

Un frequenzimetro da 1Hz a 100MHz, munito di un display LCD ed una interfaccia RS232



by Guido Socher ([homepage](#))

About the author:

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



Abstract:

Questo articolo fa parte della serie inerente il Microcontrollore AT90S4433. Vi suggerisco di leggere gli articoli precedenti sul Microcontrollore della Atmel:

1. Come installare ed utilizzare l'ambiente di sviluppo e come costruire il programmatore:
[Marzo 2002, Programmare il microcontrollore AVR con GCC](#)
2. Come crearsi i propri circuiti stampati:
[Maggio 2002, Un pannello di controllo LCD per il Vostro server Linux](#)

Questa volta ci occuperemo di un frequenzimetro che è in grado di misurare da 1Hz a 100MHz. Potete anche utilizzarlo come semplice conta eventi, per esempio sapere quante persone hanno attraversato la strada (o quello che la vostra fantasia suggerisce, purché si tratta di un impulso di tipo digitale). Il contatore ha un display LCD ed una porta seriale RS232 per poter leggere e gestire i risultati all'interno di Linux.

Introduzione

La frequenza generata da un segnale, sinusoidale od a onda quadra, è il numero di oscillazioni che essa compie al secondo. Di conseguenza per determinare la frequenza di un segnale continuo, si devono semplicemente contare il numero di oscillazioni che si hanno in un secondo. In questo modo possiamo

ottenere la misura della principale armonica di un segnale oscillante. Per misurare le frequenza non stabili di un segnale, come per esempio il suono, si dovrà ricorrere ad un analizzatore di spettro. In quest'ultimo caso però si tratterebbe di un hardware ben diverso. Quello che andremo ora a sviluppare è un oscillatore per segnali stabili. Partiremo quindi dal presupposto che questo segnale non muterà la propria frequenza durante un determinato lasso di tempo che definiremo come intervallo di campionamento.

Il nostro frequenzimetro lavora in due distinte fasi:

1. Rimuove ogni residuo di corrente continua dal segnale e poi lo converte in un segnale ad onda quadra per mezzo di un comparatore di tensione.
2. Conta il numero di impulsi per unità di tempo e poi lo divide per l' intervallo.

Quello che vi serve

Per costruire il frequenzimetro avrete bisogno dei seguenti componenti

1 x Microcontrollore Atmel At90S4433

1 x zoccolo 28pin 7,25 mm

2 x zoccoli 16pin

1 x zoccolo 1pin

1 x zoccolo 14pin

Nessuno zoccolo per il MAX903 e per lo 74F74. Saldateli direttamente sul circuito stampato!

1 x MAX232

1 x quarzo da 4,194304MHz

1 x LED (verde)

1 x BC557 (transistore PNP)

4 x condensatori (elettrolitici) da 1uF

2 x condensatori ceramici da 27pF

4 x condensatori ceramici a goccia da 10nF

3 x condensatori ceramici a goccia da 100nF

1 x condensatore ceramico a goccia da 200nF

1 x condensatore ceramico a goccia da 0,47uF

2 x resistenze 470 Ohm

1 x resistenza 470K

2 x resistenze 100 Ohm

3 x resistenze 1k

5 x resistenze 10k

3 x resistenze 47K

1 x resistenza 220 Ohm

3 x resistenze 4K7

1 x resistenza 3k3

1 x resistenza 2k2

1 x resistenza 47 Ohm

1 x Trimmer da 4K7 (il più piccolo possibile), Se volete potete sostituire il trimmer con un paio di resistenze. Questo trimmer serve a regolare il contrasto del display. Io ho utilizzato due resistenze da 100 Ohm in parallelo ed una da 1k, ottenendo un buon contrasto.

1 x diodo zener da 4.3V

2 x piccoli pulsanti

1 x condensatore (elettrolitico) da 470uF

1 x condensatore (elettrolitico) da 4,7uF

- 1 x diodo 1N4001
- 1 x Integrato in tecnologia TTL 74HC02
- 1 x Integrato in tecnologia TTL 74F74
- 1 x Integrato in tecnologia TTL 74HC390
- 1 x LM393 regolatore di tensione
- 1 x MAX903 operazionale a comparatore elettrico, in formato 8pin, plastico, tipo DIP. Se il Vs. rivenditore di fiducia non lo avesse potete ordinarlo on-line presso www.maxim-ic.com.
- 1 x regolatore di tensione fisso 7805
- 1 x Display LCD compatibile con il modello HD44780, a 20 caratteri e due righe. Esso può essere sia con che senza retroilluminazione ((16 pin o 14 pin)

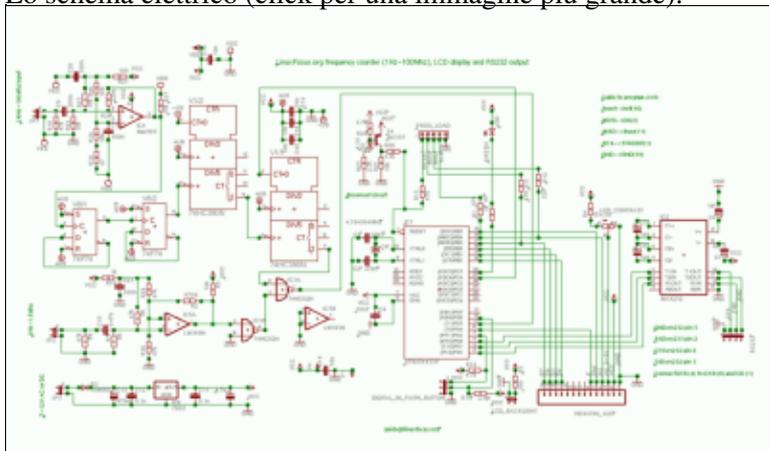
Tutti i display CD che ho provato che hanno o 14 o 16 pin al connettore risultano essere compatibili con il modello HD44780. Potete anche utilizzare un display a 3 o 4 righe, ma in tal caso dovrete modificare in parte il software.

Oltre al succitato avrete bisogno di alcuni cavi, connettori (BCD, alimentazione, seriale) e di un trasformatore a 9Volt o una qualsiasi altra forma di alimentazione da 150mA. Spesso potete trovare alimentatori molto economici, che si usano per apparati elettronici di largo consumo, che si innestano direttamente alla presa.

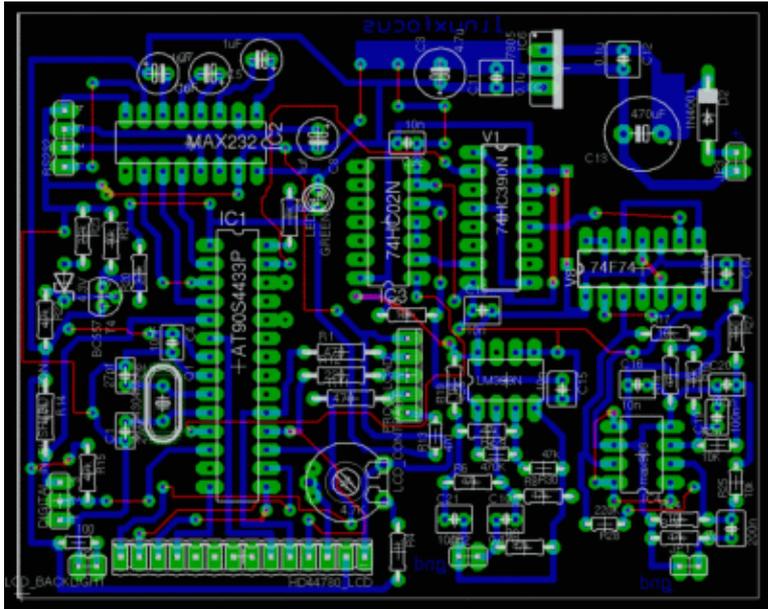
Lo schema elettrico e la basetta

Ho utilizzato eagle per Linux per disegnare lo schema elettrico e lo stampato. Tuttavia il programma presenta alcuni problemi minori, in quanto non riesce a comprendere che tutte le alimentazioni degli integrati sono a 5Volt. Otterrete degli errori se cercherete di eseguire il test di validazione elettrico. Il design risulta comunque corretto.

Lo schema elettrico (click per una immagine più grande):



Lo stampato (click su di esso per una immagine più grande):



Lo stampato è specificatamente studiato per uso hobbistico. Solo il disegno in blu verrà utilizzato per eseguire l'incisione dello stampato. Le linee rosso sono costituite da semplice filo. È di gran lunga più facile, e richiede anche meno precisione, fare una basetta monofaccia che una multistrato. Vi consiglio di studiare il percorso più breve possibile per i fili. Purtroppo con Eagle questo non mi è stato possibile.

Il disegno dello stampato su fondo bianco per facilitarne le operazioni di stampa: [Il circuito su fondo bianco](#)

(Nota: non dovrete utilizzare questo file per generare il circuito stampato.)

I file in formato eagle sono inclusi nel pacchetto software (linuxfreqcount-0.4) che potreste scaricare alla fine di questo articolo.

Il funzionamento

Il microcontrollore AT90S4433 al suo interno possiede due contatori. Uno ha una ampiezza di 16 bit e l'altro di 8 bit. Utilizzeremo il contatore ad 8 per la generazione del segnale di time base basandoci sul clock generato dal quarzo. A tal fine ricorreremo ad una quarzo da 4194304Hz e lo passeremo al contatore ad 8 bit per mezzo del divisore interno con un rapporto di 1 a 256 (potete verificarne le specifiche nel datasheet presente alla fine di questo articolo). Il contatore in questione è gestito in modo da ottenere in interrupt all'overflow. In parole povere avremmo un time base di 64Hz: $4194304\text{Hz} / (256 * 256)$. Per mezzo di un loop riusciremmo ad ottenere frequenze di campionamento da 1Hz o da 64Hz.

Ora abbiamo una funzione che può essere richiamata ad intervalli o di 1Hz o di 64Hz in funzione di come il nostro contatore software stia lavorando. Tutto quello che ora ci resta da fare è semplicemente leggere il responso del contatore a 16bit per mezzo di questa funzione e farne vedere il risultato. Il contatore a 16 bit otterrà il segnale digitale da misurare al pin PD5 del microcontrollore.

Il microcontrollore effettua un campionamento del segnale in ingresso sulla base del proprio clock interno. Se ci basiamo sui teoremi di campionamento, avremo che non potremmo misurare frequenze superiori alla metà della frequenza di risonanza del quarzo. In pratica saremo però in grado di misurare frequenze sino a 1,5MHz.

Per poter misurare frequenza superiori a questo limite, avremmo bisogno di un divisore. Questa funzione viene svolta dalla coppia di circuiti costituiti dal 74F74 e 74HC390. Il 74F74 è utilizzato come divisore asincrono a base 4 (1/4) ed il 74HC390 viene utilizzato come divisore a base 25 (1/25). Il motivo dell'utilizzo

di due divisori è dato dal fatto che il 74HC390 non è in grado di gestire frequenze superiori ai 25MHz.

Il nostro circuito avrà quindi due ingressi: uno per mezzo del divisore a 100 ed uno diretto. A seconda della scala di frequenza che andremo a misurare, sceglieremo l'ingresso appropriato, ma ricordate, _MAI_ utilizzare entrambi gli ingressi assieme.

Se lo utilizzerete solo per frequenze fino ad 1.5MHz (per esempio per verificare la risposta di un amplificatore audio), potrete omettere la sezione costituita dagli integrati MAX903, 74F74 e 74HC930. Ricorrete comunque allo stesso software ed alla stessa basetta.

La funzione svolta dai due operazionali in tensione (il MAX903, per la sezione ad alta velocità, e l'LM939) è quella di amplificare il segnale ed ottenere alla loro uscita una onda quadra, al posto di una sinusoidale.

Va usata particolare cura nella progettazione del layout della basetta per evitare oscillazioni indesiderate degli operazionali. È altresì bene che notiate il MAX903 è realmente un comparatore ad altissima velocità. Se per esempio, avete un segnale a 100KHz modulato con un rumore ad 1MHz, allora otterrete un conteggio addizionato del segnale ad 1MHz, nel momento in cui il segnale ad 1MHz passa per il livello 0.

La resistenza da 220K in prossimità del MAX903 evita che eventuali rumori possano influenzare il segnale generando un feedback positivo. È tuttavia raccomandabile che non utilizzate segnali che contengano del rumore sul nostro frequenzimetro.

Il microcontrollore può esser gestito per mezzo della porta seriale RS232 e con due pulsanti (clear e cambia tipo di conteggio).

I due segnali di ingresso sono combinati per mezzo di una porta logica OR (74HC02). La porta OR, in caso di assenza di segnale nell'ingresso 1/100, escluderà il livello logico 1 generato dall'integrato 74HC390. Tuttavia l'integrato 74HC930 viene 'ripulito' (per mezzo del pin 2) allorquando si preme il pulsante di clear o si cambi la modalità di conteggio.

Il software

Il software per il microcontrollore gestisce due contatori in modalità interrupt. Come ciò sia fatto è descritto a sufficienza nel datasheet dell'AT90S4433. Quello che dovrete fare è configurare una serie di registri. Questa operazione è importante ma non complessa. Non ne spiegherò le ragioni qui nuovamente. La maggior parte della logica del frequenzimetro è implementata nel file [linuxfreqcount.c](#). I restanti file sono "librerie" per il pannello LCD, le porte seriali,... Quando state scrivendo software per il microcontrollore dovrete prestare attenzione a non utilizzare più di 128Byte di RAM. Tuttavia è raccomandabile evitare di eseguire troppe funzioni nidificate con troppi parametri. Indici e variabili globali sono largamente preferibili.

Allorquando i registri sono correttamente configurati la funzione SIGNAL(SIG_OVERFLOW0) verrà richiamata con intervalli di 64Hz. A quel punto andremo a leggere il contatore a 16bit e ne gestiremo un flag (hflag) per processare in un secondo momento il risultato, per mezzo della funzione handlecounterresult(). Se utilizzassimo la frequenza di 1Hz potremmo rendere direttamente visibile il risultato del contatore, ma ricorrendo ai 64Hz (1/64 di secondo), dovremmo obbligatoriamente moltiplicare il risultato per 64. La matematica che si può gestire con questo microcontrollore è assai limitata (non abbiamo dei fantastici numeri interi a 32bit come su una cpu Pentium). Fortunatamente moltiplicare per 64 equivale ad effettuare uno shift (operazione binaria di spostamento di un set di bit per incrementarne il valore) di 6bit. Andremo poi a memorizzare il risultato dell'operazione in una variabile di 3x 8bit (counterval[3]).

Ora abbiamo a disposizione il corretto valore rappresentato a 24bit. Per poterlo rendere visibile lo dovremo convertire in ASCII decimale. La funzione printf potrebbe facilmente svolgere questa operazione, ma tale funzione non è presente nel microcontrollore, e includerla consumerebbe ben più delle risorse di memoria di

cui questo microcontrollore dispone. Per eseguire la conversione dovremmo dividere il numero a 24bit per 10 con relativi riporti. I riporti saranno numeri decimali. La nostra CPU però non può eseguire calcoli a 24bit, ma possiamo trattare il dato in frazioni di 8bit (function divby10()). Aggiungendo ad ogni risultato il valore ASCII del numero zero ci fornirà la rappresentazione finale del nostro numero (function longtoascii()).

Questa stringa ASCII può esser inviata sulla linea seriale RS232 ed anche al display LCD.

Creazione del circuito stampato

Il pacchetto software contiene un file postscript (linuxfcount.ps) per poter facilmente stampare il circuito. Secondo il mio punto di vista trovo che i punti di saldatura siano troppo piccoli. Vi consiglio quindi di renderli più ampi per mezzo delle vernice prima che incidiate la basetta stessa. Il procedimento per creare la propria basetta in ambiente amatoriale è descritto nell'articolo: [Maggio 2002. Un pannello di controllo LCD per il Vostro server Linux](#). Alcuni lettori mi hanno comunicato la loro difficoltà nel reperire lo spray Kontakt Pausklar-21. In tal caso potete ricorrere all'uso di petrolio per illuminazione. Avrete però bisogno di pulire la basetta con del sapone prima di inciderla con l'acido.

Quale contenitore costruire per il nostro frequenzimetro?



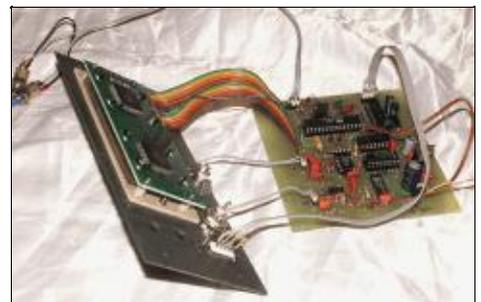
Uno dei più grossi problemi per l'elettronica amatoriale è spesso il contenitore per i circuiti che ci costruiamo. Le scatole componibili in alluminio hanno, di norma, un costo elevato. Io ho trovato un metodo economico che può garantire un aspetto professionale: ricorro all'uso del legno per la struttura



angolare e per i pannelli, eventualmente dipingendolo con dei colori scuri per darne un aspetto più esoso. Se lo tingete ricordatevi di utilizzare tintura solubile per interni. Per il pannello frontale, quello laterale e quello superiore, ricorro ad un foglio metallico tagliato a misura. Potete notare nella foto a destra la struttura principale in legno. Nella foto a sinistra potete notare il risultato complessivo (solo il coperchio superiore è stato rimosso).

Assemblaggio del circuito

Quando andrete a saldare tutti i componenti sulla basetta prestate particolare attenzione alla sezione ad alta frequenza (il MAX903 ed il 74F74, con i relativi condensatori e resistenze). Normalmente io raccomando di ricorrere all'utilizzo di zoccoli per gli integrati, ma in questo particolare caso è meglio non utilizzarli per evitare di creare effetti parastatici e capacitativi non desiderati. Per i condensatori di disaccoppiamento tra l'alimentazione e la massa, dovrete ricorrere a dei piccoli condensatori ceramici. Il valore di questi condensatori dovrebbe essere compreso tra 10nF e 100nF.



Questo è il più complesso circuito che abbiamo fino ad oggi studiato e realizzato per LinuxFocus. Vi raccomando di procedere a piccoli passi e controllare che tutti i vari componenti siano effettivamente funzionanti:

1. Iniziate saldando i componenti necessari all'alimentazione (7805, ...) sulla basetta e verificate il funzionamento.
2. Saldate tutti i rimanenti componenti, ma inserite nello zoccolo il solo microcontrollore.
3. Il pacchetto software linuxfreqcount-0.4 include un programma di controllo (avr_led_lcd_test.c) per la verifica della funzionalità e del microcontrollore, dei LED e del display LCD. Eseguite, collegate tutti i cavi, come descritto nel primo articolo della serie inerente l'AVR ([Marzo 2002, Programmare il microcontrollore AVR con GCC](#)), e poi digitate "make testload" per eseguire il software avr_led_lcd_test presente nella distribuzione. I LED dovrebbero lampeggiare ed il display dovrebbe presentare la scritta "Hello".
4. Inserite ora tutti gli integrati nei relativi zoccoli. Eseguite e caricate nel microcontrollore il software (make load). Verificate il funzionamento, prima per mezzo della sezione a bassa frequenza, e successivamente quella ad alta. Avrete bisogno di un oscillatore come origine della frequenza (i segnali ad onda quadra sono preferibili). Nota del traduttore: Come origine, per verificare il funzionamento della sezione a bassa frequenza, potete ricorrere all'uscita di un trasformatore a bassa tensione. La frequenza dell'alimentazione di casa si aggira sui 50Hz o 60Hz. Se all'uscita di questo trasformatore applicherete un ponte a diodi avrete una frequenza doppia (rispettivamente 100Hz o 120Hz).

Come utilizzare il nostro contatore

Il contatore può funzionare in svariati modi:

1. contatore continuo incrementale
2. frequenzimetro con frequenze di campionamento ad 1Hz. In questa modalità può misurare frequenze dell'ordine di 16bit di ampiezza. Potremmo quindi misurare fino a 65535Hz o 6.5535MHz, per mezzo del divisore 1/100.
3. frequenzimetro con campionamento a 64Hz. In questa modalità può misurare frequenza fino ad 1,5Mhz o 100MHz, per mezzo del divisore 1/100.

```

cat /dev/ttyS8
c:50112
c:50176
c:50112
ak
c:50161
c:50180
c:50110
c:50857
c:50110
c:50896
ret
c:50102

```

comandi ASCII sull'interfaccia RS232.

Quanto preciso?

Qualsiasi contatore digitale ha una precisione di \pm una unità di visualizzazione. L'ultima cifra tende sempre ad oscillare anche se il segnale da analizzare avrà la massima stabilità possibile. Se il segnale viene campionato a 64Hz, avremmo una precisione di \pm 64. Lo stesso vale per il divisore 1/100. In ogni caso questo contatore è assai preciso. L'unico punto debole è costituito dal cristallo di quarzo da 4194304Hz. I cristalli di quarzo oscillano più lentamente quando sono caldi. Questo concerne la fisica e può essere eliminato solo controllando la temperatura in maniera esatta!

È possibile calibrare il contatore se trovate un buon segnale di riferimento. Lo si può calibrare per mezzo del software inserendo un piccolo ritardo nella funzione handlecounterresult(). Ho confrontato il mio contatore con un segnale di riferimento molto preciso, ed ho potuto notare che il mio frequenzimetro è pressoché esatto. Vi comunico anche che i ritardi di compensazione sono attualmente commentati nel codice sorgente. Potete

influenzare la frequenza del cristallo di quarzo variando la capacità di uno dei condensatori da 27pF (dovrete però fare delle prove sperimentali per i valori o ricorrere ad un condensatore variabile).

Per un normale utilizzo amatoriale non vi è la necessità di calibrare il frequenzimetro. Vi ricordo che i normali cristalli da 4194304Hz sono molto precisi.

Il contatore in azione

Qui potete vedere il frequenzimetro/contatore in azione



Per chi vuole saperne di più

In questo articolo non viene fornito alcun software per gestire i responsi direttamente da Linux. Si potrebbe utilizzare questo frequenzimetro, per esempio, per misurare la velocità del vento in luoghi lontani o quello che la fantasia ci suggerisce, purché possa essere convertito in una frequenza digitale. Scrivere software per Linux per gestire risultati in caso di misurazione che si perpetrano nel tempo non è complesso. Potete per esempio apportare delle modifiche al programma in perl presente nell'articolo [Maggio 2002. Un pannello di controllo LCD per il Vostro server Linux](#). Se siete interessati a registrare i risultati su di un file, potete utilizzare semplicemente i seguenti comandi:

```
ttydevinit /dev/ttyS0  
cat /dev/ttyS0 > your_logfile.txt
```

In questo caso si presuppone che il frequenzimetro sia collegato alla porta seriale COM1 (=ttyS0). Semplice no?

Bibliografia

- Il software di gestione dell'AVR: www.amelek.gda.pl/avr/
copia locale: [uisp-20011025.tar.gz](#)
- Come compilare ed installare il programmatore per gli AVR:
[Marzo 2002. Programmare il microcontrollore AVR con GCC](#)
- Il codice sorgente di questo articolo [linuxfreqcount-0.4.tar.gz](#) . Lo schema elettrico, i file in formato Eagle ed alcuni screenshot sono incluse nel pacchetto.
- Tutto il software (anche gli aggiornamenti eventuali sono qui pubblicati) ed i relativi documenti: [software/datasheets](#)
- Il datasheet del 74hc390 [74hc390.pdf 48K](#)
- Il datasheet del 74f74 [sn54f74.pdf 84K](#)
- Il datasheet dell'LM393 [LM193.pdf 348K](#)

- Il datasheet del MAX903 [MAX900–MAX903.pdf](#) 164K
- Il datasheet del MAX232 [MAX220–MAX249.pdf](#) 448K
- Il datasheet dell'AT232, una variante più economica, spesso venduta al posto del vero MAX232 [st232.pdf](#) 100K
- Il datasheet dell'Atmel AT90S4433 [avr4433.pdf](#) 2356K
- Il sito web dell'atmel: www.atmel.com/
- Eagle 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>

2005-01-10, generated by lfparsr_pdf version 2.51