

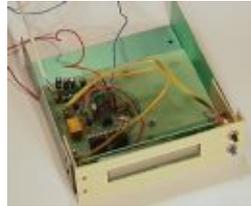
Un pannello di controllo LCD per il Vostro server Linux



by Guido Socher ([homepage](#))

About the author:

Guido adora Linux non solo per il fatto che sia divertente scoprire le grandi possibilità di questo sistema, ma anche per le persone che sono coinvolte nel suo sviluppo.



Abstract:

In questo articolo tratteremo della costruzione di un display LCD utilizzato come pannello di controllo basato sul display Hitachi HD44780 ed il microcontrollore, AVR ad 8bit in tecnologia RISC, AT90S4433 della Atmel. Entrambi i componenti sono di facile reperibilità e non molto costosi. Il pannello di controllo comprende la funzionalità di watchdog per supervisionare il computer medesimo ed ha due pulsanti che permettono all'utente di interagire con il sistema. Così potete settare l'IP, la maschera di sottorete, l'indirizzo del gateway, spegnere il computer, leggere delle statistiche, insomma più o meno quello che desiderate fare in quanto la logica è implementata per mezzo di uno script in perl che può facilmente essere modificato. Questo pannello sarà collegato al vostro server Linux per mezzo di una porta seriale (una RS232).

Per poter procedere con questo articolo dovete prima aver installato il software come descritto nell'articolo intitolato [Programmare il microcontrollore AVR con GCC](#).

Introduzione

Questo progetto unisce assieme le funzionalità di hardware che abbiamo già visto in alcuni articoli precedenti:

- [Utilizzare un display LCD con un collegamento seriale con Linux](#)
- [Spegnere il vostro computer per mezzo di un pulsante collegato alla porta seriale.](#)

Questo nostro nuovo progetto tuttavia è un ulteriore passo avanti rispetto alla semplice unione di questi componenti. Difatti aggiungeremo dei tasti che permettono all'utente di interagire ed aggiunge la funzionalità di watchdog hardware per supervisionare il nostro server Linux. In aggiunta a tutto ciò il nostro pannello fornisce una linea di ingresso analogica, che noi ora non utilizzeremo. La si potrebbe usare, per esempio, per collegare un sensore di temperatura....

Per realizzare il tutto è bene che abbiate delle conoscenze basilari di elettronica, è sufficiente a livello hobby. I componenti che utilizzeremo sono economici ed il costo complessivo dovrebbe aggirarsi sui 40 Euro.

Il concetto alla base di questo pannello è di dare all'utente la possibilità di controllare il Vostro server senza l'ausilio nè di tastiera nè di monitor. Linux è un sistema operativo molto stabile ed affidabile e può facilmente esser controllato da remoto. Tuttavia la prima volte che lo collegate alla rete avete la necessità di configurare un indirizzo IP, il relativo gateway e la sottomaschera di rete. Il nostro pannello vi permette di configurare questi indirizzi. Vi dà anche la possibilità di spegnere il server mentre siete nella stanza dei server.

Il design per questo pannello è generico. Tutte le parti inerti il Server sono implementate per mezzo di uno script in perl. Tutto l'hardware, i pulsanti di stato, il testo sul display, i LED... insomma il complesso viene controllato attraverso una serie di comandi ASCII. Potete tranquillamente utilizzare il progetto per costruire un lettore mp3 o un controllo per vostro tostapane... insomma qualsiasi cosa la vostra fantasia possa suggerire.

Il materiale necessario

Per assemblare il progetto vi serve il seguente materiale elettronico:

- 1 x Microcontrollore Atmel At90S4433
- 1 x zoccolo 28pin 7.25 mm
- 1 x zoccolo 16pin
- 1 x CI MAX232
- 1 x relay da 5V di piccole dimensioni (meglio se formato ISO DIL)
- 1 x cristallo di quarzo da 4MHz
- 2 x LED (rosso e verde)
- 1 x transistor BC547 NPN
- 1 x transistor BC557 PNP
- 4 x condensatore da 1uF (va bene sia comune che polarizzato)
- 2 x condensatore ceramico da 27pF
- 1 x condensatore da 10nF
- 1 x condensatore da 100nF
- 3 x resistenze da 4k7
- 2 x resistenze da 2k2
- 1 x resistenza da 10K
- 1 x resistenza da 3k3
- 2 x resistenze da 100 Ohm
- 3 x resistenze da 470 Ohm
- 3 x resistenze da 1k
- 1 x resistenza da 220 Ohm
- 1 x trimmer (potenziometro) da 4K7
- 1 x diodo zener da 4.3V
- 1 x basetta fotosensibile monofaccia
- 2 x tasti di piccole dimensioni
- 1 x diodo al silicio comune (per esempio un 1N4148, o altro diodo economico)
- 1 x display LCD da 2 righe e 20 caratteri compatibile con HD44780.

Tutti i display LCD che io ho sperimentato come compatibili con l' HD44780 hanno o 14 o 16 pin. Potete, se volete, utilizzare display a 3 o 4 righe, nel qual caso dovrete modificare un poco il software.

In aggiunta a tutto questo vi serviranno alcuni fili e dei connettori di alimentazione e per interfacciarsi alla porta seriale RS232. Se utilizzerete un display a due righe potrete fissarlo in un piccolo telaietto di alluminio

I file in formato eagle (compressi con gzip; vi ricordo che alcuni browser la decomprimono automaticamente durante il download):

- [linuxlcdpanel.brd.gz](#)
- [linuxlcdpanel.sch.gz](#)

Lo schema elettrico

Vi spiegherò brevemente lo schema elettrico. L'AT90S4433 ha 3 porte: PB, PC e PD. PC può essere utilizzata come ingresso sia analogico che digitale. Tutte le altre porte possono essere utilizzate sia come ingresso che uscita sia in digitale che in analogico. Queste caratteristiche sono controllate per mezzo del software per mezzo della DRR (Data Direction Register = registro della direzione dei dati). Noi utilizzeremo tutti i pin tranne il 23 che è solo per linee non digitali (0/ 5Volt – TTL). Il chip MAX232 è un adattatore di livello. La porta RS232 utilizza segnali +-10V ed il MAX232 li converte in 0/5V. Sul pin 1 (pin di reset) dell'AT90S4433 trovate il circuito Brownout. Questo circuito mantiene lo stato del reset a livello logico 0 (attivo) per prevenire che, nel caso di alimentazione insufficiente, induca la CPU ad eseguire erronee istruzioni o indurre ad un malfunzionamento generalizzato. Questo potrebbe avvenire per alcuni millisecondi durante la fase di accensione o di spegnimento. Potremmo riassumere dicendo che questo circuito ci garantisce un avvio del programma in situazione pulita.

Qualcuno di voi potrebbe chiedersi perchè quel diodo sia in parallelo alla bobina del relè e sia collegato in quello strano modo – con la polarità invertita -. Questo diodo ha una funzione essenziale! Allorquando si rilascia il relè, la bobina di quest'ultimo genera una elevata extratensione che danneggerebbe di sicuro il nostro microcontrollore. Questa alta tensione ha polarità inversa rispetto a quella applicata alla bobina. Il diodo non è un modello particolare ma un comune economico diodo, ma di importanza vitale per il nostro circuito.

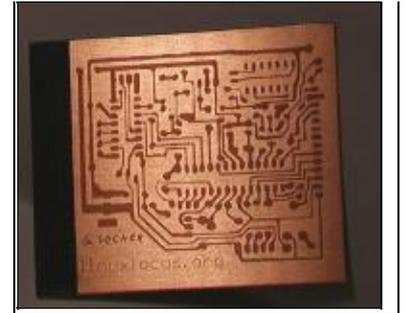
Creiamo il circuito stampato

Per incidere il circuito stampato avete bisogno di stampare questo [file postscript \(linuxlcdpanel.ps.gz\)](#) su di un lucido. Lo potete trovare nelle eliografie. Di norma viene chiamato Sinolit. Questo prodotto è della Regulus ed è comunemente usato per la stampa offset. Un'altra ottima alternativa è quella di ricorrere ad un foglio di carta da 60gr/mq ed uno spray trasparente (per esempio il pausklar 21 della Kontakt Chemie). Il vantaggio nell'utilizzo di una di queste due soluzioni è che il toner di una stampante laser si fissa bene e crea una ottima opacità nella stampa favorendo un ottima impressione della basetta.

Ho convertito il file postscript in uno [PDF](#) per coloro che non hanno un sistema di stampa postscript. Tuttavia la qualità ne risente.

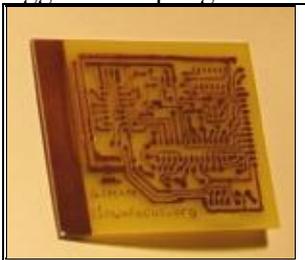
Il tempo di esposizione della basetta fotosensibile dipende dalla tipo di luce che si utilizza. Per una comune lampada solare, di uso casalingo, il tempo è circa di 1–2 minuti. Potete utilizzare anche la luce solare ma cercate di evitare una esposizione diretta alla luce (i raggi solari sono troppo forti). Vi consiglio di effettuare delle prove su dei ritagli di basetta prima di esporre alla luce tutta la basetta col circuito.





La basetta esposta

La basetta esposta deve esser sviluppata per alcuni minuti. Dovrà quindi esser immersa in NaOH (idrossido di sodio). Dopo questa operazione dovrete controllare attentamente il risultato ed eventualmente apportare delle correzioni. Nel caso alcune tracce non siano buone potete correggerle per mezzo di pennarelli appositi come l'Edding 780 (non si deve usare pennarelli permanenti ma pennarelli a vernice). Di norma creo delle basette leggermente più grandi di quanto Edge non consigli. Le ritengo un poco troppo piccole per un uso hobbistico.



La basetta finita, dopo l'immersione in acido, pronta per la foratura

Nota: Dato che non sembra esservi uno standard per la piedinatura dei relè potrebbe esser necessario modificare leggermente il layout dello stampato. Io ho utilizzato un relè della Matsushita a 5V.

Allorquando sarete soddisfatti del layout del vostro circuito stampato, potrete procedere all'acidazione del medesimo con del FeCl₃ (Cloruro Ferrico). FeCl₃ ha una buona capacità erosiva a temperatura ambiente. Si tratta di un acido semplice da utilizzare anche in ambito casalingo. Otterrete risultati migliori immergendo la basetta in verticale in un contenitore lungo e stretto. Dato che gli ioni di rame (Cu⁺) sono più pesanti degli ioni di ferro (Fe⁺) alla fine del processo troveremo nel fondo del contenitore gli ioni di rame.

Quando la basetta è pronta; ripulitela con della trementina per rimuovere le correzioni effettuate con il pennarello. Potete lasciare la pellicola fotosensibile sulle piste. Essa evaporerà naturalmente quando andrete a saldare, offrendo così una protezione per le piste di rame.

Il software per il microcontrollore

Il software per il microcontrollore è composto dai seguenti file:

- lcd.c, lcd.h, lcd_hw.h: Si tratta di una libreria generica per la gestione degli LCD per mezzo degli avr. È basata sul lavoro svolto da Peter Fleury (<http://jump.to/fleury>) Questa versione è stata leggermente adattata e più flessibile. Vi permette, infatti, di collegare il display a qualsivoglia pin del microcontrollore. Per fare ciò dovete solo cambiare le dichiarazioni nel file lcd_hw.h.
- avr-util.c, avr-util.h: funzioni per gestire i delay.

- [uart.c, uart.h](#): Questa libreria permette la gestione dell'interfaccia RS232. Sfrutta gli interrupt hardware. Ogni qualvolta un carattere è ricevuto dal computer la funzione SIGNAL(SIG_UART_RECV) viene eseguita ed i dati sono copiati dal buffer di ricezione in una stringa di buffer. Il set di comandi implementato in questo display fa sì che ogni comando termini con il carattere di nuova riga. Quando viene letto il carattere di nuova riga, viene settato il flag (uart_rx_linecomplete) ed i dati sono resi quindi disponibili al pannello. Questo significa che non potete inviare i dati al display il più rapidamente possibile, ma dovete attendere alcuni millisecondi tra una linea e l'altra. Ogni comando ha un responso di acquisizione (ACK) che può dare come risultato o ok o err (nel caso di errore). Il piccolo programma-driver scritto in perl può tuttavia utilizzare un trigger per inviare il comando seguente.
- [analog.c, analog.h](#): Codice per il convertitore da analogico a digitale. Anche questo sfrutta gli interrupt. Una singola conversione viene iniziata ed il programma attende il segnale dell'interrupt SIG_ADC per acquisirne il risultato dai registri ADC.
- [hardwarewd.c, hardwarewd.h](#): Si tratta del watchdog. Si ricorre ad un divisore interno (per 1024) per ottenere l'impulso di riferimento. Il timer è costituito da un registro a 16 bit che quando arriva all'overflow fa sì che noi si faccia un decremento di una variabile di 8 bit. Con un oscillatore a 4MHz eseguiamo questo decremento ogni 16 secondi. Il programma in perl esegue un incremento della variabile in questione segnalando lo stato di buon funzionamento del computer. Se il programma fallisce l'incremento (a causa di un blocco del computer), allorché questa variabile arriva a zero, in quanto decrementata nel tempo, il circuito si aziona attivando il relè, le cui uscite mettono in corto i pin di reset del nostro PC.
- [linuxlcdpanel.c](#): si tratta del programma principale. Eseguce un controllo continuo per eventuali comandi e pressioni di tasti sulla porta RS232.

Per comprendere appieno il software vi consiglio di leggere bene il datasheet del microcontrollore. È disponibile nella sezione riferimenti alla temine di questo articolo. (o dal sito www.atmel.com)
 Sebbene per usare questo pannello non sia necessario che comprendiate il suo software, vi serve solo decomprimere ed estrarre il codice sorgente dell'archivio (scaricate linuxlcdpanel-0.7.tar.gz from dalla [pagina dei download](#)) e digitate :

```
make
make load
```

oppure utilizzare la versione del software precompilata e caricatela nel microcontrollore con il comando
 make loadprebuild

Molto semplice, no? Potete trovare una descrizione inerente la programmazione del microcontrollore nel primo articolo di questa serie: [Programmare il microcontrollore AVR con GCC](#).

Verifica di funzionalità del pannello LCD

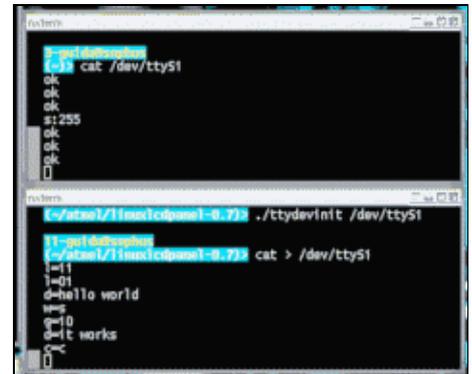
Il pannello LCD è studiato per funzionare con i 5V dell'alimentatore interno del PC (cavo rosso=5V, cavo nero=massa). Tuttavia è bene che la prova non la facciate collegando il display a questa fonte di alimentazione, in quanto il vostro alimentatore è assai robusto, ma la piastra madre ed il resto della componentistica, potrebbero fare una fumata a causa di un errore, fatto o nell'assemblaggio o durante la fase di saldatura. Verificatene la funzionalità dapprima con un alimentatore elettronico stabilizzato e limitato in corrente! Ora eseguite il download nella EEPROM come precedentemente descritto. Dopo questa operazione noterete un banner nel display che recita "linuxfocus.org". Ora collegate il cavo seriale all'interfaccia RS232:

```
pin 14 del MAX232 al segnale CTS (DB-9: pin 8)
pin 7 del MAX232 al segnale RXD (DB-9: pin 2)
pin 13 MAX232 al segnale TXD (DB-9: pin 3)
```

Dovete anche collegare il segnale DTR,DSR e CD tra di loro (DB-9: pin 4, 6 ed 1)

Questo è ben documentato nello schema elettrico precedente.

Per utilizzare una linea seriale, prima la si deve inizializzare. Il file `linuxlcdpanel-0.7.tar.gz`, contenente il codice sorgente, include un programma che ha per nome `ttydevinit` che, per l'appunto, inizializza questa linea. Per esempio, poniamo, che voi abbiate collegato il pannello alla porta COM2 (`ttyS1`), dovreste dare il seguente comando una sola volta:



```
root@kali:~# cat /dev/ttyS1
ok
ok
s:255
ok
ok
ok

root@kali:~# ./ttydevinit /dev/ttyS1
l=11
l=10
d=hello world
w=10
q=10
o=t works
c=1
```

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

Ora il driver per la linea seriale in questione è inizializzato a 9600 Baud e voi potrete iniziare a "dialogare" con il vostro pannello LCD. Aprite due xterm in X-Windows. In una digitate "`cat /dev/ttyS1`", e nell'altra "`cat > /dev/ttyS1`". Ora potete, per esempio digitare il comando `l=11` (accendi il LED nr. 1) o `l=10` (spegni il LED nr. 1). Potete vedere nella prima, delle due console xterm, l'acknowledge (ovvero la risposta) al vostro comando ("ok").

Tutti i comandi sono spiegati nel file [README.commands](#) (in lingua inglese).

L'archivio contenente il codice sorgente include uno script in perl (`tytest.pl`) la cui funzione è null'altro tranne che far lampeggiare il led rosso. L'idea di base è di fornire un semplice esempio per far vedere come si possa gestire un pannello LCD. Potete usare questo codice come base per i vostri programmi. Date liberamente una occhiata al codice in questione. È bene avere un poca di conoscenza del linguaggio perl anche se questo programma è breve.

Collegiamo il watchdog

Il sistema di supervisione è normalmente spento. Per attivarlo dovete usare il comando `w=1` e definire il suo timeout con `s=x`, dove `x` è un valore compreso tra 1 e 255. – per esempio `s=10` permetterà al watchdog di avere un tempo di timeout pari a $10 * 16\text{sec}$, ovvero 160 secondi. – È essenziale definire il tempo di timeout o il sistema potrebbe esser indotto a forzare un reboot della macchina. Difatti se il server si dovesse bloccare, il programma non ri-definirebbe più il corretto timeout inducendo il watchdog ad attivarsi. So che un server Linux non si ferma mai. Tuttavia, nei rari casi in cui si ferma, non trovate mai nessuno che possa premere il tasto di reset o nessuno ricorda più dove esso sia in quanto sono anni che lo usano ma che non vi sono stati problemi.

Il watchdog è stato studiato dimodochè possa intervenire solo una volta. La ragione di questo è da ricercarsi nel fatto che, al riavvio del sistema, quest'ultimo potrebbe impiegare molto tempo per effettuare il check del file system. Quando, alla fine, il server sarà nuovamente online. Il software a quel punto riattiverà il driver per effettuare il pool.

Per collegare fisicamente il Vostro watchdog divete trovare quali siano i fili che fanno capo al pulsante di reset del vostro server. In parallelo a questo colleghere i cavi che vanno al nostro relè.

Come utilizzare il watchdog

Il watchdog ha la funzione di garantire la capacità del sistema (inteso come Sistema Operativo) di eseguire operazioni, non che il server web o il database sia online. Per verificare la permanente funzionalità di questi

programmi dovrete ricorrere alle funzionalità del cron. Possiamo considerare sufficientemente affidabile la crontab, in quanto il watchdog assicura che, in generale, sia in esecuzione.

Potete, per esempio, creare uno script, per mezzo della crontab, che ogni 15 minuti effettui il download di una pagina. Dovrete però prestare attenzione a che, questo download, potrebbe anche portare ad un responso negativo se si tratta di un server ad elevato traffico. Dovrete quindi valutare le risposte. Per esempio se risponde una sola volta nelle ultime 10 richieste è bene che forziatelo un restart del web server, o un normale reboot, piuttosto che ricorrere ad reset fisico (non quindi un reset hardware indotto dal watchdog).

Oltre a supervisionare lo stato delle applicazioni, è essenziale monitorare lo stato d'occupazione dei dischi. Il seguente set di comandi daranno una risposta allorché uno dei filesystem superi l'80% di occupazione:

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

Anche questa serie di comandi può essere inserita come voce nella crontab per avere una costante supervisione dell'uso dei dischi.

Gli script del server

La maggior parte della logica di controllo del nostro pannello LCD è implementata per mezzo di uno script in perl: llp.pl. Copiate questo script nella cartella /usr/sbin. Copiate successivamente il programma ttydevinit nella cartella /usr/bin ed il file ifconfig_llp.txt (contenuto nella cartella etc ove si trovano i file sorgenti) in /etc. Modificate ora il file ifconfig_llp.txt ed apportate le modifiche di indirizzo IP secondo le vostre esigenze (qui trovate un esempio):

```
NETMASK=255.255.255.0  
IPADDR=10.0.0.4  
GATEWAY=10.0.0.2
```

Fate ora una copia di backup del seguente script: /etc/rc.d/init.d/network e sostituitelo con lo script etc/network preso dalla cartella dei sorgenti. Questo script ed i nomi delle cartelle sono validi solo per le distribuzioni RedHat e Mandrake di Linux. Lo script etc/network_all_distributions è basilare e dovrebbe funzionare con tutte le distribuzioni di Linux, ma dovete capire dove la vostra distribuzione abbia gli script di init/rc. Questi parametri cambiano leggermente da distribuzione a distribuzione.

Modificate il file /etc/rc.d/init.d/network e cambiate la riga

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

Se il watchdog non è collegato alla COM2.

Ora potete eseguire il seguente comando.

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

e vedere il nostro pannello LCD in azione. Nota: il nostro pannello salva e modifica l'indirizzo IP. I cambiamenti hanno effetto solo dopo il reboot seguente. Potete fare delle prove cambiandolo e poi ripristinando i valori di prima. È bene che facciate in ogni caso una prova riavviando poi il server. Alla fine potete o ripristinare la copia di backup o editare a mano il file /etc/ifconfig_llp.txt per ripristinare i valori.

I file di log

Lo script llp.pl genera un file di registro (/var/log/llp.log). Questo file crescerà molto lentamente. Non vi dovrebbe quindi essere la necessità di farne il rotate (ovvero quella procedura che crea una copia del file attuale e ne crea uno nuovo di dimensione 0). Potete effettuare il rotate di questo file con il programma logrotate, se lo desiderate. Non vi è la necessità di fare operazioni dopo il rotate. La riga di configurazione del rotate (all'interno dello script logrotate) potrebbe essere:

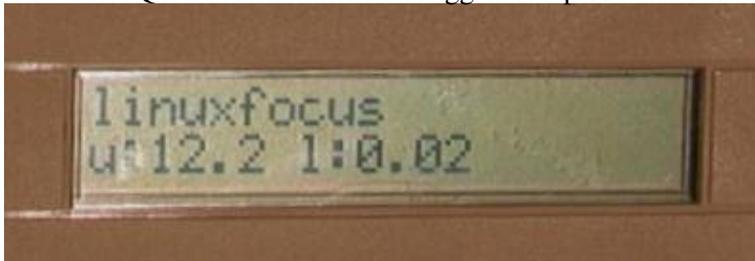
```
/var/log/llp.log {  
nocompress  
monthly  
}
```

Il file di registro contiene le voci inerenti lo spegnimento manuale ed il cambio delle configurazioni di rete (cambio di indirizzo IP, di gateway o sottomaschera di rete) o quando il watchdog ha forzato un riavvio. Ovviamente sarà impossibile avere delle voci inerenti il timeout del watchdog (difatti il sistema era bloccato!), ma troveremo una voce per questo tipo di problematica aggiunta al riavvio del sistema.

Il pannello al lavoro.

Qui trovate alcune immagini del pannello LCD in uso. Queste non sono tutte le funzioni offerte dal pannello. Ve ne sono molte altre che potrete aggiungere voi stessi.

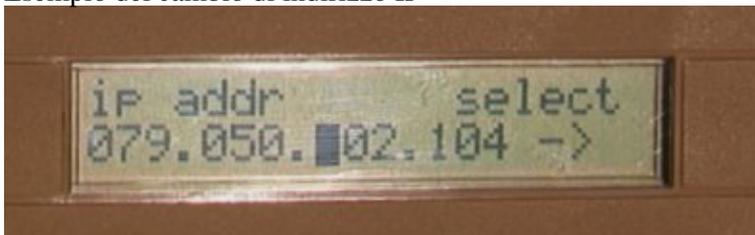
La finestra principale. Mostra alcuni nomi (in questo caso linuxfocus), il tempo di uptime ed il carico medio del server. Queste informazioni sono aggiornate periodicamente.



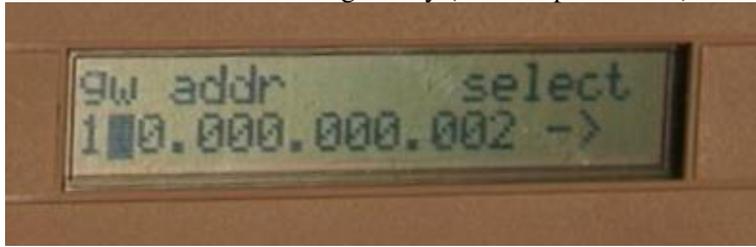
Il menu di impostazione della rete



Esempio del cambio di indirizzo IP



Come variare l'indirizzo del gateway (la route predefinita)



Conclusioni

Per assemblare questo pannello LCD è necessaria una conoscenza basilare dell'elettronica. Il nostro Pannello LCD per Linux offre moltissime funzionalità, molte di più di quelle che io abbia visto su altri prodotti simili, e, soprattutto, ha un costo modesto.

Buon lavoro :-)



Bibliografia

- Il software usip per programmare i microcontrollori AVR: www.amelek.gda.pl/avr/
- I sorgenti utilizzati in questo articolo linuxlcdpanel-0.7.tar.gz . Lo schema elettrico, i file in formato Eagle e alcune immagini sono altresì incusi.
- Tutto il software e la relativa documentazione [citata in questo articolo](#)
- Il datasheet del chip MAX232 [MAX220-MAX249.pdf 448K](#)
- Il datasheet del chip ST232, una variante economica e spesso venduta come sostituto del vero MAX232 [st232.pdf 100K](#)
- Il datasheet dell'Atmel AT90S4433 [avr4433.pdf 2356K](#)
- Il sito web della atmel: www.atmel.com/
- Eagle, versione per Linux cadsoftusa.com

Webpages maintained by the LinuxFocus Editor team

© Guido Socher

"some rights reserved" see linuxfocus.org/license/

<http://www.LinuxFocus.org>

Translation information:

en --> -- : Guido Socher ([homepage](#))

en --> it: Toni Tiveron <toni/at/amicidelprosecco.com>