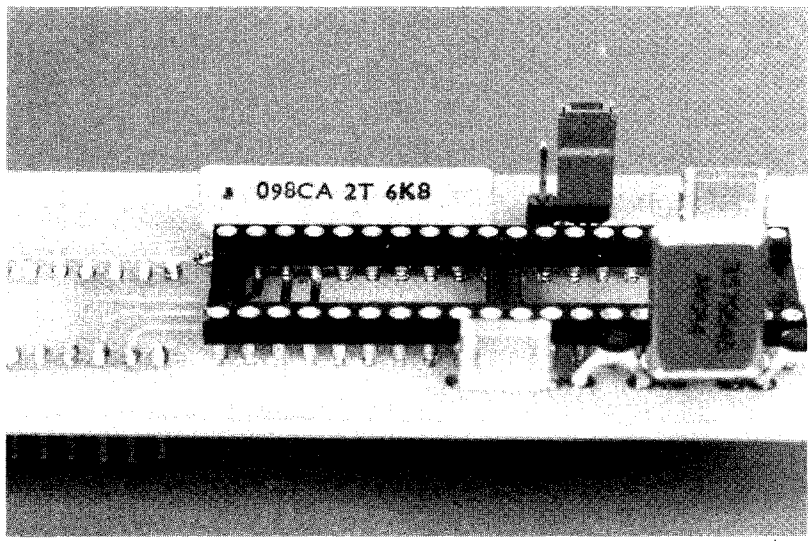
Un adaptateur de programmation pour 8751 ou 8753

Le microcontrôleur 8751 est une version du célèbre 8051 équipée d'une EPROM interne de 4 k-octets, et donc particulièrement appréciée pour la mise au point de prototypes ou la réalisation de petites séries. Le 8753, pour sa part, est doté de 8 k-octets d'EPROM, ce qui lui permet de recevoir des logiciels déjà conséquents.
Le processus de programmation ressemble beaucoup à celui de n'importe quelle EPROM, mais le boîtier DIL à 40 broches complique un peu les choses.
A défaut d'un programmateur dit "universel", il faut recourir à un adaptateur capable de "déguiser" le microcontrôleur en simple EPROM.



Faisons connaissance :

Les microcontrôleurs de la famille 8051 sont extrêmement populaires, en particulier pour les applications industrielles. Produits par de nombreux fabricants (INTEL, AMD, PHILIPS, SIEMENS, etc.), ils sont équipés de quatre ports d'entrée-sortie à huit lignes, d'une interface de communication série, de deux compteurs-timers, et d'un processeur booléen facilitant au maximum le traitement de bits individuels.
La **figure 1** montre que la plupart des broches sont affectées aux ports, à l'exception de celles servant à l'alimentation, au quartz d'horloge, au reset, et à quelques sélections de mode.
Un jeu d'instructions particulièrement riche et efficace permet de tirer le meilleur parti possible de cette infrastructure matérielle très complète.
Cette famille de processeurs est supportée par une gamme pratiquement sans équivalent d'outils de développement : émulateurs, assembleurs, simulateurs, compilateurs C ou Pascal, proposés par les meilleurs spécialistes.
Il existe des versions sans ROM, nécessitant une mémoire externe, des modèles à ROM masquée pour la production en grande série, mais surtout des exécutions à EPROM incorporée inestimables pour les tâches de développement et les petites séries.
Les programmateurs dits "universels", presque tous dotés d'un support à 40 contacts, supportent évidemment ces versions mais un sérieux problème se pose aux possesseurs d'un simple programmateur d'EPROM à 24 ou 28 broches : il faut nécessairement un adaptateur, que certains n'hésitent pas à facturer près de 3 000 F...

8751 OU 8753

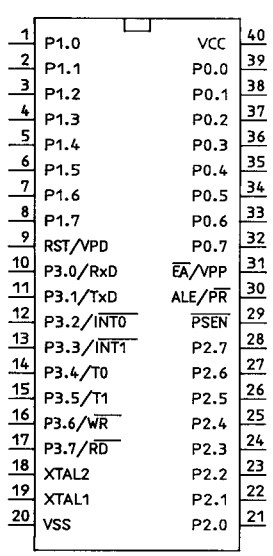
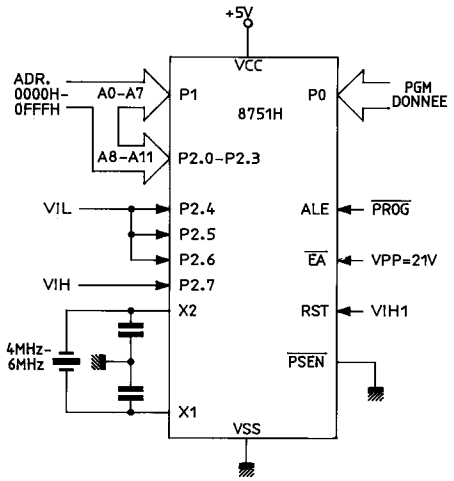


Figure 1.

UN ADAPTATEUR AU "PRIX COUTANT" :

La notice technique des 8751 et 8753 ne va pas jusqu'à fournir un schéma d'adaptateur, mais elle décrit clairement l'algorithme de programmation, très voisin de celui d'une EPROM. Notons cependant qu'il est totalement différent de celui des versions CMOS 87C51 et 87C53 !
La **figure 2** montre comment il faut procéder pour programmer un 8751 : les données à programmer doivent être appliquées sur le port P_0 et les adresses sur le port P_1 plus les quatre lignes basses du port P_2.
La **figure 3** indique pour sa part que la ligne d'adresse supplémentaire nécessaire pour "balayer" 8 k-octets au lieu de 4 doit



PROG : Impulsion de 1ms a GND/programmation
Boucle pour algorithme de programmation adapte
(Imp. de 50ms a GND pour un algorithme conventionnel)

Figure 2.

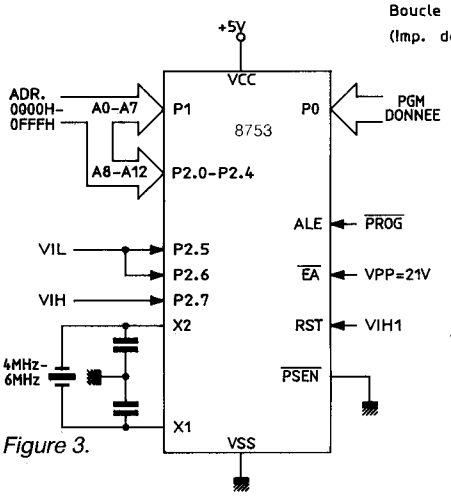


Figure 3.

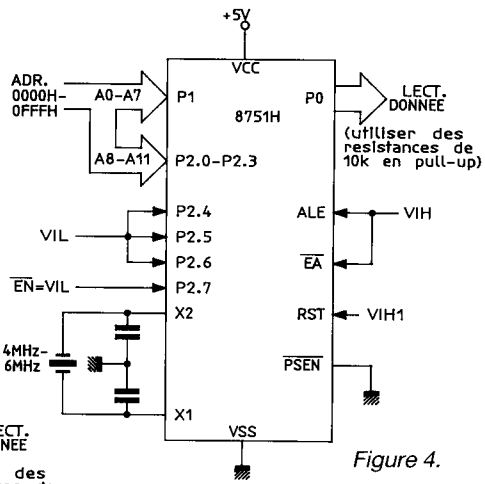


Figure 4.

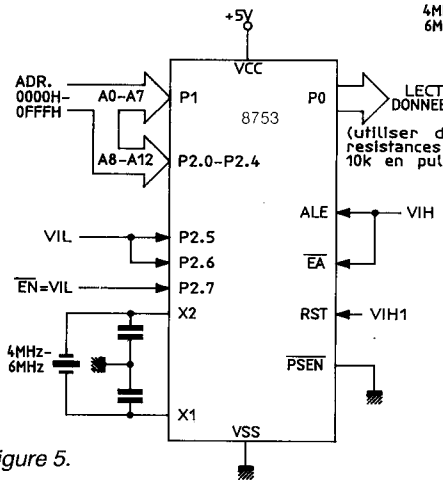
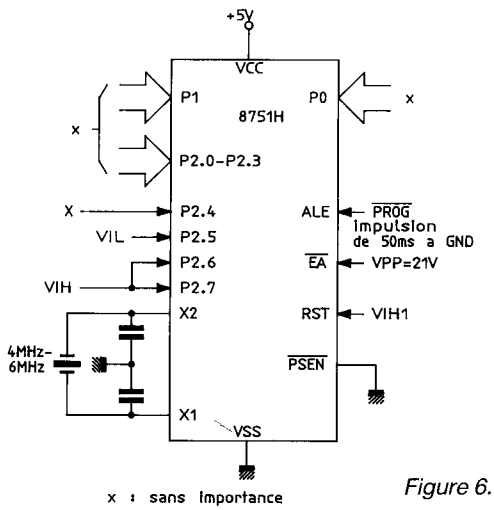


Figure 5.



x : sans importance

Figure 6.

être reliée à la ligne 4 du port 2.

La programmation de chaque adresse est exécutée à l'aide d'une impulsion négative appliquée à la broche ALE/PROG, en présence d'un Vpp de 21 V. Sa durée nominale est de 50 ms, mais peut être réduite par utilisation d'un algorithme adaptatif intercalant des lectures de contrôle entre des impulsions de programmation d'une milliseconde seulement.

Mais où est donc la différence avec une EPROM 2732 ou 2764 ? Dans le brochage, bien sûr mais aussi dans quelques niveaux hauts ou bas à appliquer à certaines broches.

Mais surtout, il est indispensable que l'oscillateur à quartz du microcontrôleur fonctionne pendant la programmation, à une fréquence comprise entre 4 et 6 MHz : cela pour cadencer les transferts internes de données et d'adresses.

Même chose en lecture, à ceci près que les niveaux appliqués à certaines broches sont différents, et que des résistances de tirage sont nécessaires sur les sorties de données (il n'y a d'ailleurs pas d'inconvénient à les laisser en place en mode programmation).

La **figure 4** détaille ainsi la configuration de lecture du 8751, et la **figure 5** celle du 8753.

La **figure 6**, pour sa part, illustre une opération qui n'existe pas avec les EPROM : la programmation d'un "bit de sécurité" dont le rôle est d'interdire la relecture de la mémoire (protection contre les copies illicites).

En fait, cette manœuvre s'apparente à la programmation d'un octet quelconque, mais en présence de niveaux particuliers sur certaines broches.

Le schéma de l'adaptateur que nous avons développé à partir de ces informations profite du fait que tous les signaux nécessaires existent, mais dans le désordre, sur le support de tout programmateur configuré pour des EPROM de type 2764.

Le montage de la **figure 7** commence donc par mettre le brochage du microcontrôleur en correspondance avec celui de la 2764, représenté à la **figure 8**, puis apporte les éléments manquants : un quartz oscillateur avec ses condensateurs auxiliaires, un réseau de résistances de tirage, quelques composants de découplage, et un inverseur permettant de choisir entre la programmation des données ou du

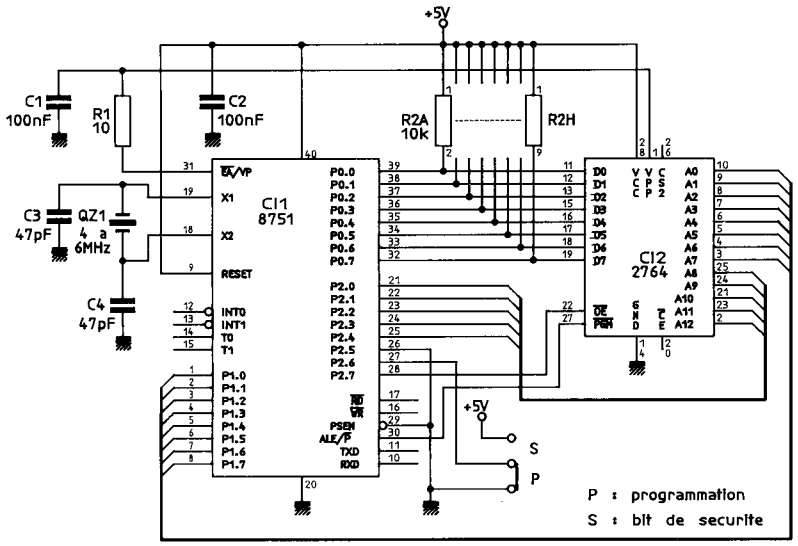


Figure 7 : Schéma de principe.

bit de sécurité. Il n'y a là rien de compliqué ni de coûteux, permettez nous d'insister !
La réalisation pratique fait appel à un petit circuit imprimé muni d'un support de bonne qualité (ZIF ou au moins "tulipe"), et à deux barrettes à 14 picots destinées à venir s'enficher dans le support à 28 broches du programmateur.
Le tracé de la face "soudures" est représenté à la **figure 9**. Celui de la face "composants" est reproduit à la **figure 10**, mais compte tenu de sa simplicité on pourra remplacer la gravure double face par un strap et trois connexions isolées, soudées comme le montre le plan de câblage de la **figure 11**.

MISE EN ŒUVRE :

Le microcontrôleur étant enfiché sur son adaptateur, il suffit de traiter le tout comme une simple 2764 21 V/50 ms, plus encombrante il est vrai.
Un 8751 pourra accueillir l'équivalent de la moitié d'une 2764, et un 8753 la totalité.
Seule la programmation du bit de sécurité nécessite un mode opératoire particulier : après avoir basculé l'inverseur sur la position prévue à cet effet, on exécutera la programmation d'un seul octet (par exemple un FFh à l'adresse 0). Il est commode, pour ce faire, de composer une fois pour toutes un fichier nommé par exemple "SECU", et contenant cet unique octet.

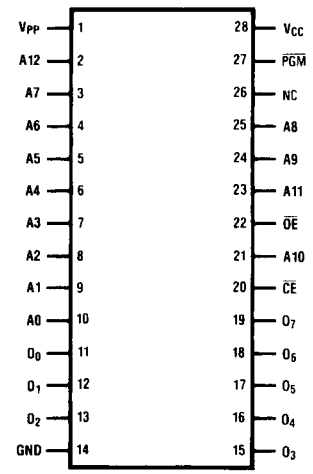


Figure 8 : Brochage d'une 2764.

Figure 9.

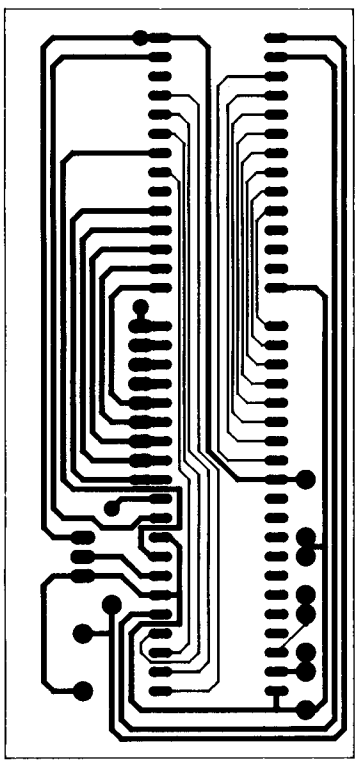


Figure 10.

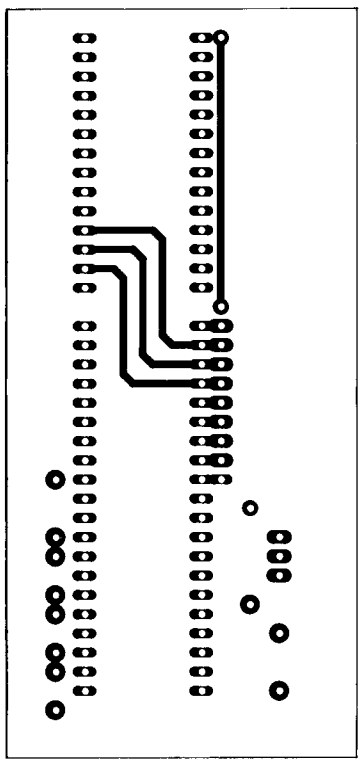
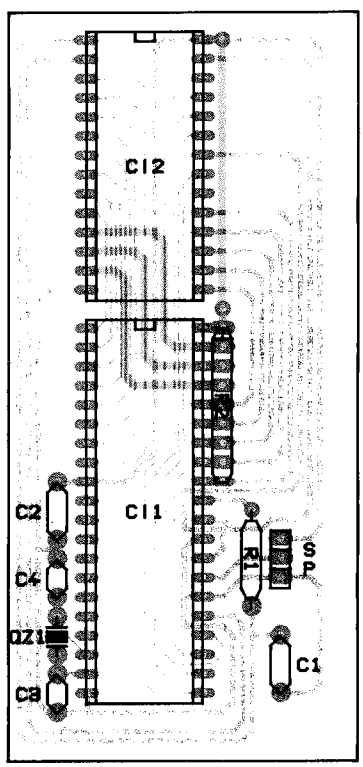
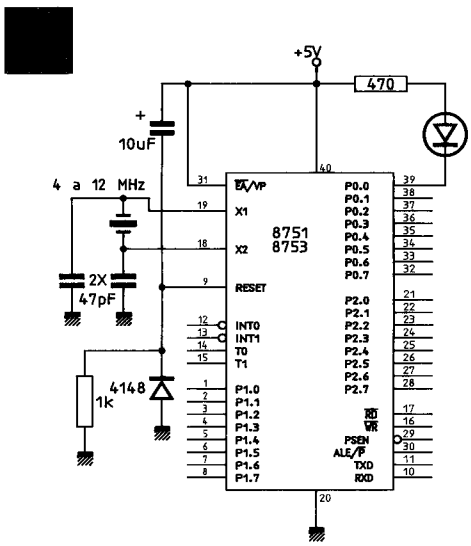


Figure 11.






```
Line 1     Col 1     Pos 1      Insert      AutoIndent                 ASCII
File       blocK     Quick      Error/brEak  Screen      Remember      Options
Delete     plaCe     Ascii      Printer      reBlock     rep Last find Esc

debut:
 clr p0.0
 mov r7,255
loopa:
 mov r6,255
loopb:
 djnz r6,loopb
 djnz r7,loopa
 setb p0.0
 mov r7,255
loopc:
 mov r6,255
loopd:
 djnz r6,loopd
 djnz r7,loopc
 jmp debut

1 Help  2 Diags 3 FPMs  4 ReAssm5 CurSkp6 Zoom  7 Watch 8 Go    9        0 Config
```

Figure 13.

```
194 128 175 255 174 255 222 254                    (décimal)
223 250 210 128 175 255 174 255
222 254 223 250 128 234
```

Figure 14.

```
C280AFFFAEFFDEFEDFFAD280AFFFAEFFDEFEDFFA80EA          (hexadécimal)
```

Figure 15.

```
:10000000C280AFFFAEFFDEFEDFFAD280AFFFAEFFF1
:06001000DEFEDFFA80EACB                            (Intel Hex)
:00000001FF
```

Figure 16.

```
S1130000C280AFFFAEFFDEFEDFFAD280AFFFAEFFED          (Motorola S)
S10900010DEFEDFFA80EAC7
S9030000FC
```

Figure 17.

Après une lecture de vérification, le microcontrôleur pourra être installé dans son circuit utilisateur, qui pourrait être aussi simple que celui de la **figure 12** : un quartz oscillateur et ses condensateurs, un circuit de reset à la mise sous tension, et les composants d'entrée-sortie nécessaires pour chaque port.

Dans notre exemple, nous nous limiterons à faire clignoter une diode LED branchée sur la ligne 0 du port 0.

Le très simple programme assembleur de la **figure 13** suffit pour obtenir ce résultat, tout en étant suffisamment court pour être assemblé par la version de démonstration du logiciel pour PC "ECAL" (voir notre nº 518).

Cette opération d'assemblage fournit 22 octets de code exécutable, qui peuvent être présentés de différentes façons selon les exigences du programmateur qui les exploitera :

La **figure 14** les représente sous la forme de simples valeurs décimales, la **figure 15** en hexadécimales directes (straight hex), la **figure 16** en code "Intel Hex", et la **figure 17** en code "Motorola S". Ces deux dernières formes sont les plus communément attendues par les programmateurs adaptables sur micro-ordinateurs : elles présentent l'avantage d'inclure des informations d'adresse d'implantation, et surtout des clefs de contrôle permettant de déceler d'éventuelles erreurs de transmission ou de recopie.

Précisons en effet que ces différentes formes du fichier à programmer dans le microcontrôleur peuvent être saisies à l'aide d'un simple éditeur de texte.

Patrick GUEULLE

Nomenclature

Résistances

R_1 : 10 Ω
R_2 : 10 kΩ
(réseau SIL)

Condensateurs

C_1, C_2 : 100 nF
C_3, C_4 : 47 pF

Divers

QZ_1 : quartz 4 à 6 MHz

Circuits intégrés

IC_1 : 8751
IC_2 : 2764