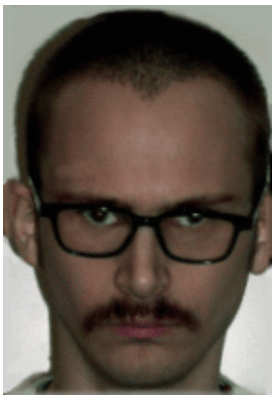
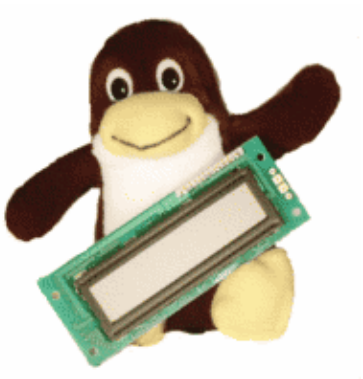
Comprendre les afficheurs LCD HD44780



par Jan Svenungson
<jan.svenungson(at)linux.nu>

L´auteur:

Jan utilise GNU/Linux depuis 1996 et n'a dû effectuer que deux redémarrages involontaires depuis lors (sans compter les redémarrages dûs aux coupures de courant).



Résumé:

Cet article va tenter de vous apprendre deux ou trois choses sur l'afficheur LCD HD44780.
Nous verrons comment le connecter au port parallèle et comment le programmer avec un petit outil nommé LCDInfo comme référence.
L'objectif n'est pas de simplement connecter l'afficheur, d'exécuter un programme et d'obtenir le résultat sur l'écran, mais plutôt de comprendre comment le matériel peut réagir selon vos souhaits.

Traduit en Français par:
Iznogood
<iznogood/at/iznogood−factory.org>

Introduction

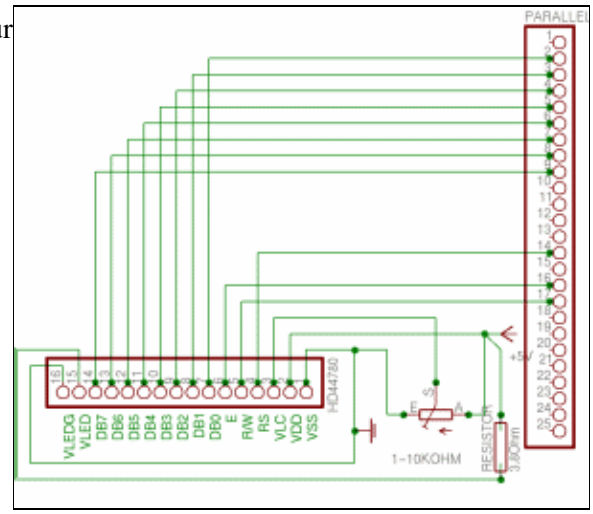
Avant tout, il nous faut du matériel et du logiciel ; nous supposons que vous possédez un ordinateur avec un port parallèle standard (imprimante) sur lequel vous pouvez faire fonctionner GNU/Linux avec gcc et glibc. Vous aurez bien sûr besoin d'un afficheur LCD HD44780, de câbles pour le connecter au port parallèle, ainsi qu'un potentiomètre si vous voulez changer le contraste. Pour alimenter l'afficheur, il vous faudra plus d'énergie que ne peut en fournir le port parallèle et vous aurez sûrement besoin de trouver une autre forme d'alimentation, ailleurs dans l'ordinateur. La meilleure façon de le faire est de prendre un connecteur standard de +5V (celui utilisé pour alimenter les disques durs et lecteurs de disquette, etc.)

Une fois la connexion de l'afficheur LCD terminée, vous devrez comprendre comment il fonctionne. Ceci est habituellement laissé de côté dans les articles sur ce sujet, mais je vais tenter de vous expliquer quelques arcanes de l'afficheur qui vous aideront à le programmer.

La dernière chose à faire est d'obtenir l'affichage de quelque chose d'utile. Comme référence, j'utiliserai un petit programme appelé LCDInfo qui supporte la plupart des fonctionnalités du HD44780 mais qui n'affiche pas grand chose pour l'instant. Ce programme est en phase alpha et je travaille dessus lorsque j'ai du temps

libre.
Si vous n'avez jamais programmé en C, envisagez de découvrir d'abord un peu le sujet. Je considère que vous êtes débutants, ce qui correspond à mon niveau actuel.

Comment le connecter ?



Avant tout, examinons les différentes broches disponibles sur le LCD et expliquons leur rôle.
La broche 1 se nomme VSS et elle est censée se connecter à GND.
La broche 2 se nomme VDD et correspond à la broche d'alimentation de +5V.
La broche 3, appelée VLC, est connectée au potentiomètre pour définir le contraste de l'afficheur.
La broche 4 est la broche RS et selon sa position, l'afficheur se prépare à recevoir des *instructions* ou des *données*.
La broche 5 est la broche R/W qui contrôle si le LCD est en mode *émission* ou *réception*.
La broche 6 est la broche d'activation (Enable). Lorsqu'elle passe du niveau bas à haut et revient à bas, le LCD lit les broches 4,5 et 7−14.
Les broches 7−14 sont les *lignes de bus de données* appelées DB0−DB7, qui sont les bits de données principaux envoyés au LCD et qui contrôlent où et quoi écrire sur l'afficheur.
Les broches 15 et 16 sont seulement présentes sur les afficheurs avec rétro−éclairage et correspondent simplement au +5V et au GND avec une résistance de 3.8 Ohm entre la broche 15 et le +5V.

Pour trouver comment les connecter au port d'imprimante, vous pouvez regarder sur le schéma de droite où j'ai tenté de rendre les explications le plus clair possible. Cliquez sur le schéma pour agrandir l'image.
Ce schéma n'est utile que si vous voulez changer le contraste de l'afficheur. J'ai simplement connecté les broches 3 et 1 à GND, ce qui a bien fonctionné; mais si vous avez un éclairage particulier dans la pièce, vous pouvez envisager d'ajouter un potentiomètre.

Lorsque vous récupérez l'alimentation du PC, soyez prudents. Si vous prenez l'alimentation à partir du mauvais câble, vous obtiendrez du +12V et vous grillerez le LCD. Vous devez choisir le rouge. Le jaune est le +12V et le noir est le GND.
Si vous avez branché correctement, le LCD doit avoir la première (et la troisième si elle existe) colonne en noir lorsque vous allumez le PC.

Comment fonctionne le LCD

Le LCD ne fait rien si vous ne lui demandez pas, il attend une variation valide (ce qui se produit lorsque la broche "enable" est mise en position haute, et qu'elle est remise en position basse après un moment).
L'afficheur détecte alors s'il s'agit de données ou d'instructions à traiter, puis s'il doit recevoir ou envoyer des informations et enfin, les bits de données sont envoyés ou reçus.
Dans cet article, nous ne recevrons jamais d'information du LCD, la broche R/W restera donc en position basse, ce qui signifie en écriture.
La broche RS sera en position basse, sauf lors de l'impression de caractères, tout le reste sera considéré comme instructions.

Ceci rend la programmation de l'afficheur réellement simple.

Une fois que nous savons ceci, nous voulons allumer l'afficheur et le préparer à recevoir des informations. Ceci est réalisé dans la séquence d'initialisation où nous indiquons à l'afficheur de démarrer, le « jeu de fonction » à utiliser, etc.
L'alimentation doit déjà être active si vous la prenez à partir d'un câble du PC, sinon, c'est la première chose à faire.
Arrive ensuite le « jeu de fonctions » selon le type d'affichage que vous avez.
Pour rendre ceci encore plus facile à comprendre, je vais expliquer exactement ce que nous faisons lors de l'utilisation du jeu de fonctions.
DB2 est le bit de police et il doit être en position *basse* indiquant une matrice de 5x7 points.
BD3 est le bit de ligne d'affichages et doit être en position *haute* signifiant 2 lignes. Que se passe−t−il si vous avez 4 lignes sur l'afficheur? Ne vous inquiétez pas, la première et la troisième ligne sont les mêmes dans la mémoire de l'afficheur, vous devez donc utiliser également la position haute.
DB4 est le bit de longueur de données et ceci détermine si vous avez 4 ou 8 DB. Si vous connectez l'affichage en fonction de mon schéma, vous devez avoir ce DB en position *haute*.
Initialisez alors *DB5* à haut pour indiquer à l'affichage qu'il s'agit bien d'une instruction du « jeu de fonctions », et vérifiez ensuite que *RS* et *R/W* sont en position *basse* et activez une variation haut et bas. Vérifiez dans le manuel pour les délais; je suppose que les temps d'attente ne sont que des microsecondes, ce qui devrait être bien plus que nécessaire.

Et à propos du code ?

Je vais parler ici des éléments du programme LCDInfo dont vous avez besoin pour comprendre comment l'interface du HD44780 fonctionne. Vous pouvez télécharger le programme LCDInfo à la fin de l'article ou allez lire les fichiers source en C iolcd.c et lcdinfo.c en cliquant ici.

Ce qu'il nous faut maintenant, ce sont les instructions en C et croyez−moi lorsque je vous dis que c'est facile. Je vais détailler le code pas à pas et même si vous êtes un débutant en C, vous comprendrez.
En premier lieu, nous incluons quelques fichiers en−tête et nous définissons quelques fonctions (voir le source pour information). Puis vient la partie la plus intéressante.

```
#define D_REGISTER 0
#define I_REGISTER 2
#define WRITE_DATA 8
#define BASE 0x378

int main(void)
{
  ioperm(BASE,3,1);
  [CUT]
}
```

C'est la première instruction dans la fonction principale qui nous donne les permissions sur le port parallèle. BASE doit être autour de 0x378 et la partie « 3 » signifie que nous avons accès à 0x378, 0x379 et 0x380, ce qui correspond basiquement au port d'imprimante.
La raison pour laquelle il y a trois adresses vient de ce que le port est divisé entre les données, les statuts et les contrôles. Pour nous, cela signifie que nous devons d'abord définir les broches de données, celles de contrôle en second et nous ne pouvons pas le faire en une seule commande.
La chose suivante, c'est le jeu de fonctions décrit ci−dessus.

```
void function_set(void)
{
  outb(56, BASE);
```

Ceci définit les broches DB à une matrice de 5x7 points, deux lignes, etc.

```
  outb(I_REGISTER + WRITE_DATA, BASE + 2);
```

Ceci initialise les broches RS et R/W en instruction et écriture. J'ai défini des variables globales de I_REGISTER et WRITE_DATA et elles sont égales à 2 et 8.
Il y a ensuite l'activation de la variation.

```
  outb(ENABLE + I_REGISTER + WRITE_DATA, BASE + 2);
  usleep(0);
  outb(I_REGISTER + WRITE_DATA, BASE + 2);
}
```

Ce que fait ce code est de basiquement mettre "enable" en position haute, attendre et le remettre en position basse. La commande usleep(0); n'est pas idéale, mais je n'ai pas fini le code pour la gestion du temps.
Quelques uns d'entre vous peuvent se demander pourquoi j'active RS et R/W dans le code, alors que j'ai dit qu'ils devaient être mis en position basse dans les instructions. C'est parce que les broches 1, 14 et 17 sont « inversées matériellement », indiquant que la broche 14 est « fermée »; pour le port d'imprimante, la broche est en position haute !
Je vous avais bien dit que c'était facile, non ?

Comment afficher des caractères

Vous voulez peut être un exemple pratique pour votre afficheur ? Pas de problème.
Le code (code comme dans commandes) est identique à l'impression d'un caractère et à la définition de fonctions. La seule chose à faire est de changer quelques variables. Nous ne voulons pas que le RS soit initialisé en instructions mais en données pour commencer. La fonction print_character() ressemble donc à :

```
void print_character(int character)
{
  outb(D_REGISTER + WRITE_DATA, BASE + 2);
  outb(character, BASE);
  outb(ENABLE + D_REGISTER + WRITE_DATA, BASE + 2);
  usleep(0);
  outb(D_REGISTER + WRITE_DATA, BASE + 2);
}
```

Comme vous pouvez le voir, nous avons changé « I_REGISTER » en « D_REGISTER » et « 56 » en « character » mais qu'est ce que cela signifie ? Si vous regardez les codes de caractères dans votre manuel, vous comprendrez.
Nous devons seulement remplir la fonction par un caractère (comme nous utilisons le C, il n'est pas nécessaire de se préoccuper de le définir d'abord comme un entier) et le caractère va alors s'afficher sur le LCD.
Chouette, non ?

Avec ce code, nous avons le squelette d'un programme LCD, adaptez−le à vos besoins, affichez la mémoire disponible, affichez les connexions http actives ou ce que vous voulez. Quelques exemples se trouvent dans le programme LCDInfo qui affiche des informations disponibles dans le système de fichiers proc sur un ordinateur GNU/Linux.

Références

- Pour avoir des informations sur le port d'imprimante, allez voir http://et.nmsu.edu/~etti/fall96/computer/printer/printer.html qui donne quelques exemples. (une copie locale de cet article est >ici<)
- Pour plus d'informations sur les programmes LCD, allez sur http://lcdproc.omnipotent.net/ qui est un bon programme LCD.

  Merci à Sven et Reinhold pour les adresses.
- Le code source du programme lcdinfo : lcdinfo−0.02.tar.bz2.
  Les mises à jour seront sur : http://savannah.gnu.org/download/lcdinfo

---

2005−02−14, generated by lfparser_pdf version 2.51