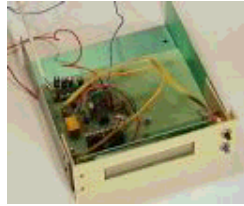


Un panneau de commande avec affichage à cristaux liquides (LCD) pour votre serveur Linux



par Guido Socher (homepage)



Résumé:

L'auteur:

Guido aime Linux non seulement parce que c'est amusant de découvrir les multiples possibilités de ce système, mais également grâce aux personnes impliquées dans sa conception.

Traduit en Français par:
Thierry Bayet
<snspsy/at/brutele.be>

Dans cet article nous concevrons un panneau de contrôle LCD basé sur l'afficheur LCD Hitachi HD44780 et le Micro-contrôleur RISC AT90S4433 AVR 8-Bit de chez Atmel. Les deux composants sont très courants et bon marché. Le panneau de contrôle comprend un système de chien de garde afin de superviser l'ordinateur et possède deux boutons pour permettre le dialogue avec l'utilisateur. Vous pouvez définir l'adresse IP, le masque de réseau, l'adresse de la passerelle, arrêter l'ordinateur, afficher les statistiques, autrement dit, tout ce que vous désirez puisque la majeure partie de la logique dépend d'un script Perl qui peut-être facilement modifié. Le panneau est connecté à votre ordinateur via le port série RS232.

Pour cet article vous avez besoin au minimum d'une installation partielle de l'environnement de développement AVR. La manière de l'installer est décrite dans cet article : Programmer le Micro-contrôleur ARV avec GCC.

Introduction

Ce périphérique combine les fonctionnalités de matériel déjà utilisé dans des articles antérieurs :

- Utiliser un affichage LCD par le port série sous Linux
- Un interrupteur sur le port série

Notre nouvelle conception va cependant beaucoup plus loin que cela. Elle ajoute des boutons pour l'interaction de l'utilisateur et inclut un chien de garde matériel pour superviser le serveur. De plus, elle

fournit une ligne d'entrée analogique. Nous ne l'emploierons pas ici mais vous pourriez, par exemple, y connecter une sonde de température.

Pour élaborer cela, il vous faut quelques connaissances d'amateur en électronique. Les pièces détachées que nous utiliserons ne sont pas chères et tout compris, coûtent moins de 40 Euro.

L'idée derrière ce panneau est qu'il devrait vous permettre de contrôler un serveur sans moniteur ou clavier. Linux est un système d'exploitation très fiable et peut facilement être contrôlé à distance. Cependant lorsque vous le connectez pour la première fois à un réseau, vous devez configurer l'adresse IP, la passerelle et le masque de réseau. Ce panneau vous permet de définir ces adresses. Il vous donne également la possibilité d'arrêter le serveur alors que vous êtes encore à côté de lui.

La conception de ce panneau est très générique. Toutes les parties "spécifiques au serveur" sont définies dans un script Perl. Le matériel, l'état des boutons, le texte de l'afficheur, les LED ..., peuvent être contrôlés par des commandes ASCII. Vous pouvez par conséquent utiliser cet ensemble pour construire un lecteur de mp3 ou contrôler votre grille-pain, ou ce que vous voulez.

Ce qu'il vous faut

Pour construire cela, vous avez besoin de ce qui suit:

- 1 x Micro-contrôleur Atmel At90S4433
- 1 x Support d'IC 28 broches de 7.25 mm
- 1 x Support d'IC 16 broches
- 1 x MAX232
- 1 x Petit relais 5V
- 1 x cristal 4MHz
- 2 x LED (verte et rouge)
- 1 x transistor BC547 NPN
- 1 x transistor BC557 PNP
- 4 x condensateur 1uF (courant ou incliné)
- 2 x condensateurs céramique 27pF
- 1 x condensateur 10nF
- 1 x condensateur 100nF
- 3 x résistances 4k7
- 2 x résistances 2k2
- 1 x résistance 10K
- 1 x résistance 3k3
- 2 x résistances 100 Ohm
- 3 x résistances 470 Ohm
- 3 x résistances 1K
- 1 x résistance 220 Ohm
- 1 x potentiomètre 4K7 (aussi petit que possible)
- 1 x diode Zener 4.3V
- 2 x petits boutons
- 1 x petite diode standard (par exemple 1N4148, toute diode bon marché)
- 1 x afficheur LCD de 2 lignes de 20 caractères avec interface compatible HD44780.

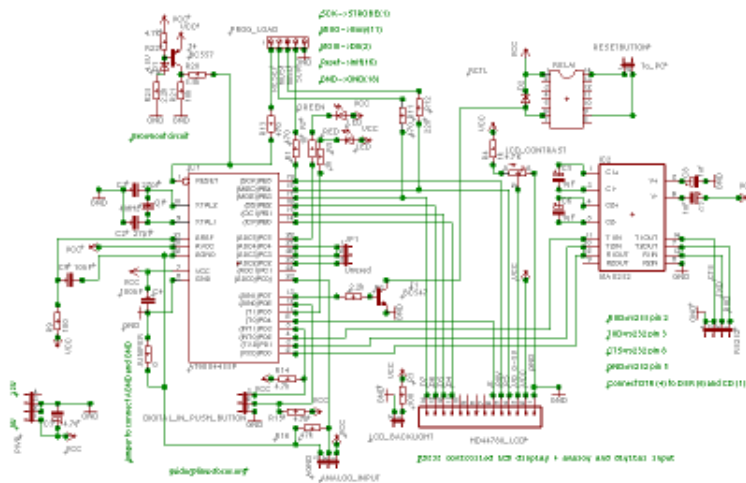
Tous les afficheurs que j'ai vus avec 14 ou 16 broches sur le connecteur étaient compatibles HD44780. Vous pouvez également utiliser un afficheur de 3 ou 4 lignes mais vous devrez légèrement modifier le programme .

De plus vous aurez besoin de quelques fils et de quelques connecteurs pour l'alimentation et la RS232. Si vous avez un afficheur de 2 lignes vous pouvez le monter dans une fine feuille d'aluminium et l'intégrer dans une baie de 5.25" sur votre serveur.

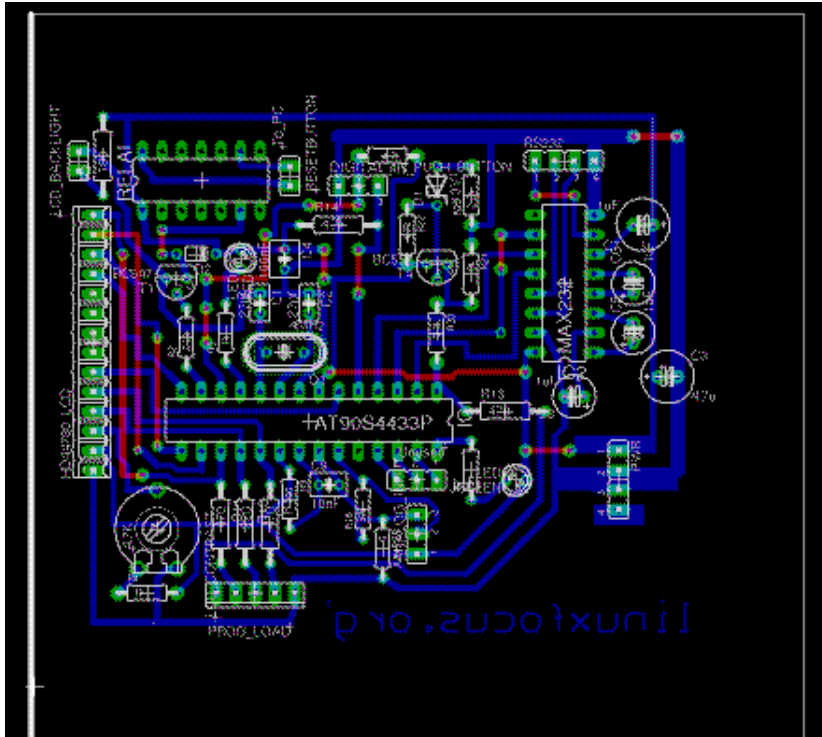
Schéma et carte

J'ai utilisé Eagle pour Linux pour concevoir le schéma et la carte. C'est un excellent programme mais il vous faudra un peu de temps pour apprendre à l'utiliser. Vous pouvez obtenir une version gratuite pour une utilisation privée sur : cadsoftusa.com.

Le schéma (cliquez dessus pour l'agrandir):



La carte (cliquez dessus pour l'agrandir):



La disposition des éléments de la carte sur fond blanc pour une meilleure impression: carte avec fond blanc (Note: Ce n'est pas le fichier requis pour élaborer le circuit imprimé de la carte.)
 Les fichiers *Eagle* (compressés avec gzip, notez que certains navigateurs décompressent pendant le téléchargement):

- linuxlcdpanel.brd.gz
- linuxlcdpanel.sch.gz

Le circuit

Je vais expliquer brièvement le diagramme du circuit ci-dessus. L'AT90S4433 a 3 ports : PB, PC et PD. PC peut-être utilisé comme port d'entrée analogique ou digital. Tous les ports peuvent être utilisés comme lignes d'entrée et de sortie digitales. C'est le programme qui contrôle cela via *DDR* (*Data Direction Register*) *Registre de Direction des Données*. Nous utilisons toutes les broches sauf la 23 comme lignes digitales (0 ou 5 V). Le Max232 est un convertisseur de tension. L'interface RS232 utilise +-10V et le Max232 le convertit en 0-5V. Sur la broche 1 (broche de remise à zéro) de l' AT90S4433 vous trouvez quelque chose nommé circuit d' *arrêt partiel* (*Brownout*). Ce circuit maintient la remise à zéro en position basse (active) dans les cas d'alimentation insuffisante afin d'empêcher le CPU d'exécuter de mauvaises instructions ou de mal fonctionner. Ceci peut se produire quelques millisecondes pendant la mise en route ou l'arrêt. Le but est de garantir que le programme du Micro-contrôleur démarre correctement.

Certains d'entre vous pourraient se demander pourquoi il y a une diode en parallèle à la bobine du relais

et que la polarité telle qu'elle est représentée n'aura jamais rien à faire. Cette diode est très importante ! Lorsque vous coupez le relais, une tension très élevée est générée par la bobine. Elle peut détruire le Micro-contrôleur. Cette tension est de polarité inverse à la tension d'alimentation de la bobine. La diode peut être une petite diode bon marché, mais il est important de l'avoir.

Les deux boutons poussoirs doivent être liés au connecteur nommé "DIGITAL IN PUSH BUTTON" dans le schéma. Ils connectent PD3 ou PD6 à la masse lorsqu'ils sont pressés.

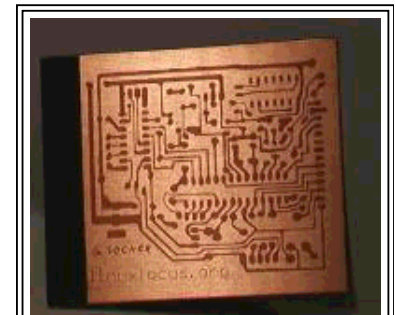
Comment réaliser le circuit imprimé

Pour graver le circuit imprimé vous devez d'abord imprimer ce fichier PostScript (linuxlcdpanel.ps.gz) sur une feuille transparente. Dans les magasins pour architectes vous pouvez trouver des feuilles de plastique semi-transparentes nommées *Sinolit*. Elles sont produites par *Regulus* et sont normalement utilisées pour les imprimantes Offset. Une alternative peut être un papier de 60g + un produit de pulvérisation transparent (pausklar 21 de Kontakt Chemie). L'avantage du papier et de Sinolit vient de ce que le toner de l'imprimante Laser colle vraiment au papier/à la feuille plastique et offre un bon contraste.

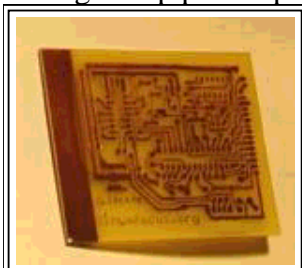
J'ai converti le fichier PostScript en PDF au cas où vous n'auriez pas d'imprimante PostScript. La qualité est cependant moindre.

Le temps d'exposition pour les plaques de cuivre avec procédé photo dépend de la source lumineuse. Pour une lumière artificielle normale il faut compter 1-2 minutes. Vous pouvez également utiliser la lumière du jour, mais essayez d'éviter la lumière directe du soleil (elle est trop forte). Vous pouvez faire l'expérience sur un petit morceau de plaque photosensible pour définir le temps d'exposition idéal avant d'utiliser la bonne plaque.

La carte exposée a besoin maintenant d'être révélée quelques minutes dans du NaOH (Natriumhydroxid). Ensuite, vous devez regarder attentivement le résultat et faire les corrections avec un marqueur de peinture noire Edding 780 (ce n'est pas le marqueur indélébile pour surcharge, c'est un marqueur contenant de la peinture). Je fais habituellement les pastilles un peu plus grosses parce que je trouve celles de Eagle trop petites pour une utilisation non professionnelle.



La carte exposée et révélée avant gravure



La carte avant de percer les trous

Note: il semble impossible aux fabricants de s'entendre sur un brochage commun ou sur des conventions de nommage pour les relais. J'utilise un petit relais 5V manufacturé par Matsushita. Votre relais peut avoir un brochage différent, modifiez donc la carte en conséquence (avec un marqueur résistant à la gravure).

Lorsque vous êtes satisfait vous pouvez plonger la carte dans du FeCl₃ (Hypochlorite de fer). Le FeCl₃ donne de bons résultats à température ambiante. Il est très facile à utiliser et donc idéal pour le bricolage à la maison.

Vous obtiendrez de meilleurs résultats si la carte est placée verticalement dans un conteneur profond. Les ions de cuivre sont plus lourds que les ions de fer et si vous remplissez de FeCl₃ un petit récipient plat, les ions de cuivre s'accumuleront au fond du récipient.

Quand la carte est prête effacez l'encre du marqueur avec de la térébenthine. Vous pouvez laisser

"l'encre" photo-résistante. Elle s'évaporerait lorsque vous souderiez dessus et protégerait le cuivre.

Le programme du Micro-contrôleur

Le programme pour le Micro-contrôleur se compose des fichiers suivants :

- `lcd.c`, `lcd.h`, `lcd_hw.h`: C'est une bibliothèque générique AVR LCD. Elle est basée sur le travail de Peter Fleury (<http://jump.to/fleury>). Cette version est un peu modifiée et plus flexible. Elle vous permet de connecter le matériel LCD à n'importe quelle broche du Micro-contrôleur. Vous n'avez qu'à changer la définition dans le fichier `lcd_hw.h`.
- `avr-util.c`, `avr-util.h`: fonctions pour obtenir divers temps de retard.
- `uart.c`, `uart.h`: C'est une bibliothèque pour l'interface RS232. Elle utilise les interruptions matérielles. Chaque fois qu'un caractère est reçu de l'ordinateur, la fonction `SIGNAL (SIG_UART_RECV)` est exécutée et les données sont copiées depuis le tampon de réception vers un tampon de chaîne de caractères. Le langage de commande pour notre panneau LCD est conçu de telle manière que chaque commande se termine par le caractère de nouvelle ligne. Lorsqu'une nouvelle ligne est trouvée alors un drapeau (`uart_rx_linecomplete`) est positionné et la donnée est rendue accessible. Cela signifie également que vous ne devez pas envoyer de commande à l'afficheur aussi vite que possible, mais attendre un peu (une milliseconde) après chaque ligne. Chaque commande recevra un accusé de réception avec un résultat `ok`, ou `err` (pour erreur). Le programme de pilotage Perl peut donc employer le résultat comme déclencheur de l'envoi de la prochaine commande.
- `analog.c`, `analog.h`: Code pour le convertisseur analogique/digital. Il utilise également les interruptions. Une conversion simple de l'analogique vers le digital est lancée et le signal attend l'interruption `SIG_AID` pour lire le résultat depuis le registre ADC.
- `hardwarewd.c`, `hardwarewd.h`: Voici le chien de garde. Nous utilisons le diviseur interne (divisé par 1024) pour gérer les impulsions du minuteur. Le minuteur est un registre 16 bits, et lorsque nous avons un dépassement nous décomptons une variable de 8 bits. Avec un cristal de 4MHz nous décompterons notre variable approximativement toutes les 16 secondes. Le programme Perl signale que l'ordinateur est bien actif en redéfinissant périodiquement la variable à une valeur haute. S'il échoue (parce que l'ordinateur s'arrête) alors la variable diminuera de sa valeur chaque fois et lorsqu'elle atteindra zéro, le relais collera et fera une remise à zéro matérielle de notre serveur.
- `linuxlcdpanel.c`: C'est le programme principal. Il vérifie continuellement les commandes de l'interface RS232 et les événements de pression des boutons.

Afin de comprendre le programme en détail, il vous est recommandé de lire les caractéristiques techniques du Micro-contrôleur. Elles sont accessibles à partir de la section références à la fin de cet article (ou sur www.atmel.com)

Cependant pour utiliser ce panneau vous n'avez pas besoin de comprendre le programme, vous avez juste besoin de décompresser les archives du code source (sélectionnez `linuxlcdpanel-0.7.tar.gz` sur la page de téléchargement) et de taper:

```
make
make load
```

ou vous pouvez éventuellement utiliser le programme pré-compilé et le charger par la commande `make loadprebuild`

Très facile. Vous trouverez une description de la manière de programmer le Micro-contrôleur dans le premier article: Programmer le Micro-contrôleur ARV avec GCC.

Test du panneau LCD

Le panneau LCD est prévu pour fonctionner avec l'alimentation interne 5V de votre serveur (fil rouge=5V, fil noir=masse). Mais ne les connectez surtout pas à l'alimentation de l'ordinateur la première fois. Vous avez peut-être fait des erreurs pendant la soudure et l'assemblage. L'alimentation de l'ordinateur est puissante, et autant votre ordinateur que votre carte peuvent partir en fumée si tout n'est pas correct. Testez d'abord avec une alimentation stabilisée externe avec limitation de courant ! Maintenant vous pouvez télécharger le programme vers l'EEPROM comme décrit plus haut. Ensuite vous devriez voir apparaître un défilement de la bannière "linuxfocus.org" sur l'afficheur LCD. Maintenant connectez l'interface RS232:

broche 14 MAX232 sur CTS (DB-9 broche 8)

broche 7 MAX232 sur RXD (DB-9 broche 2)

broche 13 MAX232 sur TXD (DB-9 broche 3)

Vous avez également besoin de connecter entre eux DTR, DSR et CD (DB-9 broches 4, 6 et 1)

Ceci est également documenté dans le diagramme du circuit ci-dessus.

Pour utiliser la ligne série vous devez l'initialiser. L'archive du code source `linuxlcdpanel-0.7.tar.gz` contient un programme appelé `ttydevinit` qui s'occupe de cette initialisation. Disons que vous avez connecté le panneau sur COM2 (`ttyS1`) alors vous pouvez lancer la commande:

```
./ttydevinit /dev/ttyS1
```

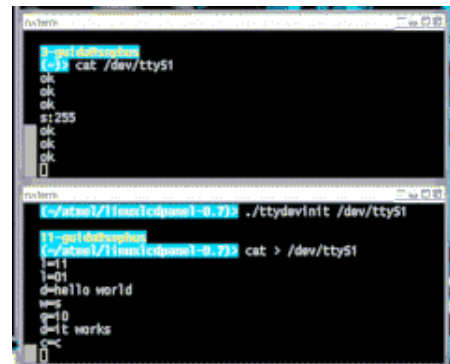
Maintenant le pilote de la ligne série est initialisé pour fonctionner à 9600 Bauds et vous pouvez commencer à "discuter" avec votre panneau LCD. Ouvrez 2 fenêtres xterm. Dans l'une vous tapez `cat > /dev/ttyS1`. Maintenant vous pouvez par exemple entrer la commande `l=11` (allumage de LED 1) ou `l=10` (extinction de LED 1). Vous pouvez voir dans le premier xterm que votre panneau LCD accuse réception de la commande par "ok".

Toutes les commandes disponibles sont expliquées dans le fichier `README.commands`.

L'archive du code source contient un script Perl nommé `ttytest.pl` qui ne fait qu'allumer et éteindre les LED rouge par intermittence. Il est censé être utilisé comme un simple programme exemple qui montre comment piloter le panneau LCD. Vous pouvez vous en servir comme base pour vos propres programmes. Lisez le code source. Il demande quelques connaissances en Perl, mais c'est un programme plutôt court.

Connection du chien de garde

Le chien de garde est coupé par défaut. Vous l'activez par la commande `w=1` et vous définissez le délai



par $s=x$ ou x est une valeur entre 1 et 255. Par exemple $s=10$ vous donnera un délai d'une valeur de $10*16\text{sec}=160\text{sec}$. Le pilote a besoin de définir le délai périodiquement pour permettre au chien de garde de fonctionner correctement. Si votre serveur se verrouille, alors le pilote n'est pas réinitialisé et le chien de garde agit. Je sais que les serveurs Linux ne décrochent jamais. Cependant si c'était le cas, il faut savoir qu'il n'y a habituellement personne sur le site pour faire une remise à zéro, ou que personne ne sait où se trouve le serveur parce qu'il n'y a jamais eu de problème depuis les 2 dernières années.

Le chien de garde est conçu de telle manière qu'il ne joue son rôle qu'une seule fois. Cela afin d'éviter une remise à zéro pendant le redémarrage du système. Lorsque le serveur est de nouveau opérationnel, le pilote doit le réactiver.

Pour connecter physiquement le chien de garde, vous devez trouver les deux fils connectés au bouton de remise à zéro de votre serveur. Parallèlement à ce bouton de remise à zéro vous devez connecter le relais à partir de votre chien de garde.

Comment utiliser le chien de garde

Le chien de garde garantit que le système est toujours capable d'exécuter les programmes. Il ne garantit pas qu'un serveur Web ou une application de base de données fonctionnent toujours correctement. Pour vérifier cela vous devriez utiliser une entrée crontab ou d'autres programmes. Vous pouvez être certain que le crontab fonctionnera probablement parce que le chien de garde s'assure que les programmes en général sont toujours actifs.

Vous pouvez par exemple concevoir un script qui est déclenché par cron et télécharge une page Web de votre serveur toutes les 15 minutes, mais vous devez être prudent : un serveur Web peut être surchargé par un tas de requêtes et il est dès lors normal qu'il ne réponde pas à toutes. Vous pourrez cependant compter le nombre de fois que le serveur ne répond pas et si, par exemple, il ne répond toujours pas après 10 vérifications, alors redémarrez le serveur ou basculez vers un redémarrage normal (pas une remise à zéro matérielle par le chien de garde).

Indépendamment de l'application, vous devriez également surveiller l'utilisation du disque. La commande shell suivante retourne une valeur si une des partitions est remplie à plus de 80% :

```
df | egrep ' (8.%|9.%|100%) '
```

Cela peut également être utilisé conjointement avec une entrée crontab afin de vérifier régulièrement l'utilisation du disque.

Les scripts sur le serveur

Presque toute la logique de notre panneau d'affichage à cristaux liquides dépend d'un script Perl nommé `llp.pl`. Copiez ce fichier dans `/usr/sbin`. Copiez ensuite le programme `ttydevinit` dans `/usr/bin` et le fichier `ifconfig_llp.txt` (depuis le répertoire `etc` des archives du code source) dans `/etc`. Editez enfin, le fichier `ifconfig_llp.txt` et changez les adresses selon les besoins :

```
NETMASK=255.255.255.0  
IPADDR=10.0.0.4
```


GATEWAY=10.0.0.2

Maintenant, faites une copie de votre script original /etc/rc.d/init.d/network et copiez le script /etc/network de l'archive du code source dans /etc/rc.d/init.d/network. Ce script et les noms de répertoire sont seulement valides sur Redhat et Mandrake. Le script /network_all_distributions est plus élémentaire et fonctionnera sur toutes les distributions Linux, mais vous devez trouver l'emplacement des répertoires init-rc de votre distribution Linux. C'est légèrement différent d'une distribution à l'autre.

Éditez le fichier /etc/rc.d/init.d/network et changez la ligne

```
/usr/sbin/llp.pl /dev/ttyS1&
```

si vous n'utilisez pas COM2.

Vous pouvez maintenant lancer :

```
/etc/rc.d/init.d/network start
```

et voir votre panneau LCD en action. Note: Il est possible de jouer avec la modification des adresses IP. Les changements ne prendront effet qu'après le prochain redémarrage. Par conséquent testez et restaurez-les avant d'arrêter votre serveur (vous pouvez aussi éditer /etc/ifconfig_llp.txt pour annuler vos changements).

Fichiers de log

Le script llp.pl crée un fichier nommé llp.log dans /var/log. Ce log ne se remplit que très lentement. Normalement, il n'est pas nécessaire de prévoir une rotation automatique. Vous pouvez le purger avec un programme tel que logrotate si vous le désirez. Il n'est pas nécessaire de gérer la post rotation. Une ligne du fichier de configuration de logrotate pourrait ressembler à ceci:

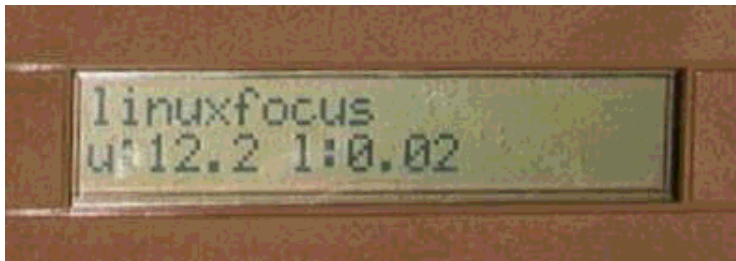
```
/var/log/llp.log {  
nocompress  
monthly  
}
```

Le fichier log contiendra des entrées lorsque le système sera arrêté manuellement, qu'une adresse IP sera modifiée (IP, GW, netmask) ou lorsque le chien de garde matériel déclenchera une remise à zéro. Naturellement vous ne pouvez pas enregistrer la valeur du délai du chien de garde lorsque cela se produit (puisque le système prend la main), par contre elle sera enregistrée au prochain démarrage.

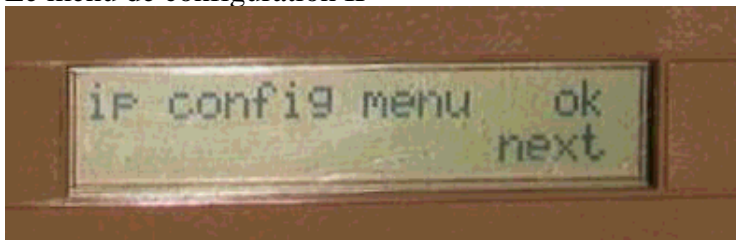
Le panneau en opération

Voici quelque "photos" du panneau LCD en action. Ce ne sont pas toutes les fonctions offertes par ce panneau. Il y en a beaucoup plus et vous pouvez y ajouter les vôtres.

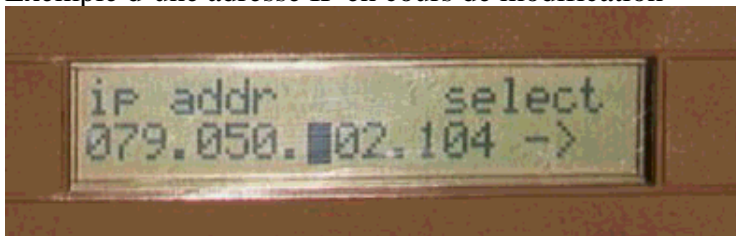
L'écran principal. Montrant un nom (linuxfocus dans ce cas), la durée de fonctionnement et la charge. Il se met à jour périodiquement.



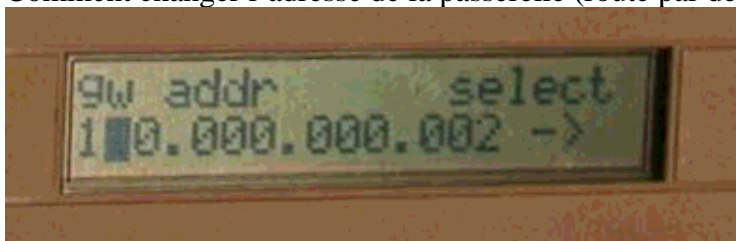
Le menu de configuration IP



Exemple d'une adresse IP en cours de modification



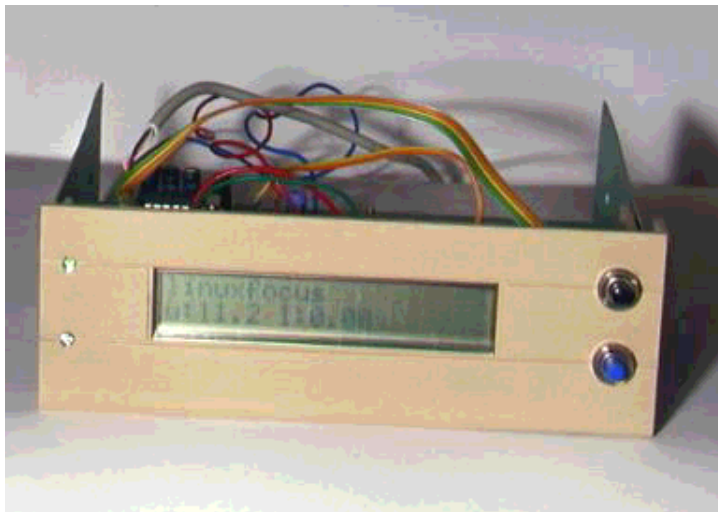
Comment changer l'adresse de la passerelle (route par défaut)



Conclusion

Pour construire ce panneau à cristaux liquides (LCD) il faut des connaissances d'amateur en électronique, mais ce n'est pas un circuit très complexe. Notre panneau LCD Linux offre plus de fonctionnalités que bien d'autres panneaux de contrôle déjà vus et il est très générique et très bon marché.

Bonne soudure :-)



Références

- Le logiciel uisp AVR de programmation: www.amelek.gda.pl/avr/
- Le code source de cet article [linuxlcdpanel-0.7.tar.gz](#) . Le schéma du circuit, les fichiers Eagle et les photos d'écrans sont également inclus.
- Tous les programmes et documents mentionné dans cet article
- Caractéristiques techniques de MAX232 MAX220-MAX249.pdf 448K
- Caractéristiques techniques de ST232, une variante bon marché, souvent vendue à la place du vrai MAX232 [st232.pdf](#) 100K
- Caractéristiques techniques d'Atmel AT90S4433 [avr4433.pdf](#) 2356K
- Le site d'Atmel: www.atmel.com/
- Eagle pour Linux cadsoftusa.com

Site Web maintenu par l'équipe d'édition
LinuxFocus

© Guido Socher

"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

Translation information:

en --> -- : Guido Socher (homepage)

en --> fr: Thierry Bayet <snspsy/at/brutele.be>