

**Guitarix** è un versatile amplificatore per chitarra rock per JACK Linux ( [Jack Audio Connection Kit](#) ) con un ingresso e uscite variabili ( *uscita Amp e uscita FX stereo* ).

Progettato per ottenere piacevoli suoni di chitarra thrash/metal/rock/blues, Guitarix fornisce un amplificatore per chitarra virtuale composto da un **preamplificatore** ( *dove controlliamo il guadagno, il drive e il volume di uscita* ) e un **rack** dove possiamo caricare molti diversi effetti mono o stereo ( *dinamiche, modulazioni, distorsioni, EQ ...* ).



Con questo software, abbiamo **il controllo totale su più parametri** : bassi, medi, alti, guadagno (ingresso/uscita), compressore, preamplificatore, valvole, drive, overdrive, distorsione, freeverb, risposta all'impulso, [\[Convoluzione\]](#) , vibrato, chorus, ritardo, piagnucolone ( *wah* ), tonestack ed eco.

**Guitarix** beneficia di **scorciatoie da tastiera** per la maggior parte delle sue operazioni e viene fornito con una **porta Jack midi-in per collegare un controller midi** e fornire una funzione di apprendimento midi per ogni controller Gui. Puoi salvare le tue impostazioni ( *e le connessioni midi* ) nei preset e caricarli o passare da uno all'altro facilmente con la tastiera o i comandi Midi e anche tramite il menu. Guitarix è dotato di un **widget di connessione con porta jack** per mantenere facilmente le connessioni in un flusso di lavoro modulare e guitarix include un **accordatore per chitarra** con un'interfaccia analogica.

## **PERCORSO DEL SEGNALE**

Il percorso del segnale all'interno di guitarix è attualmente il seguente:

Chitarra elettrica

↓

Strumento inserito sulla scheda audio (se la tua scheda audio non dispone di ingressi strumento, usa semplicemente il microfono o anche gli ingressi di linea ed esegui la preamplificazione all'interno di guitarix stesso)

↓

Porta di acquisizione del sistema (JACK)

↓

gx\_head\_amp:in\_0 (JACK)

↓

Rack mono e testata gx-2 (guitarix)

↓

gx\_head\_amp:out\_0 (JACK)

↓

gx\_head\_fx:in\_0 (JACK)

↓

Rack stereo (chitarrax)

↓

↓

gx\_head\_fx:out\_0

(JACK)

gx\_head\_fx:out\_1 (JACK)

↓

↓

Ulteriore elaborazione o direttamente alle porte di riproduzione del sistema (JACK)

Per impostazione predefinita, i rack di effetti e gli effetti (o plug-in) al loro interno sono ordinati verticalmente e il flusso del segnale avviene dall'alto verso il basso. Tienilo a mente quando inserisci i tuoi effetti o plug-in, soprattutto quando usi plug-in come Tonestack e Cabinet. Se desideri davvero emulare un impianto analogico, non vorrai nessun effetto stereo dopo quello. Gli effetti mono non possono essere utilizzati nel rack stereo e viceversa. Gli effetti stereo sono contrassegnati da un  $\sigma$  simbolo.

## **INTERFACCIA UTENTE**

### introduzione

Inizia a pensare a Guitarix solo come a un piccolo amplificatore:



I controlli principali sono

Pregain	gain used at amp input
Drive, Clean/Dist	extra distortion
Master gain	gain used at amp output

Per ottenere un segnale pulito, impostare Drive e Clean/Dist su zero, Pregain un valore basso e Master guadagnare un valore alto (trascinando verso l'alto o la rotellina del mouse si cambia il valore).

Per ottenere un suono distorto, alza Pregain (e magari abbassa il guadagno Master in modo che il livello del suono sulla scala del misuratore a destra non tocchi la linea 0). Inoltre puoi usare Drive e Clean/Dist per ulteriori e altri tipi di distorsione.

Puoi anche attivare il potenziamento dei bassi, la presenza e il riverbero e modificare i livelli con le rotelle. Oppure fai clic sulla riga con "12ax7" per ottenere la configurazione dell'amplificatore valvolare successivo (o fai clic con il pulsante destro del mouse per vedere l'intero elenco).

Per inciso, puoi anche fare clic sulle icone di stato in basso a destra (suggerimento: equivalente allo spazio sulla tastiera, ctrl-c e ctrl-l :-)

Ora possiamo provare il livello successivo: clicca su "Preset:", seleziona un banco di fabbrica (quelli con "F") e un preset:



Le preimpostazioni attualmente selezionate vengono sempre visualizzate nella riga di stato in basso. Molte azioni del programma dispongono di tasti di scelta rapida. Possono essere trovati nel menu. I primi 9 preset dall'attuale elenco dei preset possono essere selezionati con il tasto numerico corrispondente.

Se non vuoi solo ascoltare il suono ma anche cosa sta succedendo, clicca su "Show Rack" (e magari di nuovo su "Preset", per fare più spazio in basso):



Le unità rack possono essere spente/accese con l'interruttore a sinistra, piatte o espanse con il pulsante freccia a destra e spostate con la maniglia sinistra o destra (trascinare con il mouse). Clicca su "Effetti" se vuoi vedere più unità o spostare alcune unità dentro o fuori dal rack. Utilizza la "Modalità configurazione" se hai bisogno di spostare molte unità (in modalità configurazione le unità sono piatte e non puoi modificare alcun controller ma hai una visualizzazione compatta e puoi trascinare un'unità facendo clic in qualsiasi punto).

In realtà ci sono 2 Rack: un rack mono nella parte superiore e un rack stereo nella parte inferiore (o a destra se si utilizza la modalità orizzontale). Le unità stereo sono contrassegnate con il simbolo

stereo a sinistra del nome e possono essere inserite solo nel rack stereo, mentre le unità mono possono vivere solo nel rack mono.

Il flusso del segnale audio va dall'alto verso il basso e da sinistra a destra. L'ordine fa la differenza se c'è un'unità distorcente nel mezzo. Un piagnucolone suona diversamente se lo metti sopra l'amplificatore (preamplificatore) o sotto (postamplificatore).

## Creazione di preimpostazioni

Iniziamo con un preset di fabbrica. Dopo averne selezionato uno, crea un nuovo preset Scratchpad selezionando il banco di Scratchpad, quindi facendo clic sulla voce "<nuovo>" nell'elenco a destra e inserisci un nome. Le impostazioni correnti (dalla preimpostazione di fabbrica) verranno memorizzate in quella nuova preimpostazione, quindi selezionata come quella corrente.

Le preimpostazioni di Scratchpad memorizzano automaticamente tutti i valori modificati. Puoi modificare alcune impostazioni, passare a un altro preset, persino uscire da Guitarix e quando riseleziona il preset vedrai la tua ultima versione.

Al contrario, altri banchi di preset mantengono semplicemente le impostazioni memorizzate, e se cambi qualcosa e ricarichi il preset o passi a un altro preset, tutte le modifiche verranno perse (uscire da Guitarix non conta, perché salva lo stato e inizia allo stesso punto in cui l'hai lasciato). Se vuoi davvero modificare tale preimpostazione, usa il pulsante Salva. E se vuoi essere sicuro di non modificarlo accidentalmente, puoi bloccare la banca facendo clic sulla colonna a sinistra del nome della banca. La banca sarà quindi di sola lettura e una "R" rossa mostrerà questo stato.

Quindi, se hai sviluppato qualche nuova preziosa impostazione, potrebbe essere una buona idea creare una nuova banca (con "Nuova banca"), se non l'hai già fatto, e salvare l'impostazione creando una nuova impostazione in quella banca. Quando crei una banca con una o più impostazioni carine, puoi condividerla sul forum Guitarix: basta trascinare la banca dall'elenco sul desktop o nel modulo di caricamento del browser. Per importare un banco trascinalo da qualche parte nell'elenco dei banchi Guitarix.

Se sei davvero produttivo, potresti voler ripulire o riorganizzare alcune banche. Cliccando su "Organizza" si accede alla modalità organizzazione. Quindi non puoi selezionare i preset, ma rinominare banchi e preset, spostarli con il mouse ed eliminarli (ma non puoi riordinare i banchi di fabbrica).

In modalità organizzazione c'è un terzo elenco sulla destra: fai clic sulla parte superiore dell'elenco per selezionare un banco e copia i preset dall'elenco dei preset trascinandoli. Se tieni premuto il tasto Maiusc, i banchi verranno spostati anziché copiati (se la sorgente non è un banco di fabbrica o di sola lettura).

Se vuoi fare un backup dei tuoi banchi, li troverai sotto `.config/guitarix/banks` nella tua home directory.

## Riproduzione dal vivo

C'è una modalità speciale per suonare dal vivo con Guitarix: cliccando sul pulsante "Live Play" si apre una finestra a schermo intero con un grande display bank/preset, un accordatore, campi di stato

(mute/bypass) e stato del controller midi (c'è una modalità sperimentale supporto per oscurare e sfumare un brutto sfondo verde...).

Funzioni: sulla tastiera

tuner on / off	Enter
mute toggle	ctrl-M
bypass toggle	ctrl-B
bank select	keys A-Z (timeout when no preset selected)
preset select	keys 1-9 (numpad keys also work)
preset pick mode	Space

Alcune funzioni possono essere collegate ai controller midi:

tuner on / off	middle click on main switch of racktuner unit
mute toggle	middle click on engine status icon (status bar)
preset select	midi program change messages
preset pick mode	menu options "Liveplay Midi Switch"

## Modalità di selezione preimpostata

L'idea di questa modalità: hai bisogno di un solo interruttore (a pedale), che può in qualche modo avvisare Guitarix di passare a questa modalità (al momento sono supportati solo la barra spaziatrice della tastiera e i dispositivi che inviano messaggi di controllo midi, invia le tue idee per altri dispositivi) . Quindi puoi selezionare preset o funzioni selezionando le note della chitarra.

Note	String	Function
lower than G	1st	toggle Tuner
G#	1st	toggle mute
G	1st	toggle bypass
A to Bb	2nd, 3rd, 4th	bank select (max. 14 banks)
B to E	5th, 6th until octave	preset select (max. 18 presets)

Ovviamente non importa su quale corda prendi la nota purché sia la stessa nota.

Dopo essere passato alla modalità di prelievo preimpostata, Guitarix tenta di rilevare un silenzio di 40 ms (ma la soglia è più alta rispetto alla modalità di accordatura normale). Il display preimpostato viene visualizzato in rosso scuro "disabilitato" (questo stato è una protezione per non attivare accidentalmente alcune funzioni). Quindi il display passa al rosso.

Le note devono essere stabili per 40 ms. Quando viene selezionata una preimpostazione valida, il display diventa verde. Quindi Guitarix attende il "silenzio", quindi carica il preset e torna alla modalità normale. Pertanto, le "scelte bancarie" mostreranno i nomi delle banche ma non attiveranno nulla. Se suoni il legato con "plettri preimpostati" puoi scorrere i nomi dei preset, ma se ti fermi Guitarix caricherà il preset visualizzato e tornerà indietro (o se non riesci a suonare il legato ;-).



## FUNZIONALITA' COMPLETE

Questo è l'elenco completo delle funzionalità di **Guitarix** .

### **Preamplificatore**



- **Tube Switch** - [Maggiori informazioni sui tubi](#)
- **Controlli del preamplificatore** : [impara a ottenere un tono che spacca](#)
- Controlli Bass Boost, presenza e riverbero.
- **Uscita principale** ( impostata su 0 dB in questa immagine ).

## [Cremagliera]

### **RACK mono**

- Emulatori
  - [\[Tonestack\]](#) : emulatore di toni di amplificatori con equalizzazione bassi/medi/alti e un ampio elenco di "amplificatori virtuali"
  - [\[Cabinet\]](#) : emulatore EQ cabinet con manopole Bass/Treble
- [\[Effetti dinamici\]](#)
  - Compressore
  - Noise Gate ( incluso con l'amplificatore )
- [\[Unità EQ\]](#)
  - Basso Alto Filtro
  - EQ scalabile con 10 bande
  - Filtro BiQuad
- [\[Modulazione Effetti\]](#)
  - Coro
  - Flangiatore
  - Faser
  - Piangi tesoro ( Wah )
- [\[Ripeti Effetti\]](#)
  - Eco
  - Ritardo
  - Verbo libero

- Tremolo
- Effetti di distorsione
  - Overdrive
  - Distorsione multibanda
- Risposta impulsiva
- Oscilloscopio
- Feedback
- Uscita MIDI

### RACK stereo

- Postamp
- [Equalizzatore a 3 bande](#) : equalizzatore shelving con manopole bassi/medi/alti
- [\[Modulazione Effetti\]](#)
  - Coro
  - Flangiatore
  - Faser
- [\[Ripeti Effetti\]](#)
  - Ritardo
  - Eco
  - Riverbero stereo
  - Zita Riverbero
- Filtro Mogg: per aggiungere risonanza alla frequenza di taglio.
- [Filtro convolvere](#)



### E altro ancora!

- [Controllo MIDI](#)



## INSTALLAZIONE

# Distribuzioni che includono Guitarix

A meno che tu non voglia vivere al limite, utilizzare la versione di guitarix fornita dalla tua distribuzione GNU/Linux è il modo più semplice e veloce per iniziare con guitarix.

Sebbene poche distribuzioni includessero guitarix nell'installazione predefinita, la maggior parte ne ha una copia nei propri repository. Sono note le seguenti distribuzioni:

- [Debian](#)
- [Ubuntu](#)
- [Fedora](#)
- [Susé](#)
- [Frugalware](#)
- [Gentoo](#)
- [ArcoLinux](#)
- [Mandriva](#)
- [Foresightlinux](#)

*Nota: se la tua distribuzione non è elencata sopra ma sai che guitarix è incluso nei suoi repository, aggiungila all'elenco sopra.*

## Costruire Guitarix dal codice sorgente

**Guitarix** utilizza "waf" per l'ambiente di creazione. Il modo più semplice e veloce per costruire Guitarix è riportato di seguito:

Scompatta l'archivio che puoi scaricare sulla nostra pagina Sourceforge:

```
tar -xvjf gx_head-0.14.0.tar.bz2
```

oppure recupera la versione di sviluppo corrente da git:

```
git clone git://git.code.sf.net/p/guitarix/git guitarix
```

Tieni presente che **la versione git non è una versione ufficiale**. È all'avanguardia, non testato e non dovrebbe creare un'applicazione per l'uso quotidiano. Quindi, se costruisci Guitarix da git e funziona per te, ci congratuliamo, in caso contrario, recupera (usa "git pull"), ricostruisci, controlla di nuovo e invia una segnalazione di bug.

Indipendentemente dal fatto che tu abbia recuperato i sorgenti da git o li abbia decompressi dal nostro tarball di rilascio ufficiale, i passaggi successivi sarebbero:

```
cd chitarraix_source_directory
./waf configurare
./wafbuild
sudo ./waf install
```

Per impostazione predefinita, questo installerà Guitarix e i file correlati nel filesystem /usr/local, ad esempio /usr/local/bin. Il prefisso di installazione può essere fornito in fase di configurazione, ad esempio:

```
./waf configure --prefix=/usr
./waf build
sudo ./waf install
```

Pertanto, Guitarix verrà installato in /usr/bin.

## Dipendenze

Per ottenere l'ultima versione di git, è necessario installare il pacchetto ' **git** '. Come il sistema di compilazione, questo è disponibile nella maggior parte dei repository.

Per compilare ed eseguire guitarix correttamente, avrai bisogno di un sistema di compilazione ( *build-essentials*, *gcc ecc.* ) e dei seguenti pacchetti extra ( *binari / librerie runtime e pacchetti di sviluppo, ovvero pacchetto-dev o pacchetto-devel* ):

```
python
libjack-dev
libsndfile1-dev
libgtk2.0-dev
ladspa-sdk
libgtkmm-2.4-dev
libboost-dev
intltool
```

(Questi sono i nomi dei pacchetti Debian. In altre distribuzioni i pacchetti hanno spesso nomi simili.)

Nelle distribuzioni basate su Debian che contengono una versione precedente di Guitarix puoi spesso iniziare con l'installazione delle dipendenze di build di quella versione:

```
sudo apt-get install build-essential sudo apt-get build-dep guitarix
```

"./waf configure" ti dirà quali pacchetti sono ancora mancanti o quali versioni più recenti sono necessarie. Cerchiamo di mantenerci compatibili con l'ultima versione stabile di Debian.

### UTILIZZO DEI WIDGET DI CONTROLLO

Utilizzare la rotellina o trascinare il mouse (per le manopole utilizzare su/giù) per apportare modifiche. Premi il tasto ctrl o maiusc durante il trascinamento per apportare modifiche precise. Facendo doppio clic si imposta il controllo sulla posizione del mouse.

Fare clic sulla visualizzazione di un valore (o fare clic sul controllo con il pulsante destro del mouse) per aprire una piccola finestra popup per l'immissione del valore. Utilizzare il tasto Invio o ESC per chiudere la finestra.



Facendo clic su un selettore di elenco si imposta il valore successivo, facendo clic con il pulsante destro viene visualizzato un elenco popup. La rotella può essere utilizzata anche su interruttori e selettori di elenco.

La maggior parte dei controlli può essere [collegata all'apparecchiatura MIDI](#) facendo clic con il pulsante centrale del mouse.

Quando un controllo ha il focus della tastiera (ad esempio dopo aver fatto clic su di esso) puoi utilizzare i tasti cursore, PgUp, PgDn, Home, End per modificare il valore (ma nell'attuale interfaccia utente di Guitarix il widget del focus non è visivamente distinto).

## COME OTTENERE UN TONO SPAVENTOSO

"Scrivere di musica è come ballare di architettura"  
- Frank Zappa

- [introduzione](#)
  - [Avviare](#)
  - [Il RACK](#)
  - [Creare il nostro tono](#)
    - [Preamplificatore](#)
    - [Porta del rumore](#)
    - [Compressore](#)
    - [Il Tonestack: Amp Modeler](#)
    - [Cabinet: emulazione del cabinet degli altoparlanti](#)

## introduzione

### Avviare

Dopo il primo avvio vedrai una piccola testina con i seguenti parametri:

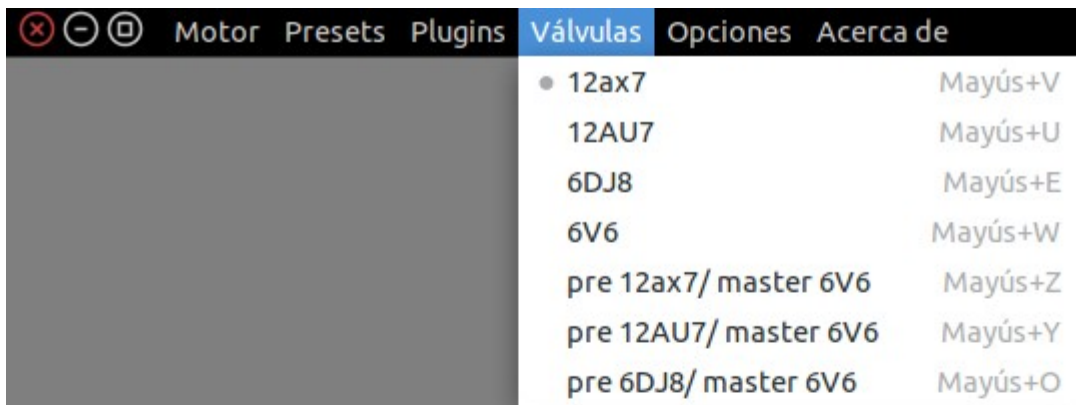


- **Pregain** : livello di ingresso della chitarra.
- **Drive** : guadagno del segnale.
- **Clean/Dist** : quantità di distorsione.
- **Master Gain** : quantità di segnale dal preamplificatore alla sezione di potenza.
- **Aumento dei bassi e dei medi** ( *presenza* ) e **riverbero** ( *con un cursore per controllare il rapporto dry/wet del riverbero* )

- **Convolver** : Una fantastica funzione che porta su Guitarix una serie di **curve caratteristiche di famosi amplificatori/microfoni e riverberi** che si possono trovare su internet ( in formato wav ). Per ulteriori informazioni, vedere [\[Convoluzione\]](#)

*Le funzioni “dry/wet” presenti nei riverberi, nelle distorsioni... trattano contemporaneamente il segnale “pulito” e quello “elaborato” regolando il mix tra il segnale originale e quello elaborato, controllando il mix tra loro. “Clean/Dist” è lo stesso per il segnale pulito e distorto*

Ma abbiamo bisogno di più cose per processare la nostra chitarra/basso, no? Almeno il nostro amplificatore “avrà” alcune valvole (o valvole) che daranno al nostro strumento quel tono “analogico”...



- **12AX7** : è la valvola principale utilizzata nei preamplificatori per chitarra, famosa per essere utilizzata da **Marshall** ( insieme all'inimitabile *EL34* nella sezione di potenza ) e **VOX** , producendo ( sempre con l'*EL34* ) il noto "**suono britannico**" .
- **12AU7** : una valvola più modesta ( fornisce solo il 19% del segnale di una 12AX7 ), è stata utilizzata in alcuni amplificatori **Gibson** e **Marshall vintage**.
- **6DJ8** : Conosciuto anche in Europa come **ECC88** , ha una storia interessante da raccontare. Fu usato prima nei sintonizzatori VHF e poi in molti amplificatori Hi-Fi dell'epoca... a parte questo non conosco nessun amplificatore delle solite marche che lo utilizzi...

- **6V6** : “*alti cristallini e bassi potenti e definiti*” è ciò che caratterizza il “**suono americano**” degli amplificatori **Fender** dell'epoca. Al giorno d'oggi il loro fratello maggiore 6L6 viene utilizzato sui più potenti stadi di potenza "Hi-Gain".

Possiamo distinguere i suoni in: **suono britannico** ( *medi potenti* ), **vintage** ( *suono meno saturo delle 12AU7 e 6DJ8* ) e **suono americano** ( *maggiore risposta in alto e in basso delle 6V6* ). I restanti suoni della versione 0.18 Guitarix disponibili nei repository sono un mix tra loro:

- Preamplificatore 12AX7 e stadio di potenza 6V6
- Preamplificatore 12AU7 e stadio di potenza 6V6
- Preamplificatore 6DJ8 e stadio di potenza 6V6

## II RACK



Per ottenere un bel suono, come in qualsiasi amplificatore, dobbiamo **“girare alcune manopole”** . Guitarix è progettato come un amplificatore modulare quindi se vogliamo sfruttarlo al meglio dobbiamo aggiungere alcuni moduli e giocherellare con loro.

Nel menu dei plugin facciamo clic su **“mostra rack stereo”** , **“mostra rack mono”** ( così possiamo vedere tutti gli effetti che aggiungiamo ) e **“mostra barra dei plugin”** . Allo stesso modo, selezioniamo il **sintonizzatore** nel menu delle opzioni ( potremmo avviarlo anche dalla barra dei plugin )



Se abbiamo già sbagliato le impostazioni, nello stesso menu delle opzioni c'è la possibilità di **ripristinare tutti i parametri** se non ci piace come funziona.

Sono molti i moduli che si possono aggiungere al rack, ad esempio:

- **Emulazione di amplificatori e cabinet** : plugin “Tonestack” e “Cabinet”.
- **Noise gate** già incluso nel rack: noise gate, noise shaper e limiter inclusi.
- **Distorsione** : Overdrive e una potente distorsione multibanda
- **Equalizzatori** : filtri passa-basso e passa-alto, equalizzatore grafico tri-banda (come negli amplificatori più comuni), parametrico (EQ a 10 bande con controlli di frequenza e fattore Q), anche nei plugin di distorsione ed emulazione
- **Compressore**
- **Effetti** : Wah Crybaby, Feedback, delay, chorus, flanger, riverbero...



Tutti i plugin di effetti possono funzionare **in pre-elaborazione** ( *normalmente utilizzata per correggere un segnale* ) o **in post-elaborazione** ( *normalmente utilizzata solo per migliorarlo* ). **Gli effetti stereo funzionano tutti in post-elaborazione** .

Tralasciando per ora gli effetti, ne parleremo più avanti, i punti principali per ottenere un buon suono sono l' **equalizzazione** ( *tramite emulatore e/o equalizzatori* ), il **noise gate** , la distorsione multibanda ( *per un ottimo suono ad alto guadagno* ) e il **compressore** .

Ecco che arriva...

## Creare il nostro tono

Proviamo lo scenario più difficile: vogliamo che un metal distortore come quelli che usano all'inferno **sia più metallaro della moto dei Painkiller** . Vuol dire che dobbiamo imparare a maneggiare l'amplificatore e il noise gate, cosa basilare negli emulatori digitali. Vedrai, rallegrati!

Se poi vorremo un tono classico più croccante, abbasseremo alcune manopole, mentre se vorremo qualcosa di più pulito sapremo mantenerlo grazie a quanto abbiamo imparato.

Per creare il nostro tono è fondamentale controllare il più possibile i seguenti punti:

1. Preamp
2. Noise gate
3. Compressor
4. Amp emulator
5. Cabinet emulator

Inizierò con impegno, così non perderai interesse. Il 2° e il 3° punto sono i più complessi ma anche fondamentali, **soprattutto la compressione** . Niente panico: esercitati, leggi e rileggi tutte le volte che vuoi finché non capisci perché è qualcosa che ti servirà con qualsiasi strumento e in tutti gli studi, domestici e non.

### Preamplificatore



Poiché cerchiamo innanzitutto un tono potente, dobbiamo prima proteggere le nostre orecchie, quindi abbassiamo il **guadagno principale** ( *solitamente noto come "post-guadagno"* ) al minimo (-20 dB). Ora spingiamo al massimo il preamplificatore in modo che le valvole distorcano: **Pre-gain, drive e Clean/dist al massimo** . Non allarmarti, probabilmente sembra terribile. Ora fai clic sull'amplificazione dei bassi e sugli interruttori di presenza. Suona meglio?

Quando cerchiamo un suono meno saturo ( *ad esempio l'hard rock degli anni '70* ) esploreremo le impostazioni di **Drive e Clean/Dist** . Non appena otteniamo il tono che ci piace, possiamo aumentare il Master Gain.

Cerca di **evitare di tagliare** il suono quando modifichi il preamplificatore perché potrebbe essere fuorviante. In linea di principio modifichiamo il pre-gain solo se il segnale in ingresso ( *direttamente dalla chitarra o tramite pedali tra questa e l'ingresso della scheda audio* ) è troppo alto.

## Porta del rumore



- **Noise gate** : taglia i segnali non desiderati. La rotella controlla la soglia, cioè se un segnale supera questa soglia viene tagliato. Se esageriamo, inizierà a funzionare anche mentre giochiamo. Puoi provarlo portandolo al massimo e suonando una nota sostenuta. Vedrai come il noise gate taglia rapidamente la nota.

Il gate può essere così brutale da poter interpretare un forte vibrato della mano sinistra come rumore, quindi usatelo in modo sensato. A volte è meglio usare un po' meno guadagno (o il controllo successivo nel noise gate, il noise shaper) se vuoi un suono potente ma rotondo (gli amplificatori aggiungono ronzio dai valori di guadagno di 7 o 8 in su uccidendo la tua dinamica).

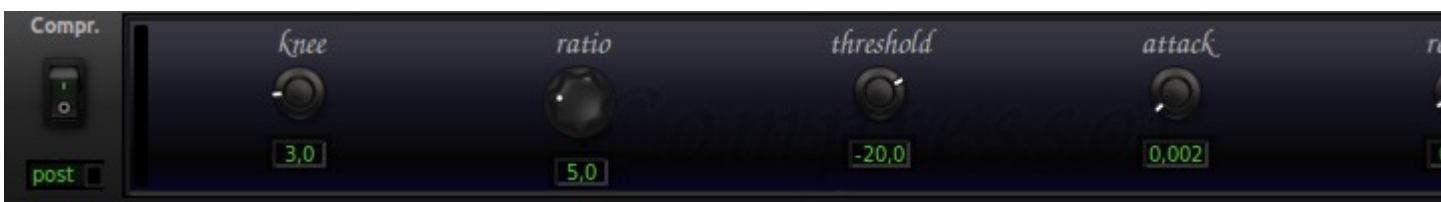
- **Noise shaper** : agisce contro il rumore generale del segnale, eliminando parte del “ronzio” derivante dalla distorsione digitale. È molto utile abbassare il guadagno non desiderato senza rovinare gli altri controlli.

Nel mio caso, se il noise shaper è impostato molto alto, si nota una perdita di luminosità nello strumento. Non è molto importante ma è una cosa che dovremo correggere poi con il pareggio e quello che comincia bene resta buono.

- **Uscita mono livello** : controlla l'uscita del noise gate. Se il nostro guadagno è così alto da intercettare anche il noise gate, questo piccolo abbasserà il segnale fino a 20 dB. Puoi ottenere risultati straordinari con enormi distorsioni e poi abbassando di 10 o 15 dB con l'uscita mono. Attenzione però, potresti essere accusato di mancanza di naturalezza.
- **Clip** : imposta un limite all'uscita del segnale e non consente alla chitarra di superarlo.

[Scopri di più sul Noise Gate](#)

## Compressore



Controllare la dinamica della chitarra o del basso può essere ( *in effetti lo è* ) molto importante, e farlo bene darà un po' di “**incisività**” al nostro tono. Nella produzione musicale è necessario controllarla.

Fin qui tutto bene. Cosa offre il compressore Guitarix e cosa può sposarsi con esso?

- **Knee** : controlla la transizione dal segnale originale a quello compresso.
- **Ratio** ( *rapporto di compressione* ): controlla la quantità di compressione ( *riduzione* ) applicata al segnale. Si scrive X:1 e si misura in dB.

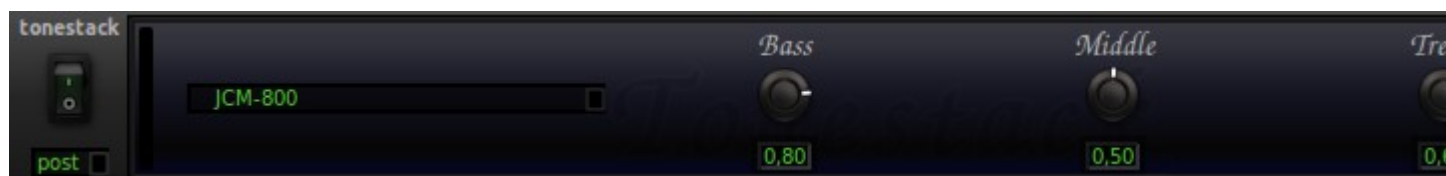
Ad esempio, se avessimo un **rapporto 6:1** il segnale che supera la soglia di 6 dB verrebbe compresso di 1 dB sopra la stessa soglia. Se il segnale lo superasse di 18 dB, verrebbe compresso a 3 dB sopra di esso.

- **Soglia** : è il punto a partire dal quale il segnale viene compresso. Tutto ciò che è al di sotto di questo livello non sarà influenzato.
- **Attack** : è il tempo che impiega il segnale per essere compresso a partire dal momento in cui supera la soglia. I valori bassi solitamente sono compresi tra 50 e 500 microsecondi e valori alti tra 20 e 100 millisecondi.
- **Rilascio** : è l'opposto dell'attacco. È il tempo che impiega il compressore per passare da un segnale compresso ad un segnale sotto la soglia.

I valori giusti dipendono dallo strumento e dal modo di suonare di ciascuno. I valori usuali per la chitarra sono **il rapporto 5:1 e l'attacco da 5 a 10 ms** . Per i bassi vengono utilizzati **valori di attacco simili e un rapporto di 8:1** .

[Ulteriori informazioni sul compressore](#)

## Il Tonestack: Amp Modeler



Non ci sono molte cose da spiegare a riguardo: **diversi modelli di amplificatori insieme ad un equalizzatore a 3 bande** (bassi, medi e alti). Devi armeggiare con le piccole manopole e, sai, se vuoi il maledetto tono dei Metallica, tanti bassi, pochi medi e alcuni alti.

Nel mio caso scelgo un Mesa Boogie, imposto l'equalizzatore su 0,7-0,5 e 0,3 e accendo il tonestack con l'interruttore sinistro.

Il volume è stato leggermente abbassato accendendo il tonestack così in alto con il guadagno principale a 0 dB ( *era su -20 dB* ) e ho già il tipico suono roccioso del Mesa.

Bene, **ti frego un po' con questa frase** : ora spegni il controllo di presenza sulla testa. Sembra “*sgranocchiare un pacchetto di snack*” , non è vero? Quindi **accendi il modulo cabinet** e vedrai quanto velocemente migliora. **Cosa stavi pensando di usare un amplificatore senza collegarlo a un altoparlante?** . Ora sai come suona il *testo corsivo* “**line out**” di un amplificatore.

[Scopri di più sul Tonestack](#)

## Cabinet: emulazione del cabinet degli altoparlanti

Ogni cabinet emetterà un tono diverso e i controlli bassi/alti influenzeranno il suono. Ricorda che in questo modulo è presente un potenziometro del volume aggiuntivo.

## CONSIGLI E SUGGERIMENTI

Stiamo scrivendo un bellissimo e simpatico manuale introduttivo per l'utente...  
Quando avremo finito avrai un bel PDF o ODT per iniziare a suonare con Guitarix2-0.19

A cosa stiamo lavorando:

- **Imparare Guitarix creando il nostro primo suono** . ( *Pronto* ).
- **EQ** : filtri, shelving a 3 bande ed EQ grafico. ( *Pronto* ).
- **Dinamica: "Il Compressore"** . ( *Pronto* ).
- **"La Porta del Rumore"** . ( *lavorando* ).
- **"Distortion"** : Overdrive e, in generale, la distorsione multibanda.
- **"Effetti di modulazione"** : il Phaser, il Flanger, il Chorus...
- **"Plugin LV2"** : Lavorare con Guitarix dalla nostra DAW preferita!!!
- **Convolver** : **"Convoluzione mediante risposte all'impulso"** .

Qualunque cosa tu abbia bisogno di essere spiegata, dacci il tuo feedback sul forum Guitarix  
<http://sourceforge.net/apps/phpbb/guitarix/>

- Gaius Baltar

## COME USARE I NUOVI PLUGIN LADSPA

I nuovi plugin Guitarix [LADSPA](#) avvolgono il motore sonoro Guitarix. Guitarix è costituito da una catena di effetti mono, un punto di insert per il collegamento di effetti esterni e una catena di effetti stereo. I plugin con un ingresso e due uscite non sono gestiti bene dalla maggior parte degli host LADSPA, quindi forniamo un plugin "Guitarix Amp" con un ingresso e un'uscita e un plugin "Guitarix Stereo Fx" con due ingressi e uscite.

L'interfaccia LADSPA è progettata principalmente per i plugin che racchiudono un algoritmo dsp. I plugin Guitarix stanno ampliando un po' questo concetto.

- [Come farlo funzionare](#)
  - [Assegnazioni dei parametri](#)
  - [Automazione DAW](#)
  - [Suggerimenti](#)
- [A cosa servono i plugin Guitarix LADSPA](#)
- [Cosa non puoi fare con loro](#)
- [Punti fini](#)
- [Host testati](#)

## **Come farlo funzionare**

Per prima cosa devi avviare il programma Guitarix ed effettuare alcune impostazioni che verranno memorizzate nel file di stato (~/.gx\_head/gx\_head\_rc) quando selezioni Presets -> save main setting. Questo sarà preimpostato su 0 nel plugin LADSPA. Facoltativamente è possibile definire le

preimpostazioni nel file delle preimpostazioni standard (~/.gx\_head/gx\_headpre\_rc). Questi sono i preset 1 .. N nel plugin LADSPA.

Ora puoi avviare un host LADSPA e inserire un plugin Guitarix. Il preset definito nel file di stato verrà caricato per impostazione predefinita e potrai selezionare diversi preset modificando il parametro LADSPA (selezionando un numero di preset troppo alto non verrà generato un messaggio di errore ma semplicemente non verrà caricato nulla). Il parametro di regolazione del livello modifica il livello del suono in uscita.

## Assegnazioni dei parametri

Successivamente ci sono cinque parametri. Per associarli alle variabili Guitarix, fare clic su un controllo in Guitarix con il pulsante 2 del mouse (pulsante centrale) e inserire un numero nel campo "numero controller midi" (vedi anche [\[Guitarix MIDI Control\]](#) ). Per i parametri LADSPA è più semplice utilizzare semplicemente i numeri da 1 a 5 (se si utilizzano altri numeri, verranno utilizzati i primi 5 in ordine crescente; è possibile visualizzare un elenco di controller definiti nel menu Motore -> Controller Midi). Per aggiornare le impostazioni nel plugin, passa avanti e indietro a un'altra preimpostazione.

I parametri che non vengono utilizzati nel tipo di plugin selezionato verranno saltati. Potrebbe essere meno confuso definire preset con parametri midi sia per "Guitarix Amp" (utilizzare i parametri di tutti i plugin mono, pre-gain, drive e distorsione), sia per "Guitarix Stereo Fx" (utilizzare i parametri di tutti i plugin stereo, master livello e il controllo di livello sul misuratore). Se vuoi definire i parametri per "Guitarix Amp", nascondi il rack stereo e mostra il rack mono.

Le variabili assegnate ai parametri LADSPA non verranno impostate da un preset (come tutti i parametri midi assegnati). Quindi suonerà diverso dal tuo preset, perché i parametri LADSPA sono inizializzati con 0 all'inizio. L'intervallo del parametro LADSPA 0..100 può essere interpretato come valori percentuali dell'intervallo inferiore..superiore della variabile Guitarix. Se hai collegato un controllo di livello, l'uscita LADSPA probabilmente inizialmente sarà silenziosa. Modifica i parametri in valori percentuali corrispondenti ai valori nel tuo preset.

È possibile utilizzare qualsiasi parametro controllabile midi con intervallo continuo, ad eccezione del controllo del cabinet e del tono di presenza.

**Importante** : per definire le assegnazioni dei parametri per i preset nel file di preset standard, è necessario impostare "memorizza midi nei preset" nelle opzioni.

## Automazione DAW

È possibile utilizzare il livello di uscita e i 5 parametri assegnabili per l'automazione DAW. La modifica della preimpostazione attiva un caricamento in background della nuova preimpostazione con tempistiche indefinite. Potresti essere in grado di usarlo per l'automazione (come nella riproduzione dal vivo), ma dovresti consentire un intervallo sufficientemente ampio durante il passaggio per ottenere risultati riproducibili.

## Suggerimenti

- Quando avvii LADSPA l'host dalla riga di comando, vedrai alcune informazioni di registrazione dal plugin, ad esempio il preset caricato e l'assegnazione dei parametri.



- Se desideri utilizzare file preimpostati diversi per Guitarix, utilizza Preset -> Esporta file preimpostato per creare un nuovo file preimpostato. È possibile passare da uno all'altro dei due file preimpostati con Preimpostazioni -> Carica file preimpostato.
- Oppure imposta la variabile d'ambiente LADSPA\_GUITARIX\_PRESET (nell'ambiente di processo dell'host LADSPA), o ancora più specifiche LADSPA\_GUITARIX\_MONO\_PRESET e LADSPA\_GUITARIX\_STEREO\_PRESET (prevalgono le impostazioni più specifiche). Queste variabili devono essere impostate sul percorso (assoluto) del file preimpostato da utilizzare.
- In generale non modificare gli ultimi quattro parametri; sono spiegati nella sezione "Punti particolari".

## A cosa servono i plugin Guitarix LADSPA

- combina alcuni effetti Guitarix con altri plugin ladspa quando hai già un host LADSPA in esecuzione.
- se lavori con una DAW e registri il segnale grezzo della chitarra, invece di collegare diverse istanze Guitarix esterne all'inserito DAW puoi aggiungere i plugin Guitarix LADSPA direttamente alle tracce interessate.
- ovviamente tutto il resto che puoi immaginare e che funziona per te :-)

## Cosa non puoi fare con loro

- Guitarix si sta evolvendo, così come i plugin Guitarix LADSPA. Ciò significa che non puoi aggiungerli a una traccia DAW, attendere un anno, installare la versione corrente e aspettarti che funzioni (o abbia lo stesso suono).
- È sufficiente aggiungere semplici plugin LADSPA a una traccia, magari regolare alcuni parametri e il gioco è fatto. Al contrario, i plugin Guitarix necessitano di un file preset che viene generato e modificato da Guitarix, e il significato dei 5 parametri LADSPA assegnabili dipende dalle impostazioni del preset.

## Punti fini

LADSPA è progettato per algoritmi che non dipendono dalla dimensione del blocco; Funziona bene con Guitarix purché non si utilizzi una delle unità che funzionano con "convolving" (cabinet, presenza o convolver). Queste unità necessitano di una dimensione di blocco fissa e creano thread aggiuntivi che necessitano di priorità speciali per l'elaborazione RT (in tempo reale). Per risolvere questo problema utilizziamo il buffering, che introduce una latenza aggiuntiva (questa latenza aggiuntiva viene annunciata tramite il parametro di output "latency", utilizzato almeno dalla DAW Ardour).

Se disattivi il buffering, non hai latenza aggiuntiva ma i convolver non possono essere utilizzati (cioè verranno bypassati). La dimensione del buffer è sempre la potenza di due che è maggiore o uguale al valore del parametro "buffersize". Se "buffersize" è 0, il plugin seleziona un valore predefinito, che è la dimensione corrente del frame jack se il demone jack può essere contattato dal plugin quando è attivato, oppure 64.

Quando si utilizza jackd non è necessario impostare una dimensione del buffer maggiore della dimensione del frame jackd. Se il jack può essere contattato dal plugin, la dimensione del buffer sarà limitata da quel valore. È possibile utilizzare dimensioni del buffer più piccole, che ridurranno la latenza aggiuntiva (dei campioni buffersize-1) ma aumenteranno il sovraccarico di elaborazione. Per l'automazione DAW, la dimensione del blocco è spesso divisa in sezioni tra i punti di automazione; se vuoi che il plugin segua i parametri di automazione in modo più accurato, usa una dimensione del buffer più piccola (ogni buffer viene calcolato con un set fisso di parametri).

L'interruttore "Non-RT Mode" può essere utilizzato se i thread del convolver non devono funzionare in modalità RT, ad esempio se jackd funziona in modalità non-rt. Se "RT Priority" è 0, la priorità del thread verrà impostata sul valore corretto per jackd quando è possibile contattare jackd.

La strategia di buffering e le priorità dei thread cambieranno solo quando il plugin viene attivato, cosa che avviene sempre all'inizio. Se succede in altri punti dipende dall'host. jack-rack sembra attivarsi quando il bypass è disattivato, Ardour solo dopo che il trasporto è stato interrotto ed è impostata l'opzione "prestazioni -> interrompi plugin con trasporto".

## Host testati

- ardour2
- audacious
- audacity
- jack-rack
- qtractor

## OPZIONI DA RIGA DI COMANDO

Opzioni di aiuto:

```
-h, --help Mostra le opzioni di aiuto
--help-all Mostra tutte le opzioni della guida
--help-style Opzioni di configurazione dello stile GTK
--help-jack Opzioni di configurazione JACK
--help-file Opzioni del file
--help-debug Opzioni di debug
--help-gtk Mostra le opzioni GTK+
```

Opzioni di configurazione in stile GTK:

```
-c, --clear Usa lo stile GTK 'predefinito'
-r, --rcset=STYLE Stile da utilizzare, 'gx1-alloy', 'gx2-emerald', 'gx3-
dezert', 'gx4-black', 'gx5-gree', 'gx6-blue', 'gx7-blues'
```

Opzioni di configurazione JACK:

```
-i, --jack-input=PORTA Ingresso JACK Guitarix
-o, --jack-output=USCITE PORT Guitarix JACK
-m, --jack-midi=PORT Guitarix JACK controllo midi
-n, --name=NAME nome dell'istanza (gx_head predefinito)
-U, --jack-uuid ID sessione Jack
-A, --jack-uuid2 ID sessione Jack
```

## Opzioni file:

-f, --load-file=FILE carica il file di stato all'avvio  
-P, --plugin-dir=Directory DIR con i plugin guitarix (file .so)

## Opzioni di debug:

-B, --builder-dir=DIR directory da cui vengono caricati i file .glade  
-S, --style-dir=Directory DIR con le definizioni dello stile skin (file .rc)  
-t, --log-terminal stampa il registro sul terminale

## Opzioni GTK+:

--class=CLASS Classe di programma utilizzata dal gestore delle finestre  
--gtk-name=NOME Nome del programma utilizzato dal gestore delle finestre  
--screen=schermo SCREEN X da utilizzare  
--sync Rende sincrone le chiamate X  
--gtk-module=MODULI Carica moduli GTK+ aggiuntivi  
--g-fatal-warnings Rende fatali tutti gli avvisi

## Opzioni di applicazione:

-v, --version Stampa la stringa della versione ed esce  
--display=VISUALIZZA X display da utilizzare

## Opzioni non documentate

-D, --dump-parameter Dump della tabella dei parametri in formato json (DEBUG)

## Opzioni interne

Le seguenti opzioni sono opzioni interne (per la gestione di JackSession) ed è meglio non utilizzarle al di fuori di un contesto JackSession:

-U, --jack-uuid ID sessione Jack  
-A, --jack-uuid2 ID sessione Jack  
-f, --load-file=FILE carica il file di stato all'avvio

## GIT

### Opzioni di configurazione JACK:

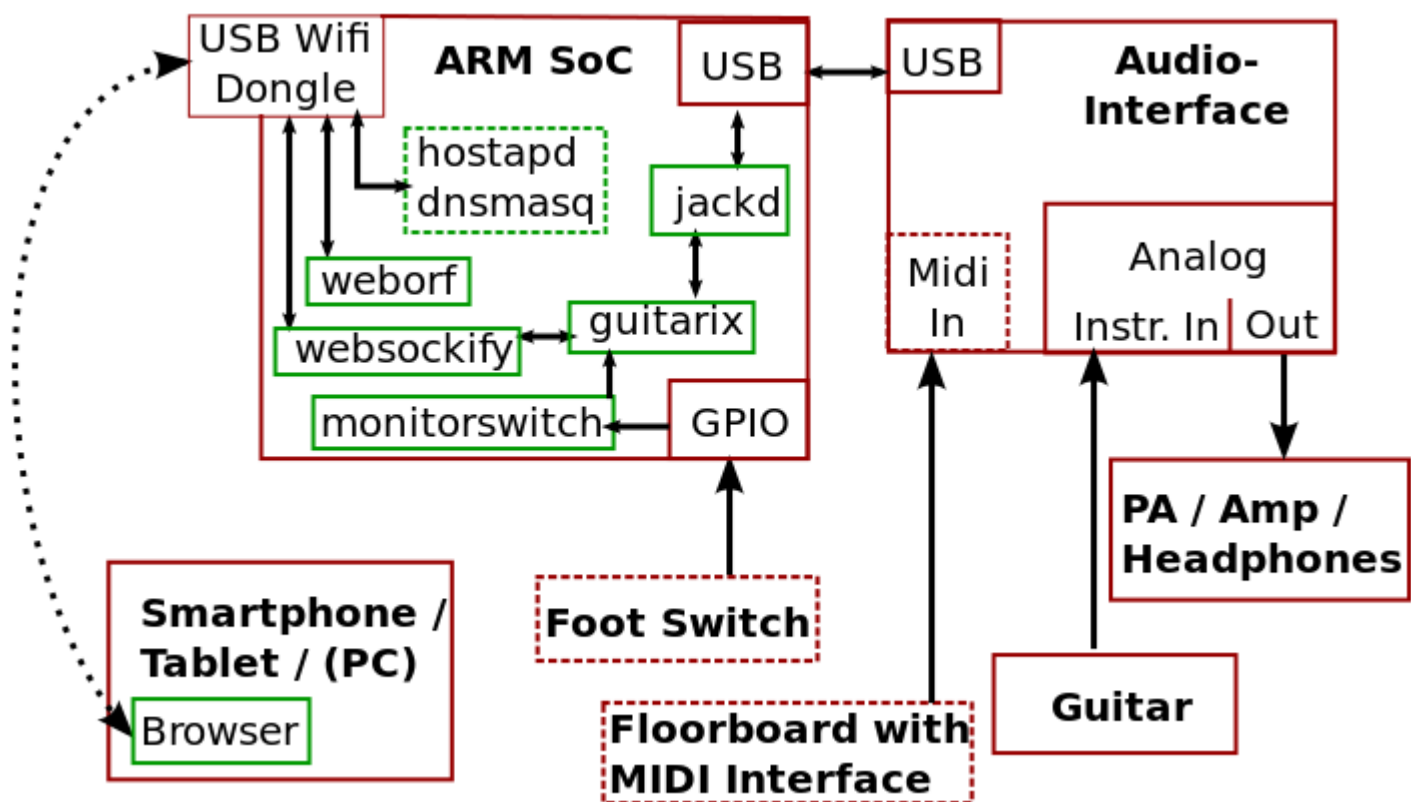
-s, --server-name Nome del server JACK a cui connettersi  
-J, --jack-no-connect Disabilita le porte JACK di autoconnessione  
-K, --disable-auto-save Disabilita il salvataggio automatico nel file di stato all'uscita

Questa guida è una descrizione compatta di come costruire un sistema Guitarix embedded. Nello specifico questa descrizione è per la piattaforma pcDuino <http://www.pcDuino.com>. Sono necessarie alcune conoscenze dettagliate di Linux, se non hai mai compilato un kernel questo potrebbe essere un po' troppo avanzato. (Alla fine potrei fornire un'immagine del filesystem, ma al momento c'è troppo lavoro in corso).

L'obiettivo è avere una scatola contenente un amplificatore per chitarra virtuale da poter utilizzare sul palco. La chitarra va collegata all'ingresso e il PA/monitor all'uscita. Il box è controllato da una pedaliera con uscita MIDI (controllo MIDI Guitarix con ad esempio FCB1010) o da un singolo interruttore a pedale. L'interruttore a pedale pone Guitarix in una modalità in cui è possibile eseguire operazioni come il passaggio a un'altra impostazione o l'attivazione dell'accordatore

pizzicando una corda della chitarra. Il passaggio a un'impostazione diversa può essere effettuato in meno di mezzo secondo con un po' di pratica.

È possibile utilizzare uno smartphone o un tablet (collegato tramite WLAN) per visualizzare l'impostazione corrente, l'accordatore, o anche modificare le impostazioni o definirne di nuove (nota: modifica/definizione remota di impostazioni non ancora implementate in Guitarix).



Il sistema ha una latenza end-to-end di 14 ms e può elaborare tutti i preset di fabbrica Guitarix. Per essere utilizzato sul palco avrebbe bisogno di una custodia robusta.

== Hardware ==

Per riferimento includo l'hardware che utilizzo tra parentesi.

=== Necessario per eseguire il sistema ===

- scheda con SoC ARM (pcDuino)
- Adattatore di alimentazione CA
- interfaccia audio (Presonus USB AudioBox)

Il pcDuino è un sistema ARM Cortex A8 con clock da 1 GHz e 1 GB di RAM (il SoC è un Allwinner A10).

Dovrebbe funzionare anche su altri sistemi simili come <http://cubieboard.org/> Cubieboard e <http://www.marsboard.com/> MarsBoard . Il requisito minimo è qualcosa come un processore ARM single core da 1 GHz con un'unità vettoriale NEON. 500 MB di RAM sono sufficienti (anche 256 MB dovrebbero funzionare). Il Raspberry Pi non sembra essere abbastanza potente da gestire tutti i preset di fabbrica di guitarix (non ha un'unità NEON e inoltre ha un'implementazione hardware USB scadente che rifiuta di funzionare con molte interfacce audio USB).

Finora non ho ancora messo insieme un'immagine per la memoria flash integrata , quindi mi serve la scheda micro SD, necessaria anche per lo sviluppo. Ma dovrebbe essere possibile ridurre la dimensione dell'immagine a 1 GB (basta eliminare tutti i pacchetti necessari solo per lo sviluppo) e inserirla nella memoria flash del dispositivo.

Esistono interfacce audio più economiche dell'AudioBox, ma per Guitarix ne hai bisogno di una con un buon SNR (rapporto segnale rumore), soprattutto per impostazioni ad alto guadagno. Ma vedi anche la sezione successiva..



pcDuino in una semplice custodia con presa di corrente, interruttore a pedale, dongle USB (nella presa USB sinistra) e interfaccia audio.

==== Combattere il rumore ====

L'alimentazione per Presonus AudioBox viene fornita tramite l'interfaccia USB. È fantastico perché non è necessario un adattatore di alimentazione CA aggiuntivo, ma è un male perché il rumore sulla linea di alimentazione USB verrà alimentato nel preamplificatore dell'interfaccia audio.

Ho dovuto giuntare 2 cavi USB e utilizzare un alimentatore aggiuntivo per eliminare il rumore in impostazioni ad alto guadagno: tagliare il filo rosso e collegare l'estremità che va all'interfaccia audio con l'alimentatore pulito (+5V), collegare il filo nero con GND (non è necessario tagliarlo, basta rimuovere un po' di isolamento) e non toccare i cavi bianco e verde (il colore dei fili utilizzati all'interno dei cavi USB è definito dallo standard).

Inoltre ho dovuto mettere a terra i circuiti collegando GND al pin di messa a terra della presa a muro CA.

Quindi avevo bisogno anche di:

- cavo USB giuntato
- presa di corrente aggiuntiva
- messa a terra

Raccomandazione: è meglio procurarsi un'interfaccia con alimentazione separata...



Cavo USB giuntato con spina di alimentazione CA aggiuntiva (messa a terra aggiuntiva non mostrata).

=== Hardware opzionale ===

- interruttore a pedale (Yamaha FC 5)  
\*\* materiale aggiuntivo (connettore femmina per jack da 6,3 mm ecc.)
- Chiavetta Wi-Fi USB (EDIMAX EW-7811UN)
- smartphone (testato con il telefono Android Nexus S)
- custodia per scheda pcDuino

È possibile utilizzare qualsiasi interruttore a pedale che chiuda un contatto.

Collegamento dell'interruttore a pedale a GPIO 8 e GND.

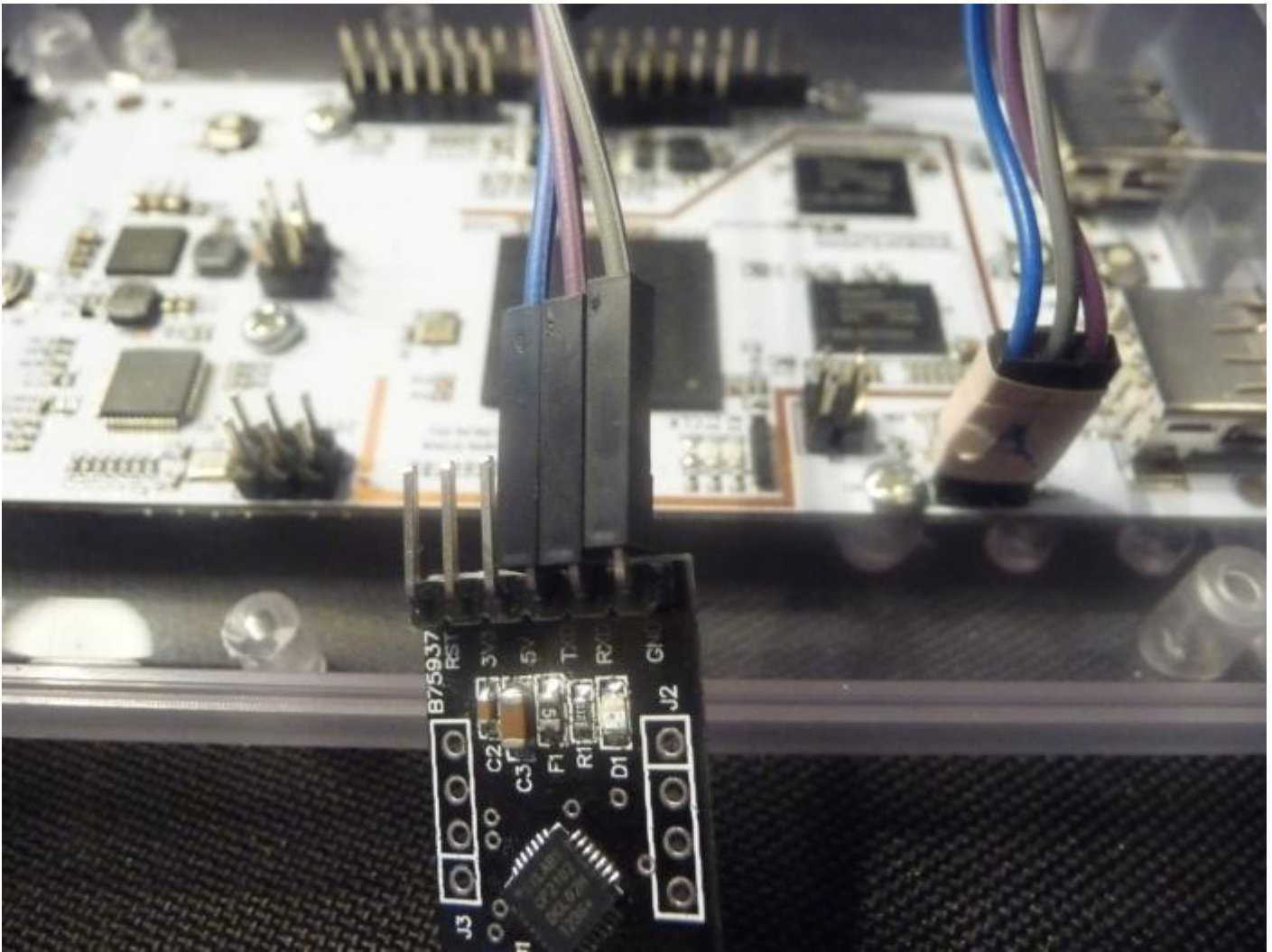
Nota: il dongle Wifi Edimax utilizza RTL8192CU che non è ben supportato dal driver standard nel kernel. Il driver del produttore (che viene utilizzato qui) supporta solo un'interfaccia del kernel precedente e non è supportato dall'hostapd standard.

=== Hardware aggiuntivo necessario per sviluppare il sistema ===

- scheda micro SD (minimo 4 GB, 8 GB potrebbero essere la scelta migliore; preferibile una veloce (classe 10))
- Convertitore RS232 da USB a TTL per collegare il laptop all'interfaccia seriale (modello senza marchio basato sul chip CP2102)
- PC portatile o desktop con Linux, lettore di schede SD
- cavo patch di rete

Nota: il mio convertitore seriale/USB stranamente non riporta sull'etichetta i segnali che fornisce, ma quelli che devono essere collegati ad esso. TxD non è l'uscita dati del circuito, ma l'ingresso a cui deve essere collegato il pin TxD del pcDuino.

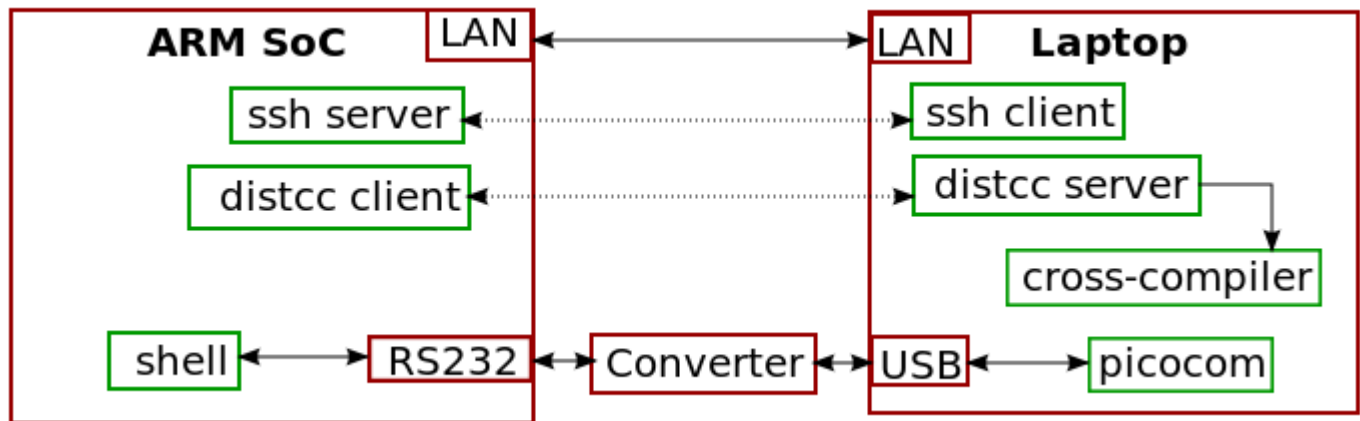




== Configurazione del sistema di sviluppo ==

Poiché utilizzo il pcDuino senza display (senza testa), è necessario un altro sistema per lavorarci. Inoltre non è una buona idea provare a compilare il kernel Linux sul sistema ARM (anche se è possibile compilare piccoli programmi direttamente su di esso).

Utilizzo un laptop con un cavo patch di rete per connettermi al sistema integrato (pcDuino supporta <http://en.wikipedia.org/wiki/Auto-MDIX#Auto-MDIX> Auto-MDIX, non è necessario alcun cavo incrociato). Per guardare i messaggi di avvio, eseguire il debug di un nuovo kernel o accedere al dispositivo quando l'interfaccia di rete non funziona è necessario un convertitore da seriale a USB.



Osservazioni: ho aggiunto una riga a `~/.bash_aliases` per avviare l'emulatore di terminale con 'picocom' dopo aver collegato il convertitore (testato su un sistema Ubuntu).

```
<code>alias picocom='picocom -b115200 -ej /dev/ttyUSB0'</code>
```

L'utilità 'script' (pacchetto: bsdutils) può essere utilizzata per acquisire la sessione, ad es

```
<code>script -c 'picocom -b115200 -ej /dev/ttyUSB0' /tmp/session.txt</code>
```

=== Toolchain del compilatore incrociato, distcc ===

Sul sistema di sviluppo sono necessari il pacchetto distcc, un cross compiler armhf (una "toolchain", composta da compilatore, assembler, linker...) e un semplice emulatore di terminale (picocom).

Il kernel dovrebbe essere compilato in modo incrociato sul sistema di sviluppo. Per gli altri programmi l'impostazione consigliata è lavorare sul sistema ARM (con ssh) e utilizzare distcc per spostare il duro lavoro del compilatore sul "sistema di sviluppo".

- Arch Linux ARM ha un buon articolo sulla <http://archlinuxarm.org/developers/distcc-cross-compiling> .
- Ubuntu: installa il pacchetto `g++-arm-linux-gnueabi`
- Debian: <http://emdebian.org/Emdebian> , <http://wiki.debian.org/BuildingCrossCompilers>

In ogni caso, il compilatore incrociato sul sistema di sviluppo deve essere la stessa versione del compilatore sul sistema ARM se si desidera utilizzare distcc (per la compilazione del kernel andrebbe bene una versione diversa). Se non riesci a trovarne uno precompilato devi compilarne uno tu stesso come descritto nel documento Arch Linux di riferimento.

=== Configurazione di rete ===

Saltare questa sezione se si collega il sistema integrato a un router di rete.

Questa sezione spiega come configurare il sistema di sviluppo per connettersi al sistema integrato utilizzando un cavo patch di rete. Le connessioni dal tuo sistema integrato a Internet verranno instradate (ma non saranno accessibili da Internet). L'indirizzo IP dell'interfaccia sul sistema di sviluppo sarà "192.168.33.1". La configurazione di rete sul sistema integrato utilizzerà un indirizzo in questa sottorete (192.168.33.0).

Lo script `/usr/local/sbin/pointpoint_conn` prevede che le variabili d'ambiente `IFACE` e `MODE` (start o stop) siano impostate. Si prega di dare un'occhiata allo script prima dell'uso. Puoi impostare il nome dell'interfaccia che si connette a Internet (`ext_iface`) e potresti voler modificare le regole di iptable (per ottenere la configurazione corretta la tua connessione Internet deve essere già impostata prima che lo script sia avviato o `ext_iface` o `ext_iface_default` devono essere impostati sul nome dell'interfaccia corretto).

`/usr/local/sbin/pointpoint_conn`:

[file:pointpoint\\_conn](#)

È pensato per essere utilizzato nel file `/etc/network/interfaces` su sistemi Ubuntu o Debian (sostituisci eth0 con il nome dell'interfaccia a cui è connesso il sistema ARM):

```
iface eth0 inet manuale
up /usr/local/sbin/pointpoint_conn
down /usr/local/sbin/pointpoint_conn
```

È possibile assegnare un nome di rete in base alla configurazione di rete del sistema integrato in `/etc/hosts`, ad esempio:

```
192.168.33.3 pcduino
```

Inoltre puoi installare netplugd (e impostare il nome dell'interfaccia da monitorare in `/etc/netplug/netplugd.conf`), per configurare automaticamente l'interfaccia quando colleghi il cavo.

== Configurazione del sistema integrato ==

Utilizzo <https://www.archlinux.org/Arch Linux> sul sistema ARM. Il sistema di base e alcuni pacchetti aggiuntivi possono essere trovati su <http://archlinuxarm.org/Arch Linux ARM>.

Ottieni il sistema embedded di base in esecuzione, incluso l'accesso ssh e la connessione di rete al tuo PC di sviluppo: <http://archlinuxarm.org/platforms/armv7/pcduino Arch Linux ARM per pcDuino>

Una descrizione migliore su come partizionare la scheda SD (cercare il passaggio 3): <http://pcduino.com/forum/index.php?topic=3642.msg3837#msg3837> Partizionare la scheda SD

Dopo aver inserito la scheda SD e avviato il sistema, dovresti essere in grado di accedere (tramite console su RS232 o anche ssh su rete se hai collegato il sistema a un router di rete). La password di root è "root".

Aggiorna l'installazione: `<code>pacman -U</code>`

C'erano alcuni piccoli conflitti che dovevano essere risolti manualmente (non ricordo esattamente).

Crea un account utente:

`<code>useradd -G audio <user></code>`

=== Pacchetti necessari per eseguire il sistema ===

I pacchetti precompilati su [archlinuxarm.org](http://archlinuxarm.org) possono essere installati con `<code>pacman -S <name></code>`.

- installa: openssh, fake-hwclock, systemd-rtirq
- installare facoltativamente: rtl8188-hostapd, dnsmasq
- costruire programmi modificati: weborf, websockify, jackd
- creare un programma personalizzato per notificare a Guitarix i cambiamenti di stato del pin 8 GPIO: monitorswitch

Risorse: \* [file:weborf-0.13-4.src.tar.gz](http://file:weborf-0.13-4.src.tar.gz) \* [file:websockify-cmod-0.4.1m-2.src.tar.gz](http://file:websockify-cmod-0.4.1m-2.src.tar.gz) \* [file:monitorswitch.tar.gz](http://file:monitorswitch.tar.gz) \* [file:rtl8188-hostapd-1.1-1.src.tar.gz](http://file:rtl8188-hostapd-1.1-1.src.tar.gz)

Osservazioni: si consiglia fake-hwclock poiché pcDuino non dispone di un orologio hardware.

=== Configurazione del pacchetto ===

Modifica in `<code>/etc/dnsmasq.conf</code>`:

`<pre>`

interfaccia=wlan0

bind-interfaces

dhcp-range=192.168.34.2,192.168.34.10,12h

`</pre>`

Modifica in `<code>/etc/hostapd/hostapd.conf</code>`:

`<pre>`

Interface=wlan0

driver=rtl871xdrv

ssid=Guitarix

`</pre>`

Aggiungi a `<code>/etc/sudoers</code>`:

`<code>ad ALL=(ALL) NOPASSWD: ALL</code>`

=== Pacchetti aggiuntivi necessari per lo sviluppo ===

- `<code>pacman faust distcc git</code>`
- qualche editor (ad esempio jed)

=== Compilazione del kernel Linux ===

Il sorgente del kernel che ho usato è l'ID commit 22628c725d nell'archivio <https://github.com/linux-sunxi/linux-sunxi>. Puoi scaricarlo direttamente da github come file zip: <https://github.com/linux-sunxi/linux-sunxi/archive/22628c725dd2ab655b288be2ecdcc1a6688a7a75.zip> file zip sorgente del kernel

Quindi applicare una patch Linux rt adattata: <file:patch-3.4.29-sunxi-rt42.diff.gz>

```
Applica la patch:
<pre>
--- drivers/gpio/gpio-sunxi.c~      2013-03-28   21:44:46.000000000   +0100
+++ drivers/gpio/gpio-sunxi.c      2013-05-08   23:50:29.297033842   +0200
@@@                                -320,7          +320,8          @@@
(gpio_eint_list [i] .gpio >= 0)) {
status          &= ~((1 <<
SUNXI_CLEAR_EINT(sgpio->gaddr, i);
-          generic_handle_irq(sgpio->irq_base + i);
+          //ad#2013-05-08# generic_handle_irq(sgpio->irq_base + i);
+          handle_nested_irq(sgpio->irq_base + i);
}
}
</pre>
```

Sostituisci il file `<code>.config</code>` nella directory di build di livello superiore con <file:config> .

Ora puoi apportare modifiche alla configurazione:

```
<code>make ARCH=arm menuconfig</code>
```

È incluso tutto il supporto hardware necessario per il sistema descritto. Alcuni moduli sono configurati ma non utilizzati. Se hai bisogno di driver o funzionalità specifici, generalmente è più semplice includerli semplicemente nella configurazione del kernel e ricostruire il kernel invece di utilizzare i moduli.

```
Compila il kernel con:
<pre>
PATH="path-to-cross-tools-bin-directory/bin:$PATH" \
make ARCH=arm CROSS_COMPILE=arm-unknown-linux-gnueabi- -j5 uImage
</pre>
```

Quindi copia `<code>arch/arm/boot/uImage</code>` nella prima partizione della scheda SD.

=== Configurazione del sistema integrato ===

```
Modificare le righe corrispondenti in <code>/etc/makepkg.conf</code>:
<pre>
BUILDDENV=(fakeroot distcc color !ccache check !sign)
DISTCC_HOSTS="192.168.33.1"
CFLAGS="-mcpu=cortex-a8 -mfloat-abi=hard -mfpu=neon -O2 -pipe -fstack-protector --
param=ssp-buffer-size=4"
```

```
CXXFLAGS="-mcpu=cortex-a8 -mfloat-abi=hard -mfpu=neon - O2 -pipe -fstack-protector --
param=ssp-buffer-size=4"
MAKEFLAGS="-j8"
</pre>
```

MAKEFLAGS può essere impostato leggermente più alto del numero di core nella tua macchina di sviluppo. Se non utilizzi la connessione tramite cavo patch dovrai modificare l'indirizzo ip DISTCC\_HOSTS.

```
Aggiungi a <code>/etc/environment</code>:
<pre>
DISTCC_HOSTS=192.168.33.1/6
CC='distcc gcc'
CXX='distcc g++'
</pre>
```

==== Compila Guitarix ====

```
<pre>
$ git clone git://git.code.sf.net/p/guitarix/git guitarix-git
$ cd guitarix-git/trunk
$ ./waf configure --no-ladspa --no-new-ladspa \
> --faust --faust-float --faust-vettorize --faust-options='-vs 128' \
> --convolver-ffmpeg \
> --cxxflags-release='-mcpu=cortex-a8 -mfpu=neon -O3 -ffast-math -funsafe-math-optimizations' \
> -j8
$ ./waf build -j8
</pre>
```

==== Avvio del sistema ====

Disabilita la limitazione RT:

```
<code>echo -1 > /proc/sys/kernel/sched_rt_runtime_us</code>
```

```
Impostazioni priorità irq in <code>/etc/conf.d/rtirq</code>:
<pre>
RTIRQ_NAME_LIST="timer clock rtc ohci dma snd nand mmc i8042 eth0 ehci"
RTIRQ_PRIO_HIGH=90
RTIRQ_PRIO_DECR=3
RTIRQ_PRIO_LOW=11
RTIRQ_RESET_ALL=0
RTIRQ_NON_THREADED="rtc snd usb"
</pre>
```

```
<code>/etc/systemd/system/jackd.service</code>: file:jackd.service <br />
<code>/etc/systemd/system/guitarix.service</code>: file:guitarix.service <br />
<code>/etc/systemd/system/monitorswitch.service</code>: file:monitorswitch.service <br />
<code>/etc/systemd/system/weborf.service</code>: file:weborf.service <br />
<code>/etc/systemd/system/websockify.service</code>: file:websockify.service <br />
<code>/etc/systemd/system/hostapd.service.d /configure-wlan0.conf</code>: file:configure-
```

[wlan0.conf](#)

<br

/>

<code>/etc/systemd/system/dnsmasq.service.d/hostapd.conf</code>: [file:hostapd.conf](#)

== Configurazione dello smartphone ==

- installare FireFox
- installa l'estensione FireFox a schermo intero
- disattivare lo schermo spento
- connettersi al punto di accesso guitarix
- apri <code>http://guitarix/</code>



AUTORE:

SOURCEFORCE Username: brummer10

TRADUZIONE: hopfrog