Université de Liège

Systèmes Programmés Enfouis : Laboratoire 1

Hugues Smeets

Année académique 2001-2002

1 L'objectif du travail

L'objectif de ce laboratoire est de se familiariser avec le PIC 16f877 et son assembleur. Dans ce but, vous réaliserez un montage simple qui comptera de 0 à une valeur qui reste à préciser, et qui affichera son état sur un afficheur à 7 segments classique. Pour ne pas trop compliquer les choses, l'affichage se fera en hexadécimal. Une première version comptera de 0 à F. Elle n'utilisera donc qu'un seul afficheur. Les plus courageux d'entre-vous s'attaqueront à une version à deux afficheurs qui seront commandés à l'aide d'une technique de multiplexage (un seul afficheur est allumé à la fois mais le raffraîchissement est suffisamment rapide pour donner l'illusion que les deux afficheurs sont allumés simultanément). Cette technique, qui a été décrite au cours des répétitions, permet d'utiliser le même circuit de commande pour les deux afficheurs et réduit donc le nombre de composants utilisés. Si vous terminez rapidement cette deuxième version, vous ne vous ennnuierez pas pour autant : en effet, je vous propose de réaliser une version décimale de ce compteur (ce qui vous permettra de vous entraîner au codage d'opérations arithmétiques comme la division par exemple, non présente telle quelle dans les instructions du PIC).

Pour résumer, voici les trois objectifs possibles de ce travail (je vous suggère de remplir au moins le premier) :

– compteur en hexadécimal de 0 à F,

– compteur en hexadécimal de 00 à FF,

– compteur en décimal de 00 à 99.

2 Le cycle de développement

Vous devrez réaliser un fichier source en Assembler. Ce fichier sera "assemblé", c'est-à-dire transformé en une suite de valeurs numériques qui correspondent aux codes opératoires des instructions du PIC. Le fichier qui résulte de l'assemblage doit être transféré dans la mémoire de programme du PIC (cette mémoire, de type "flash", est effaçable électriquement). Grâce à un système de programmation présent sur la puce du PIC, cette opération peut se faire sans retirer le PIC de son circuit d'application. Vous disposerez d'un circuit d'interface entre le PIC et le port parallèle d'un PC pour mener à bien cette programmation.

Voici les différentes étapes que vous devrez suivre :

1. analyser le problème calmement,
2. réaliser le circuit,
3. écrire le code source,
4. l'assembler à l'aide de "gpasm <fichier.asm>" (disponible sur les machine ms800),
5. sauver le fichier binaire résultant sur une disquette,
6. récupérer le fichier sur l'ordinateur réservé à la programmation,
7. programmer le PIC à l'aide de "prog84 <fichier.hex>",
8. essayer l'application.

3 Le matériel

Voici le matériel que vous aurez à votre disposition pour réaliser ce travail :

- un PIC 16f877,
- un oscillateur à quartz fonctionnant à 20Mhz,
- deux afficheurs 7 segments,
- des résistances,
- un 7805 et les composants connexes (condensateurs, pont redresseur),
- des fils rigides,
- un "breadboard".

L'afficheur utilisé (voir figure 1) est à anodes communes (les deux lignes communes doivent être reliées à l'alimentation positive).

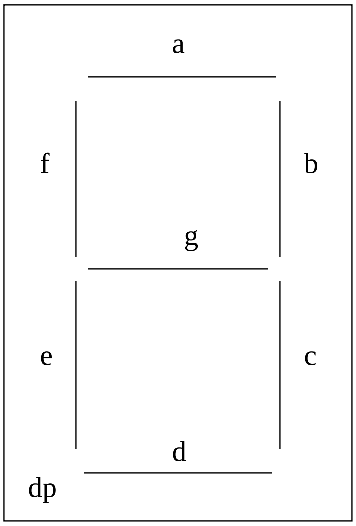

FIG. 1 – Afficheur 7 segments

Chaque segment est constitué d'une diode LED (light emitting diode). Pour qu'une led s'allume, il faut placer sa cathode à la masse de l'alimentation (l'anode, commune, est reliée au "+" de l'alimentation). Sous une tension de 5 Volts, les segments s'allument avec une luminosité suffisante si ils sont mis en série avec une résistance de 1 KOhms. Remarquez que si vous oubliez d'alimenter un segment à travers une résistance de valeur suffisante, vous détruirez le segment en question. Le brochage de l'afficheur est donné dans le tableau suivant :

broche	fonction
1	a
2	f
3	+
6	dp
7	e
8	d
10	c
11	g
13	b
14	+

4 Le logiciel

Vous devrez mettre en jeu un certain nombre de mécanismes dont certains ont été vus aux répétitions. Des exemples de codes source vous seront aussi fournis.

Parmis les mécanismes à utiliser, on trouve :

– la définition correcte des états et des types des broches d'entrée/sortie du PIC,

– l'utilisation de constantes codées en mémoire de programme,

– la programmation de temporisations, éventuellement par interruption.

Bon travail !