

## Icdproc -- affichages LCD faciles à utiliser et faciles à construire



par Guido Socher ([homepage](#))

### L'auteur:

Guido apprécie Linux parce que c'est un vraiment bon système pour développer son propre matériel.

*Traduit en Français par:*  
Jean-Etienne Poirrier  
([homepage](#))



### Résumé:

Que les lecteurs qui pourraient avoir attendu maintenant un article sur les microcontrôleurs ne se tracassent pas. Je continuerai certainement. Le prochain article sur du matériel que j'écrirai sera de nouveau sur les microcontrôleurs. J'ai aussi créé une page où vous pourrez trouver tous les articles récents d'électronique : <http://www.tuxgraphics.org/electronics/> J'espère que cela rendra le tout facile à suivre et à en avoir un aperçu.

Les lecteurs de longue date de LinuxFocus se rappelleront encore mon vieil article de Mars 2003 : Affichage LCD USB sous Linux et celui de Mai 2002 : Un panneau de contrôle LCD pour votre serveur Linux. C'étaient des panneaux LCD « de luxe », avec chien de garde et ADC, etc. Avec le temps, j'ai noté que toutes ces fonctionnalités supplémentaires étaient des choses que je n'utilisais pas souvent.

Cet article présente maintenant un modèle « de base » : un affichage LCD et 2 boutons, simple et facile à construire pour tout le monde.

---

## Qu'est-ce que Icdproc?

Au tout début, Icdproc était utilisé comme programme pour afficher quelques statistiques (charge processeur, uptime, temps), ... sur un affichage LCD externe. Avec le temps, il a cependant évolué vers une solution beaucoup plus générique. De nos jours, le paquet Icdproc contient LCDd, un serveur générique et le pilote LCD, accompagné de nombreux clients. Un de ces clients est toujours l'exécutable appelé Icdproc qui affiche toujours des statistiques du serveur, bien qu'il y en ait d'autres. Cette architecture client serveur a le grand avantage que vous ne devez pas écrire votre client dans un langage



spécifique. Vous avez juste besoin d'utiliser le protocole ascii simple entre le client et le serveur. Voici un exemple qui implémente une horloge en juste 36 lignes de code Perl. Ça affiche l'heure et la date et le coin supérieur droit montre un icône appelé « battement de coeur ». Le « battement de coeur » est ajouté par le serveur LCDd. Cet icône « battement de coeur » clignote une fois de temps en temps pour indiquer que l'affichage est toujours vivant.

Avec lcdproc, vous pouvez avoir plus que juste des chaînes. Vous avez des barres de graphique, des barres de défilement, des barres de titre, ... et des chaînes de caractères. Au début du programme, vous définissez ces objets (ils sont appelés « widgets ») et, ensuite, vous les remplissez avec des données. Tout cela est expliqué dans le Guide du Développeur de LCDproc, aux pages 7 et suivantes.

La partie principale de la petite horloge LCD du programme d'exemple ressemble à ceci :

```
#!/usr/bin/perl -w  
....
```

```

# initialise la communication:
print $remote "hello\n";
....
# défini notre nom et défini un widget de type chaîne de texte
print $remote "client_set name lcdtime\n";
print $remote "screen_add scr1\n";
print $remote "widget_add scr1 str1 string\n";
....
while(1)
{
    ....
    $date = scalar localtime;
    # maintenant écrit sur le LCD:
    print $remote "widget_set scr1 str1 1 1 \"\$date\"\n";
    sleep 10;
}

```

Assez facile, non ? OK, maintenant, construisons quelque chose de plus utile.

## Le Matériel

Selon les spécifications, le port parallèle d'un PC travaille à 5V. Si vous jetez un oeil à la feuille de données d'un affichage LCD moderne (par exemple, à partir de tuxgraphics : <http://shop.tuxgraphics.org/electronic/lcd.html>), vous vous apercevrez que même si les affichage sont grands, ils nécessitent extrêmement peu d'énergie. Seulement 2 mA ! Avec si peu de courant, il devrait être possible d'alimenter l'affichage directement par le port parallèle. L'avantage est qu'alors, vous n'avez plus besoin d'un connecteur de courant supplémentaire. C'est juste un affichage LCD avec un connecteur au port parallèle. Très commode.

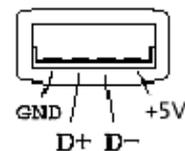
Tout ceci est possible avec un port parallèle qui fournit réellement 5V. Le problème est qu'il y en a quelques uns qui ne respectent pas les spécifications. Ainsi, comment savez-vous si votre port parallèle est suffisamment bon pour une solution « auto-alimentée » ? Voici un petit test :

1. Installez lcdproc avec le patch « parallel port power » (voyez les références).
2. Configurez le fichier /etc/LCDD.conf comme décrit ci-dessous.
3. Lancez le programme LCDD en tant que root.
4. Prenez un voltmètre digital et mesurez la tension sur la broche 1. Il devrait être de 4.90V ou plus (mesurez entre la broche 1 et la broche 18, c.à.d. la terre (GND)).
5. Maintenant, connectez une résistance de 33K sur le port parallèle, entre les broches 1 et 18. Le voltage devrait descendre à 4,25V. Si vous obtenez plus, alors votre port parallèle est très bon. Si vous obtenez moins, votre solution « auto-alimentée » ne fonctionnera pas.

Si le test est réussi, alors vous pourrez utiliser la solution à port parallèle auto-alimenté en utilisant l'affichage 20x2 de tuxgraphics (ou un équivalent). Celui-là est testé. Il fonctionnera.

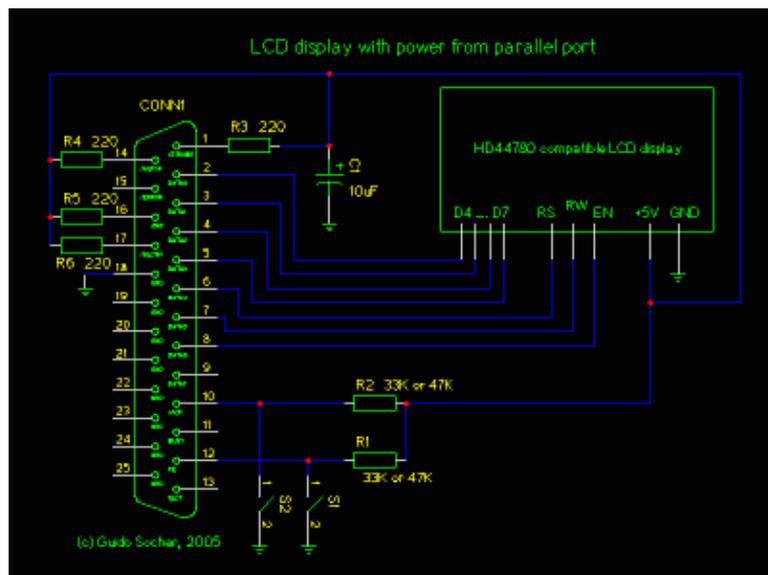
Si votre port parallèle est un peu faible, ne vous en faites pas trop. Vous pouvez soit utiliser les 5V internes de votre PC (vous devrez l'ouvrir pour cela) soit vous pouvez prendre 5V d'un connecteur USB libre.

4 pin USB A socket at the PC



# Le diagramme du circuit

Le premier diagramme montre les connexions pour un affichage LCD avec 2 boutons, utilisant la solution du port parallèle auto-alimenté.



Port parallèle

pin 1 , strobe = 220, supply voltage

pin 2 , D0 = LCD pin D4

pin 3 , D1 = LCD pin D5

pin 4 , D2 = LCD pin D6

pin 5 , D3 = LCD pin D7

pin 6 , D4 = LCD pin RS

pin 7 , D5 = LCD pin RW

pin 8 , D6 = LCD pin EN

pin 10, ACK = bouton A, bouton rouge au-dessus

pin 12, PE = bouton C, bouton noir en-dessous

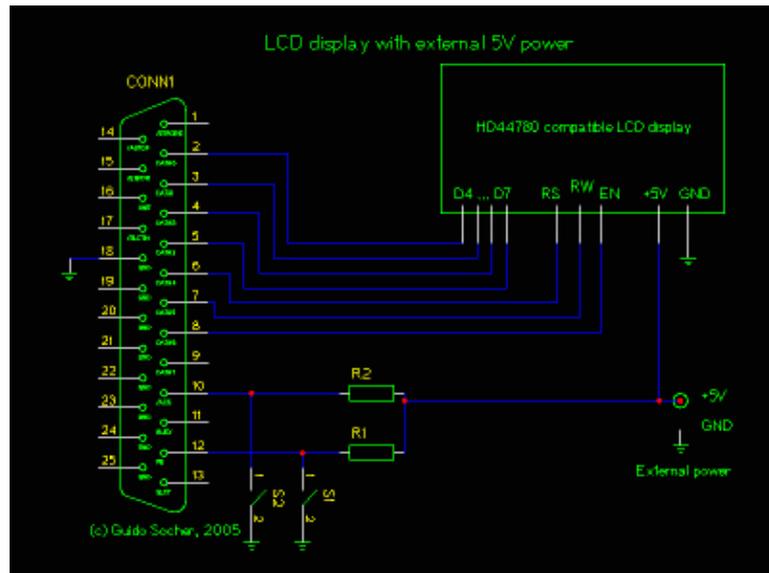
pin 14, strobe = 220, supply voltage

pin 16, strobe = 220, supply voltage

pin 17, strobe = 220, supply voltage

pin 18, ground = 0V, GND

Ce second schéma montre les connexions pour un affichage LCD avec deux boutons et une source de courant extérieure :



Comme pour tout nouvel article d'électronique, vous pouvez de nouveau obtenir un kit avec tous les composants sur [shop.tuxgraphics.org](http://shop.tuxgraphics.org)

J'ai monté l'affichage LCD dans un cadre de plastique noir :



## Le « lcdproc parallel port power patch »

LCDd envoie toujours des données à l'affichage. On pourrait dès lors utiliser des diodes et dériver le courant des lignes de signal. Le problème est que les diodes vont générer une chute de tension d'au-moins 0,5V, même à des courants faibles. Nous ne pouvons pas nous le permettre. Il y a cependant assez de lignes de sortie

libres si vous utilisez LCDd avec juste un seul affichage. Rien n'est plus simple que de modifier le logiciel LCDd pour toujours maintenir ces broches libres sur la valeur logique « 1 ». On insère toujours quelques résistances pour avoir une protection minimale contre les courts-circuits (ou une protection contre les fautes stupides ;-)). Le patch pour lcdproc-0.4.5 est disponible dans la section téléchargement. Vous l'appliquez comme ceci :

```
tar zxvf lcdproc-0.4.5.tar.gz
cd lcdproc-0.4.5
patch -p 1 < lcdproc-0.4.5-pp-powerpatch-v1.txt
./configure --enable-drivers=hd44780
make

make install
(ou vous pouvez juste faire "copy LCDd.conf /etc" et copier server/LCDd vers
/usr/bin)
```

## Le fichier LCDd.conf

Le fichier /etc/LCDd.conf configure nos pilote et serveur LCD (serveur = le processus LCDd). Dans le fichier /etc/LCDd.conf, vous devez enlever les commentaires sur les lignes suivantes pour utiliser l'affichage compatible HD44780 avec les connections décrites dans cet article :

```
[server]
Driver=HD44780

[HD44780]
# Hitachi HD44780 driver

Port=0x3BC
# Port where the LPT is. Usual values are 0x278, 0x378 and 0x3BC

ConnectionType=4bit
Keypad=yes
Backlight=no

Size=20x2
# Specifies the size of the LCD.

DelayMult=4
DelayBus=true
```

La partie difficile est l'adresse du port (0x278, 0x378 et 0x3BC). Comment savez-vous à quelle adresse d'E/S correspond quel port parallèle ? Le plus simple est de charger le module noyau parport (modprobe parport) et d'ensuite regarder dans /proc/ioports:

```
modprobe parport
modprobe parport_pc

cat /proc/ioports

...
0388-0389 : OPL2/3 (left)
038a-038b : OPL2/3 (right)
03bc-03be : parport0
03c0-03df : vga+
03f6-03f6 : ide0
...
```

dans ce cas, vous utilisez : 0x3bc

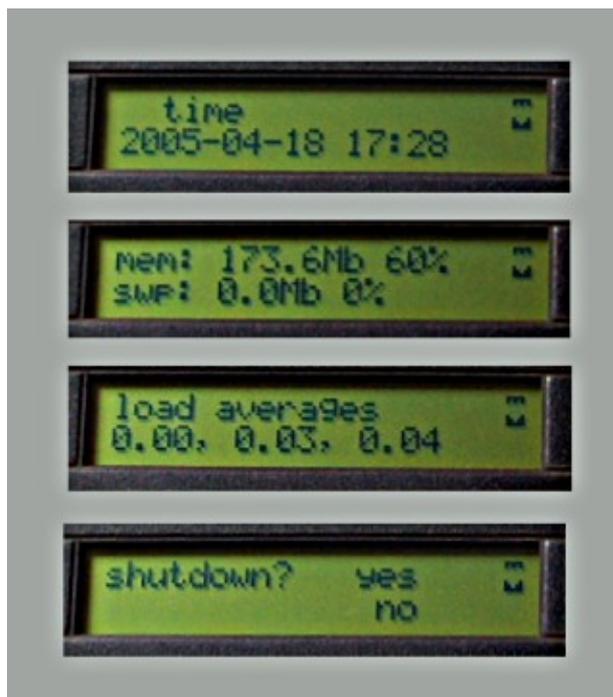
## Notre propre application

Vous avez vu, au début de cet article, combien il était facile d'envoyer un texte de chaînes quelconque vers l'affichage. Cela devient un peu plus compliqué si vous voulez un affichage interactif (avec les deux boutons). Dans ce cas, nous avons besoin d'éviter le signal de sommeil, sinon les boutons ne répondront plus durant la période de sommeil. En même temps, nous avons besoin d'un signal d'arrêt qui arrête la boucle. Ce dont nous avons besoin est l'appel système « select ». Il arrête l'exécution tant qu'un timeout (délai) n'est pas dépassé ou jusqu'à ce qu'un bouton soit pressé sur l'affichage.

Notre application va donner quelques statistiques (mem, swap, cpu, temps) et vous pouvez utiliser les boutons pour éteindre la machine.

Le script s'appelle servermenu\_pl et vous le trouverez dans la section de téléchargement

Je présente juste ici une application de base car l'expérience des articles précédents m'a montré que la plupart des lecteurs souhaitent juste savoir comment faire quelque chose et, ensuite, ils ont leur propre applications pour cela. J'ai reçu des e-mail avec toutes sortes d'idées : station de gravure de CD, PC station météo, systèmes d'alarme, ... Tout cela peut être réalisé. Il suffit de modifier un peu le code. C'est vraiment génial et procure beaucoup de plaisir de jouer avec cet affichage et lcdproc.



## Références

- [Section téléchargement](#) pour cet article
- Page web de lcdproc : <http://www.lcdproc.org>

- Le magasin en ligne tuxgraphics. Vous pouvez commander toutes les pièces nécessaires dans ce magasin (<http://shop.tuxgraphics.org>).

---

<p><u>Site Web maintenu par l'équipe d'édition LinuxFocus</u> © Guido Socher "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Translation information: en --&gt; -- : Guido Socher (<a href="#">homepage</a>) en --&gt; fr: Jean-Etienne Poirrier (<a href="#">homepage</a>)</p>
--	---

2005-09-06, generated by lfparsr\_pdf version 2.51