Université de Liège

Aide à la conception de circuits électroniques

Hugues Smeets

Année académique 2001-2002

Table des matières

Chapitre 1

Introduction

La conception de circuits électroniques, qu'ils soient digitaux, analogiques, ou les deux, nécessite, pour donner les meilleurs résultats, le respect de certaines règles.

J'ai remarqué, au cours de mes études à l'Ulg, que ces règles ne sont pas systématiquement données aux étudiants. Pourtant, on leur demande de concevoir et réaliser des circuits électroniques, parfois complexes. Un circuit mal conçu, même simple, ne fonctionnera pas ou, pire, se comportera de manière correcte en général mais, sporadiquement, sera défaillant, sans qu'on puisse trouver facilement la cause du problème...

Le but de ce document est de présenter un ensemble de problèmes classiques qui se posent au concepteur d'un circuit électronique et de donner des solutions éprouvées à ces problèmes. Je tenterai, dans la mesure du possible, de satisfaire tant le lecteur pressé (qui désire une solution facile à mettre en œuvre pour un problème précis mais sans chercher à comprendre tous les détails techniques de cette solution) et le lecteur pointilleux (qui désire plus qu'une solution empirique qui a fait ses preuves en exigeant aussi une base théorique pour comprendre de manière approfondie comment les choses se passent).

Ce tutoriel est fourni dans le cadre du cours de "Systèmes programmés enfouis" donné par Bernard Boigelot. Les étudiants qui suivent ce cours ont des compétences diverses en informatique et en électronique (voir figure 1.1).

Certaines notions abordées ici paraîtront triviales à certains mais ardues à d'autres. Il est difficile dans ce contexte de contenter tout le monde. Bien sûr, toute remarque est la bienvenue si elle peut permettre d'améliorer ce document. Ce dernier ne prétend aucunement se substituer à un cours d'électronique. Il s'agit uniquement d'un complément destiné à vous aider à

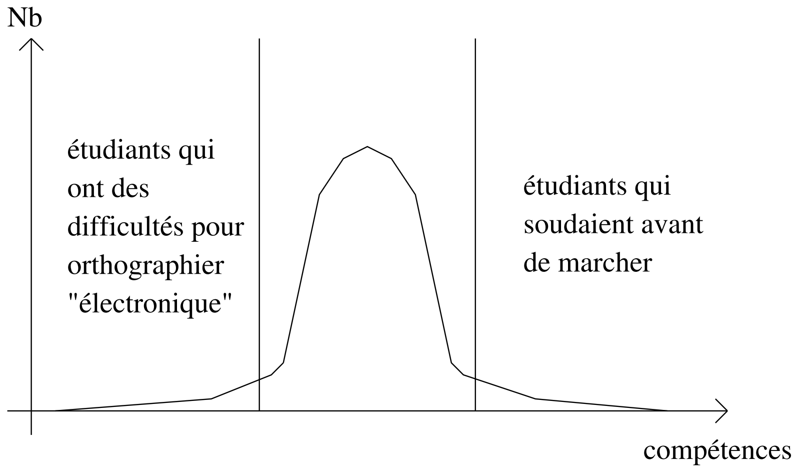

FIG. 1.1 – La répartition du niveau des étudiants

passer des notions théoriques à une réalisation pratique.

Les remarques ou corrections peuvent être envoyées à l'adresse suivante : *smeets@montefiore.ulg.ac.be*

Il ne me reste plus qu'à vous souhaiter de rencontrer de nombreuses difficultés (car c'est dans l'échec qu'on apprend, dit-on) mais aussi de trouver très vite les solutions (car l'Homme a besoin de succès, dit-on aussi), peut-être en partie grâce à ces notes...

Bon travail !

Chapitre 2

La manipulation des composants et leur soudage

2.1 Les décharges électrostatiques

Un individu qui marche sur un tapis peut accumuler des charges électrostatiques dont le potentiel peut atteindre 15 kV, sur une capacité de l'ordre de 100pF [?]. Quand on sait qu'une tension de 40 V peut détruire une couche d'oxyde de 50 nm, on se rend compte qu'une simple manipulation, sans précaution, peut détruire un circuit intégré. La plupart des circuits modernes sont protégés contre les décharges électrostatiques. Cela ne veut pas dire qu'on peut les manipuler impunément ! En effet, les protections intégrées dans les circuits actuels minimisent les risques mais ne les suppriment pas ! De plus, même si on ne détruit pas un circuit intégré, on peut le fragiliser, introduisant dans le système des causes de pannes sporadiques (les pires). Dans le cas des composants programmables, les décharges électrostatiques peuvent être la cause d'une réduction de la durée de rétention des informations par rapport aux spécifications du fabricant (EPROM, Flash,...).

2.2 Les précautions à prendre

Si vous êtes souvent victime de décharges statiques (quand vous sortez d'une voiture ou quand vous serrez la main d'une autre personne, par exemple), vous êtes une personne *à risques* ! Dans ce cas, les précautions qui suivent seront d'autant plus importantes pour vous. Même si vous êtes peu victime de décharges, je vous conseille de les suivre malgré tout !

Par ordre d'importance, les précautions élémentaires à prendre pour limiter les risques de destruction de circuits intégrés lors de leur manipulation sont :

1. limiter les manipulations (ça paraît idiot mais de nombreuses personnes aiment manipuler, sans raison apparente, les objets qui passent à leur portée, ce qui est à proscrire dans le cas de circuits intégrés !).

2. stocker le plus souvent possible les circuits intégrés sur des mousses conductrices antistatiques.

3. utiliser un tapis antistatique avec bracelet, lors des opérations de soudure.

4. travailler dans une pièce dont le sol n'est pas recouvert de tapis plein et utiliser un plan de travail en bois.

5. si on ne dispose pas de tapis, on peut se décharger à une prise de terre quelconque. On peut aussi utiliser de la mousse antistatique, ou, à défaut, un essuie en *coton*, et poser les composants dessus (surtout si le plan de travail est en matière synthétique).

6. porter, lors des opérations de manipulation, des vêtements en coton et éviter les matières synthétiques.

7. éviter de toucher les broches des circuits intégrés (si on doit plier les broches d'un circuit, on peut le tenir par le boîtier et appuyer les broches sur le plan de travail).

2.3 Comment souder correctement les broches des composants

Bien que la technique de soudure de composants s'acquière avec l'expérience, un débutant peut arriver à de bons résultats en respectant quelques règles simples. Premièrement, il convient d'utiliser un fer à souder de puissance adéquate : on utilise un fer de 30 Watts environ pour réaliser des circuits électroniques. La soudure qui convient à ce type de travail est en général constituée de 60 pour cent d'étain et de 40 pour cent de plomb. Dans le futur proche, le plomb devrait être remplacé par un autre métal, vu le contexte actuel de respect de l'environnement.

Ensuite, quand on soude ensemble deux parties, il faut s'arranger pour les amener toutes deux à la même température. Pour cela, on touche les *deux* pièces avec la pointe du fer à souder. On approche alors la soudure de manière à ce qu'elle fonde et coule en recouvrant les deux pièces. Dès que cela se passe, on retire la soudure et le fer (dans cet ordre, sinon la soudure reste collée), pour ne pas prolonger inutilement le réchauffement des composants. Certaines pièces ont besoin d'être étamées pour pouvoir être soudées avec d'autres. C'est le cas des fils de cuivre émaillés, où l'émail sert d'isolant électrique. Dans ce cas, on approche la pointe du fer à souder de l'extrémité du fil qui doit être soudée et on fait fondre l'émail. Puis, on fait fondre un peu de soudure sur le bout de fil. On peut à présent souder ce bout de fil avec un autre élément.

Chapitre 3

L'alimentation

3.1 Construction d'une alimentation simple

Tout circuit électronique doit être alimenté correctement. En général, les circuits sont alimentés sous une tension constante. Pour la plupart des circuits, une alimentation fournissant une simple tension régulée suffit. En particulier, les circuits digitaux basés sur des composants TTL[1] se contentent d'une alimentation de 5 Volts. Ces composants TTL, qui sont les circuits logiques les plus répandus, exigent une tension d'alimentation qui se situe dans la plage 4,75 Volts - 5,25 Volts. D'autres circuits logiques sont plus tolérants (2 Volts - 16 Volts, par exemple) mais je vous conseille d'utiliser des composants logiques qui fonctionnent avec une alimentation de 5 Volts, car la plupart des circuits logiques de base (portes logiques, bascules,...) sont disponibles dans cette version et surtout, la plupart des microcontrôleurs et circuits périphériques sont conçus pour fonctionner avec une telle tension d'alimentation.

Une solution simple existe pour réaliser une telle alimentation. Elle est illustrée sur la figure 3.1.

3.1.1 Le module secteur

Le rôle du module secteur est de transformer la tension alternative du secteur (220V alternatifs ou “AC”) en une basse tension stabilisée quasi continue (ou “DC”). Cette tension n'est pas *régulée* et vaut de 8 à 16 Volts environ. Cette valeur dépend du module. Sur certains d'entre-eux, elle peut être choisie à l'aide d'un sélecteur. On dit de la tension de sortie de ce module qu'elle

[1]Transistor Transistor Logic : les entrées et les sorties des circuits logiques de cette famille sont construits à l'aide de transistors, par opposition aux circuits DTL (obsolètes) dont les étages d'entrées sont réalisés à partir de diodes

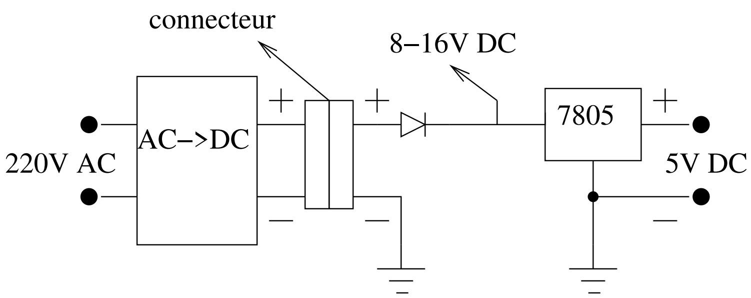

FIG. 3.1 – Synoptique d'une alimentation 220V AC vers 5V DC

est stabilisée [2] parce qu'elle subit des variations relativement faible (mais malgré tout trop importantes pour alimenter les circuits logiques).

L'autre caractéristique importante d'un module est le courant maximal qu'il peut fournir. Cette valeur est très importante. Vous devrez avoir une idée de la consommation totale de votre circuit avant de vous procurer un module secteur. Mais rassurez-vous, les fiches de caractéristiques des composants électroniques vous permettront facilement de déterminer la consommation de votre circuit.

Le lecteur qui désire uniquement réaliser une alimentation fiable le plus rapidement possible peut passer cette section.

Ce qui se cache derrière un module secteur se trouve sur la figure 3.2.

Le transformateur abaisse la tension du secteur selon le rapport du nombre de ses spires au primaire et au secondaire. De plus, il isole le secondaire galvaniquement par rapport à la tension du secteur, empêchant tout risque d'électrocution. La tension alternative qui apparaît au secondaire est redressée par le pont de diodes. Ensuite, un condensateur de lissage est utilisé pour stabiliser cette tension redressée. Si vous deviez réaliser vous-même un tel module, voici comment dimensionner les différents composants selon vos besoins :

1. la tension de sortie est déterminée par le transformateur et vaut

$$220/U_{sortie\,eff} = \frac{Nb_{spires\,au\,primaire}}{Nb_{spires\,au\,secondaire}},$$

[2] souvent, le terme "stabilisé" et "régulé" sont utilisés pour décrire la même chose. Ici, nous dirons qu'une tension est *stabilisée* si elle est relativement stable mais dépend du courant fourni. Une tension *régulée*, par contre, doit être pratiquement indépendante du courant fourni par l'alimentation.

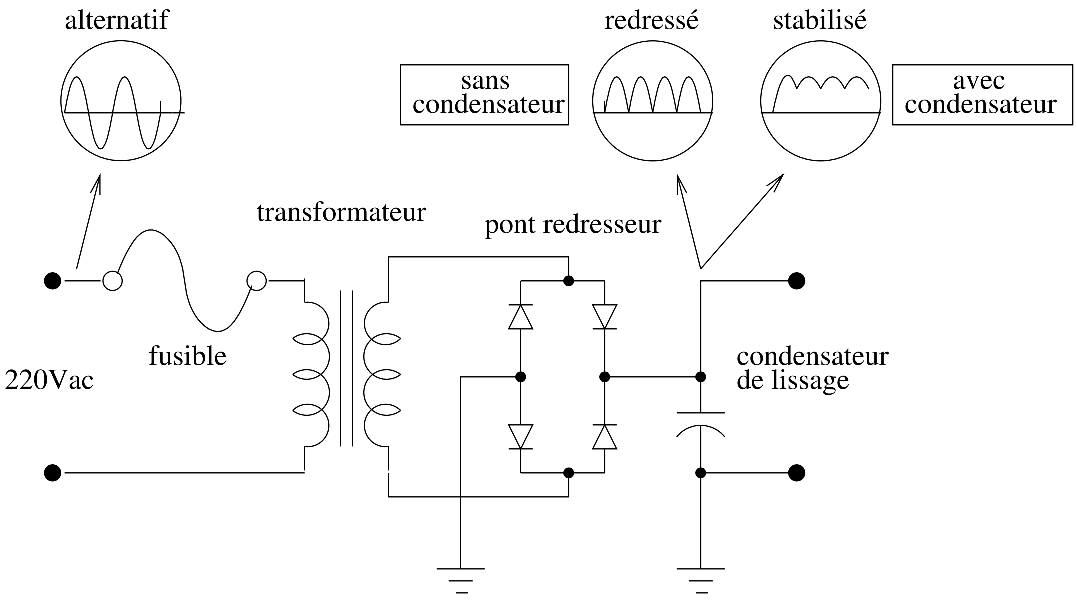

FIG. 3.2 – Schéma d'un module secteur

mais nous n'avons pas à nous préoccuper de bobinage car la tension de sortie est indiquée sur le transformateur. Le fabriquant indique la tension efficace qui vaut :

$$\sqrt{2}\,U_{crête\ à\ crête}.$$

Les résistances série[3] des bobines du transformateur sont loin d'être négligeables, ce qui entraîne une diminution de la tension de sortie aux courants importants. Le fabriquant garantit que son transformateur fournira un signal de tension efficace donnée à la puissance maximale qu'il peut fournir,

$$P_{max} = U_{sortie\,max\,eff}\,I_{sortie\,max\,eff}.$$

A faible puissance, la tension de sortie peut donc être sensiblement plus importante que prévu...

2. Le courant maximal que doit pouvoir fournir le module détermine le courant maximal qui doit passer par les diodes :

$$I_{max\,diode\,eff} = \frac{I_{max\,module\,eff}}{2}\,;$$

[3]la bobine réelle peut être modélisée par une self idéale en série avec un résisteur idéal.

en effet, les diodes ne sont passantes qu'une alternance sur deux, donc elle ne doivent supporter, chacune, que la moitié du courant de sortie.

3. Le fusible est placé dans la partie 220 Volts de manière à éviter aussi le risque d'incendie qui serait dû à un court circuit entre des spires du primaire du transformateur. Il doit laisser passer

   $$220\frac{U_{sortie\,max\,eff}}{I_{sortie\,max\,eff}}.$$

   Les fusibles existent en deux versions : à action retardée ou rapide. Ici, il faut prendre un fusible lent car l'inductance de la bobine du primaire provoque, lors de l'allumage et de l'extinction du montage, des pointes de courant qui dépassent largement les spécifications du système en fonctionnement nominal.

4. La valeur du condensateur de lissage se calcule facilement. Si on imagine que la charge consomme 1 Ampère en permanance, le condensateur se décharge à courant constant. Vu la faible résistance des diodes du pont redresseur, il se charge très vite. La tension aux bornes du condensateur évolue alors en “dents de scie” : elle monte brutalement puis, elle descend doucement. . .Ensuite, le cycle recommence. Le temps de la décharge est environ de 10 millisecondes, car la tension alternative du secteur, qui a une période de 20 millisecondes, a été redressée et donc, vu la forme du signal, sa fréquence a été doublée. La différence de charge aux bornes du condensateur se calcule comme suit :

   $$\triangle Q = C\triangle V,$$

   où V est la tension aux bornes du condensateur et C est sa capacité. De plus, la différence de charge stockée dans le condensateur vaut

   $$\triangle Q = I\triangle T,$$

   où I est le courant qui “traverse” le condensateur et $\triangle T$ le temps de la décharge, ici de 10 millisecondes. Si on veut, par exemple, une tension de ronflement $\triangle V$ de 3 Volts, le condensateur doit avoir une valeur qui est supérieure ou égale à 3300 microfarads par ampère du courant de sortie.

3.1.2 Le connecteur de liaison

Le lecteur pressé peut reprendre sa lecture ! Revenons à la figure 3.1.

Pour rendre votre réalisation modulaire, il convient de la doter d'un connecteur destiné à brancher le module secteur au reste du montage. Le choix du type de connecteur vous est laissé. Pour éviter toute erreur de connexion (connecteur placé à l'envers, si cela est possible), on intercale une diode entre la sortie positive du module et le reste de l'alimentation. Ainsi, si une inversion de polarité malencontreuse survient, le montage ne fonctionnera pas, tout simplement. En l'absence de cette diode, de nombreux composants de votre montage partiraient en fumée en cas d'inversion des bornes positive et négative du module secteur...

3.1.3 Le régulateur de tension

Pour réguler la tension qui provient du module secteur, on utilise souvent un circuit intégré spécialement conçu à cet effet : le *régulateur de tension positive fixe* "7805". Il appartient à une famille de circuits qui ne diffèrent que par leur tension de sortie, indiquée par les deux derniers chiffres de leur nom. le 78*05* est donc un régulateur qui fournit une tension de sortie régulée de 5 Volts à partir d'une tension d'entrée stabilisée mais non régulée dont la valeur doit être comprise entre 8 Volts et 35 Volts.

Encore une fois, vous pouvez passer la fin de cette section si vous manquez de temps.

Le schéma de principe d'un régulateur de tension classique est donné en figure 3.3.

L'amplificateur opérationnel (représenté par un triangle) est un circuit intégré qui amplifie la *différence* de potentiel qui apparaît à ses broches "+" et "-". Ce qui est important est de noter que le signe de cette différence est pris en compte : quand la tension sur la broche "+" devient plus élevée que la tension sur la broche "-", la tension à la sortie de l'amplificateur *augmente* et dans le cas contraire, elle diminue...

La tension de sortie commande la base du transistor de puissance qui reproduit cette tension (à 0,6 Volts près) sur son émetteur. Une fraction de cette tension est appliquée à la broche "-" de l'amplificateur. La broche "+" est maintenue à une tension constante (c'est la tension au bornes de la diode *zener*, qui varie fort peu.

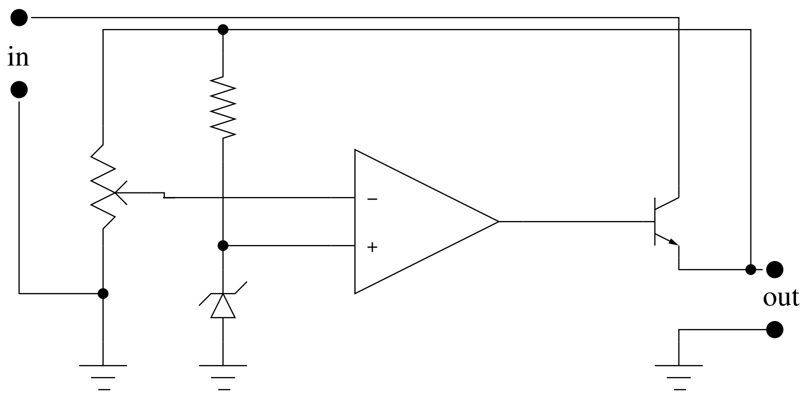

Fig. 3.3 – Schéma d'un régulateur de tension classique

A présent, imaginez que la charge (non représentée sur le schéma), consomme plus de courant. A cause de la résistance des câbles de liaison, la tension de sortie de l'alimentation va avoir tendance à diminuer. La tension sur la broche "-" diminue alors et provoque une augmentation de la tension de sortie, ce qui compense l'effet initial. On voit apparaître le phénomène de régulation. Si la tension de sortie tend à augmenter, la tension de sortie de l'amplificateur va diminuer, ce qui va ramener la tension de sortie de l'alimentation à sa valeur de consigne.

Un 7805 comporte un circuit similaire. De plus, il est aussi doté de plusieurs systèmes de protection (surchauffe, court-circuit, ...).

Si les amplificateurs opérationnels et les régulateurs de tension vous intéressent, deux excellentes références sont [3] et [4].

3.2 Le calcul du radiateur de l'alimentation

Le 7805 est soumis à une différence de potentiel qui vaut $U_{entre} - 5\,Volts$, il doit donc dissiper une puissance de $(U_{entre} - 5\,Volts)\,I_{sortie}$, par effet Joule. Le courant maximal qu'il peut fournir est de 1 Ampère (cette valeur est donnée dans la fiche de caractéristiques du 7805). Imaginons que votre module secteur fournisse un tension stabilisée de 12 Volts, par exemple. Le 7805 devra alors dissiper $(12\,V - 5\,V)\,1A = 7\,W$. La figure 3.4 illustre cette situation.

La température maximale que peut atteindre la puce du 7805 est donnée dans la fiche de caractéristiques et vaut 125^oC environ. La puissance dissipée par cette puce doit être évacuée de manière correcte vers l'air ambiant pour que sa température ne dépasse pas les 125^oC fatidiques. Pour calculer

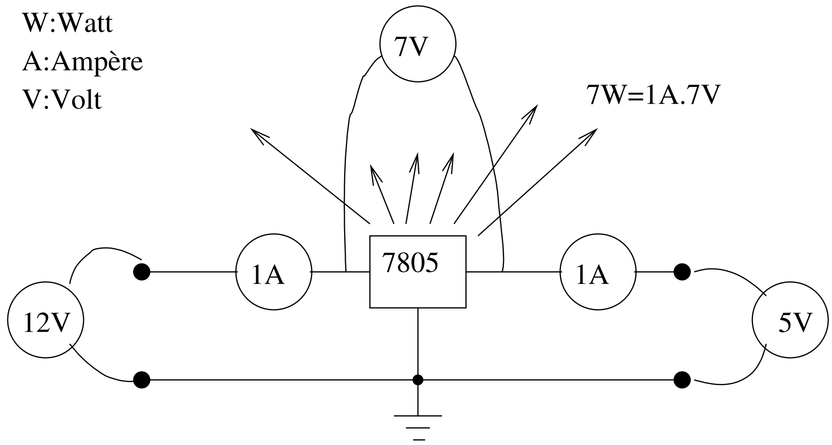

FIG. 3.4 – Courant, tension et dissipation !

la résistance thermique du radiateur à fixer sur le 7805, on procède de la manière suivante :

1. On suppose que la jonction a une température maximale de 125^oC. On calcule ensuite la température entre le boîtier du 7805 et l'extérieur, qui dépend de la résistance thermique du boîtier ($5^oC/W$, d'après la fiche de caractéristiques). La température vaut donc $125^oC - (7W\ 5^oC/W) = 90^oC$.

2. De la même manière, on calcule la température de la zone de contact du radiateur et du boîtier du 7805. Le 7805 n'est pas mis directement en contact avec le radiateur ; on améliore le transfert de chaleur si on place un peu de pâte thermoconductrice entre les deux. On considère en général de manière empirique que cette façon de procéder introduit une résistance thermique de $1^oC/W$ sur le trajet du flux de chaleur. La température de la zone de contact du radiateur et du boîtier du 7805 est donc $90^oC - (7W\ 1^oC/W) = 83^oC$.

3. Pour connaître maintenant la résistance thermique *maximale* du radiateur, il suffit de calculer $(83^oC - 30^oC)/7W = 7.57^oC$.

Des radiateurs ayant cette caractéristique existent dans le commerce et sont faciles à trouver. Le schéma de la figure 3.5 illustre comment la puissance à dissiper est évacuée.

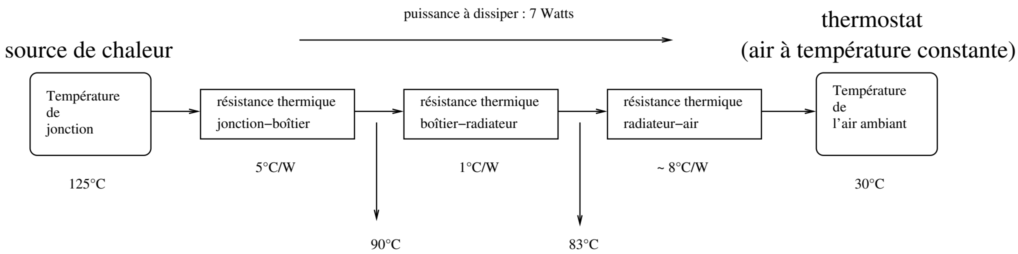

FIG. 3.5 – Dissipation de puissance à l'aide d'un radiateur

3.3 Le découplage de l'alimentation

3.4 Surveillance de la tension fournie par l'alimentation

3.5 Un exemple

Chapitre 4

Le microcontrôleur PIC 16F877

Avant tout, je vous conseille vivement de lire la documentation réalisée par le constructeur[1] concernant ce circuit [1]. Comme le document cité contient la plupart des informations nécessaires pour programmer un PIC avec succès, nous nous contenterons ici d'un survol, destiné aux personnes qui ne connaissent pas ce circuit. Néanmoins, quelques informations intéressantes concernant certaines particularités du PIC, et non présentes dans la documentation, seront fournies.

4.1 Architecture

Dans le cadre du cours de Systèmes Programmés Enfouis, nous utiliserons ce microcontrôleur pour illustrer les applications typiques de ce genre de circuit. Avant d'en arriver là, il peut être intéressant de jeter un coup d'œil sur son architecture (figure 4.1).

Le PIC est un microcontrôleur 8 bits à architecture pseudo-RISC[2]. Pratiquement toutes les instructions, codées sur un mot de 14 bits, s'exécutent en 4 cycles d'horloge. Le processeur connaît un jeu de 35 instructions (d'où le terme RISC). Cette faible quantité d'instructions a permis de réaliser un processeur simple, qui exécute les instructions en un faible nombre de cycles. Malheureusement, l'impact d'une telle approche a quelques inconvénients : comme le code opératoire d'une instruction doit tenir dans 14 bits, les champs d'adresse qui permettent d'accéder aux différentes zones de mémoire du processeur ont une taille limitée. Des bits supplémentaires sont donc présents dans des registres spéciaux pour compléter ces adresses. Le programmeur

[1]Microchip ; la documentation concernant le 16f877 se trouve à l'adresse suivante : http ://www.microchip.com/1010/pline/picmicro/families/16f87x/devices/16f877/index.htm

[2]Reduced Instruction Set Computer. En fait, chaque instruction prend quatre cycles d'horloge sauf les instructions de saut qui prennent malgré tout deux cycles du processeur, ou 8 cycles d'horloge.

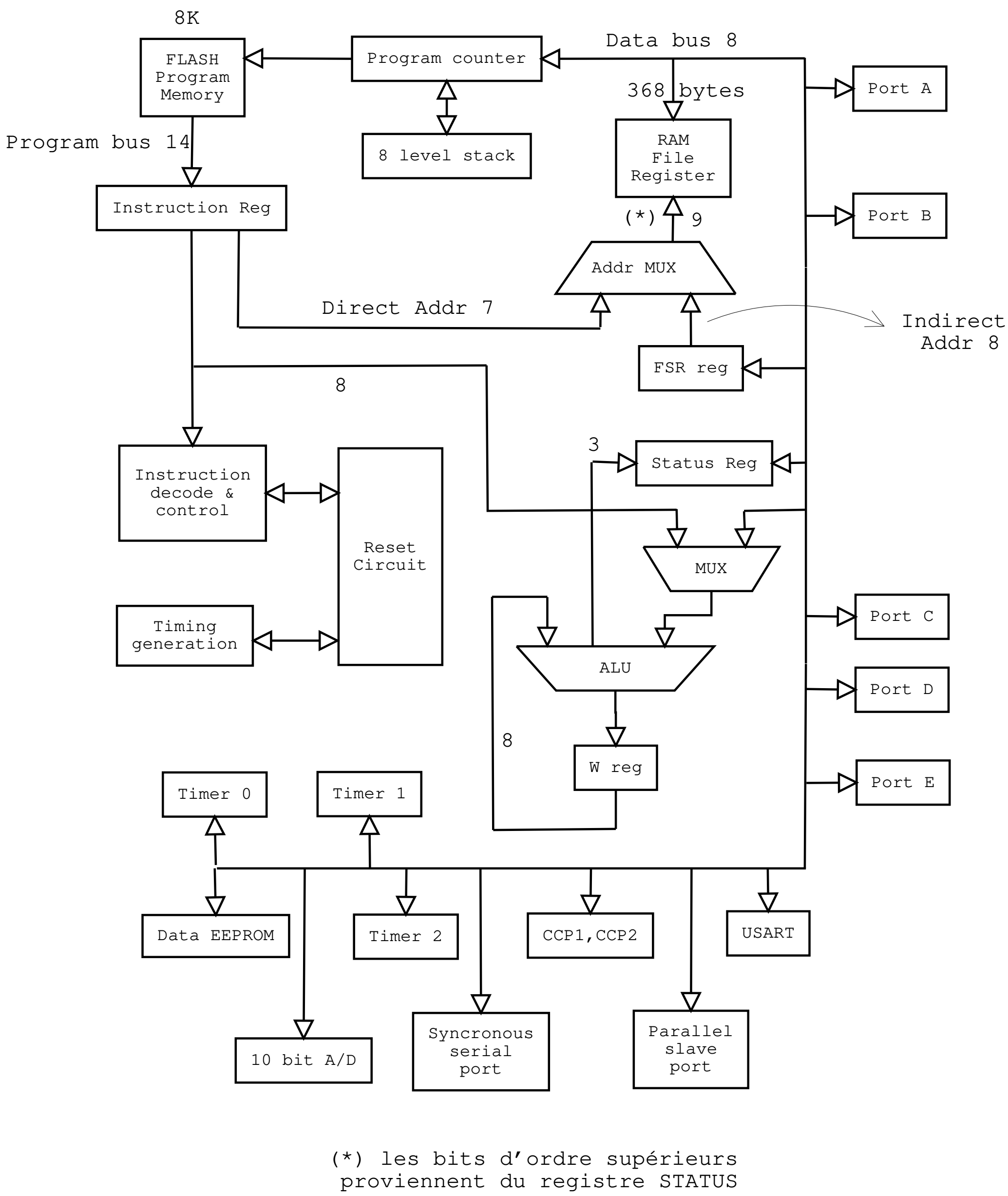

FIG. 4.1 – Architecture du PIC 16F877 de Microchip

doit modifier ces bits de manière à sélectionner le bon *banc* de mémoire lorsqu'il veut accéder à une variable (RAM) où à une sous-routine (ROM).

D'un point de vue logique, il y a deux zones de mémoire dans ce circuit : la mémoire de programme et la mémoire de données. On est donc en présence d'une architecture de type "Harvard" : les deux zones sont indépendantes et peuvent être adressées pendant le même cycle d'horloge.

Comme nous sommes en présence d'un microcontrôleur, des périphériques sont aussi présents sur la puce du microprocesseur : des compteurs/temporisateurs, un chien de garde, une logique de génération d'interruption, une EEPROM de 64 octets et des circuits d'entrée/sortie,...

4.1.1 La mémoire de programme

Le compteur de programme du PIC (PC), a une taille de 13 bits : il permet donc d'adresser une mémoire de 8192 mots. Les mots de la mémoire de programme ont une taille de 14 bits : chaque mot contient le code opératoire d'une instruction. Sur le 16F877, les 8192 mots sont physiquement présents.

Cette mémoire est de type FLASH, c'est-à-dire qu'elle peut être reprogrammée électriquement. Un système de programmation *in situ* est présent sur la puce du PIC. On peut donc reprogrammer le circuit sans l'extraire de la carte d'application sur laquelle il se trouve.

Les instructions de saut (GOTO) et d'appel de sous-routine (CALL) contiennent dans leur code opératoire un champ de 11 bits pour déterminer l'adresse du saut. Comme le compteur de programme fait 13 bits, les deux bits supplémentaires sont chargés à partir des bits 4 et 3 d'un registre spécial : PCLATH. Le registre PCLATH[3] est utilisé quand la cible d'une instruction est la partie basse de PC : PCL. Ce registre étant constitué de 8 bits, la partie haute de PC (PCH) est chargée à partir des 5 bits de poids faible de PCLATH.

4.1.2 La mémoire de données

Cette mémoire est constituée de quatre bancs, chacun permettant l'accès à 128 adresses. Les registres à fonction spéciales et des zones de RAM se partagent ces adresses. De nombeuses zones ne contiennent physiquement rien. L'utilisateur dispose de 368 octets de RAM qui sont *mappés* dans les quatre bancs. Cette mémoire peut être adressée de manière directe ou indirecte. Deux bits dans le registre spécial STATUS fournissent l'éventuelle partie manquante de l'adresse.

4.1.3 la pile matérielle

Le microprocesseur dispose d'une pile matérielle à 8 niveaux. Cette pile ne peut mémoriser que des adresses de programme et n'est accédée que lors

[3]PCLATH signifie Program Counter Latch High

de l'exécution d'une instruction CALL, de l'appel de la routine de traitement d'interruption, de l'exécution d'une instruction RETURN (retour d'une sous-routine), RETFIE (retour de la routine de traitement des interruptions) ou RETLW (retour d'une sous-routine avec chargement d'une valeur immédiate dans l'accumulateur (W)). Dans les deux premiers cas, l'adresse de l'instruction qui suit celle qui est en cours d'exécution est placée au sommet de cette pile. Dans les trois derniers cas, l'adresse qui est au sommet de la pile est placée dans le registre PC.

Cette pile n'est pas présente dans l'espace d'adressage du PIC. Le programmeur n'a donc aucun moyen d'accéder à son contenu.

4.1.4 Les interruptions

Il n'y a, d'une certaine manière, qu'une seule interruption matérielle sur le PIC. Plusieurs périphériques peuvent la déclencher et le programme de traitement de l'interruption devra interroger les différents périphériques pour savoir lequel a réellement déclenché l'interruption. Si plusieurs périphériques demandent à être servis en même temps, l'ordre des tests présents dans la routine de traitement leur donnera un priorité. Les interruptions sont d'office interdites lors de l'appel de la routine de traitement. Si l'utilisateur le désire, il pourra réactiver ces interruptions explicitement au début de cette routine de traitement. Si on veut réellement que des périphériques aient des priorités différentes, on devra procéder de la sorte. En effet, imaginons qu'un périphérique à faible priorité soit en train d'être servi (le test qui lui correspond dans la routine de traitement d'interruption est placé *après* les tests de périphériques plus prioritaires). Si pendant ce temps, un périphérique plus prioritaire génère une interruption et que les interruptions sont interdites, il ne sera servi qu'après la terminaison de la routine de traitement d'interruption du périphérique qui est en train d'être servi.

Il faut noter ici une particularité importante du PIC : le drapeau de demande de service d'un périphérique est positionné, le cas échéant, indépendamment de l'état du masque d'interruption de ce périphérique. Si, par exemple, le temporisateur déborde, il positionnera son drapeau de débordement, même si il n'est pas autorisé à générer une interruption (et il n'en génèrera d'ailleurs pas). Le code C suivant, qui pourrait être utilisé pour programmer une routine de traitement d'interruption, est donc faux :

```
void interrupt(void) {

  if (TIMER_FLAG) {    /* interruption du timer ? */
```

```
    handleTimer();     /* traite la demande du timer */
    return;
  }
  if (OTHER_PERIPH_FLAG)
...
}
```

Il faut écrire :

```
void interrupt(void) {

/* il faut ensuite vérifier que le timer
   peut déclencher une interruption : */

  if (TIMER_FLAG && (TIMER_INTERRUPT_FLAG == ENABLED)) {
    handleTimer();
    return;
  }
  if (OTHER_PERIPH_FLAG)
...
}
```

J'ai bien sûr été victime de ce problème et, bien que des avertissements figurent en grand nombre dans la documentation de Microchip, on ne pense pas toujours à y faire attention !

4.1.5 Les périphériques

Voici les périphériques présents sur la puce du 16F877 :

1. Un temporisateur 8 bits : il s'agit d'un compteur 8 bits qui peut déclencher une interruption lorsque sa valeur passe de 0xFF à 0. L'horloge qui incrémente ce compteur peut être interne ou externe. Un prédiviseur est disponible ; cependant, il est utilisable soit par le temporisateur soit par le chien de garde.

2. Un chien de garde : c'est un compteur autonome qui déclenche une réinitialisation du processeur lorsque son contenu déborde. Le programme de l'application doit le réinitialiser périodiquement pour éviter que cette condition n'apparaisse. Si, pour une raison ou une autre, le programme ne se comporte pas comme prévu (parasite sur les lignes

d'alimentation, erreur dans le programme,...), il y a de grandes chances pour que le chien de garde ne soit pas réinitialisé, son débordement replacera le système dans un état connu (RESET). Quand le PIC sort de la chaîne de fabrication, le chien de garde est activé. Si vous ne voulez pas l'utiliser, pensez à le désactiver (WDT_ON à 0 dans le mot de configuration, voir plus loin), sinon, il se déclenchera intempestivement et vous risquez de mettre longtemps à comprendre pourquoi votre PIC fonctionne de manière bizarre !

3. Les lignes d'entrée/sortie : le PIC fournit 35 lignes qui peuvent chacune être dynamiquement configurées comme entrée ou comme sortie. Certaines de ces lignes peuvent aussi remplir d'autres fonctions (RB0 peut déclencher une interruption, par exemple).

4. L'EEPROM : cette mémoire de 256 octets est programmable électriquement par le programme de l'application (et par le programmateur externe). Elle permet de stocker des données (pas de code).

5. le temporisateur 1 : c'est un compteur 16 bits qui peut générer une interruption quand il déborde de 0xFFFF à 0. La source d'horloge peut être interne ou externe. Il dispose aussi d'un prédiviseur.

6. le temporisateur 2 : c'est un compteur 8 bits doté d'un prédiviseur et d'un postdiviseur dont la sortie peut produire une interruption. Le compteur compte (c'est normal, pour un compteur !) de 0 à la valeur d'un registre spécial (PR2). Quand cette valeur est atteinte, la valeur du compteur est remise à 0.

7. les deux unités de comparaison/capture : ces unités fonctionnent en collaboration avec les temporisateurs 1 et 2. La comparaison consiste à déclencher une interruption quand un compteur atteint une certaine valeur. La capture consiste à mémoriser la valeur d'un compteur lorsqu'un événement a lieu (par exemple le passage d'un niveau bas à un niveau haut d'un broche d'entrée).

8. le convertisseur analogique $\longrightarrow$ numérique : ce convertisseur à 10 bits permet de convertir une tension analogique en une valeur binaire. Un multiplexeur permet de sélectionner une entrée parmi huit (huit lignes de ports d'entrée/sortie peuvent être configurées comme entrées analogiques).

9. l'USART : c'est un circuit de communication sérielle synchrone ou asynchrone. On peut l'utiliser pour implémenter une interface RS-232,

par exemple, ce qui est très pratique.

10. le SSP : c'est un circuit de communication synchrone exclusivement qui peut fonctionner, entre autres, aux normes I^2C, le bus inventé par Philips, et que nous utilisons dans notre réalisation. Bien que ce type d'interface puisse être simulé par programme, le fait de disposer d'une interface matérielle apporte un avantage certain : le processeur n'est plus sollicité pendant les transferts. Il ne doit intervenir que pour fournir un octet à envoyer ou lire un octet reçu. Le périphérique signale ces deux conditions au processeur par la génération d'une interruption.

11. le port parallèle esclave : ce système permet au PIC d'apparaître aux yeux d'un processeur externe sous la forme d'un périphérique intelligent auquel il est connecté par un bus de données de huit bits.

La fréquence d'horloge maximale du 16F877 est de 20 Mhz, un cycle d'instruction prenant 4 période d'horloge (ce qui donne environ 5 méga-instructions par seconde, la plupart des instructions s'exécutant en un cycle d'instruction et certaines instructions prenant 2 cycles).

Ces informations sont tirées de [1].

Chapitre 5

Utilisation correcte des entrées et sorties des circuits digitaux

Chapitre 6

Utilisation de boutons et interrupteurs

Chapitre 7

Utilisation de dispositifs d'affichage simples

Chapitre 8

Autres problèmes d'interfaçage

Chapitre 9

Conception et réalisation d’un circuit imprimé

Chapitre 10

Annexes

Bibliographie

[1] Microchip, *PIC16F87X Data Sheet, 28/40-Pin 8-Bit CMOS FLASH Microcontrollers*, 2001.

[2] Texas Instruments, *TL7702A, TL7705A, TL7709A, TL7712A, TL7715A, SUPPLY-VOLTAGE SUPERVISORS*, 1999.

[3] Malvino, *Electronic Principles*, Mac Graw Hill, fourth edition, 1989.

[4] D.E. Pippenger et E.J. Tobaben, *Circuits linéaires et d'interface, applications, tome 1*, Texas Instruments, 1989.

[5] http ://www.superpic.com/Datasheets.htm