

Un alimentatore in CC basato su microcontrollore



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:

Questo è il quarto articolo della serie presentata su **LinuxFocus** inerente il microcontrollore AT90S4433. Vi suggerisco di leggere i precedenti articoli che trattano del microcontrollore della Atmel:

1. Come installare ed utilizzare l'ambiente di sviluppo e costruire l'hardware necessario alla programmazione del microprocessore: [Marzo 2002, Programmare il microcontrollore AVR con GCC](#)
2. Come creare il vostro circuito stampato: [Un pannello di controllo LCD per il vostro server Linux](#)
3. Come costruire un contenitore per il vostro alimentatore: [Settembre 2002, Un frequenzimetro da 1Hz a 100MHz, munito di un display LCD ed una interfaccia RS232](#)

L'elemento più importante per i vostri 'lavoretti' domestici è un buon alimentatore in corrente continua. In questo articolo andremo a costruirne uno. Sarà gestito da un microcontrollore. Avrà un piccolo display LCD, e potrete pilotarlo da remoto con Linux per mezzo della porta seriale. Ha anche un design sobrio ma efficace.

Questo articolo ci dimostrerà quanto versatili siano i Microcontrollori. Tuttavia quello che andremo a costruire non sarà il circuito più semplice della serie. Se state cercando un semplice alimentatore in corrente continua potete dare un'occhiata all'articolo ["Un semplice alimentatore in CC"](#). Questo semplice alimentatore è ottimo per delle modeste necessità, come spesso richiedono i circuiti presentati su : **LinuxFocus**. È da precisare che questo semplice alimentatore non ha nulla a che fare con Linux e con il software. Anche se andrete a costruire il "semplice alimentatore in CC" potete ugualmente leggere questo articolo ed apprendere molte cose interessanti sui microcontrollori.

Introduzione

Questo alimentatore CC è basato su di un circuito che non è certo dei più semplici ma, vi posso assicurare che il tempo che vi dedicherete non lo rimpiangerete. È molto affidabile e robusto. Esso è anche molto interessante dal punto di vista tecnico in quanto vi spiegherà come costruire un alimentatore controllato da microprocessore che generi una tensione analogica senza ricorrere ad un integrato–convertitore DA.

Vi serviranno molti componenti per questo articolo ma la maggior parte è economica e di facile reperibilità. L'alimentatore in sé non è particolarmente costoso.

Quello che vi serve

Date un'occhiata alla [lista dei componenti](#) per avere le idee chiare su cosa vi serve di preciso. Potete veder anche i componenti nel seguente schema elettrico.

Il nostro alimentatore è disponibile in 3 varianti. A parte il trasformatore ed una resistenza le uniche modifiche sono di tipo software. Tutto il resto dei componenti non varia per le tre versioni:

1. 0–16V $I_{max}=2.2A$
acquistate un trasformatore da 15V 2.5A
2. 0–24V $I_{max}=2.2A$
acquistate un trasformatore da 24V 2.5A
3. 0–30V $I_{max}=3A$
acquistate un trasformatore da 30V 3A

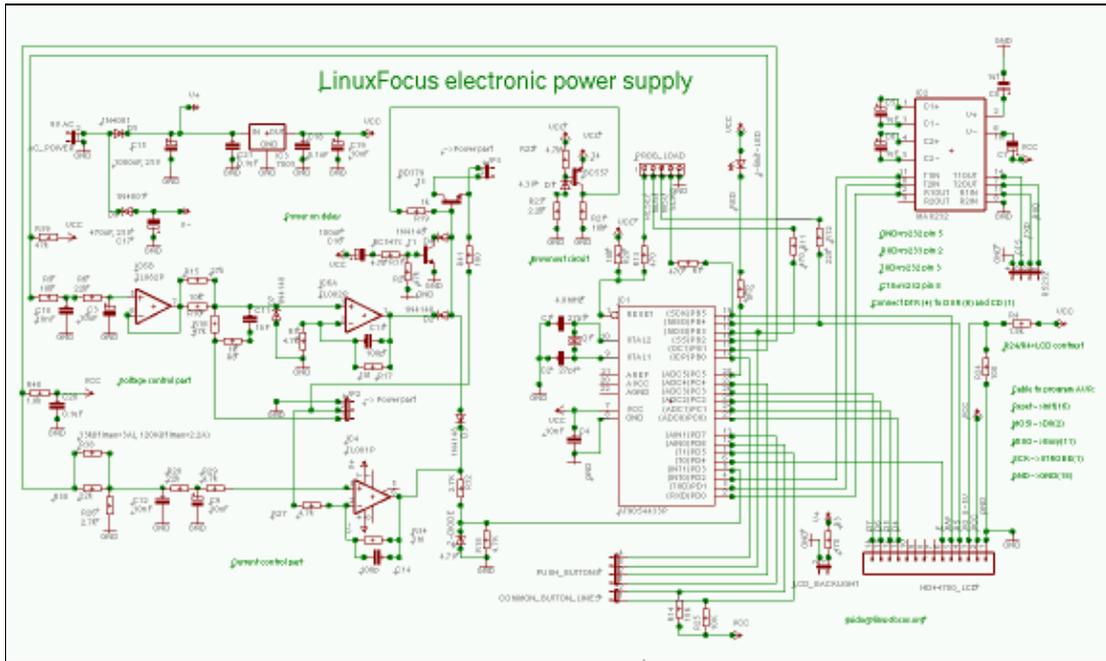
In tutti e 3 i casi avrete bisogno anche di un piccolo trasformatore ausiliario da 9V 100mA per alimentare il circuito di controllo.

Schema elettrico e circuito stampato

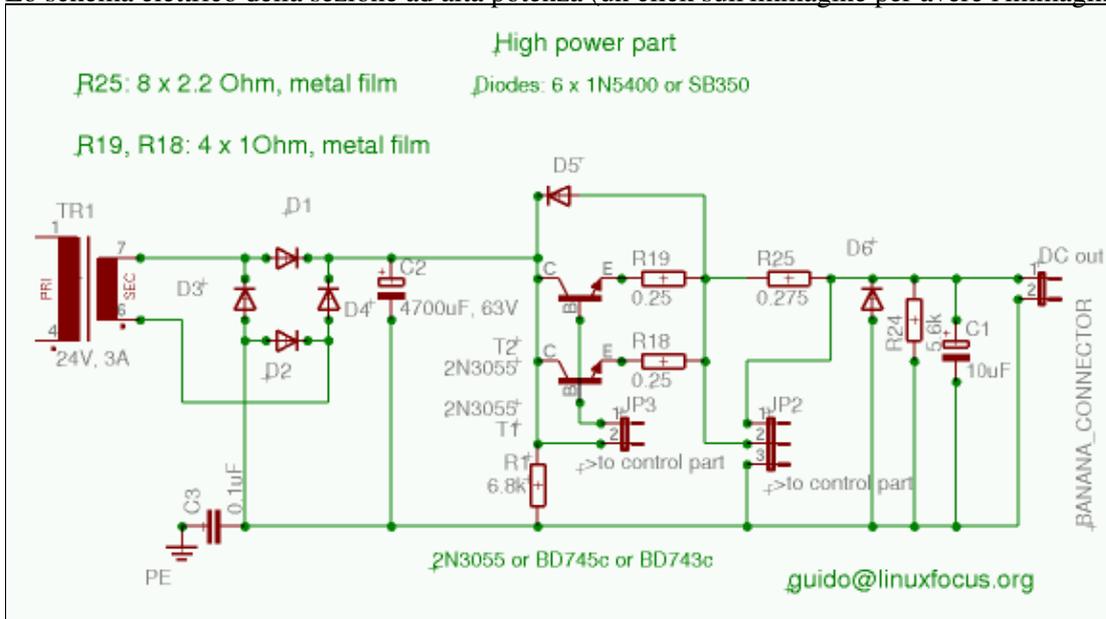
Ho utilizzato [eagle](#) per Linux per la creazione dello schema elettrico e del circuito stampato. I file in formato eagle sono inclusi nel file tar.gz assieme al software. Lo potete scaricare alla fine dell'articolo.

Il circuito è diviso in due parti. Una parte principale ed una parte che si situa vicino ai transistor di potenza. Qui potete vedere i due schemi elettrici delle due parti, che verranno interconnessi per mezzo di spezzoni di cavo elettrico.

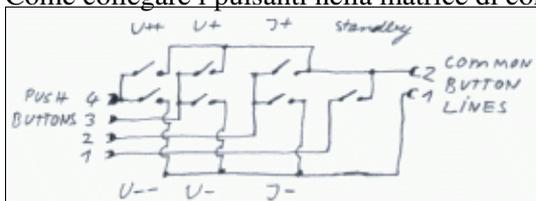
Lo schema elettrico della sezione principale (un click sull'immagine per avere l'immagine più grande):



Lo schema elettrico della sezione ad alta potenza (un click sull'immagine per avere l'immagine più grande):

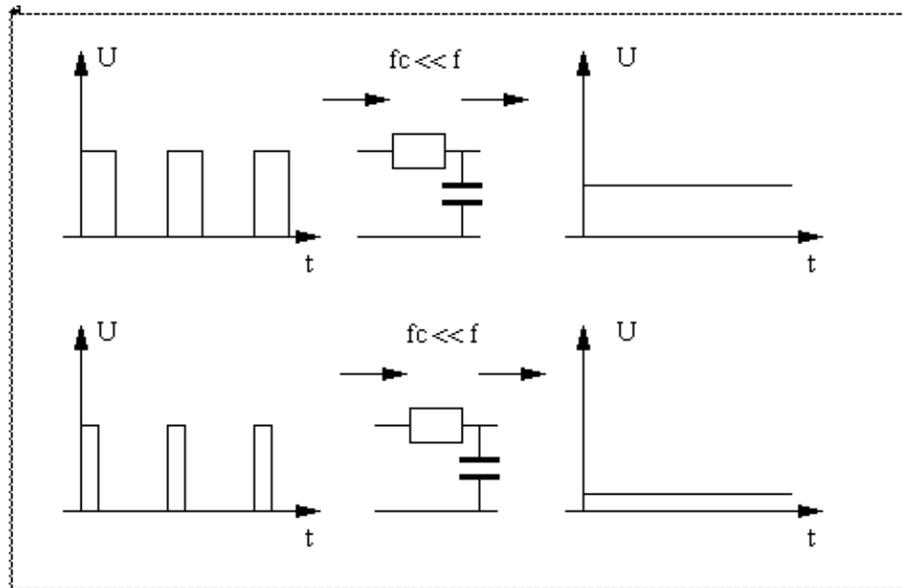


Come collegare i pulsanti nella matrice di comando (un click sull'immagine per avere l'immagine più grande):



Il circuito di controllo visto dal lato componenti un click sull'immagine per avere l'immagine più grande):

Per ottenere la massima corrente o per regolare l'uscita dell'alimentatore dobbiamo solo fornire la corretta alimentazione ai due punti di potenza (pin 5 dell'IC6B ed al resistore R30). Questo è quanto fa il microcontrollore... ma come è in grado di generare e regolare propriamente una tensione di riferimento in corrente continua? Date un'occhiata al seguente schema:



Quello che potete notare in questo schema è come un segnale ad impulsi possa essere trasformato in un segnale in corrente continua. Tutto quello che si deve fare è far transitare il segnale attraverso un filtro passa-basso con una frequenza di cut-off di circa 10 volte (o più) della frequenza del segnale. Dato che il nostro microcontrollore lavora ad una frequenza di 4MHz, risulta abbastanza semplice costruire questo genere di filtro passa-basso. Anche se implementassimo il segnale via software otterremo comunque una frequenza di qualche KHz ed il filtro risulterebbe ancora molto compatto e semplice.

La differenza tra lo schema in alto e quello in basso viene definita come modulazione in ampiezza d'impulso (PWM). Variando la lunghezza degli impulsi possiamo cambiare la tensione all'uscita del filtro.

Carino vero?? Già... possiamo generare una precisa tensione in corrente continua partendo da un segnale digitale!

Il microcontrollore AT90S4433 ha due contatori interni. Uno di 1bit di ampiezza ed uno di 8bit. Il contatore a 16bit ha la possibilità di lavorare in PWM, che viene quindi implementata in maniera hardware all'interno dell'AT90S4433 con una risoluzione di 10bit. Il contatore ad 8bit non ha questa funzionalità, ma possiamo emularla ed implementarla per mezzo del software. Sarà ancora sufficientemente veloce. Utilizzeremo il contatore a 16bit per la regolazione della tensione, ottenendo una definizione di 1023 passi (10bit sono, difatti, pari a 1023). Il controllo della corrente in uscita viene demandato al contatore ad 8bit, che ci permetterà di ottenere 155 passi di controllo della corrente (da 1 a 3000 mA). Questo significa che avremo una precisione dell'ordine di circa 12mA. Questo è un valore più che sufficiente per un controllo in corrente di un alimentatore.

Tutto il resto della componentistica nel circuito serve per l'alimentatore e per generare la nostra tensione di riferimento (il circuito integrato 7805, è difatti il nostro elemento di riferimento) e per garantire che il nostro alimentatore non si comporti in maniera impropria all'accensione o allo spegnimento.

Il software

Il software per il microcontrollore utilizza molti degli aspetti che dovrete avere già appreso dai precedenti articoli (uart per le porte seriali rs232, display LCD, contatori ed modalità di interrupt). Potete dare un'occhiata qui:

[linuxdcp.c](#).

La parte più interessante probabilmente è il circuito PWM (Pulse Width Modulation – modulazione ad ampiezza d'impulso). La variabile `ipwm_phase`, congiuntamente a `ipwm_h` gestisce la tecnologia PWM per quel che concerne la parte della sezione di potenza in corrente. Possiamo gestire il contatore ad 8bit in modalità interrupt ogni qualvolta si genera un overflow, chiamando la funzione "SIGNAL(SIG_OVERFLOW)". A questo punto si valuterà se la variabile `ipwm_phase` debba produrre un valore di 0 o di 1 come responso della verifica prima di far ripartire il timer.

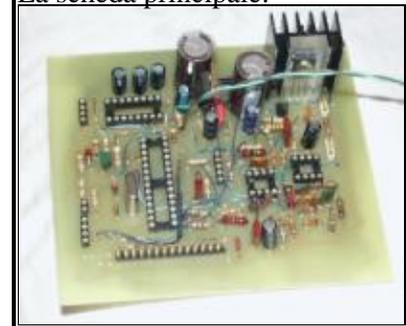
Il software non è per nulla complesso ma per riuscirlo a comprendere appieno è importante comprendere perfettamente il data sheet del 4433 (date una occhiata a fine articolo, nella sezione della bibliografia per la documentazione appena citata).

Il 4433 è un microprocessore ad 8bit e quindi la sua capacità matematica è assai limitata. Le funzioni `divXbyY` e `multiXbyY` richiedono una implementazione ad almeno 24bit per poter calcolare la corretta ampiezza d'impulso, partendo dalla tensione definita dall'utente.

Il nostro alimentatore ha 7 pulsanti. 6 di questi servono per aumentare e ridurre corrente e tensione ed il settimo lo utilizziamo per mettere l'unità nello stato di standby. Nello stato di standby potete temporaneamente spegnere l'alimentatore e cambiare i limiti della corrente e della tensione. Lo stato dei pulsanti viene ciclicamente rilevato dal ciclo continuo del programma principale. La variabile `ignorebutton` viene utilizzata per evitare il bouncing dei pulsanti. Questo in quanto, quando le vostre dita andranno a premere il pulsante non saranno fermissime e potrebbero generare più impulsi. Come esseri umani non lo notiamo, ma il nostro microcontrollore è talmente veloce nell' eseguire le operazioni che rivelerebbe questa imperfezione. La variabile `ignorebutton` ignora per una frazione di tempo lo stato del pulsante alla pressione, per evitare questo tremolio.

La creazione del circuito stampato

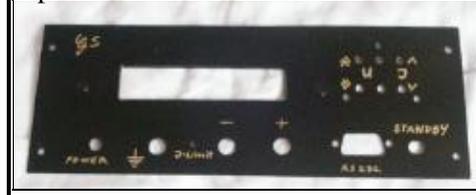
La scheda principale:



Il contenitore. Legno sulle pareti, un foglio di metallo per la parte dei pulsanti, il fondo ed il coperchio:



Il pannello frontale:



Il pacchetto software contiene al suo interno un file PostScript (linuxDCpower.ps) per il circuito stampato. Dal mio punto di vista, trovo che le piazzole di saldatura siano un poco troppo piccole. Vi consiglio di ingrandirle con un pennarello per incisioni. Il procedimento per realizzare un circuito stampato lo potete trovare nell'articolo [Un pannello di controllo LCD per il vostro server Linux](#).

Per costruire un economico ma funzionale contenitore per il vostro alimentatore potete leggere l'articolo [Un frequenzimetro da 1Hz a 100MHz, munito di un display LCD ed una interfaccia RS232](#). Potete vedere il contenitore ed il pannello frontale come li ho fatti io. Potete clickkare sull'immagine per vederla ingrandita.

Verifica di funzionamento

Come ogni circuito che, viene realizzato in maniera hobbistica, è bene non venga direttamente collegato ad una fonte a piena potenza. Vi consiglio quindi, di collegarlo ad una fonte di alimentazione più ridotta, per effettuare un test in maniera più sicura. Questo al fine di trovare difetti che si possono aver generato durante l'assemblaggio del medesimo.

1. Assemblate la basetta principale con tutti i componenti ma non inserite gli integrati negli zoccoli.
2. Utilizzate una batteria da 9Volt. Collegate il positivo della medesima al pin2 ed il negativo al pin1 del connettore che ha come nome AC_POWER nello schema elettrico. Con un multimetro controllate di avere +5V sui piedini 8 e 16 dell'integrato max232 e sui piedini 8 e 7 del microcontrollore. Sul pin di alimentazione dell'operazionale dovrete avere all'incirca 9Volt.
3. Ora invertite la polarità della batteria (collegate il positivo al pin 1 e il negativo al pin 2) e verificate che ci siano all'incirca 9volt sul piedino dell'alimentazione negativa dell'integrato operazionale.
4. Se queste verifiche hanno avuto esito positivo potete inserire gli integrati nei relativi zoccoli (tranne l'operazionale).
5. Ricorrete ancora alla batteria da 9volt e collegatela in modo da avere l'alimentazione a 5Volt. Collegare il programmatore alla porta parallela per mezzo del cavo. Scompattate il software (per scaricarlo consultate la fine dell'articolo), entrate nella cartella che si è creata e digitate:

```
make avr_led_lcd_test.hex  
make testload  
make ttydevinit
```

Ora il software di test ciclicamente esser caricato nel circuito. Sul display LCD dovrete leggere

"hello", il LED rosso dovrebbe lampeggiare e se il vostro computer è collegato all'interfaccia rs232 dovrete veder una serie di "ok" (inizializzate la linea seriale con il comando ttydevinit, e osservatene il responso con cat /dev/ttyS0, se lo avete collegato alla porta COM1, e cat /dev/ttyS1, se lo avete collegato a COM2).

6. Ora assemblate il circuito di potenza ma non collegate ancora il trasformatore di potenza. Collegate invece la batteria a 9Volt ai cavi che sarebbero, normalmente, collegati al trasformatore. Non importa la polarità della batteria, in ogni caso il condensatore da 4700 microfarad si dovrebbe caricare a circa 9Volt. Controllate questo valore con un VoltMetro.
7. Quando anche l'ultimo test ha avuto esito positivo, collegate il trasformatore con i relativi cavi ed alimentatelo. Senza l'integrato operativo, dovrete avere all'uscita dell'alimentatore la massima tensione possibile. Misurate questa tensione, ma prestate attenzione a non generare dei corto- circuiti, che andrebbero a danneggiare irrimediabilmente i transistor di potenza, in quanto non vi è ancora alcuna limitazione alla massima corrente erogabile.
8. Spegnete tutto ed inserite l'amplificatore operativo e collegate nuovamente il cavo di programmazione, alimentate e digitate:
make
make load
9. Ora l'alimentatore dovrebbe essere completamente funzionante. Vi prego di notare che fintatoché il cavo di programmazione è collegato l'uscita dell'alimentazione è imprecisa. Scollegate questo cavo per ottenere una alimentazione stabile e precisa.

Eccoci alla fine: Il nostro alimentatore

Come avrete visto precedentemente vi sono 3 possibili opzioni, che dipendono dal trasformatore che andrete ad utilizzare. Il software è nominalmente configurato per la tensione di 16Volt e 2.2Ampere. Per cambiare questa tensione dovrete modificare il file [linuxdcp.c](#) cerando le stringhe:

MAX_U, IMINSTEP, MAX_I, e nella funzione set_i dovrete cambiare il valore per avere un massimo di 3Ampere. Il codice è ben commentato e vi trovate i cambiamenti da apportare.

Ora potrete vedere alcune fotografie dell'alimentatore che io ho costruito. Ho impiegato del tempo, ma direi che ne è valsa la pena, dato che ho ottenuto un alimentatore robusto. Il tempo lo considero ben investito, in quanto, un alimentatore da laboratorio è uno degli strumenti più utilizzati.





Utilizzare l'alimentatore

È probabilmente molto ovvio come si utilizzi questo alimentatore. Avete 4 pulsanti per regolare la tensione d'uscita. Due pulsanti per incrementare o ridurre di 1Volt e due per 0.1Volt. La limitazione in corrente si può gestire con gli altri 2 pulsanti. Qui il fattore di incremento/decremento non è in scala lineare: per piccoli valori i passi sono di 50mA, per valori superiori ai 200mA sono di 100mA e per valori superiori all'Ampere sono di 200mA. Per questa ragione è semplice variare la limitazione con solo due pulsanti.

Il pulsante di standby può essere utilizzato per spegnere momentaneamente l'alimentatore senza aver poi la necessità di configurare nuovamente i valori impostati.

Il LED rosso si accenderà quando avrete superato il limite di carico in corrente e lampeggerà quando sarà in modalità stand-by.

L'alimentatore può anche essere completamente controllato per mezzi di comandi ascii inviati presso la porta seriale rs232. Questi sono i comandi utilizzabili:

u=X per definisce la tensione (e.g u=105 porta la tensione a 10.5V)

i=Xmax definisce la massima corrente (e.g i=500 porta il massimo assorbimento a 500mA)

s=1 or s=0 lo porta in modalità stand-by

u=? or i=? or s=? ci va vedere lo stato e le configurazioni. Questo produce un responso di questo tipo:

u: 50 s:0 i: 100 l:0

u: tensione pari a 5V, s:0 l'unità è operativa (non in modo stand-by), i: 100 significa massima corrente a 100mA, e l:0 ci dice che non è mai stato raggiunto il limite di carico.

Per mezzo di questi semplici comandi ascii sarete anche in grado di creare una interfaccia grafica per questo alimentatore. Per poter utilizzare la linea seriale la dovrete inizializzare con il comando `ttydevinit`, che è incluso nel pacchetto software. Questo comando è stato descritto nell'articolo [Settembre 2002. Un frequenzimetro da 1Hz a 100MHz, munito di un display LCD ed una interfaccia RS232.](#)

Come avrete notato nello schema elettrico noi utilizziamo 2 trasformatori ed il segnale di massa (prelevato dall'alimentazione ad alta tensione, è collegato al positivo dell'uscita del nostro alimentatore. I due trasformatori isolano le varie tensioni e, quindi, non vi sono di norma problemi con questa configurazione. Sono dovuto ricorrere a questa configurazione per avere la corretta polarità nel circuito di ritorno dell'amplificatore operazionale. **Una piccola avvertenza:** questa configurazione implica anche che la linea di massa della porta rs232 è collegata all'uscita positiva dell'alimentatore. In parole povere NON potrete utilizzare la linea seriale se volete utilizzare l'alimentatore con altri apparati che in qualche modo siano collegati alla linea di terra del vostro impianto e quindi, per mezzo di quest'ultima, anche al vostro computer. Potrebbe esser una buona idea applicare una etichetta che rechi questa avvertenza. Se volete essere completamente sicuri si non creare alcun cortocircuito, vi consiglio di ricorrere all'uso di un computer

portatile alimentato a batterie, o siate sicuri che il circuito che andrete ad alimentare non ha alcun collegamento con il vostro impianto di terra, oppure non collegate l'alimentatore alla linea seriale mentre alimentate l'apparato. Non siate troppo turbati da questo avvertimento. Se non supererete l'intensità massima di 250mA, il LED rosso vi avvertirà immediatamente se qualcosa non funzionasse correttamente, ed, in questo caso, il vostro computer non subirà alcun danno se commetterete degli errori.

Sicurezza

Il circuito contiene dei trasformatori che sono collegati alla linea di alimentazione di casa vostra (220Vo 110V, a seconda del paese in cui vi trovate), quindi verificate che il tutto sia correttamente isolato. Se non avete mai avuto a che fare con alimentatori, chiedete a qualcuno che abbia una maggiore esperienza affinché controlli il vostro operato per quel che concerne la sicurezza d'uso e l'isolamento del tutto. Questo prima che voi colleghiate il circuito all'impianto elettrico di casa vostra.

Ottimizzazione

Il software per il controllo dell'alimentatore è già di per sè calibrato, Nella maggior parte dei casi non dovrete cambiare nulla. La calibrazione dal punto di vista hardware dipende solo da pochi componenti: 7805, R15, R10, R16, R38,R30 e R26. Sono solo questi che influenzano la tensione e la corrente. Se volete effettuare una perfetta ottimizzazione potete cambiare questi resistori o modificare il software. Vi ricordo anche che il cavo per la programmazione influenza la precisione dell'alimentatore. Prima di effettuare qualsiasi misura, vi consiglio di scollegare questo cavo. Nel software potete apportare dei cambiamenti alle sezioni `set_u` e `set_i`. Il codice [linuxdcp.c](#) è sufficientemente commentato.

Riferimento e bibliografia

- Il software per programmare l'avr: www.amelek.gda.pl/avr/
una copia locale del medesimo: uisp-20011025.tar.gz
- Come costruire il programmatore ed installare il compilatore per l'AVR
[Marzo 2002, Programmare il microcontrollore AVR con GCC](#)
- I sorgenti citati in questo articolo [linuxdcpower-0.1.tar.gz, 1201K](#). Lo schema elettrico, i file in formato Eagle, ed alcune immagini dimostrative sono incluse nel file
- Tutto il software (eventuali aggiornamenti saranno si potranno trovare qui) e la documentazione:[software/datasheets](#)
- Le caratteristiche tecniche del bd379 [bd379.pdf 44K](#)
- Le caratteristiche tecniche del TL082 [TL082.pdf 110K](#)
- Le caratteristiche tecniche del TL071 [TL071.pdf 268K](#)
- Le caratteristiche tecniche del 2n3055 [2n3055.pdf 64K](#)
- Le caratteristiche tecniche dell'integrato MAX232 [MAX220-MAX249.pdf 448K](#)
- Le caratteristiche tecniche dell'integrato ST232, una variante economica dell'integrato MAX232, spesso venduto come valida alternativa del MAX [st232.pdf 100K](#)
- Le caratteristiche tecniche dell'Atmel AT90S4433 [avr4433.pdf 2356K](#)
- Il sito web della atmel: www.atmel.com/
- Eagle per Linux cadsoftusa.com

<p>© <u>Guido Socher</u> "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>en --> -- : Guido Socher (homepage) en --> it: Toni Tiveron <toni(at)amicidelprosecco(dot)com></p>
--	---

2005-01-10, generated by lfparsr_pdf version 2.51