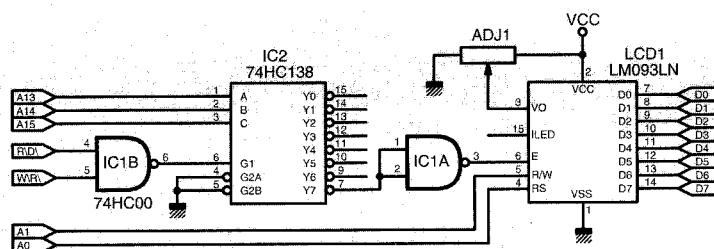


# COMMANDE DE LCD PAR $\mu$ C

Les afficheurs à cristaux liquides alphanumériques sont universellement utilisés pour afficher des informations diverses. Ces afficheurs utilisent tous le même type de contrôleur qui permet d'afficher l'ensemble des caractères ASCII ainsi que quelques symboles et un jeu de caractères Japonais stylisé. Les caractères accentués français ne font pas partie du jeu de ces contrôleurs.

Fort heureusement on peut tout de même afficher tous les caractères désirés grâce à des caractères définissables par l'utilisateur. Ceux-ci sont au nombre de huit et correspondent aux caractères de 020H à 07DH (inclus) sont les caractères ASCII classiques excepté le caractère \ qui est remplacé par le signe Yen. Les caractères supérieurs à 07DH sont des caractères Japonais et des symboles.



■ Schéma 1

La routine `lcdputchar` permet de convertir les caractères IBM-PC pour les afficher sur le LCD. Certains caractères sont simplement transcodés, les caractères IBM-PC dont le code se trouve dans la table `IBM_LCD` sont simplement traduits dans l'équivalent LCD (table `IBM_CAR`). La majorité des caractères accentués ne sont pas disponibles dans le jeu de caractères associé au contrôleur LCD. Les caractères IBM-PC dont le code est dans la table `IBM` sont recréés avec le bitmap associé présent dans la table `PATTERN`. Ces caractères sont créés dans une matrice 5 X 8, chaque ligne de huit octets de la table `PATTERN` correspond au caractère IBM-PC dont on désire recréer la forme. Cette forme est recréée de haut en bas, seul les cinq bits de poids les plus faibles étant représentatifs. On pourra créer n'importe

quelle forme en ajoutant (ou en modifiant) huit octets à la table `PATTERN` et en ajoutant à la table `IBM` le code du caractère auquel on souhaite associer la nouvelle forme créée. Les tables `IBM_LCD` et `IBM` contiennent les codes des caractères utilisés sous DOS. Nos lecteurs qui développent des logiciels sous Windows ou sous un autre système d'exploitation devront modifier ces tables en conséquence.

La routine `lcdputchar` ne gère pas les caractères de déplacement de curseur. Il est facile de modifier `lcdputchar` pour lui faire prendre en compte ces caractères de contrôle. Le seul point délicat est que le même contrôleur sert à gérer des afficheurs de type différent. Ainsi les afficheurs 2 X 16, 2 X 20, 4 X 16, 4 X 20 utilisent le même contrôleur et l'adresse qui correspond au 17ème caractère sur un afficheur de deux lignes

de vingt caractères, correspond au premier caractère de la troisième ligne sur un afficheur de quatre lignes de seize caractères. Ces modifications dépendent donc du type d'afficheur choisi. Le sous-programme `lcd_str` affiche une chaîne de caractères pointée par le `DPTR` en utilisant `lcdputchar`. La fin de la chaîne est indiquée par le caractère zéro (comme dans le cas du langage C). Ces différents sous-programmes font appel à des sous-programmes plus simples qui constituent les routines de bas niveau. L'afficheur à cristaux liquides sera connecté sur le bus du 8051 suivant le schéma 1. Contrairement à ce que l'on pourrait penser, il ne faut pas relier la ligne `R/W` du LCD à la ligne `WR` du microprocesseur. En effet, les timings respectifs de l'afficheur et du microprocesseur ne permettent pas cet arrangement. La solution la plus simple est de relier `RS` et `R/W` à des lignes d'adresses, la ligne `E` (qui joue le rôle d'un chip select) est connectée au décodage d'adresses et est validée à la lecture et à l'écriture. Normalement, l'impulsion positive envoyée sur la ligne `E` devrait durer au moins 450ns. En fait, cette ligne s'accorde de durée plus courte, on pourra donc sans inconvénient utiliser un microcontrôleur cadencé à 12 MHz. Le présent listing (au verso) est disponible sur le serveur ERP.

