l'absence des circuits d'entrées-sorties et leurs nombres.

1. Avantages des circuits microcontrôleurs :

L'utilisation des microcontrôleurs pour les circuits programmables à 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. Nous allons voir que le nombre d'entre eux découle du simple sens.

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

- Coûts de main d'œuvre diminués.

Ces privilèges font que les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications de grand public ou professionnelles. Ainsi, nous trouvons les microcontrôleurs dans les appareils électroménager (machines à laver, démodulateurs numériques, climatiseurs, ...), l'industrie automobile (système de freinage, amortissement, localisation par GPS, ...), télécommunications (téléphones portables, ...), loisirs (jouets, ...), ... etc.

Nous nous proposons dans ce mémoire d'étudier et de concevoir un système de commande à base d'un circuit microcontrôleur pour la gestion d'un ascenseur.

2. Le choix du microcontrôleur : Pourquoi le pic16F877 ?

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

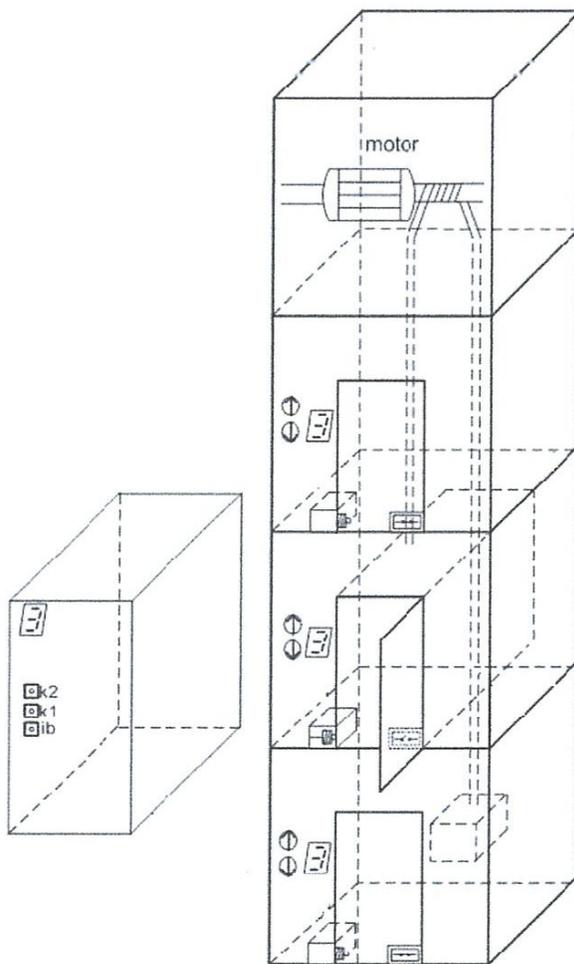
Le PIC 16F877 possède en plus des instructions très puissantes donc un programme à développer réduit, une programmation simple grâce au mode série.

Ce rapport est subdivisé en 6 Chapitres. Le premier est consacré à une introduction dans laquelle nous expliquons les raisons qui nous ont poussés à choisir ce sujet. Le deuxième chapitre est consacré à la description de la "technologie" des ascenseurs. Les différentes techniques utilisées pour commander les ascenseurs sont abordées dans le troisième chapitre. Le quatrième chapitre décrit le microcontrôleur 16F877 utilisé dans cette carte de commande. Nous décrivons le principe de fonctionnement de cette carte, tant du point de vue matériel que logiciel, dans le cinquième chapitre. Nous terminons finalement ce rapport par une conclusion.

Cahier de charges

1. Sujet :

Conception et réalisation carte de commande d'un ascenseur à base de PIC 16F877aA qui permet de commander la cabine de 1 ascenseur a 3 étages



2. Description :

La maquette doit contenir :

- ✓ UN ensemble de LEDs.
- ✓ 3 boutons poussoirs pour l'appel cabine ,une pour chaque étage.
- ✓ 3 boutons poussoirs pour l'appel étage .
- ✓ Un afficheur 7 segments.
- ✓ Commande d'un moteur à courant contenu.

Chapitre 1 :

Les ascenseurs

1. Introduction

Monter et descendre des personnes et des charges sur des grandes hauteurs était une entreprise particulièrement hasardeuse jusqu'à l'invention du monte-charge à parachute en 1852. Pour lever les charges lourdes, l'homme s'était, jusqu'alors, battu avec des moyens divers (la plupart du temps sans succès ou, tout au moins, sans sécurité). Qu'il s'agisse d'un treuil ou d'une simple poulie la corde pouvait casser . . . imaginez le résultat !

Les experts estiment que les Égyptiens ont dû utiliser des treuils pour construire les Pyramides. La plus grande d'entre elles, édifiée 2600 ans avant Jésus-Christ, culminant à plus de 150 mètres et est construite à partir de nombreux blocs de pierres pesant plus de 100 tonnes chacune !!!

Aujourd'hui, en France à titre d'exemple, l'ascenseur transportent cent millions d'utilisateurs par jour, il occupe à ce titre le rang de premier moyen de transport collectif en France. Pour ce faire, chaque machine parcourt en moyenne 3 000 kilomètres par an, sous la libre utilisation de ses passagers. Avec un volume de marché annuel de 9 400 ascenseurs et un parc de 430 000 machines en usage, l'ascenseur est un produit industriel de grande série dans ce pays.

2. Définitions

Un ascenseur est un dispositif assurant le déplacement en hauteur d'une charge. Il n'y a pas de distinction nette entre ce terme et d'autres : comme le monte-charge ou l'élévateur. Toutefois, on réserve souvent le mot '*ascenseur*' aux appareils destinés au transport vertical de personnes dans les bâtiments à niveaux multiples. Le terme désigne aussi couramment la cabine de l'ascenseur qui n'est qu'un élément du dispositif.

Autre définition : Les ascenseurs sont des appareils de levage verticaux destinés au transport des personnes ou des marchandises. On distingue techniquement trois types d'appareils :

Les *ascenseurs* destinés à l'usage unique des personnes ;

Les *monte-charges* destinés aux personnes et marchandises ;

Les *monte-charge industriels* destinés aux marchandises uniquement.

3. Applications

Suivant la nature de la charge transportée on rencontre :

- ✓ les monte-voitures (garages).
- ✓ les monte-chariots (usines).
- ✓ les monte-décors (théâtres).
- ✓ les monte-scories (aciéries).
- ✓ les monte-fûts (magasins), etc.

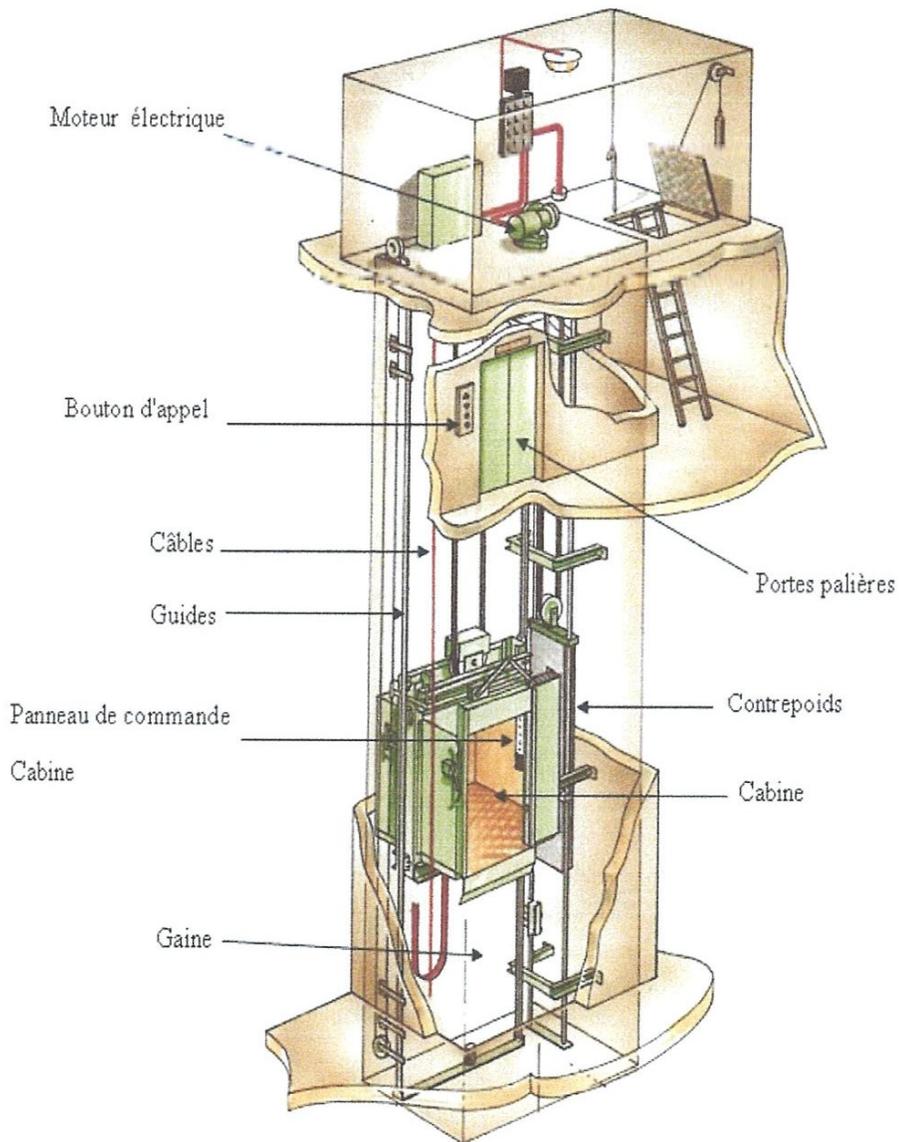


Figure 1 : Schéma des différents composants d'un ascenseur.

Certains de ces appareils peuvent également transporter en même temps des personnes :

- ✓ les monte-charges accompagnés.
- ✓ les monte-malades.

4. Les composants principaux d'un ascenseur

4.1. La gaine

La gaine est un volume dans lequel se déplacent la cabine et son contrepoids. Ce volume est matériellement délimité par le fond de la cuvette, les parois et le plafond. La gaine est équipée de guides en acier (profilés en acier, généralement en forme de T) destinés à guider la suspension de la cabine et du contrepoids.

4.2. Les guides

La cabine et le contrepoids circulent le long de guides en acier dont l'importance est importante pour le confort des usagers, figure 2. Les guides de cabine sont constitués par des barres d'acier de section ronde de 60 mm environ de diamètre. Ces barres comportaient deux rainures longitudinales qui permettaient leur propre fixation et également celle d'organes d'arrêt ou de sécurité disposés sur la gaine.



Figure 2 : Suspension de la cabine, on y distingue le guide en forme de T et la gaine (mur en briques)



Figure 3 : L'intérieur de la cabine. On y distingue le panneau des boutons de commande.

4.3. La cabine

Organe de l'ascenseur ou du monte-charge destiné à recevoir les personnes et/ou les charges à transporter, figure 3. Elle est munie de portes à fermeture automatique destinées à confiner l'utilisateur dans la cabine pendant le déplacement de celle-ci, lui interdisant tout contact avec les parties extérieures à la cabine.

4.4. Les commandes d'appel

Elles ont pour but de faire venir l'appareil aux étages d'où elles émanent respectivement, figure 4.

4.5. Les commandes d'envoi

Leur rôle est d'envoyer la cabine du niveau où elle se trouve au niveau auquel la personne qui effectue l'envoi le désire, figure 5. La commande d'envoi s'effectue par l'intermédiaire d'un bouton qui est placé à l'intérieur de la cabine dans la majorité des cas ou encore sur le palier pour certains monte-charges.

Lorsqu'une commande d'envoi est effectuée depuis la cabine, elle doit évidemment avoir la priorité sur la commande extérieure. L'utilisateur qui est à l'intérieur de la cabine doit disposer, pour appuyer sur le bouton de son choix, d'un certain de temps après la fermeture des portes avant qu'une commande d'appel, faite de l'extérieur, puisse être exécutée.

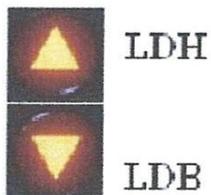


Figure 4 : les boutons d'appel de montée et de descente

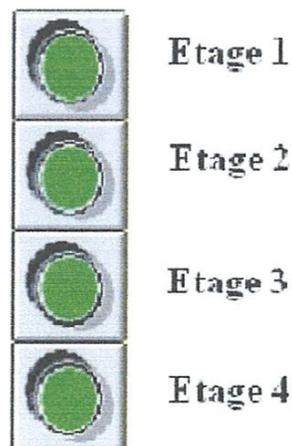


Figure 5 : les boutons d'appels de la cabine placés à chaque étage.

4.6. Les portes palières

Les portes obturent à chaque niveau les ouvertures dans la gaine servant d'accès à la cabine d'ascenseur. Afin d'éviter les chutes de personnes dans la gaine, leur ouverture n'est possible qu'en présence de la cabine. Ainsi, les portes prennent de plus en plus corps avec l'appareil lui-même pour des raisons impérieuses de sécurité.

4.7. Le contrepoids

Elément destiné à contre balancer le poids de la suspension de la cabine augmenté de la moitié de la charge utile. Le contrepoids est constitué d'une suspension métallique contenant des gueuzes en fonte destinées à l'alourdir. Lorsque la cabine de l'ascenseur monte, le contrepoids descend. Les masses peuvent varier de 25 à 120 kg.

4.8. Les câbles de traction

Les câbles assurent la liaison entre la cabine et le contrepoids, figure 6. Les câbles modernes, constitués de 6 ou 8 torons, sont fabriqués à partir de fils en acier qui subissent trois opérations distinctes : la trempe, la préparation des fils et le tréfilage.



Figure 6 : La cabine suspendue par les câbles.

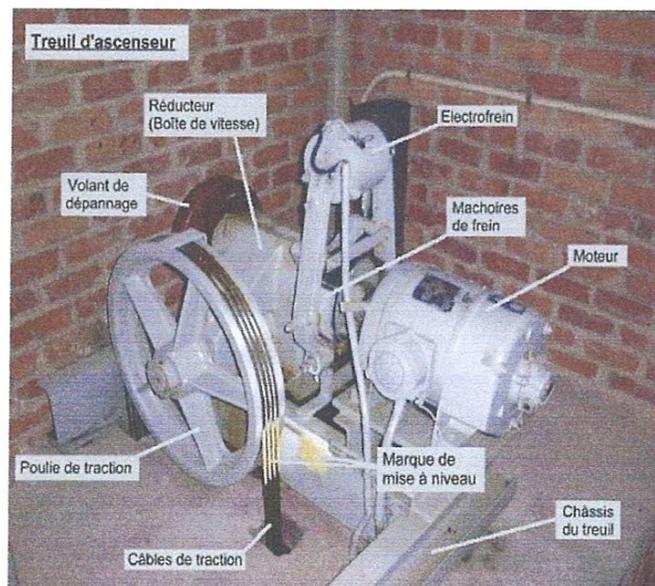


Figure 7 : Le treuil d'un ascenseur. La marque jaune sur les câbles indique le positionnement précis de la cabine en face d'un palier

4.9. Le treuil

Le treuil entraîne la cabine et le contrepoids dans leur mouvement. Il est composé des éléments suivants :

- ✓ un moteur électrique.
- ✓ un réducteur roue et vis sans fin.
- ✓ une poulie de traction.
- ✓ un volant d'inertie.
- ✓ un embrayage frein à tambour.

4.10. Le moteur

Le moteur d'ascenseur est une machine de fabrication spéciale. Les conditions de fonctionnement sont en effet très particulières.

Le moteur le plus couramment employé est actuellement le moteur asynchrone à double cage. En raison de son faible coût (y compris pour des fortes puissances), de sa robustesse (pas de pièces de contacts) et de son bon rendement.

4.11. Parachute

Organe mécanique placé sur la suspension de la cabine et est commandé par un câble limiteur, figure 8. En cas de rupture des câbles de traction ou de survitesse exagérée en descente, le mécanisme du parachute se déclenche en assurant un blocage mécanique de la suspension dans les guides et évitant la chute libre de la cabine.



Figure 8 :

Le parachute: mis au point dès 1853 par Otis. Ce système de Sécurité empêche la chute de la cabine en cas de rupture de Câble.

4.12. Amortissement

Organe constitué d'une butée déformable en fin de course et comportant un système de freinage par fluide ou par ressort, figure 9. Ces organes sont destinés à ralentir la suspension cabine ou le contrepoids en cas de dépassement des "fin de course" de sécurité.

Dans le cas d'un ascenseur à grande vitesse, on utilise des amortisseurs à huile, sinon on utilise des ressorts puissants placés dans la cuvette.

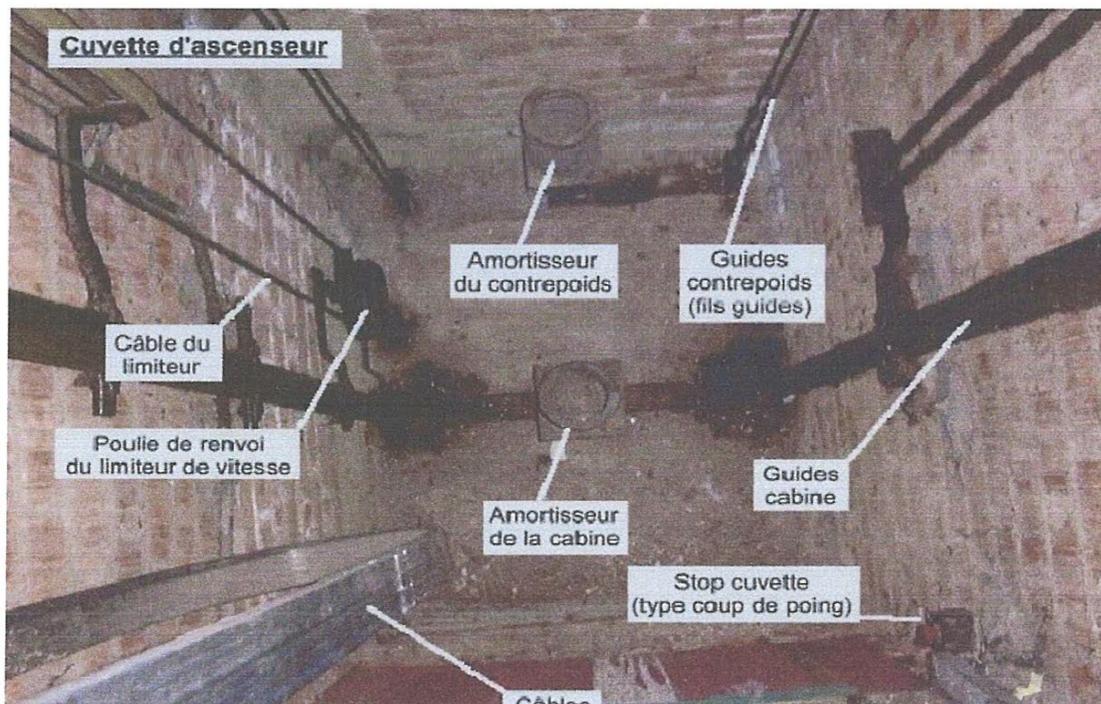


Figure 9 : Le fond de la gaine (la cuvette). On distingue les amortisseurs de la cabine constitués par des puissants ressorts.

5. La sécurité

Pour assurer la sécurité des usages, différents dispositifs sont prévus pour un ascenseur pour provoquer l'arrêt du moteur en cas d'anomalie. On peut citer :

- ✓ Le parachute : destinée à arrêter et à maintenir à l'arrêt la cabine sur ces guides en cas de rupture de la gaine ou de survitesse à la descente.
- ✓ Verrouillage des portes. Lorsque la cabine se met en mouvement, l'ouverture de ces dernières n'est permise que lorsque la cabine est en face du palier.
- ✓ Un bouton-poussoir d'urgence, à placer à l'intérieur de la cabine, qui permet d'activer

une alarme sonore attirant ainsi l'attention du technicien ou du concierge leur demandant d'aller secourir les utilisateurs ;

- ✓ Une cellule photo-électrique à placer aux niveaux des portières. Si jamais un utilisateur entre dans la cabine au moment de la fermeture des portes, ces dernières s'ouvrent de nouveau pour permettre à ce retardataire d'entrer dans la cabine en toute sécurité.

6. Différentes technologies des ascenseurs

Pour la production de l'énergie mécanique qui permet de soulever et déplacer la cabine, on distingue deux grandes familles d'ascenseur :

- ✓ Les ascenseurs à traction à câble ;
- ✓ Les ascenseurs hydrauliques

6.1. Les ascenseurs hydrauliques

Comme toute machine hydraulique, la pompe met sous pression l'huile qui pousse le piston hors du cylindre vers le haut. Lorsque la commande de descente est programmée, le bypass (vanne) de la pompe permet de laisser sortir l'huile du cylindre vers le réservoir, fig. 10.

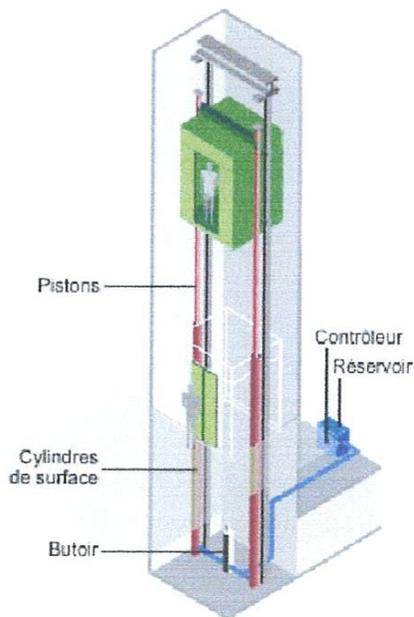


Figure 10 : ascenseur hydraulique

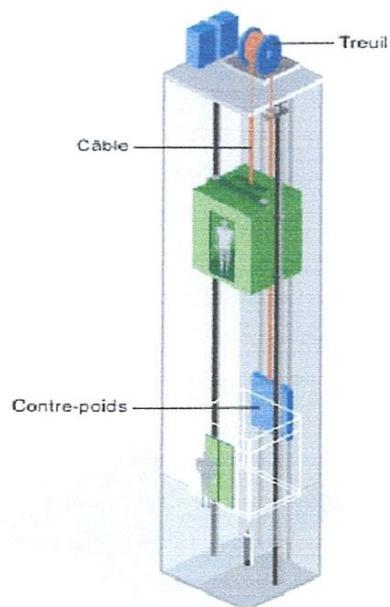


Figure 11 : ascenseur à traction à câbles

6.2. Les ascenseurs à traction à câbles

Les ascenseurs à traction à câbles sont les types d'ascenseurs les plus fréquemment utilisés, notamment dans les bâtiments, fig. 11. Ce type d'ascenseur a été largement explicité dans les paragraphes ci-dessus.

Chapitre 2 :

Differentes techniques pour commander un ascenseur

1. Différentes techniques de commande d'un ascenseur

Qu'il soit de type hydraulique ou à traction à câbles, les ascenseurs se composent principalement d'une cabine, de guides et d'un moteur électrique. Leur fonction essentielle est de desservir des personnes ou des charges à des niveaux définis. Pour réaliser cette tâche, nous disposons de 4 techniques distinctes :

- Commande d'un ascenseur par une logique câblée ;
- Commande d'un ascenseur par un automate programmable ;
- Commande d'un ascenseur par un microprocesseur ;
- Commande d'un ascenseur par un microcontrôleur.

2. Commande d'un ascenseur par une logique câblée :

Dans cette technologie, la réalisation de la carte de commande fait essentiellement appel à des circuits intégrés logiques de type combinatoire et séquentiel. A titre d'exemple, nous présentons dans la figure ci-dessous une armoire comportant les différentes cartes de commande et de contrôle de la cage de l'ascenseur. Le technicien peut directement avoir accès aux différents signaux d'entrée et signaux de sortie.

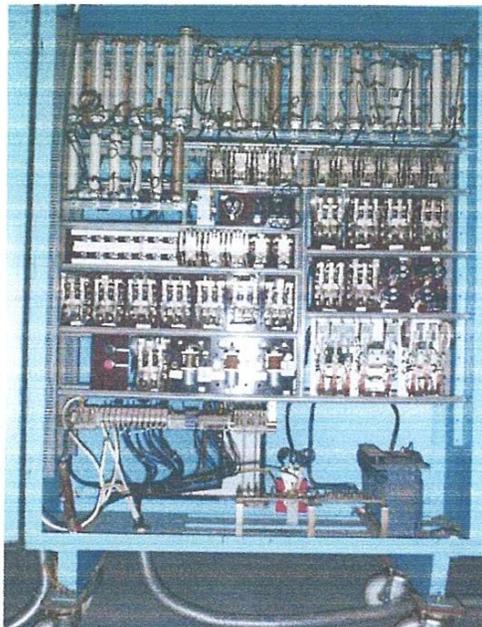


Figure 1 : Commande d'un ascenseur par une logique câblée

2.1. Avantages :

Simple manipulation grâce au fils

Coût réduit

Disponibilité des pièces.

2.2. Inconvénients :

Complexité du circuit (encombrement) ;

Circuit imprimé très grand et grande consommation électrique ;

La difficulté pour modifier la carte en cas d'amélioration (il faut une nouvelle carte).

Remarque :

Ce type de cartes de commande n'est plus utilisée de nos jours sinon il peut encore exister dans les très vieux bâtiments.

3. Commande d'ascenseur par un automate programmable :

L'automate fait partie de la technologie programmée. Il dispose d'un grand nombre d'entrées-sorties facilitant ainsi sa mise en œuvre. Nous présentons ci-dessous une illustration de ce type de commande.



Figure 2 : Commande d'ascenseur par un automate programmable

3.1. Avantages :

Fiabilité.

Simplicité de mise en œuvre.

Souplesse d'adaptation.

3.2. Inconvénients :

Cout très élevé.

Exclusivité d'utilisation des automates dans l'industrie.

Remarque :

Ce type de commande n'est pas utilisée en pratique en raison des couts très élevés de cette solution.

Nous avons eu l'opportunité d'étudier, de concevoir et de réaliser avec succès l'année dernière ce type de cartes dans le module de Projets d'Automates.

4. Commande d'ascenseur par un microprocesseur :

Cette technologie est également de type programmée. L'élément principal dans cette technique est le microprocesseur. Il est entouré de circuits mémoires et de circuits d'entres-sorties. C'est la carte à microprocesseur.

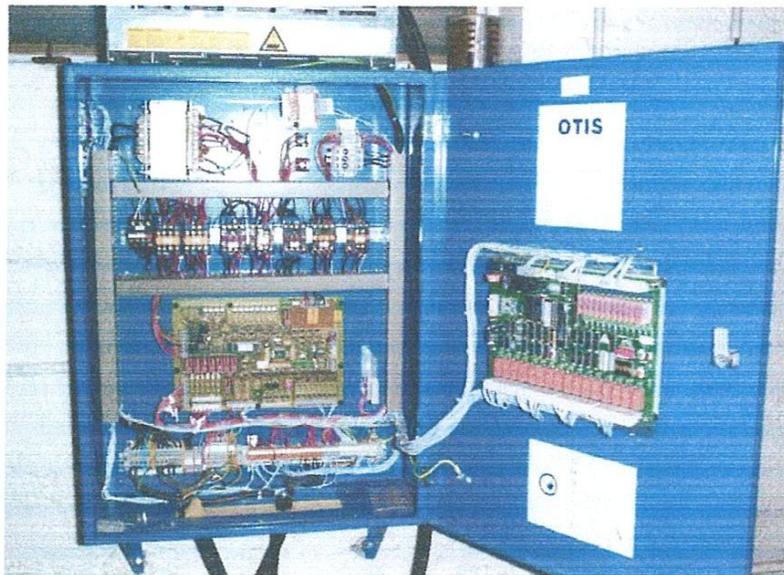


Figure 3 : Commande d'ascenseur par un microprocesseur

4.1. Avantages :

Rapidité.

Encombrement réduit.

4.2. Inconvénients :

Un cout très élevé.

Nécessite d'utiliser beaucoup de ports d'entrées/sorties.

Remarque :

Ce type de commande n'est pas aussi utilisée en pratique en raison des couts très élevés de cette solution.

5. Commande d'ascenseur par un microcontrôleur :

Le système à microprocesseur étant une carte à usage universel, on peut le faire remplacer par un système à microcontrôleur d'un cout nettement plus réduit mais spécifique à notre application, ce qui n'est pas un inconvénient.

5.1. Avantages :

Cout réduit

La possibilité d'ajouter des autres étages à tout moment.

La possibilité d'améliorer le programme à tout moment.

Disponibilité des pièces.

5.2. Inconvénient :

La carte ainsi réalisée ne servira qu'à la commande exclusive de l'ascenseur et pas à autre chose. C'est ce que nous voulons en fait !! ... Une solution programmée spécifique à notre application, ce n'est donc pas vraiment un inconvénient dans notre cas.

Remarque :

C'est cette solution que nous avons décidé finalement d'étudier et de concevoir. Elle fait l'objet de ce présent travail

Chapitre 3 :

etude d'un microcontrôleur PIC

Etude d'un microcontrôleur PIC :

1. Introduction :

Les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications de grand public ou professionnelles, il existe plusieurs sociétés qui fabriquent des microcontrôleurs. La société américaine *Microchip Technology* a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (*Peripheral Interface Controller*). Ce composant, encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

Dans ce chapitre, nous donnons une description assez détaillée du microcontrôleur dans la mesure où il constitue l'élément de base dans notre carte de commande de l'ascenseur.

1.1 Généralités sur les microcontrôleurs PICs

Le 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.

Les microcontrôleurs PICs sont des composants dits à architecture **RISC** (**R**educed **I**nstructions **S**et **C**omponent) ou encore composant à jeu d'instructions réduit. Sachant que plus le nombre d'instructions plus il est réduit, plus facile et plus rapide en est le décodage et plus vite le composant fonctionne.

L'horloge, fournie au microcontrôleur **PIC**, est pré-divisée par 4 au niveau de celle-ci. C'est cette base de temps qui donne le temps d'un cycle. Si on utilise par exemple un quartz de 4 MHz, on obtient donc 1 000 000 de cycles/seconde. Comme le microcontrôleur **PIC** exécute pratiquement une instruction par cycle, hormis les sauts, cela donne une puissance de traitement de l'ordre de 1 MIPS (**1** Million d'**I**nstructions **P**ar **S**econde).

1.2. Les différentes familles du microcontrôleur PICs :

Il existe trois grandes familles de PICs :

- ✓ La famille **Base Line** qui utilise des mots d'instructions de 12 bits.
- ✓ La famille **Mid-Range** qui utilise des mots d'instructions de 14 bits (les PICs 16F84 et 16F876 en font partie).
- ✓ La famille **High-End** qui utilise des mots d'instructions de 16 bits.

Les microcontrôleurs PICs Mid-Range ont un jeu de 35 instructions.

Etude d'un microcontrôleur PIC :

1.3. Identification du microcontrôleur PIC :

Pour identifier un microcontrôleur PIC, on utilise simplement son numéro. Les 2 premiers chiffres indiquent la catégorie du microcontrôleur **PIC**, **16** indique un PIC de la famille **Mid-Range**. Vient ensuite une lettre

- ✓ **L**: Celle-ci indique que le microcontrôleur **PIC** peut fonctionner avec une plage de tension plus tolérante.
- ✓ **C** indique que la mémoire programme est une **EPROM** ou plus rarement une **EEPROM**
- ✓ **CR** pour indiquer une mémoire de type **ROM**
- ✓ **F** pour indiquer une mémoire de type **FLASH**

Les microcontrôleurs **PICS** sont des composants **STATIQUES**, c'est à dire que la fréquence d'horloge peut être abaissée jusqu'à l'arrêt complet sans perte de données et sans dysfonctionnement. Ceci par opposition aux composants **DYNAMIQUES**, donc la fréquence d'horloge doit rester dans des limites précises.

2. Architecture générale d'un microcontrôleur PIC :

Le schéma ci-dessous présente les principaux blocs fonctionnels présents à l'intérieur d'un PIC ainsi qu'une courte description. Pour un schéma plus précis, se référer au datasheet du constructeur.

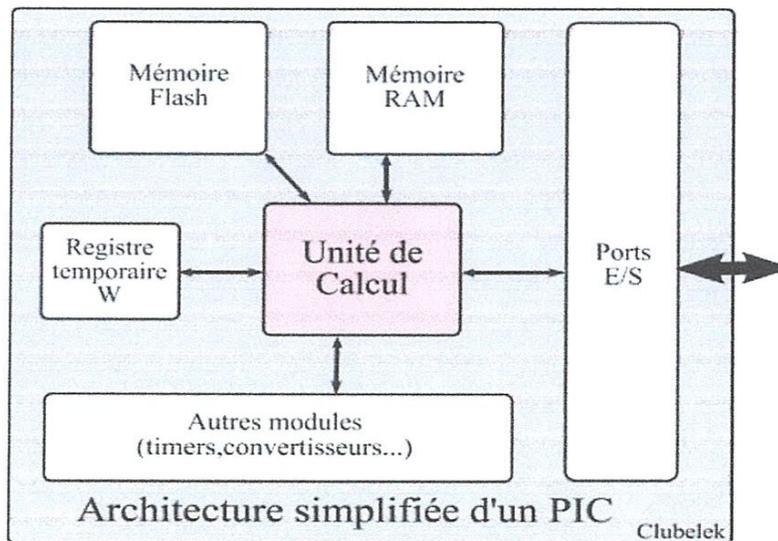


Fig. 1 : Architecture interne du PIC

Etude d'un microcontrôleur PIC :

Description fonctionnelle des blocs :

- ✓ **Mémoire flash** : C'est une mémoire réinscriptible qui conserve ses données lorsque le PIC n'est pas alimenté. Elle est utilisée pour stocker le programme. A chaque ligne du programme est attribuée une adresse qui permettra à l'unité de calcul de se repérer.
- ✓ **Mémoire RAM** : C'est une mémoire volatile (qui s'efface quand le PIC n'est plus alimenté). Les variables utilisées au cours du programme sont stockées à cet endroit.
- ✓ **Unité de Calcul** : C'est le cœur du microcontrôleur. Ici se déroulent toutes les opérations à une vitesse définie par la fréquence d'horloge (fréquence d'oscillation divisée par 4).
- ✓ **Registre temporaire W** : C'est ici qu'est stockée la valeur nécessaire pour une opération de l'unité de calcul. Ce registre sert aussi à la réception du résultat d'un traitement donné.
- ✓ **Ports E/S (Entrées/Sorties)** : Ce sont les unités qui font le lien entre ce qui se passe à l'intérieur du PIC et l'extérieur.
- ✓ **Modules annexes** : Toutes fonctions annexes tels que le timer, le comparateur, le convertisseur analogiques/numérique, des ports d'entrées-sorties série).

3. Choix du microcontrôleur : X

3.1. Pourquoi un microcontrôleur PIC 16F877 ?

Le choix d'un microcontrôleur est important car c'est de lui que dépendent, en grande partie, les performances, la taille, la facilité d'utilisation et le prix du montage.

Le 16F877 est un microcontrôleur de MICROCHIP. Il fait partie intégrante de la famille des Mid-Range (16) dont la mémoire programme est de type flash (F) de type 877 et capable d'accepter une fréquence d'horloge maximale de 20 MHz.

3.2. Description matérielle du microcontrôleur PIC 16F877

Le microcontrôleur 16F877, en boîtier DIP 40 broches, peut fonctionner à une fréquence maximale de 20 MHz (pour un cycle d'instruction de 200 ns). Il est constitué des éléments suivants :

Etude d'un microcontrôleur PIC :

| PIC | FLASH | RAM | EEPROM | I/O | A/D | PORT // | Port série |
|--------|-------|---------------|---------------|-----|-----|---------|---------------|
| 16F877 | 8K | 368 octets | 258 octets | 33 | 1 | 5 | USART/ SSP |

- ✓ 8 K mots de 14 bits de mémoire-programme du type Flash,
- ✓ 256 octets de mémoire EEPROM,
- ✓ 368 octets de mémoire RAM,
- ✓ 2 *Timers* à 8 bits et un *Timer* à 16 bits,
- ✓ Un convertisseur A/N à 10 bits avec 5 entrées multiplexées,
- ✓ Un chien de garde (*WatchDog Timer*),
- ✓ Une interface série synchrone (SSP),
- ✓ Une interface série asynchrone (USART).
- ✓ 5 ports E/S :
 - ❖ Un port A de 6 bits (RA0 à RA5).
 - ❖ Un port B de 8 bits (RB0 à RB7).
 - ❖ Un port C de 8 bits (RC0 à RC7).
 - ❖ Un port D de 8 bits (RD0 à RD7).
 - ❖ Un port E de 3 bits (RE0 à RE3).

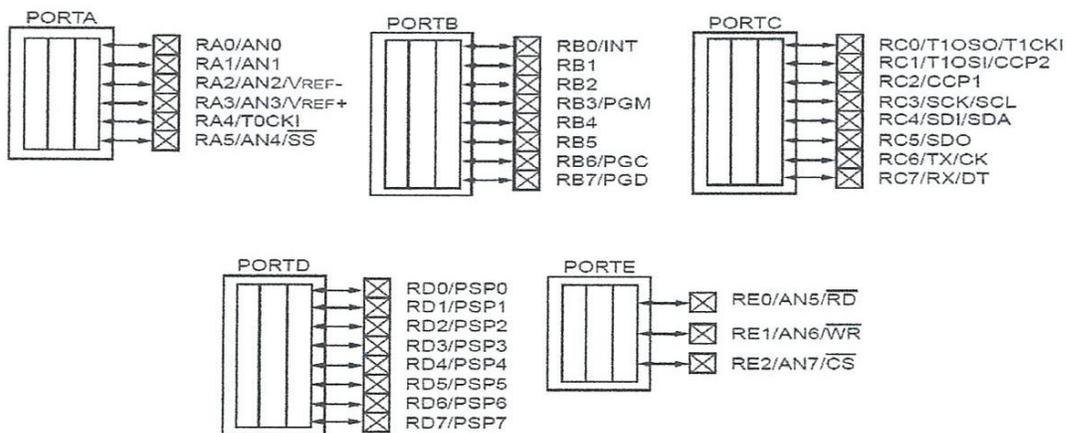


Fig. 3: Les ports d'entrées/sorties du PIC 16F877

Etude d'un microcontrôleur PIC :

Les figures 4 et 5 donnent les signaux et brochage général du PIC 16F877 ainsi que son architecture interne générale dans laquelle nous trouvons les organes (blocs) que nous avons rappelé ci-dessus.

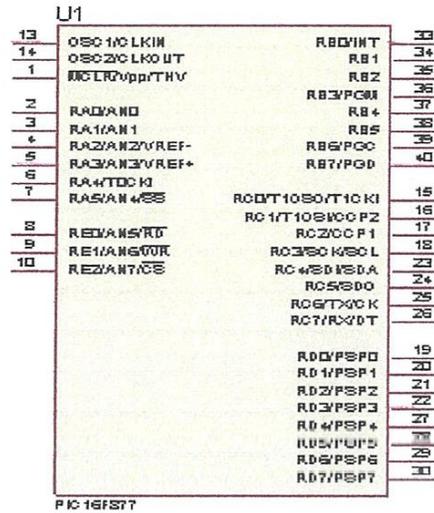


Fig. 4 : Structure générale d'un pic 16F877

Etude d'un microcontrôleur PIC :

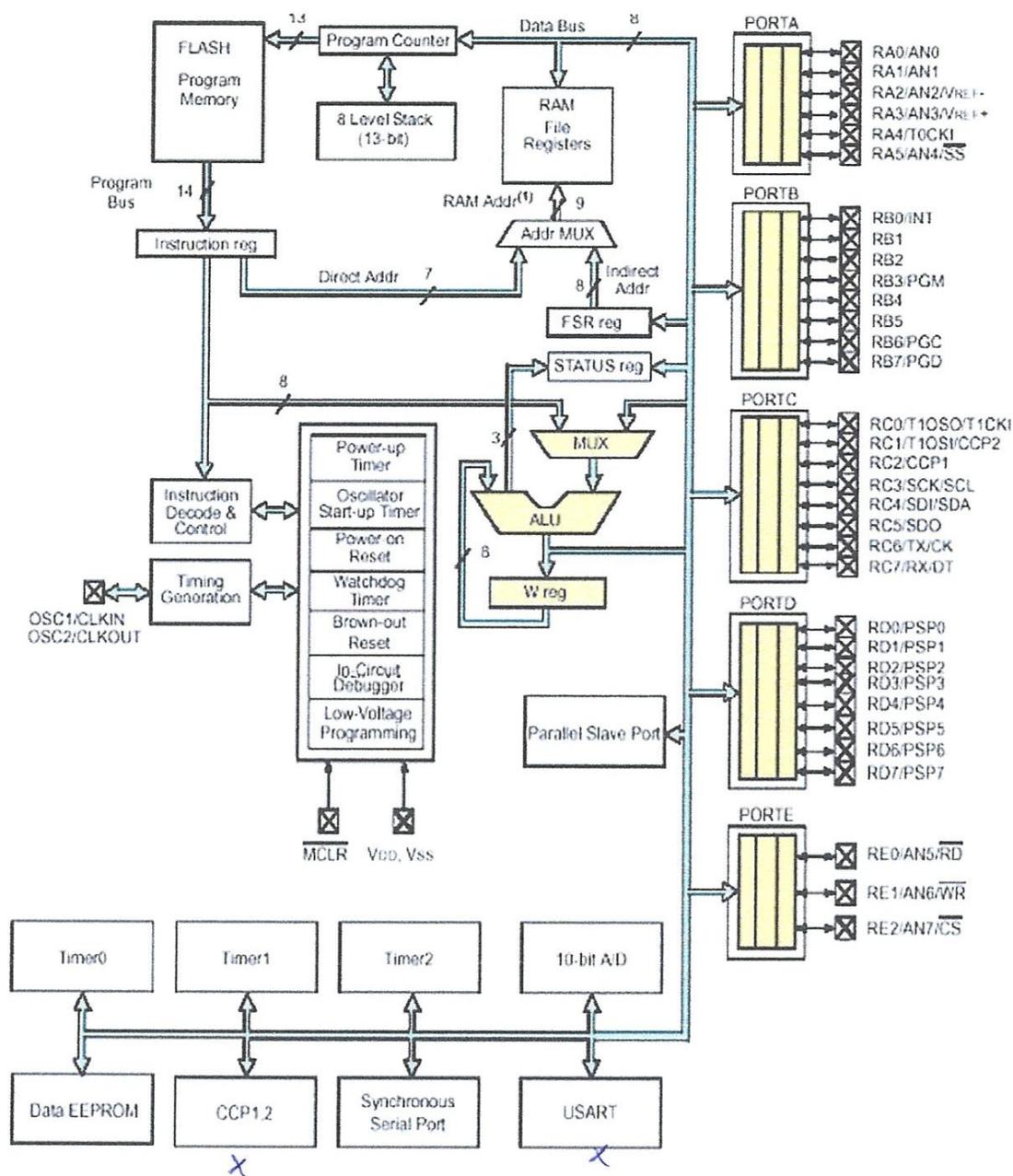


Fig. 5 : Structure interne du 16F877

Chapitre 4 :

Aspects pratiques :

Commande d'un ascenseur a l'aide d'un PIC 16F877

Position du problème

1. Position du problème :

Nous voulons commander la montée et la descente d'une cabine d'un ascenseur qui dessert un bâtiment à trois étages (Rez de Chaussée, étage 1 et étage 2), figure 1.

Pour cela, nous disposons d'un moteur à courant continu réversible qui tourne dans un sens ou dans un autre en fonction du sens du courant appliqué à ce moteur. L'utilisateur de l'ascenseur dispose de trois boutons d'appel (Appel cabine) situés à chaque palier du bâtiment et de trois boutons poussoirs de sélection de l'étage de destination situés à l'intérieur de la cabine.

On suppose que la cabine de l'ascenseur est déposée au niveau 0 (Rez de Chaussée : RDC) lors de son installation. Par la suite, dès qu'elle entre en service, sa position n'est plus importante. En effet, le programme se chargera par la suite de traiter tous les cas possibles.

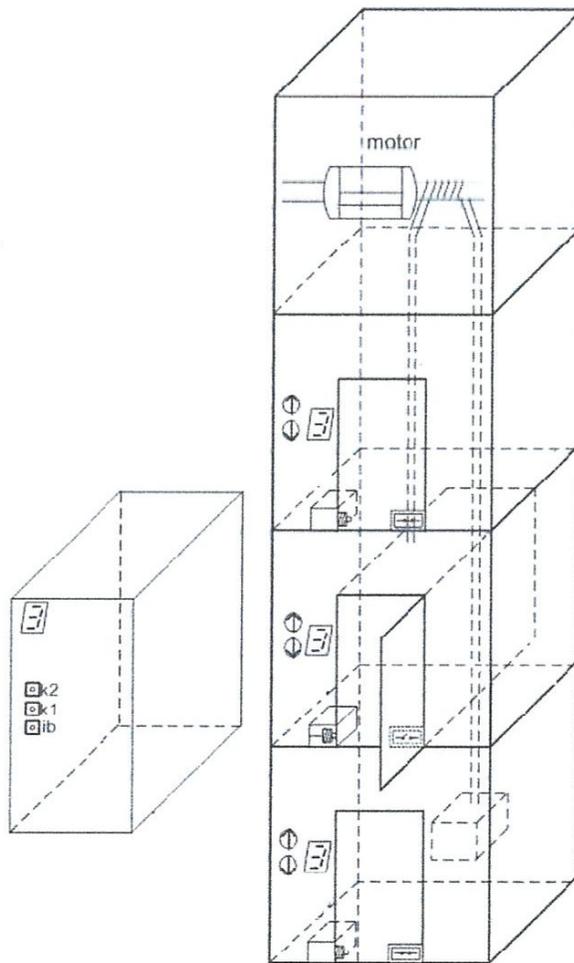


Figure 1 : Schéma d'un système ``ascenseur + cabine``. Nous distinguons, à chaque palier, les boutons d'appel de la cabine ainsi qu'un afficheur qui indique la position instantanée de cette dernière. Ce même afficheur ainsi que des boutons de sélection de l'étage se trouvent à l'intérieure de la cabine.

Position du problème

Supposons que l'utilisateur se trouve au 2^e étage et qu'il désire descendre au 1^e étage. Pour cela, il va appuyer sur le bouton d'appel de la cabine se trouvant au 2^e étage. Le programme se chargera alors d'identifier l'étage d'appel (il enregistre cet étage : ETAGE = 2) et va le comparer au numéro de l'étage dans lequel se trouve la cabine (la position de la cabine est toujours stockée en mémoire dans une case appelée PRESENCE. A l'état initial : PRESENCE = 0). En fonction du résultat de cette comparaison, le programme va actionner le moteur soit dans le sens de la descente soit dans le sens de la montée selon les configurations du tableau suivant :

| Etage ou appel est actionné | Etage ou se trouve la cabine | Sens du moteur |
|------------------------------------|-------------------------------------|----------------|
| Etage supérieur | Etage inférieur | Montée |
| Etage inférieur | Etage supérieur | Descente |

Lorsque la cabine est ramenée à l'étage d'appel (le deuxième étage dans ce cas et donc PRESENCE = 2), le moteur est arrêté. Quelques secondes plus tard (2 s) la porte s'ouvre et reste ouverte pendant 5 secondes. Ensuite la porte se ferme. Voir organigramme : appel de la cabine de la figure 2.

Légendes :

Nous utilisons dans le programme, l'organigramme et dans le schéma électrique un certain nombre de noms et de symboles définis ci-dessous :

- ✓ App cab : registre en mémoire adresse h'22'.
- ✓ Présence : registre en mémoire adresse h'21'.
- ✓ App etg : registre en mémoire adresse h'23'.

Nous utilisons également dans le programme les noms suivants qui représentent les :

- ✓ **Entrées** : Boutons-poussoirs

App cab : bouton-poussoir situé à chaque palier. Il sert à appeler la cabine.

App etg_i : bouton-poussoir situé à l'intérieur de la cabine. Il sert à choisir l'étage ``i`` de destination.

- ✓ **Sorties** : Moteur + diodes LEDs + Afficheur

présence cabine : Trois voyants situés aux trois étages du bâtiment (LED1, LED2, LED3) qui indiquent la présence ou non de la cabine dans l'étage.

Porte : Voyant LED4 qui simule l'ouverture et la fermeture de la porte (LED allumée = porte ouverte et inversement).

Sens du moteur (montée ou descente) : Deux voyants indiquant soit la montée soit la descente de la cabine (LED5 : montée, LED6 : descente).

Afficheur 7 segment : Il affiche l'étage dans lequel se trouve la cabine.

Position du problème

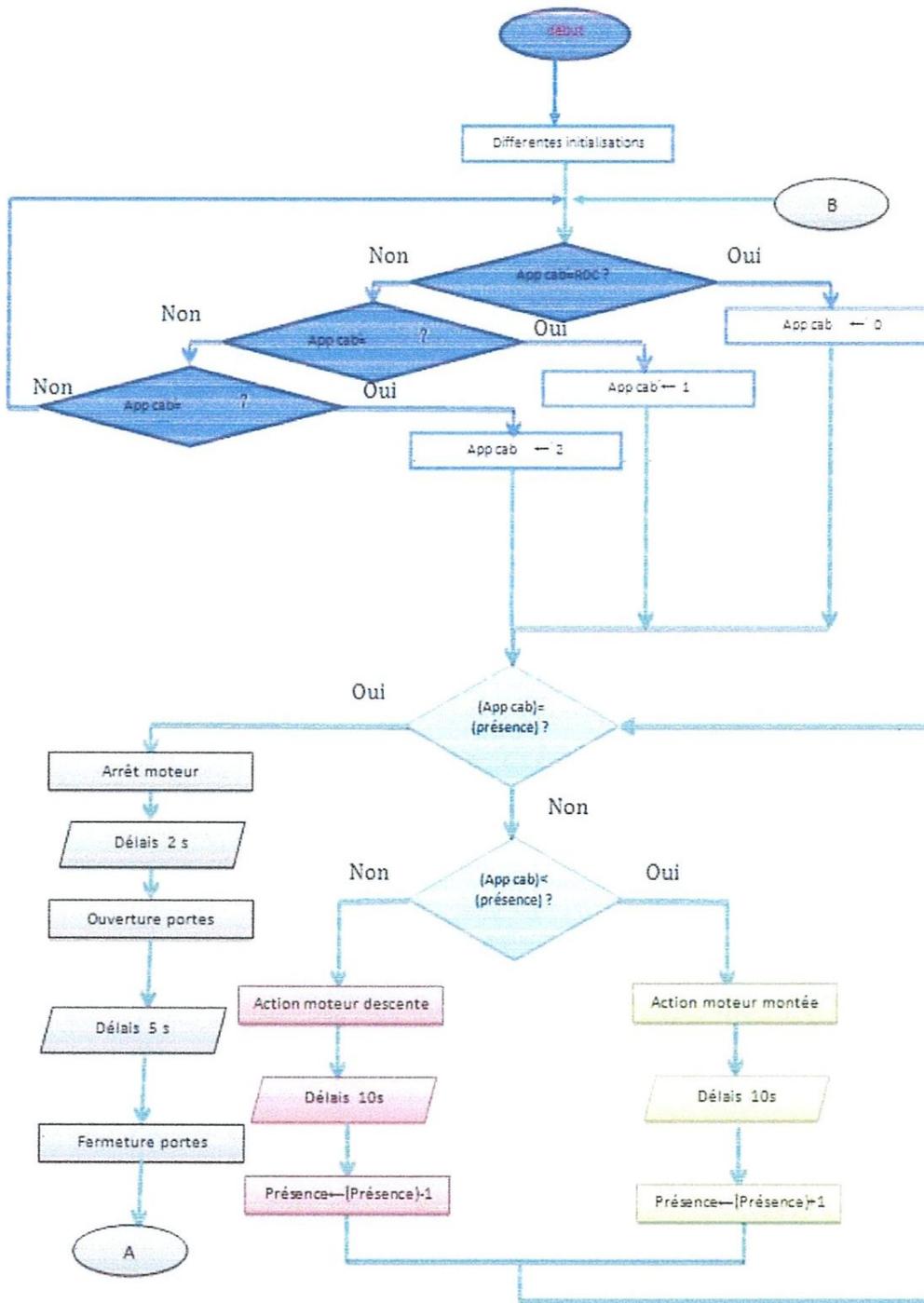


Figure 2 : Organigramme de l'appel de la cabine.

Position du problème

A ce moment l'utilisateur, après son entrée dans la cabine, est sensé avoir appuyé sur le bouton 1^o étage qui se trouve à l'intérieur de la cabine. Le programme se chargera ensuite de déterminer le numéro du bouton appuyé (étage destination = 1 dans notre cas ou ETAGE = 1, tout simplement) et le comparera à l'étage dans lequel se trouve maintenant la cabine (PRESENCE = 2) et fera démarrer le moteur dans le sens de la descente jusqu'à ce que : ETAGE = PRESENCE = 1, auquel cas le moteur s'arrête. Deux secondes plus tard, les portes s'ouvrent et restent ouvertes pendant 5 s et puis se referment. Le programme est terminé, voir organigramme (sélection étage destination) de la figure 3.

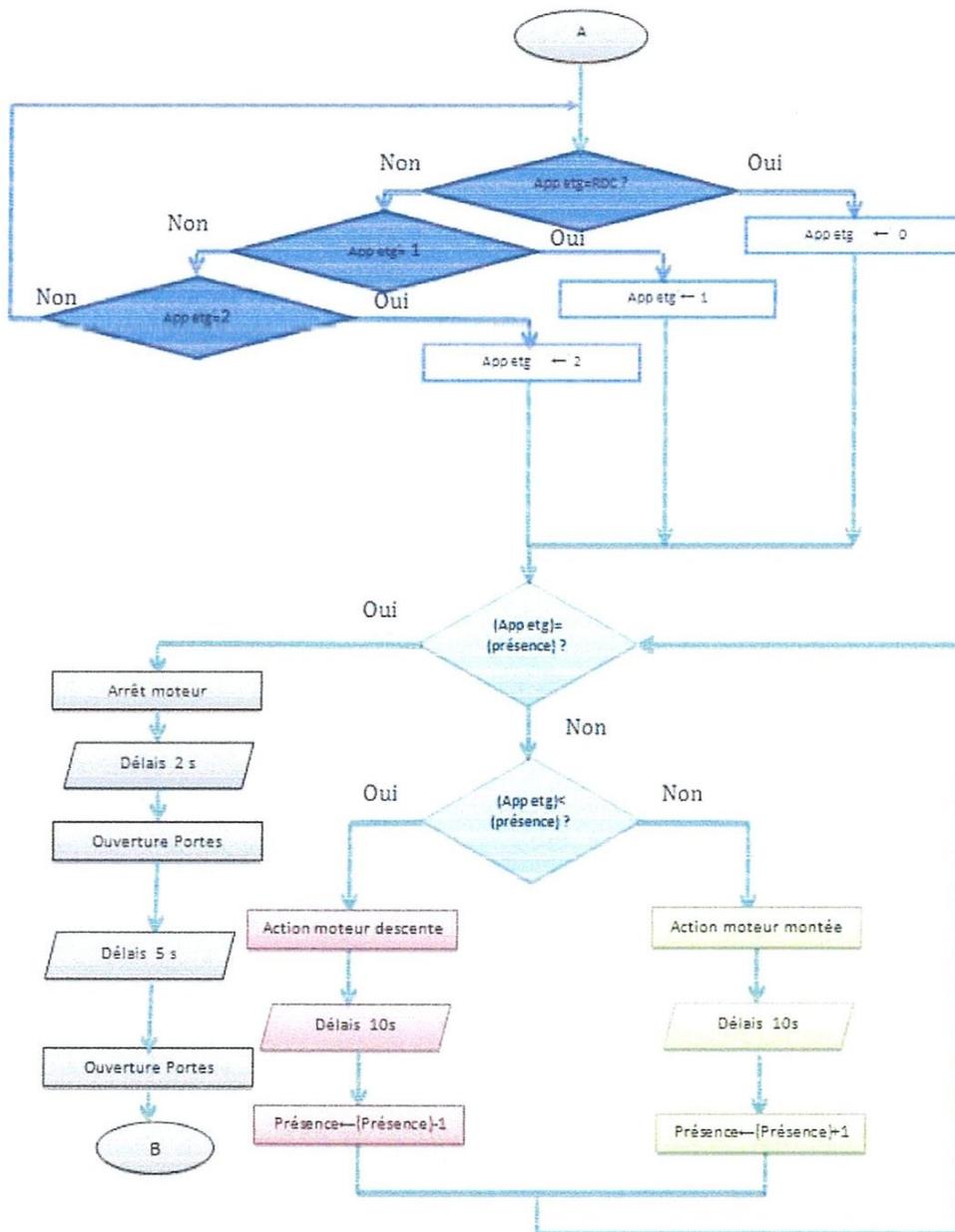


Figure 3 : Organigramme de la sélection de l'étage de destination.

L'organigramme donne le détail des différents registres à initialiser dès le lancement du programme.

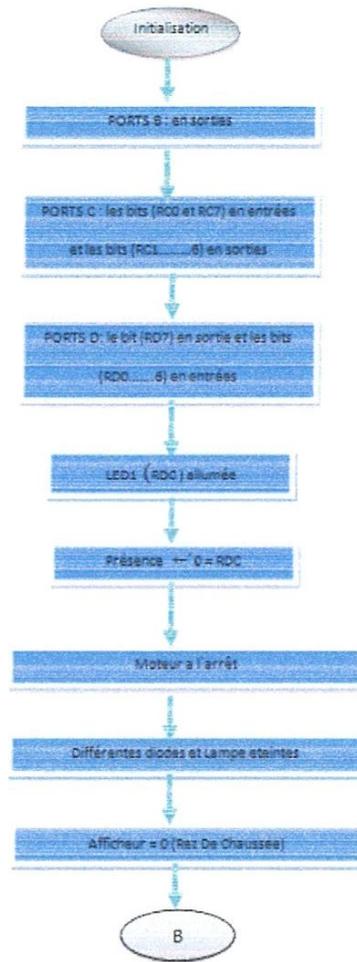


Figure 4 : Organigramme d'initialisation.

Position du problème

Le passage d'un étage à un autre est déterminé soit après l'écoulement d'un délai de 10 s par exemple entre un étage et un autre soit au moyen de contacts de présence placés au niveau de chaque palier. La première solution est intéressante pour les ascenseurs à peu d'étages délai constant entre un étage et un autre, c'est le cas de notre application. La seconde solution s'adapte mieux pour les bâtiments à plusieurs dizaines d'étages. La vitesse de montée ou de descente de la cabine n'est plus constante. Cette solution est meilleure du point de vue pratique mais nécessite cependant plusieurs broches d'entrées réservées aux capteurs pour la détermination de l'étage.

Une lampe pour éclairer la cabine a été prévue. Elle sera allumée dès que l'ouverture des portes de la cabine (suite à un appel) et s'éteindra lorsque l'usager est ramené à l'étage de destination et que les portes se ferment derrière lui. (Lampe connectée à la broche RC1 du PIC). Voir schéma de la figure 4.

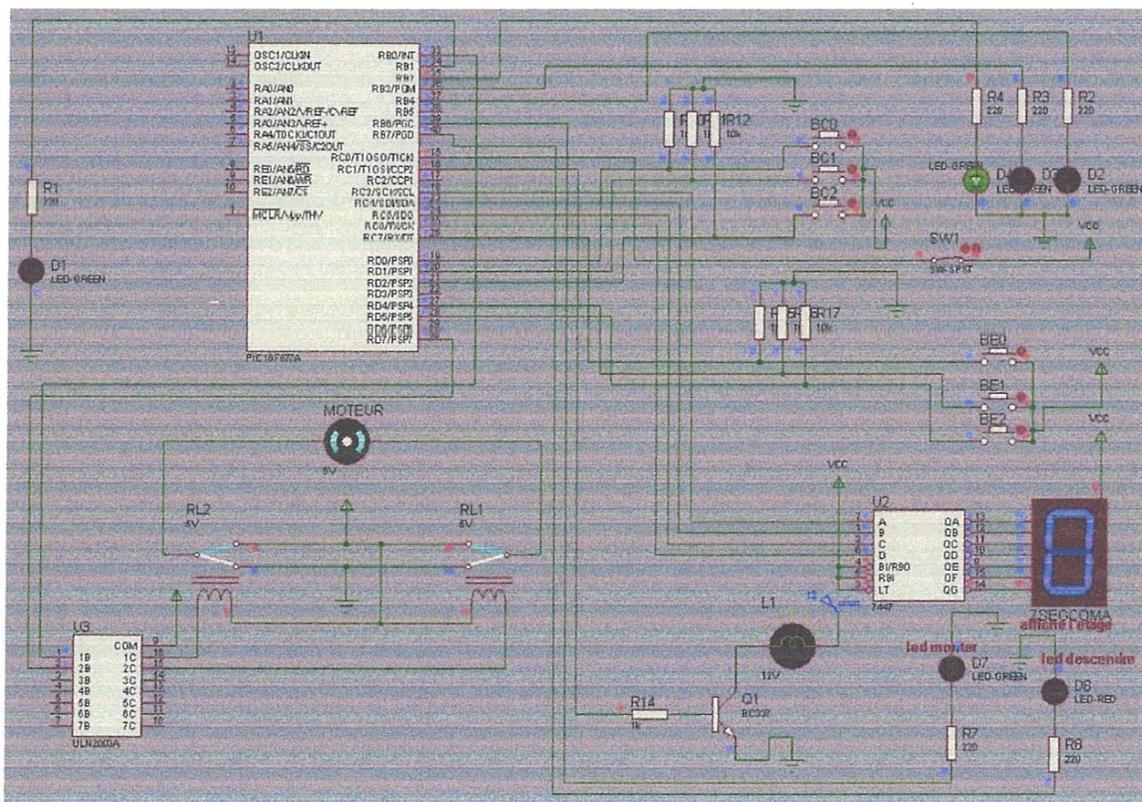


Figure 5 : Schéma électrique de gestion d'un système "ascenseur + cabine" contrôlé par un microcontrôleur PIC 16F877 de Microchip.

Position du problème

Deux diodes D7 et D8, liées aux broches RB6 et RB7 du PIC, servent à indiquer respectivement le sens de rotation du moteur en montée ou en descente.

Un afficheur 7 segments est connecté, via un convertisseur binaire/7 segments, au port C du PIC. Il affiche le numéro de l'étage dans lequel se trouve la cabine.

Une diode D1, reliée à RB1, simule l'ouverture et la fermeture de la porte de la cabine.

Deux lignes du PIC (RB0 et RD7) servent à commander le sens de rotation du moteur. Pour isoler électriquement le PIC du moteur, nous avons placé 2 relais et un circuit ULN 2003 qui sert à la fois à isoler le PIC du moteur et à amplifier le courant délivré par le PIC.

Dans le même ordre d'idées, 3 LEDs : D2, D3 et D4, connectées au port B, servent à indiquer l'étage dans lequel se trouve la cabine.

Cinq lignes du port D et une ligne du port C (RC7) sont utilisées pour véhiculer l'état des boutons-poussoirs - d'appel de la cabine (BC0, BC1 et BC2) et de sélection de l'étage (BE0, BE1 et BE2) – et informer le PIC s'ils sont activés ou non.

Remarque :

Cette carte peut être étendue à la commande d'un bâtiment à plusieurs étages sans problème. Il suffit d'ajouter des boutons-poussoirs et des LEDs de présence de la cabine et de modifier quelque peu le programme en conséquence.

Nous donnons finalement le programme écrit dans le langage assembleur correspondant à cette application.

2. Le programme :

```
__CONFIG _XT_OSC & _CP_OFF & _LVP_OFF & _WDT_OFF
```

```
LIST P=16F877A
```

```
INCLUDE <P16F877A.INC>
```

```
Presence EQU h'21'
```

```
Appcab EQU h'22'
```

```
Appetg EQU h'23'
```

```
Reg0 EQU h'24'
```

```
Reg1 EQU h'24'
```

```
Reg2 EQU h'24'
```

```
reset org 0x00
```

Position du problème

```
goto    init
```

```
;configuration des ports :
```

```
init
```

```
bsf     status,5
```

```
movlw   0x37
```

```
movwftrisd
```

```
movlw   0x81
```

```
movwftrisc
```

```
clrw
```

```
movwftrisb
```

```
bcf     status,5
```

```
clrfportb
```

```
clrfportc
```

```
clrfportd
```

```
bsf     portb,4
```

```
movlw   0x10
```

```
movwfpresence
```

```
;Démarrage du programme:
```

```
;Partie appel cabine :
```

```
START  btfsc   portd,0
```

```
        goto   config0
```

```
        btfsc  portd,1
```

```
        goto   config1
```

```
        btfsc  portd,2
```

```
        goto   config2
```

```
        goto   start
```

```
Config0 movlw   0x00
```

Position du problème

```
Movwf  appcab
goto   test
Config1 movlw  0x01
        movwfappcab
        goto   test
Config2 movlw  0x02
        movwfappcab
        goto   test
test    movfpresence,w
        subwfappcab,w
        btfsstatus,z
        goto   porte1
        btfsstatus,c
        gotodescender
        goto   monter
descendre call  afficheur
        bsf   portb,7
        bsf   portd,7
        call  delay1
        bcf   portb,7
        bcf   portd,7
        decf  presence,1
        call  afficheur
        goto  test
Monter   call  afficheur
        bsf   portb,6
        bsf   portd,0
```

Position du problème

```
    call    delay1
    bcf     portb,6
    bcf     portd,0
    incf   presence,1
    call    afficheur
    goto   test
porte1  bsf     portb,1
        call   delay2
        bcf   portb,1
        goto  ascenseur
;partie appel etage :
ascenseur  btfsc  porte,0
           goto  config00
           btfsc  portd,5
           goto  config11
           btfsc  portd,6
           goto  config22
           goto  ascenseur
config00  movlw  0x00
           movwf  appetg
           goto  test2
config11  movlw  0x01
           movwf  appetg
           goto  test2
config22  movlw  0x02
           movwf  appetg
           goto  test2
```

Position du problème

```
test2    movf    presence,w
         subwf   appetg,w
         btfsc  status,z
         goto   porte1
         btfsc  status,c
         goto   descender2
         goto   monter2
descendre2 call   afficheur
         bsf    portb,7
         bsf    portd,7
         call   delay1
         bcf    portb,7
         bcf    portd,7
         decf   presence,1
         call   afficheur
         goto   test2
monter 2 call   afficheur
         bsf    portb,6
         bsf    portd,0
         call   delay1
         bcf    portb,6
         bcf    portd,0
         incf   presence,1
         call   afficheur
         goto   test2
```

Position du problème

```
afficheur  movf      presence,w
           rlfw,w
           rlfw,w
           rlfw,w
           movwf     porte
```

DELAY1

```
movlw     .43
movwf     REG0
PLoop0
movlw     .226
movwf     REG1
PLoop1
movlw     .205
movwf     REG2
PLoop2
clrwdt
clrwdt
decfsz   REG2, 1
goto     PLoop2
decfsz   REG1, 1
goto     PLoop1
decfsz   REG0, 1
goto     PLoop0
clrwdt
return
```

Position du problème

DELAY2

movlw .165

movwf REG0

PLoop3

movlw .41

movwf REG1

PLoop4

movlw .147

movwf REG2

PLoop5

clrwdt

clrwdt

decfsz REG2, 1

goto PLoop5

decfsz REG1, 1

goto PLoop4

decfsz REG0, 1

goto PLoop3

return

END

Conclusion

L'objectif de ce travail était de faire l'étude et la programmation d'un ascenseur.

Et cela a base du langage de programmation MPLAB. Pour arrive a ce but nous avons commencé par une étude du fonctionnement du PIC et de son logiciel de programmation MPLAB

Le PIC est un équipement qui peut être utilisé a l'industrie pour piloter des chaines de montages, productions, manutentions, robots industriels, machines outils....leurs avantages sont :

- Simplification du câblage.
- Modification du programme faciles a effectue par rapport a une logique câblée.
- Enormes possibilités d'exploitation.
- Fiabilité professionnelle.

Ce travail nous a permis d'explorer le coté pratique de l'électronique et de voir les difficultés réelles au sein de ce milieu.

D'autre part, ce travail nous a permis d'avoir une idée générale sur le domaine de systèmes électroniques, et de comprendre les différentes étapes suivies pour ces types des systèmes.

A la fin de notre projet nous avons réussi a écrire le programme complet en langage assembleur. Et afin de valider ce programme, un essai pratique est réalise sur un vérin de translation double effet, des switches et des boutons poussoirs, et cela en utilisent pic 16F877A disponible au niveau du laboratoire d'électronique.

Bibliographie

Sites Web :

<http://www.microchip.com>

<http://bigonoff.com>

<http://www.alldatasheet.com>

<http://www.datasheet.com>

<http://www.abcelectronique.com>

<http://320volt.com/>

<http://www.diselc.es/>

<http://www.technologuepro.com/>



Dans de nombreux affichages numériques, les dix chiffres 0 à 9, et parfois les caractères hexadécimaux A à F, sont donnés au moyen de 7 segments. Chaque segment est constitué d'un matériau qui émet de la lumière quand il est traversé par un courant. Les matériaux les plus utilisés sont les diodes électroluminescentes (DEL) et les filaments incandescents.

LES AFFICHEURS 7 SEGMENTS A DEL

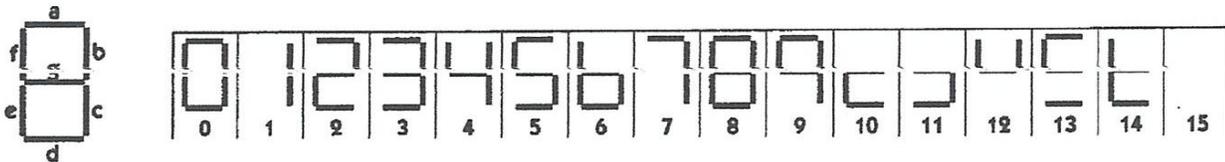


figure 1 : identification des segments et visualisation sur les afficheurs

■ **Constitution générale**

Chacun des 7 segments est éclairé par une DEL. Une huitième DEL permet l'éclairage du point décimal. Les afficheurs sont de deux types :

- à cathode commune (voir (1) figure 2) : les cathodes de toutes les DEL sont connectées ensemble et sont à relier au 0V;
- à anode commune (voir (2) figure 2) : les anodes de toutes les DEL sont connectées ensemble et sont à relier au +Vcc.

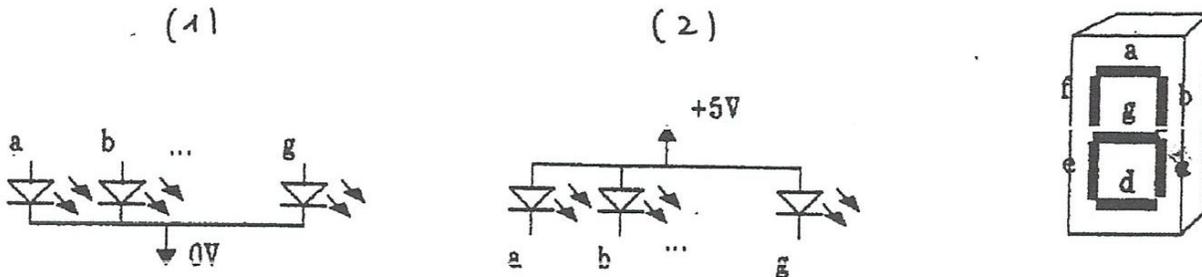


figure 2 : afficheur à cathode - anode commune

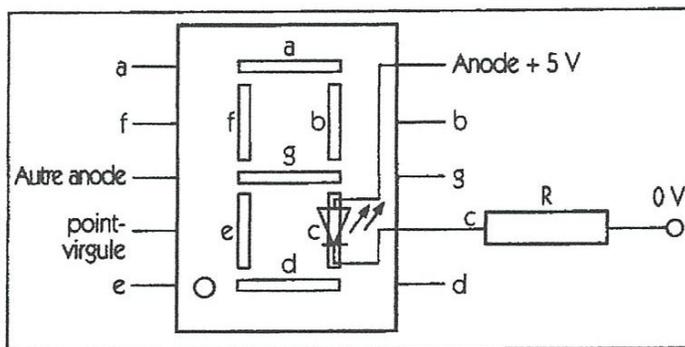


figure 3 : exemple d'afficheur 7 segments à anode commune

■ Principe de l'affichage

Pour utiliser un afficheur 7 segments, il est nécessaire de disposer d'un décodeur qui traduit le code BCD en code d'allumage des segments de l'afficheur.

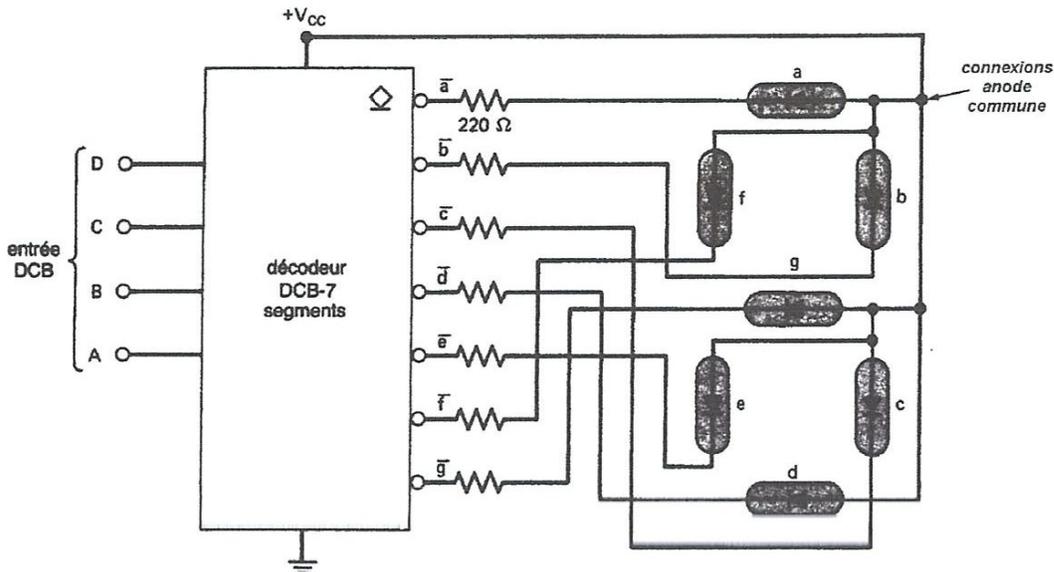


figure 3 : décodeur BCD / 7 segments attaquant un afficheur à DEL à 7 segments à anode commune. Les résistances sont nécessaires pour limiter le courant dans les segments.

■ Références de décodeur DCB / 7 segments

- technologie MOS : 4511, 4543 : décodeur BCD / 7 segments à mémoire;
- technologie TTL : 7446, 7447, 74347 : décodeur BCD / 7 segments pour afficheur à anode commune (c.o.);
- 7448 : décodeur BCD / 7 segments pour afficheur à cathode commune (2kΩ pull-up);
- 7449 : décodeur BCD / 7 segments pour afficheur à collecteur ouvert.

■ Brochage d'afficheurs 7 segments à DEL

| | A | B | C | D |
|----|---------------|---------------|------------|-----------------|
| 1 | cathode A | cathode A | anode C, D | anode F |
| 2 | cathode F | cathode F | - | anode G |
| 3 | anode commune | anode commune | anode C, D | - |
| 4 | - | - | - | cathode commune |
| 5 | - | - | - | - |
| 6 | N.C. | cathode DP | - | anode E |
| 7 | cathode E | cathode E | cathode D | anode D |
| 8 | cathode D | cathode D | cathode C | anode C |
| 9 | cathode DP | N.C. | - | anode DP |
| 10 | cathode C | cathode C | cathode B | - |
| 11 | cathode G | cathode G | cathode A | - |
| 12 | - | - | - | cathode commune |
| 13 | cathode B | cathode B | - | anode B |
| 14 | anode commune | anode commune | anode A, B | anode A |

Les relais :

Nous avons fait appel, dans notre application, à deux relais, fig. 1, pour commander le moteur soit dans le sens de la montée de la cabine soit dans le sens de la descente de celle-ci.



Figure 1 : Un relais

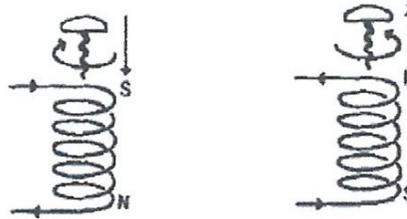


Figure 2 : La règle du tire-bouchon qui indique le sens du champ magnétique

1. Principe du relais :

Un conducteur électrique parcouru par un courant crée un champ magnétique au voisinage de ce conducteur. La polarité de ce champ dépend du sens du courant et du sens de l'enroulement des spires de la bobine, fig. 2.

La règle du tire-bouchon est un moyen simple de connaître la polarité magnétique. En bobinant un fil autour d'un matériau, ce matériau constitue, lors du passage du courant dans la bobine, un aimant capable d'attirer des matériaux ferreux.

2. Définition d'un relais :

Qu'est-ce que les relais ? Comment sont-ils faits ? Quelle est leur schéma ?

Les relais sont des dispositifs électromécaniques qui ferment un interrupteur sur commande lorsqu'un courant électrique leurs parvient.

Pour cela, les relais sont formés de deux parties principales : une lame métallique et un électro-aimant, d'où le schéma symbolique suivant du relais.

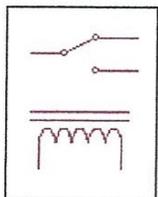


Figure 3 : Symbole électrique d'un relais

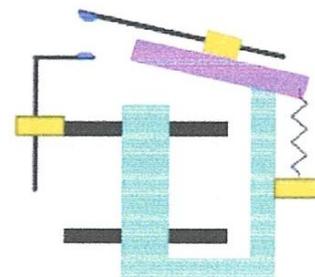


Figure 4 : Différentes parties d'un relais.

3. Fonctionnement du relais :

Lorsqu'un courant parcourt la bobine, il se produit un champ magnétique qui attire l'interrupteur: celui-ci se ferme. On dit alors que le relais est en position "Travail" (T).

Lorsqu'aucun courant ne passe dans la bobine, un petit ressort fait remonter le contact. On dit alors que le relais est en position "Repos" (R).

Si le relais est en position repos, la lame est reliée à un premier contact. Lorsque le relais est en position travail, la lame est dirigée vers un deuxième contact.

Un relais a donc 5 pattes au minimum : 2 pour la bobine, et 3 pour la lame. Une patte pour chaque contact (position travail notée T, position repos notée R et une borne commune : celle qui est reliée au générateur dans l'exemple ci-dessus)

Dans les figures ci-dessous, le courant passe ou ne passe pas : il n'y a qu'un seul circuit, la lame se comporte comme un interrupteur simple.

Mais dans un véritable relais, lorsque la lame est au repos, le courant passe dans une autre direction : la lame se comporte comme un interrupteur inverseur.

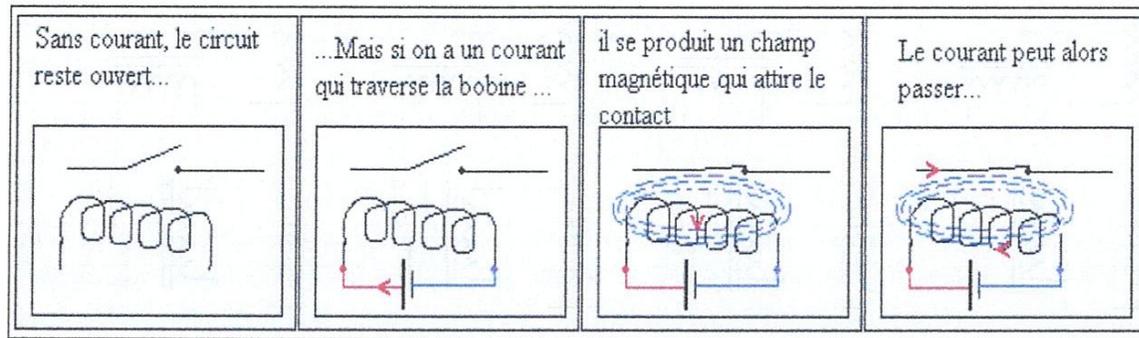
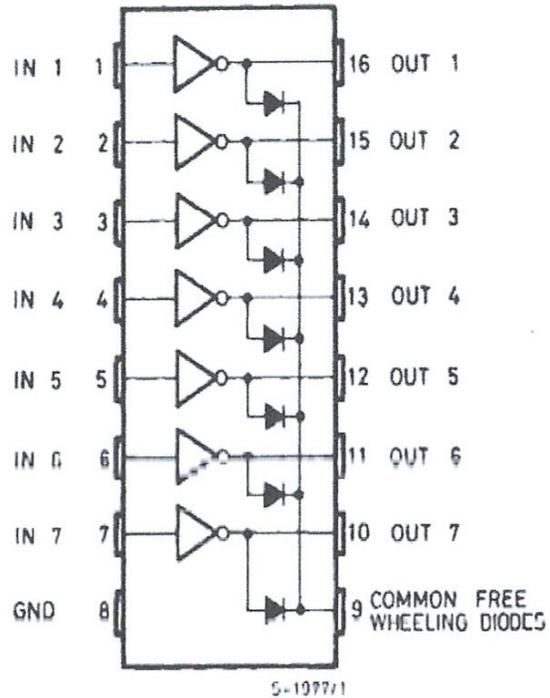


Figure 5 : Le passage du courant dans la bobine crée un champ magnétique dans la bobine ce qui a pour effet d'attirer la lame du relais.

Les Darlington de la série ULN200xA

La série ULN200xA est composée de cinq types de composants, chacun dédié à une application précise; en fonction de la logique avec laquelle il est utilisé. Ils possèdent tous en commun les mêmes caractéristiques:

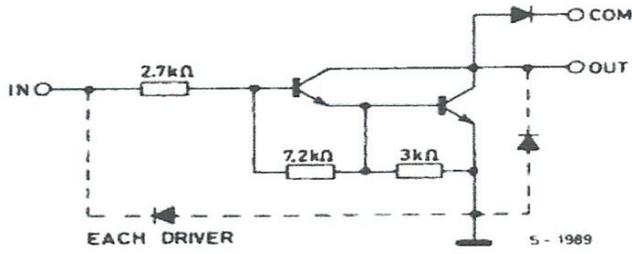
- ils sont proposés en boîtier DIP à 16 broches, dont le brochage et le schéma interne sont donnés ci-contre;
- 7 Darlington internes en émetteur commun;
- le courant de sortie disponible est de 500 mA, et peut atteindre 600 mA en pointe;
- tension d'utilisation de 50V; diodes de protection intégrées;
- les sorties peuvent être mises en parallèle afin de disposer d'un courant plus important;
- les broches d'entrées sont opposées en ligne aux broches de sortie afin de faciliter l'implantation du composant.



Cinq versions sont disponibles afin de simplifier leur interfaçage avec les différentes

familles logiques:

- ULN2001A: utilisations générales avec une résistance de limitation;
- ULN2002A: il dispose d'une résistance d'entrée de 10,5 k Ω et d'une zener pour être compatible avec la logique P1v10S (14 - 25V);
- ULN2003A: c'est l'un des plus utilisés de la série. Il dispose d'une résistance d'entrée de 2,7 k Ω et est compatible TTL et CMOS;
- ULN2004A: sa résistance d'entrée a une valeur de 10,5 k Ω et il est utilisé avec la logique CMOS (6 - 15V);
- ULN2005A: il est conçu pour fonctionner avec la famille TTL standard et TTL SCHOTTKY, familles où un courant plus important est nécessaire.



Le schéma ci-contre représente une des entrées du circuit ULN2003A, tandis que le tableau ci-dessous donne les caractéristiques électriques détaillées de chacun des composants de la famille.

ULN2003 (each driver)

Table 3. Electrical characteristics
($T_A = 25^\circ\text{C}$ unless otherwise specified).

| Symbol | Parameter | Test | Min. | Typ. | Max. | Unit |
|----------------------|--|---|------|------|------|---------------|
| I_{OEX} | Output leakage current | $V_{CE} = 50\text{V}$, (Figure 3.) | | | 50 | μA |
| | | $T_A = 70^\circ\text{C}$, $V_{CE} = 50\text{V}$ (Figure 3.) | | | 100 | |
| | | $T_A = 70^\circ\text{C}$ for ULN2002, $V_{CE} = 50\text{V}$, $V_I = 6\text{V}$ (Figure 4.) | | | 500 | |
| | | $T_A = 70^\circ\text{C}$ for ULN2002, $V_{CE} = 50\text{V}$, $V_I = 1\text{V}$ (Figure 4.) | | | 500 | |
| $V_{CE(SAT)}$ | Collector emitter saturation voltage (Figure 5.) | $I_C = 100\text{mA}$, $I_B = 250\mu\text{A}$ | | 0.9 | 1.1 | V |
| | | $I_C = 200\text{mA}$, $I_B = 350\mu\text{A}$ | | 1.1 | 1.3 | |
| | | $I_C = 350\text{mA}$, $I_B = 500\mu\text{A}$ | | 1.3 | 1.6 | |
| $I_{(ON)}$ | Input current (Figure 6.) | for ULN2002, $V_I = 17\text{V}$ | | 0.82 | 1.25 | mA |
| | | for ULN2003, $V_I = 3.65\text{V}$ | | 0.93 | 1.35 | |
| | | for ULN2004, $V_I = 5\text{V}$ | | 0.35 | 0.5 | |
| | | $V_I = 12\text{V}$ | | 1 | 1.45 | |
| $I_{(OFF)}$ | Input current (Figure 7.) | $T_A = 70^\circ\text{C}$, $I_C = 500\mu\text{A}$ | 50 | 65 | | μA |
| $V_{(ON)}$ | Input voltage (Figure 8.) | $V_{CE} = 2\text{V}$, for ULN2002 $I_C = 300\text{mA}$ | | | 13 | V |
| | | for ULN2003 $I_C = 200\text{mA}$ | | | 2.4 | |
| | | $I_C = 250\text{mA}$ | | | 2.7 | |
| | | for ULN2004 $I_C = 300\text{mA}$ | | | 3 | |
| | | $I_C = 125\text{mA}$ | | | 5 | |
| | | $I_C = 200\text{mA}$ | | | 6 | |
| | | $I_C = 275\text{mA}$ | | | 7 | |
| $I_C = 350\text{mA}$ | | | 8 | | | |
| β_{FE} | DC Forward current gain (Figure 5.) | for ULN2001, $V_{CE} = 2\text{V}$, $I_C = 350\text{mA}$ | 1000 | | | |
| C_i | Input capacitance | | | 15 | 25 | pF |
| $t_{d(on)}$ | Turn-on delay time | 0.5V_I to 0.5V_O | | 0.25 | 1 | μs |
| $t_{d(off)}$ | Turn-off delay time | 0.5V_I to 0.5V_O | | 0.25 | 1 | μs |
| I_S | Clamp diode leakage current (Figure 9.) | $V_A = 50\text{V}$ | | | 50 | μA |
| | | $T_A = 70^\circ\text{C}$, $V_A = 50\text{V}$ | | | 100 | |
| V_F | Clamp diode forward voltage (Figure 10.) | $I_S = 350\text{mA}$ | | 1.7 | 2 | V |

Chacun des Darlington peut dissiper une puissance de 1 W, tandis que la dissipation totale du boîtier est de 2,25 W. Il est donc évident que l'on ne pourra faire débiter en même temps un courant de 500 mA à chacun des transistors ! D'où la possibilité de mettre en parallèle deux ou plusieurs des composants.

Le circuit intégré ULN2003A :



ULN2001A-ULN2002A
ULN2003A-ULN2004A

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (800mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT

DESCRIPTION

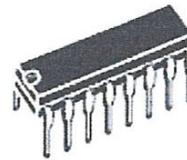
The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 800mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

The four versions interface to all common logic families :

| | |
|----------|---------------------------------------|
| ULN2001A | General Purpose, DTL, TTL, PMOS, CMOS |
| ULN2002A | 14-25V PMOS |
| ULN2003A | 5V TTL, CMOS |
| ULN2004A | 6-15V CMOS, PMOS |

These versatile devices are useful for driving a wide range of loads including solenoids, relays, DC motors, LED displays, filament lamps, thermal print-heads and high power buffers.

The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.



DIP 16

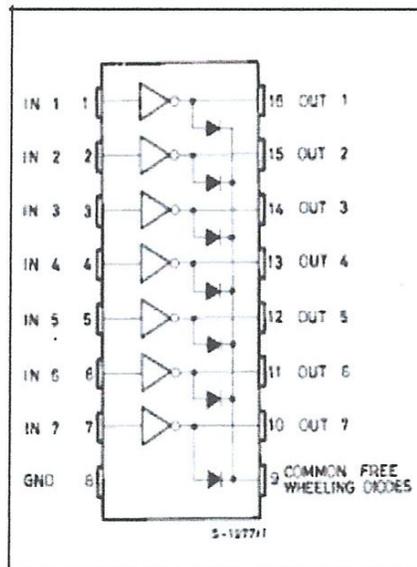
ORDERING NUMBERS: ULN2001A/2A/3A/4A



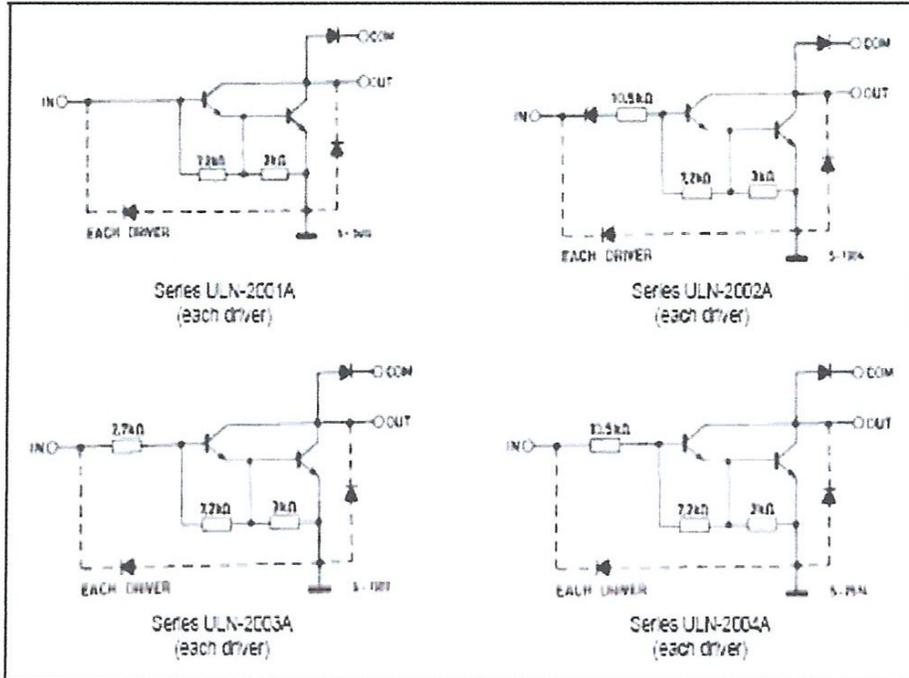
SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

PIN CONNECTION



SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

| Symbol | Parameter | Value | Unit |
|-----------|--|-------------|------|
| V_o | Output Voltage | 50 | V |
| V_{in} | Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D) | 30 | V |
| I_c | Continuous Collector Current | 500 | mA |
| I_b | Continuous Base Current | 25 | mA |
| T_{amb} | Operating Ambient Temperature Range | - 20 to 85 | °C |
| T_{stg} | Storage Temperature Range | - 55 to 150 | °C |
| T_j | Junction Temperature | 150 | °C |

THERMAL DATA

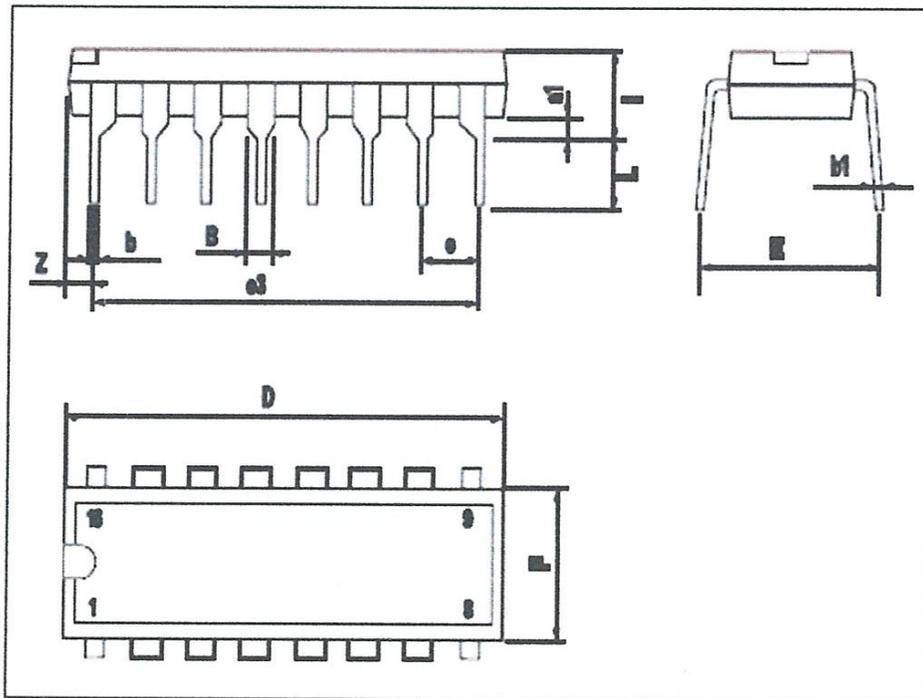
| Symbol | Parameter | DIP16 | SO16 | Unit |
|--------------------|-------------------------------------|---------|------|------|
| $R_{\theta j-amb}$ | Thermal Resistance Junction-ambient | Max. 70 | 120 | °C/W |

| DIM. | mm | | | Inch | | |
|------|------|-------|------|-------|------|-------|
| | MIN. | TYP. | MAX. | MIN. | TYP. | MAX. |
| a1 | 0.51 | | | 0.020 | | |
| B | 0.77 | | 1.65 | 0.030 | | 0.065 |
| b | | 0.5 | | 0.020 | | |
| b1 | | 0.25 | | 0.010 | | |
| D | | | 20 | | | 0.787 |
| E | | 8.5 | | 0.335 | | |
| e | | 2.54 | | 0.100 | | |
| e3 | | 17.78 | | 0.700 | | |
| F | | | 7.1 | | | 0.280 |
| I | | | 5.1 | | | 0.201 |
| L | | 3.3 | | 0.130 | | |
| Z | | | 1.27 | | | 0.050 |

OUTLINE AND MECHANICAL DATA



DIP16



Moteur à courant continu

Dans notre projet, nous avons utilisé un moteur à courant continu qui tourne soit dans un sens (montée) soit dans l'autre (descente) en fonction du sens du courant dans le moteur.

1. Définition :

Le moteur à courant continu est une machine électrique tournante constituée de deux parties principales : le stator (la partie fixe) et le rotor (la partie mobile). L'énergie électrique est transformée en énergie mécanique.

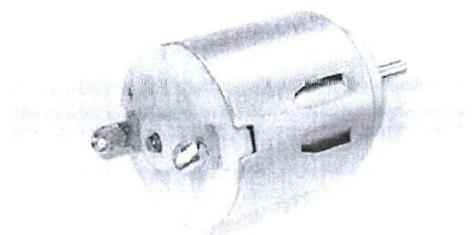


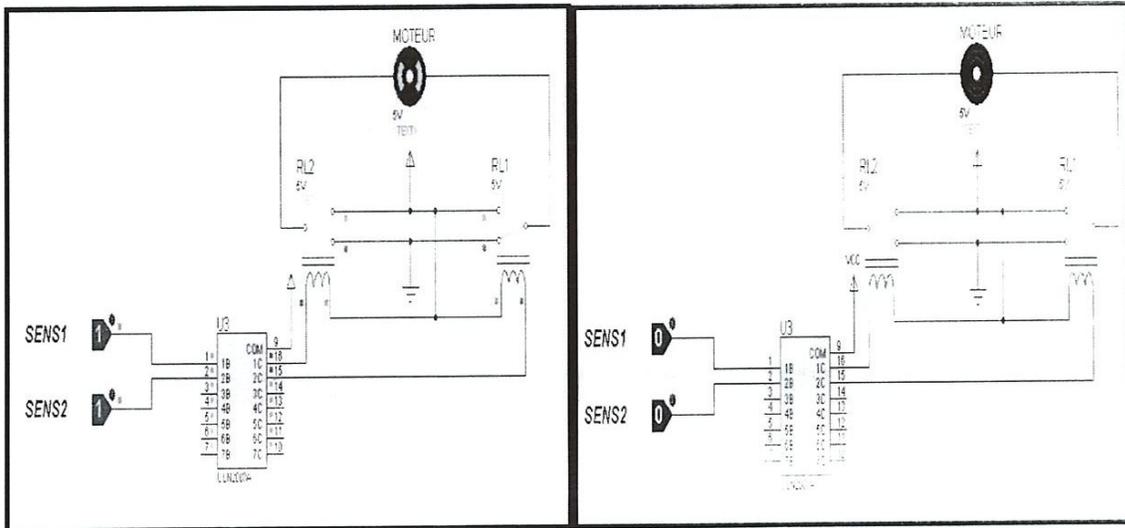
Figure 1 : Photographie d'un moteur à courant continu

2. Fonctionnement :

Le moteur qui fait monter ou descendre la cabine est commandé par le PIC à travers un circuit ULN 2003 et de deux relais. Le PIC utilise les broches RB0 et RD7 pour commander l'arrêt ou la rotation du moteur dans un sens ou un autre.

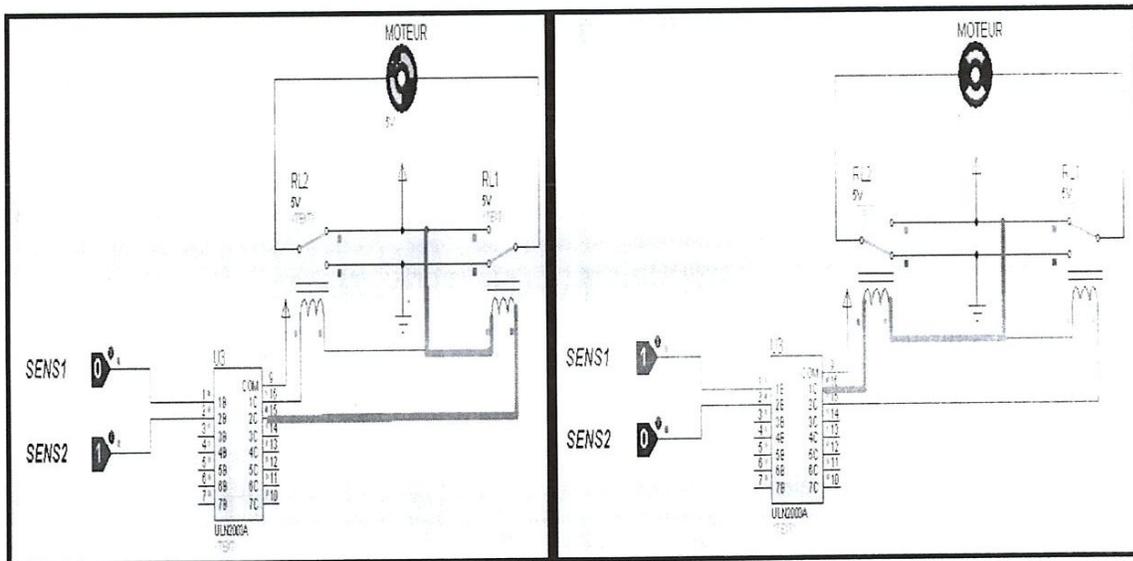
L'écriture d'un 1 logique dans les broches RB0 et RD7 du PIC génèrent des potentiels bas à la sortie des broches 1C et 2C du circuit ULN 2003. Ces niveaux produisent le maintien des lames du relais vers le bas (aucun champ magnétique n'est produit dans les bobines des relais). Le moteur est alors à l'arrêt, figure 2.

Les mêmes résultats sont obtenus lorsque des niveaux logiques bas sont appliqués sur les broches RB0 et RD7 du PIC, figure 3.



Figures 2 et 3 : Moteur au repos

L'application de niveaux logiques complémentaires (niveau haut et niveau bas) sur les broches RB0 et RD7 du PIC provoquent l'excitation d'une seule bobine (1 seul relais excité). Ceci provoque le passage du courant dans le moteur dans un sens ou dans l'autre. Figures 4 et 5.



Figures 4 et 5 : Moteur en marche (sens1 et sens2)

3. Association moteur + réducteur :

Les moteurs à courant continu sont construits pour fonctionner en permanence dans une plage de vitesse proche de leur vitesse à vide.

Cette plage de vitesse est généralement trop élevée pour la majorité des applications. Pour réduire cette vitesse, il est possible d'associer un système de pignons et d'engrenages qui permettent de réduire cette vitesse (solution mécanique).

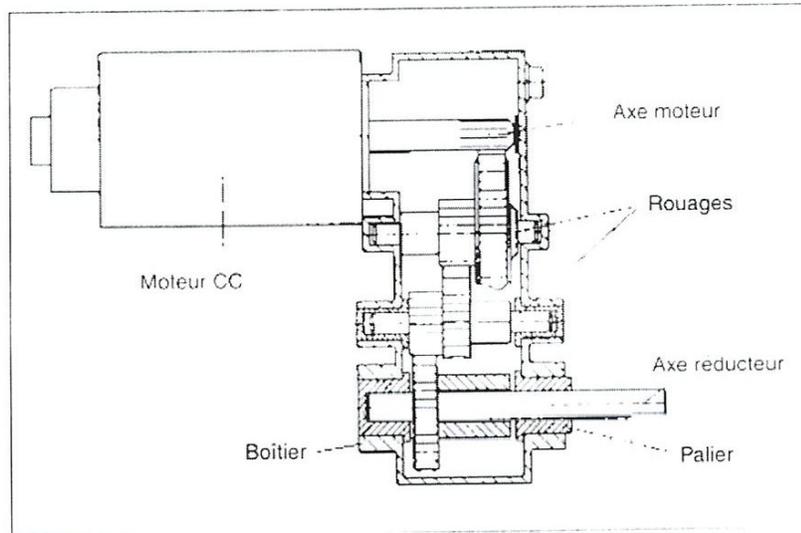


Figure 6: Motoréducteur

Une solution électronique serait de faire varier la durée des impulsions générées par le PIC.

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PIC16F87X Mid-Range Reference Manual, DS90C03.

3.1 PORTA and the TRISA Register

PORTA is a 8-bit wide bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (=1) will make the corresponding PORTA pin an input, i.e., put the corresponding output driver in a high-impedance mode. Clearing a TRISA bit (=0) will make the corresponding PORTA pin an output, i.e., put the contents of the output latch on the selected pin.

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. A write operation is read-modify-write operation. Therefore a write to a port implies that the port pins are read, this value is modified, and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and push-pull output drivers.

Other PORTA pins are multiplexed with analog inputs and analog watch input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register).

Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

```

BCF STATUS, RPO ; Initialize PORTA by
CLR PORTA       ; clearing output
                ; data latches
BCF STATUS, RPO ; Select Bank 1
MOVW 0x0F      ; Value used to
                ; initialize data
                ; direction
MOVWF TRISA    ; Set RA-0:3 as inputs
                ; RA-5:4 as outputs
                ; TRISA<7:6> are always
                ; read as '0'.
    
```

FIGURE 3-1. BLOCK DIAGRAM OF RA3, RA0 AND RA5 PINS

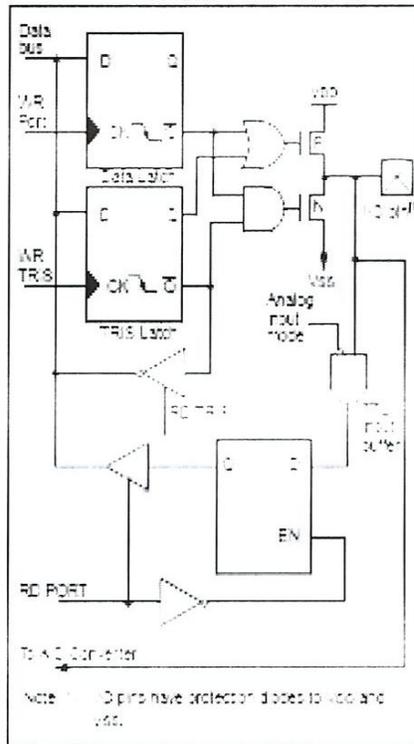
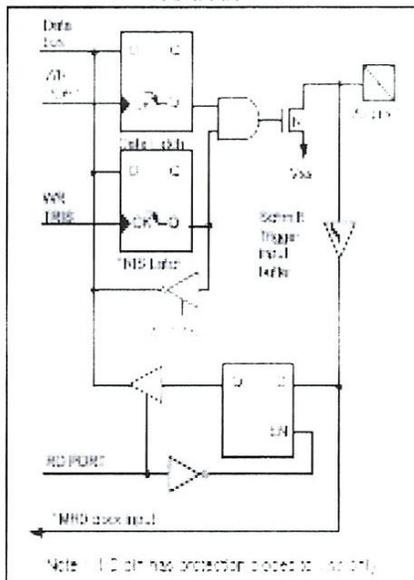


FIGURE 3-2. BLOCK DIAGRAM OF RA4/T0CKI PIN



3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit to '1' will make the corresponding PORTB pin an input. (i.e., put the corresponding output driver in a high-impedance mode.) Clearing a TRISB bit to '0' will make the corresponding PORTB pin an output. (i.e., put the contents of the output latch on the selected pin.)

Three pins of PORTB are multiplexed with the Low Voltage Programming function: RB3/RGM, RB6/RGO and RB7/RGC. The alternate functions of these pins are described in the Special Features Section.

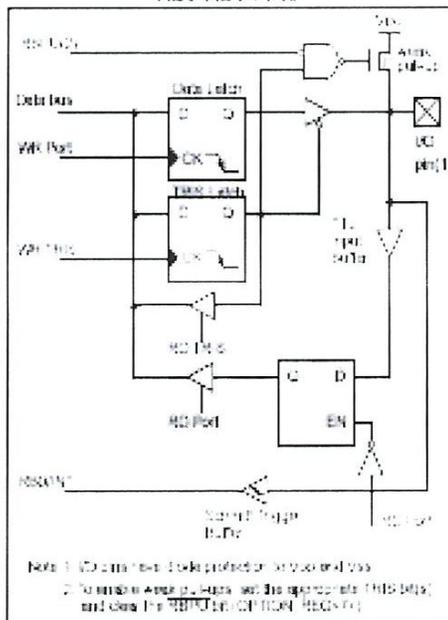
EXAMPLE 3-1: INITIALIZING PORTB

```

007 STATUS, RPO :
007 PORTB      : Initialize PORTB by
                : clearing output
                : data latches
02F STATUS, RPO : Select Bank 1
MOVWF 0x0F     : Value used to
                : initialize data
                : direction
MOVWF TRISB    : Set RB3:0 to an output
                : RB5:4 to an output
                : RB7:6 to an input
    
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBFU (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

FIGURE 3-3: BLOCK DIAGRAM OF RB3:RB0 PINS



Four of PORTB's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur. (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison.) The inputs pins for RB7:RB4 are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

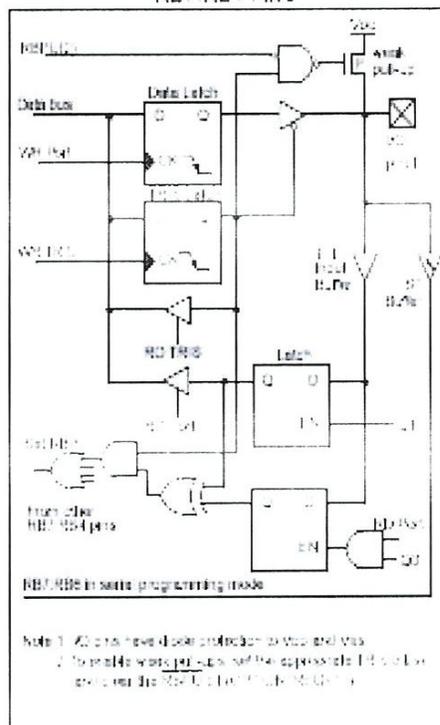
This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

- Any read or write of PORTB. This will end the mismatch condition.
- Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

The interrupt on change feature is recommended for wakeup on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 3-4: BLOCK DIAGRAM OF RB7:RB4 PINS



3.3 PORTC and the TRISC Register

PORTC is an 8-bit wide bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin an input, i.e., put the corresponding output driver in a high-impedance mode. Clearing a TRISC bit (=0) will make the corresponding PORTC pin an output, i.e., put the contents of the output latch on the selected pin.

PORTC is multiplexed with several peripheral functions (Table 3-5). PORTC pins have Schmitt Trigger input buffers.

When the I/O module is enabled, the PORTC (34) pins can be configured with normal I/O levels or with SMBUS levels by using the OKE bit (SSPSTAT<6>).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Once the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (RBF, BCF, XORWF) with TRIS as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

EXAMPLE 3-1: INITIALIZING PORTC

```

BCF  STATUS, RP0      ; Select Bank 0
MOVF  PORTC, WREG     ; Initialize PORTC by
                       ; clearing output
                       ; data latches
BCF  STATUS, RP0      ; Select Bank 1
MOVLW 0x0F           ; Value used to
                       ; initialize data
                       ; direction
MOVF  TRISC, WREG     ; Set RC<3:0> as outputs
                       ; RC<5:4> as outputs
                       ; RC<7:6> as inputs
    
```

FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<0:2> RC<5:7>

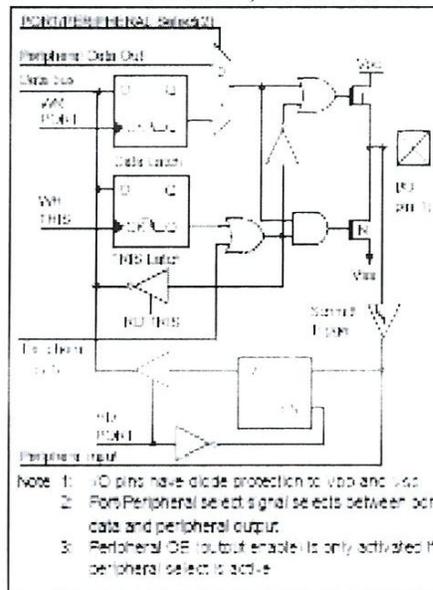
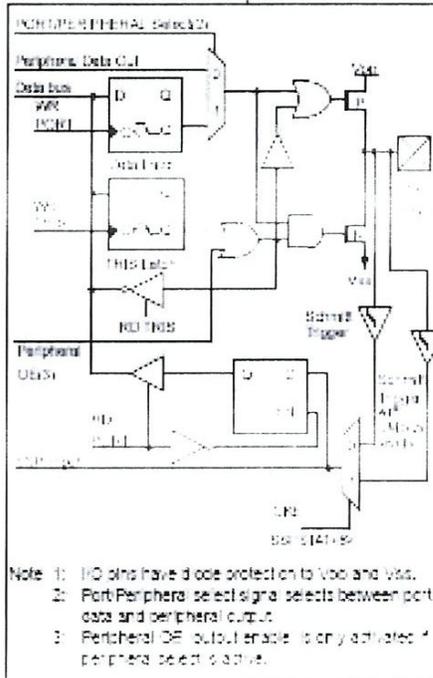


FIGURE 3-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<3-4>



3.4 PORTD and TRISD Registers

This section is not applicable to the 28-pin devices.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTD can be configured as an 8-bit wide microprocessor port (parallel slave port) by setting control bit PSMODE (TRISD<4>). In this mode the input buffers are TTL.

FIGURE 3-7 PORTD BLOCK DIAGRAM (IN I/O PORT MODE)

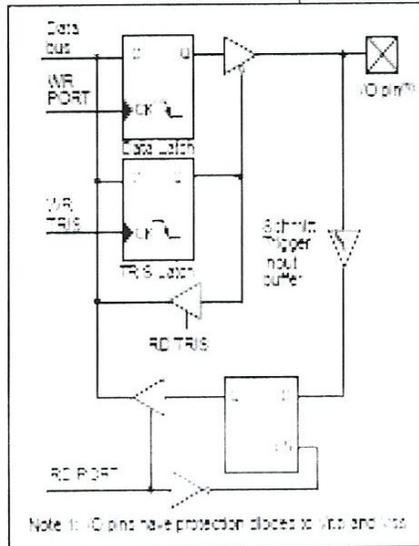


TABLE 3-7 PORTD FUNCTIONS

| Name | Bit# | Buffer Type | Function |
|---------|------|------------------------|--|
| RD0/DP0 | b10 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 0 |
| RD1/DP1 | b11 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 1 |
| RD2/DP2 | b12 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 2 |
| RD3/DP3 | b13 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 3 |
| RD4/DP4 | b14 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 4 |
| RD5/DP5 | b15 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 5 |
| RD6/DP6 | b16 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 6 |
| RD7/DP7 | b17 | Schmitt ⁽¹⁾ | Input/output port pin or parallel slave port bit 7 |

Legend: ST = Schmitt Trigger Input TTL = TTL Input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 3-8 SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other resets |
|---------|-------|-------------------------------|-------|-------|---------|-------|---------------------------|-------|-------|-------------------|---------------------------|
| 5Fh | PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | xxxx xxxx |
| 5Fh | TRISD | PORTD Data Direction Register | | | | | | | | 0000 0000 | 0000 0000 |
| 5Fh | TRISF | BF | CBF | DBF | PSMODEF | — | PORTD Data Direction Bits | | | 0000 1111 | 0000 1111 |

Legend: x = Unknown, u = unchanged, — = Unimplemented read as '0'. Graded cells are not used by PORTC.

PIC16F87X

3.5 PORTE and TRISE Register

This section is not applicable to the 28-pin devices.

PORTE has three pins: REDRESIAN6, RE1/WRIAN6 and REDRESIAN7, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

I/O PORTE becomes control inputs for the microprocessor port when bit P0PMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs). Ensure ADCON1 is configured for digital I/O. In this mode the input buffers are TTL.

Figure 3-8 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. When selected as an analog input, these pins will read as 0.

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

Note: On a Power-on Reset these pins are configured as analog inputs.

FIGURE 3-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)

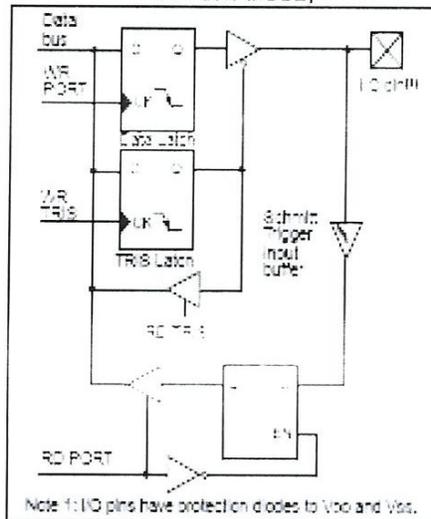


FIGURE 3-9: TRISE REGISTER (ADDRESS 89h)

| R/W | R/W | R/W | R/W | U | R/W | R/W | R/W |
|---------------------------|--|-----|---------|-----|-----|-----|-----|
| BF | OBP | IOV | P0PMODE | — | BI2 | BI1 | BI0 |
| | | | | bit | | | |
| 0x7 | | | | | | | |
| bit 7 | | | | | | | |
| BF | Input Buffer Full Status bit | | | | | | |
| 1 | = A word has been received and is waiting to be read by the CPU. | | | | | | |
| 0 | = No word has been received. | | | | | | |
| bit 6 | | | | | | | |
| OBP | Output Buffer Full Status bit | | | | | | |
| 1 | = The output buffer still holds a previously written word. | | | | | | |
| 0 | = The output buffer has been read. | | | | | | |
| bit 5 | | | | | | | |
| IOV | Input Buffer Overflow Detect bit (in microprocessor mode) | | | | | | |
| 1 | = A write occurred when a previously input word has not been read (must be cleared in software). | | | | | | |
| 0 | = No overflow occurred. | | | | | | |
| bit 4 | | | | | | | |
| P0PMODE | Parallel Slave Port Mode Selection | | | | | | |
| 1 | = Parallel slave port mode. | | | | | | |
| 0 | = General purpose I/O mode. | | | | | | |
| bit 3 | | | | | | | |
| Unimplemented. Read as 0. | | | | | | | |
| PORTE Data Direction Bits | | | | | | | |
| bit 2 | | | | | | | |
| BI2 | Direction Control bit for pin REDRESIAN7 | | | | | | |
| 1 | = Input. | | | | | | |
| 0 | = Output. | | | | | | |
| bit 1 | | | | | | | |
| BI1 | Direction Control bit for pin RE1/WRIAN6 | | | | | | |
| 1 | = Input. | | | | | | |
| 0 | = Output. | | | | | | |
| bit 0 | | | | | | | |
| BI0 | Direction Control bit for pin REDRESIAN6 | | | | | | |
| 1 | = Input. | | | | | | |
| 0 | = Output. | | | | | | |

PIC16F87X

3.6 Parallel Slave Port

The Parallel Slave Port is not implemented on the 28-pin devices.

PORTD operates as an 8-bit wide Parallel Slave Port on microprocessor ports when control bit PPSMODE (TRISB44) is set. In slave mode it is asynchronously readable and writable by the external world through RD control input pin RE0/RD and WR control input pin RE1/WR.

It can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PPSMODE enables port pin RE0/RD to be the RD input, RE1/WR to be the WR input and RE2/OD to be the OD (tri-state) input. For this functionality, the corresponding data direction bits of the TRISB register (TRISB<0:2>) must be configured as inputs (set). And the AD port configuration bits PCFG3:PCFG0 (ADCON1<3:0>) must be set to configure pins RE2/RE0 as digital I/O.

A write to the PSP occurs when both the WR and RD lines are first detected low. A read from the PSP occurs when both the OD and RD lines are first detected low.

FIGURE 3-10: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)

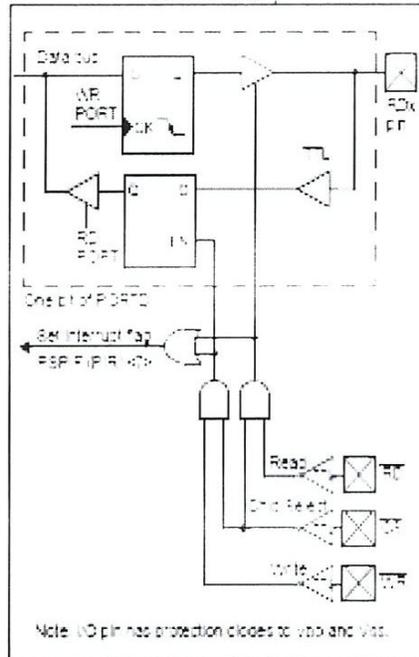
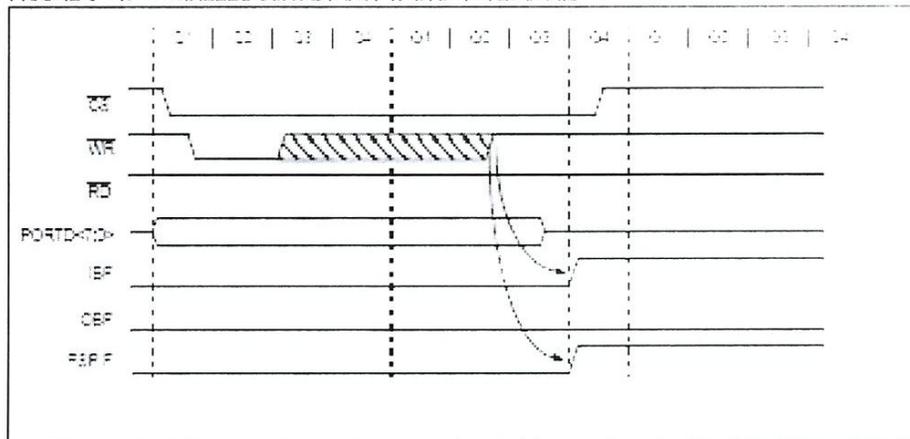


FIGURE 3-11: PARALLEL SLAVE PORT WRITE WAVEFORMS



5.0 TIMER0 MODULE

The Timer0 module timer counter has the following features:

- 8-bit timer counter
- Readable and writable
- Internal or external clock select
- Edge select for external clock
- 8-bit software programmable prescaler
- Interrupt on overflow from FFh to 00h

Figure 5-1 is a simplified block diagram of the Timer0 module.

Additional information on timer modules is available in the PICmicro™ Mid-Range Reference Manual, DS33023.

5.1 Timer0 Operation

Timer0 can operate as a timer or as a counter.

Timer mode is selected by clearing bit T0CS (OPTION_REG<6>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instructions. The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting bit T0CS (OPTION_REG<6>). In counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed below.

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (T0PCL). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

Additional information on external clock requirements is available in the PICmicro™ Mid-Range Reference Manual, DS33023.

5.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a prescaler for the Watchdog Timer (Figure 5-2). For simplicity, the counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The prescaler is not readable or writable.

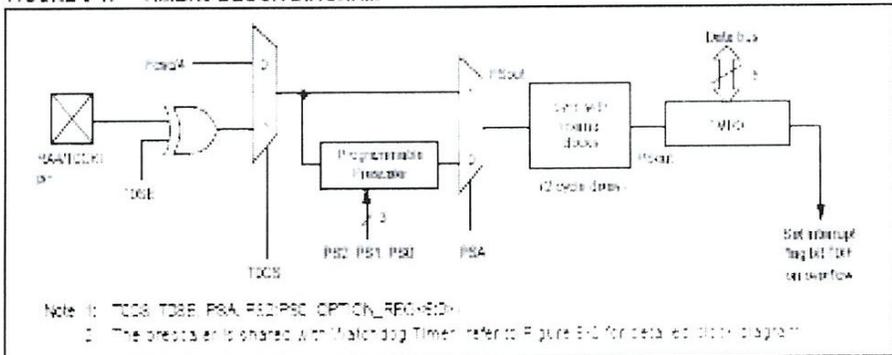
The PSA and PSC bits (OPTION_REG<5:0>) determine the prescaler assignment and prescale ratio. Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable.

Setting bit PSA will assign the prescaler to the Watchdog Timer (WDT). When the prescaler is assigned to the WDT, prescale values of 1:1, 2, ..., 1:128 are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF, MOVWF, MOVF, BSF, CMRCL, etc.) will clear the prescaler. When assigned to WDT, a CLSWDT instruction will clear the prescaler along with the WDT.

Note: Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count, but will not change the prescaler assignment.

FIGURE 5-1: TIMER0 BLOCK DIAGRAM



PIC16F87X

5.1.1 TIMER1 COUNTER OPERATION

In this mode, Timer1 is being incremented via an external source; increments occur on a rising edge after

Timer1 is enabled in counter mode; the module must first have a falling edge before the counter begins to increment.

FIGURE 6-2: TIMER1 INCREMENTING EDGE

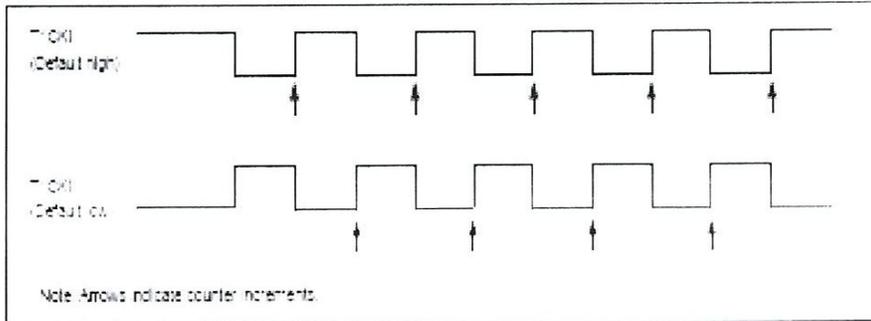
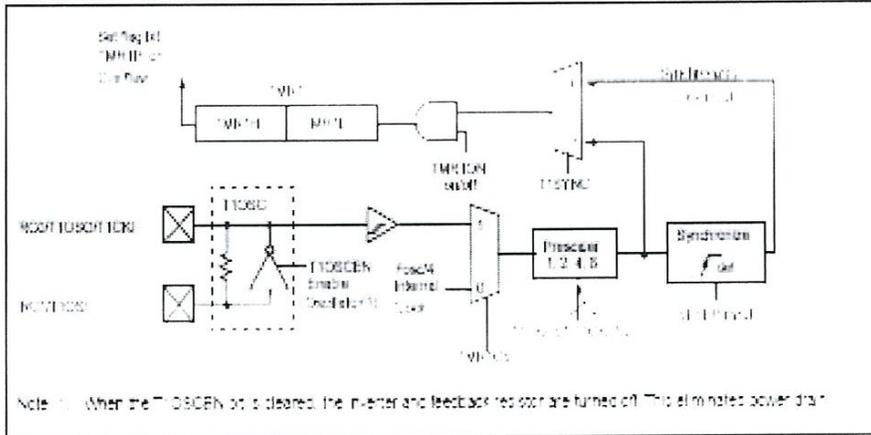


FIGURE 6-3: TIMER1 BLOCK DIAGRAM



PIC16F87X

7.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon reset.

7.3 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module which optionally uses it to generate shift clock.

FIGURE 7-2: TIMER2 BLOCK DIAGRAM

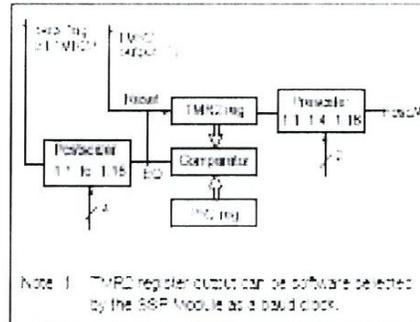


TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

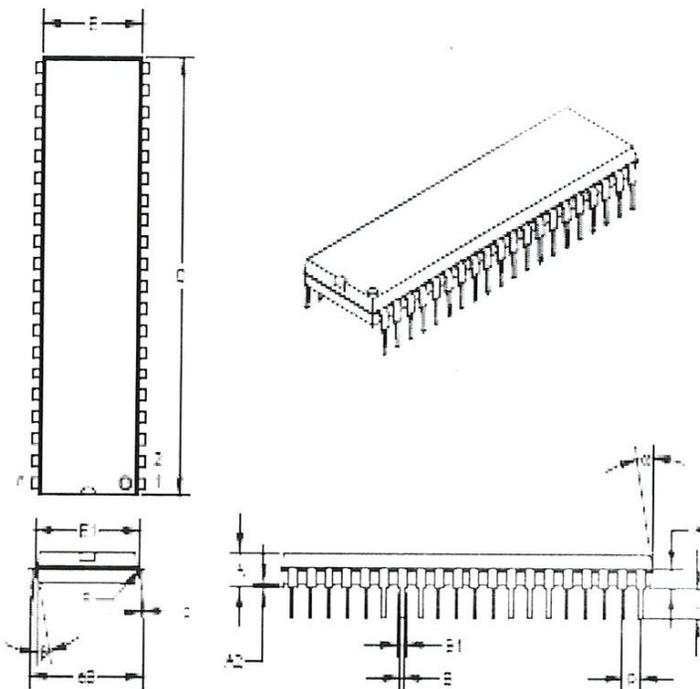
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR | Value on all other resets |
|----------------------|-----------------|------------------------|---------|---------|---------|---------|--------|---------|---------|--------------|---------------------------|
| 40h-50h (PR2-PR5) | PRn (n=2,3,4,5) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00h-00h | 00h-00h |
| 5Ch | PR2 | PSPP ⁽¹⁾ | ADP | RDY | TAPE | SSPTE | CCP1TE | TMR2E | TMR1E | 0000 0000 | 0000 0000 |
| 5Dh | PR3 | PSPP ⁽¹⁾ | ADP | RDY | TAPE | SSPTE | CCP1TE | TMR2E | TMR1E | 0000 0000 | 0000 0000 |
| 11h | TMR2 | Timer2 module register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | 0000 0000 | 0000 0000 |
| 5Eh | PR2 | Timer2 Period Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = Unknown, u = unchanged, — = unimplemented read as '0'. Shaded cells are not used by the Timer2 module.

Note: 1. These bits are reserved on the 16-pin devices. Always maintain these bits clear.

PIC16F87X

17.4 K04-016 40-Lead Plastic Dual In-Line (P) – 600 mil



| Units | INCHES ¹ | | | MILLIMETERS | | |
|------------------------------|---------------------|-------|-------|-------------|------|-------|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| PCB Row Spacing | | 0.600 | | 15.24 | | |
| Number of Pins | n | 40 | | 40 | | |
| Pitch | p | 0.100 | | 2.54 | | |
| Lower Lead Width | B | 0.016 | 0.018 | 0.40 | 0.46 | 0.5 |
| Upper Lead Width | B ² | 0.046 | 0.050 | 0.055 | 1.27 | 1.40 |
| Shoulder Radius | R | 0.000 | 0.005 | 0.010 | 0.00 | 0.13 |
| Lead Thickness | t | 0.006 | 0.010 | 0.01 | 0.03 | 0.05 |
| Top to Seating Plane | A | 0.110 | 0.160 | 0.180 | 0.75 | 4.08 |
| Top of Lead to Seating Plane | A ³ | 0.073 | 0.093 | 0.113 | 1.85 | 3.35 |
| Base to Seating Plane | A2 | 0.020 | 0.020 | 0.040 | 0.51 | 0.61 |
| Tip to Seating Plane | L | 0.125 | 0.130 | 0.135 | 3.18 | 3.30 |
| Package Length | D ⁴ | 0.013 | 0.018 | 0.023 | 0.13 | 0.26 |
| Molded Package Width | E ⁵ | 0.630 | 0.615 | 0.640 | 3.43 | 15.89 |
| Radius to Radius Width | F | 0.645 | 0.655 | 0.665 | 3.34 | 16.35 |
| Overall Row Spacing | eE | 0.630 | 0.610 | 0.610 | 3.35 | 15.43 |
| Mold Draft Angle Top | α | 5 | 0 | 5 | 5 | 0 |
| Mold Draft Angle Bottom | β | 5 | 10 | 5 | 5 | 0 |

¹ Controlling Parameter

² Dimension "B" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.008" (0.152 mm) more than dimension "B".

³ Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimension "D" or "E".