

# PipeWire (autore: Vittorio Albertoni)

## Premessa

PipeWire è un server multimediale di nuova generazione, che sostituisce i server audio PulseAudio e JACK nei sistemi operativi Linux e fa molto altro.

Esso estende la portata del server audio durante l'elaborazione di qualsiasi flusso multimediale e può mescolare e reindirizzare stream con video, fornendo anche opzioni per la gestione di sorgenti video, come dispositivi di acquisizione video, webcam o contenuto dello schermo generato dall'applicazione.

È un software libero rilasciato con licenza MIT, creato da Wim Taymans alla Red Hat.

Inizialmente si chiamò PulseVideo, in quanto nato per offrire una gestione dei flussi e delle periferiche video simile a quella che PulseAudio offriva per l'audio.

Dal 2021 Fedora lo ha implementato nelle sue distro in sostituzione di PulseAudio e JACK per gestire l'audio.

Dal 2023 praticamente tutte le distro Linux lo utilizzano con la stessa finalità.

Nei sistemi dove è installato si avvia per default all'avvio del sistema e ci mette a disposizione tutto ciò che serve per avere l'audio di tutti i contenuti multimediali del nostro computer, indipendentemente dal fatto che il software per produrlo sia stato concepito per il server PulseAudio o per il server JACK.

Chi lo ha progettato si è infatti preoccupato di garantire retrocompatibilità con questo software.

In realtà, nel momento in cui scrivo (aprile 2024) sono riuscito a sperimentare questa retrocompatibilità solo in casa Fedora.

Su altre distro tutto funziona bene per il software prodotto per il server PulseAudio ma il software prodotto per il server JACK presenta seri problemi di malfunzionamento.

In questo manualetto mostro come usare PipeWire per lavorare con le due DAW Linux a portata di dilettante Qtractor e Rosegarden e quella che forse è la più grande novità che ci porta PipeWire: darci la possibilità di regolare la latenza del computer nel trattare l'audio.

## Indice

<b>1</b>	<b>Come si presenta</b>	<b>2</b>
<b>2</b>	<b>Uso con Qtractor</b>	<b>3</b>
<b>3</b>	<b>Uso con Rosegarden</b>	<b>4</b>
<b>4</b>	<b>Regolazione della latenza</b>	<b>5</b>

# 1 Come si presenta

Per avere un'idea di come funziona PipeWire e per utilizzarlo quando necessario per creare collegamenti diversi da quelli che si creano per default dobbiamo usare un piccolo tool, che a volte troviamo preinstallato. Se non è così possiamo installarlo con il gestore dei programmi della distro. Si chiama `qpwgraph`.

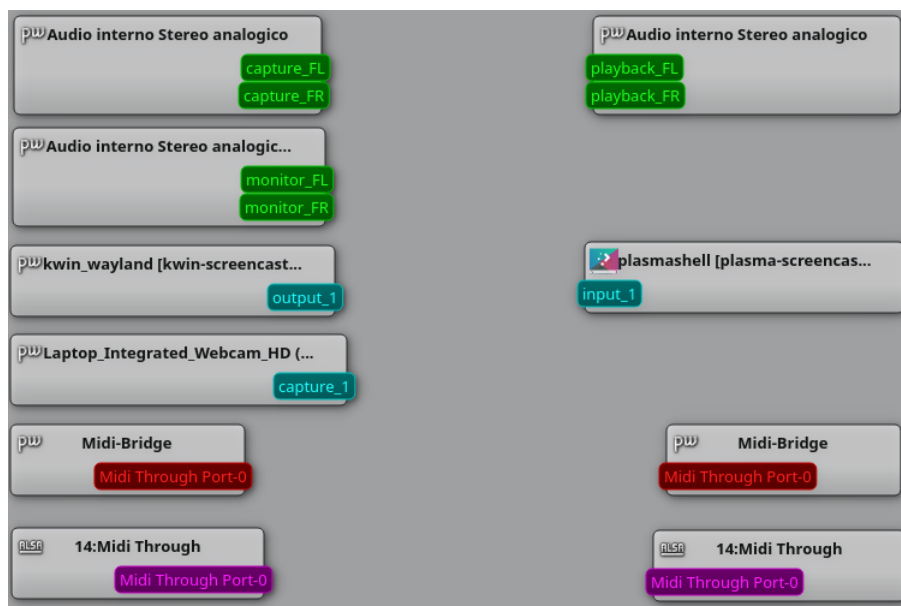
## Qpwgraph

In tutto simile all'interfaccia grafica del server JACK serve per mostrare i collegamenti tra le varie componenti di un progetto multimediale e per poterli modificare.

Nuovi collegamenti si creano per trascinamento del mouse tra i punti da collegare.

Per disattivare un collegamento lo si seleziona con il puntatore del mouse e, cliccando destro sul collegamento selezionato, si sceglie DISCONNECT nel menu a discesa.

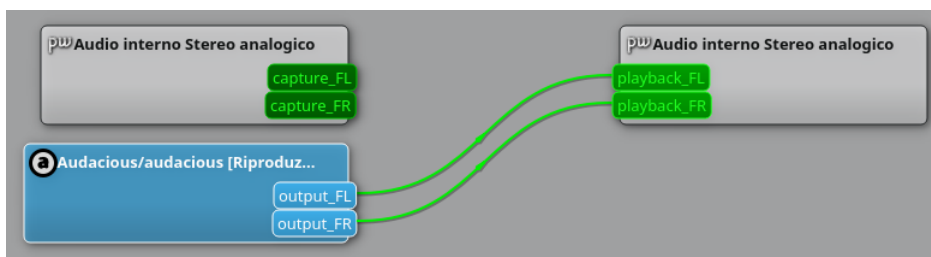
Ci mette a disposizione questi elementi per i collegamenti



Le illustrazioni provengono dalla distro Fedora Jam.

All'attivazione di una qualsiasi funzione inerente l'audio, che è l'aspetto che qui ci interessa, si instaura automaticamente un collegamento di base affinché si produca l'effetto voluto.

Per esempio, attivando il lettore audio Audacious per riprodurre un brano musicale, automaticamente si instaura il collegamento



che indirizza l'output del lettore Audacious all'apparato di sistema per la riproduzione del suono analogico.

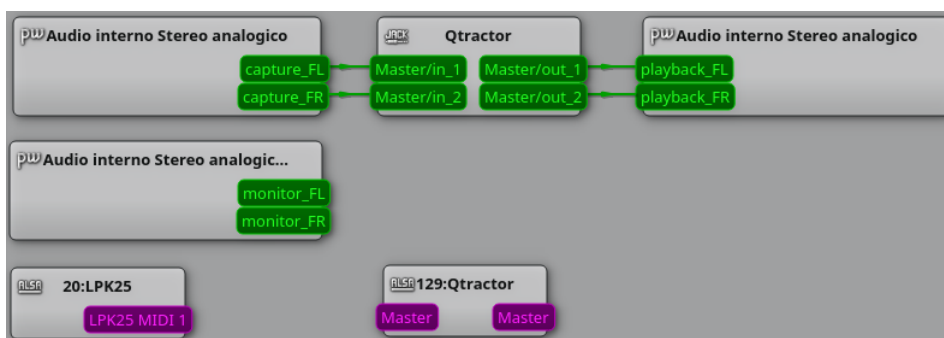
Se ci limitiamo a cose «normali», come riprodurre un file audio o un filmato con l'audio o un file MIDI con Timidity, non dobbiamo fare assolutamente nulla in quanto i necessari collegamenti si instaurano automaticamente.

Se dobbiamo fare cose un po' più complicate dobbiamo intervenire con qualche manualità.

Qui di seguito illustro alcune cose attinenti l'uso dei software Qtractor e Rosegarden, in aggiornamento ai manualetti che si trovano sul mio blog all'indirizzo [www.vittal.it](http://www.vittal.it), che trattano l'uso di quei software in ambiente PulseAudio/JACK.

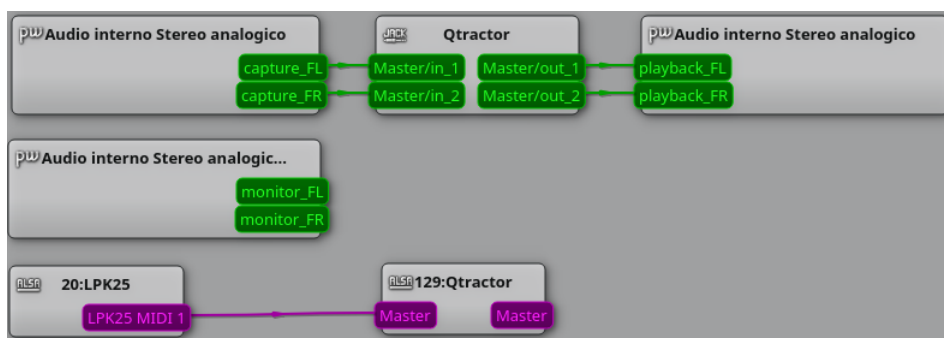
## 2 Uso con Qtractor

Al lancio di Qtractor con collegata una tastiera MIDI (nel caso la LPK25 della Akai) si creano automaticamente questi collegamenti

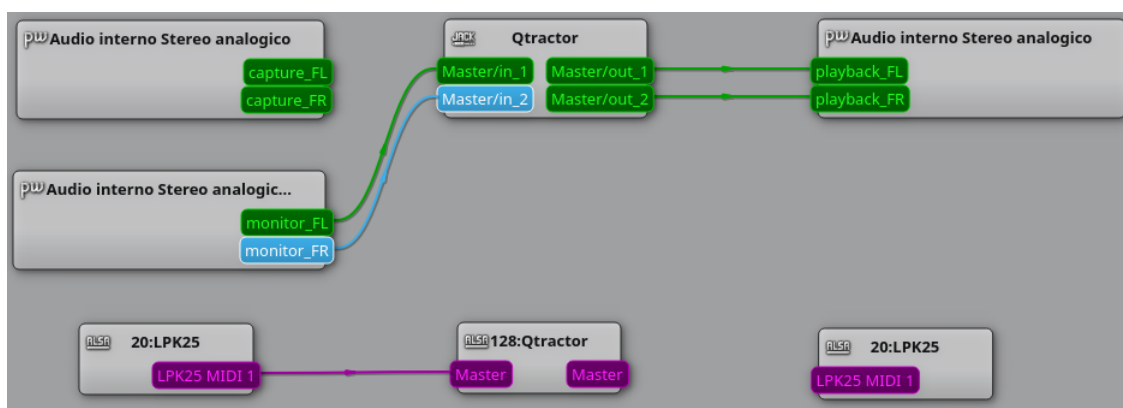


E' quanto basta per registrare audio dal microfono incorporato, per creare una clip MIDI con l'editor MIDI e per ascoltare ciò che si è fatto.

Se vogliamo creare una clip MIDI inserendo le note con la tastiera MIDI dobbiamo collegare la tastiera a Qtractor



Per registrare su traccia audio clip MIDI e altre tracce audio non ammutolite, e per farlo senza che il microfono incorporato generi disturbi, dobbiamo creare questo collegamento

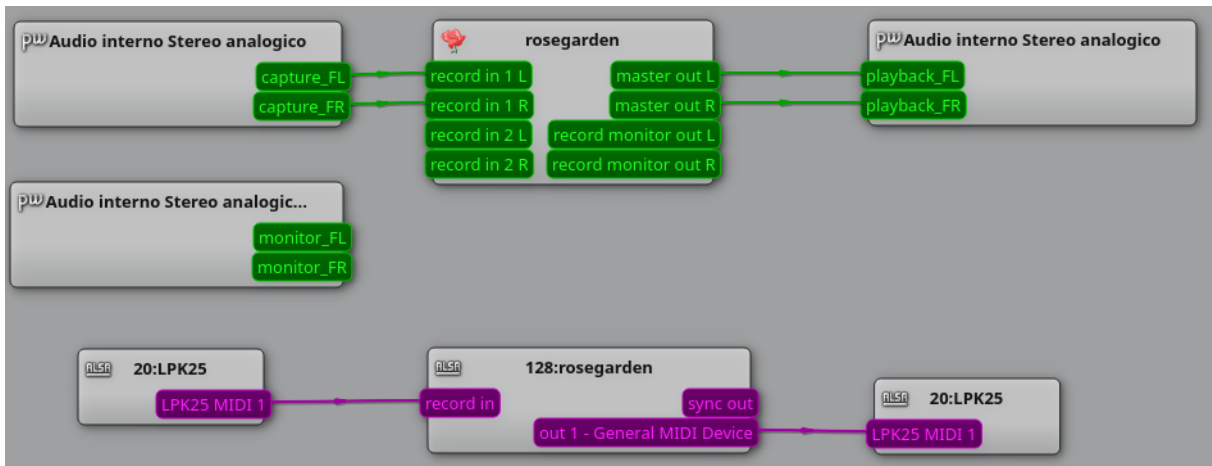


dove sostituiamo la modalità monitor alla modalità capture, quella che cattura da microfono incorporato.

Tutto, soprattutto la registrazione da clip MIDI, più facile rispetto a ciò che dovremmo fare con JACK.

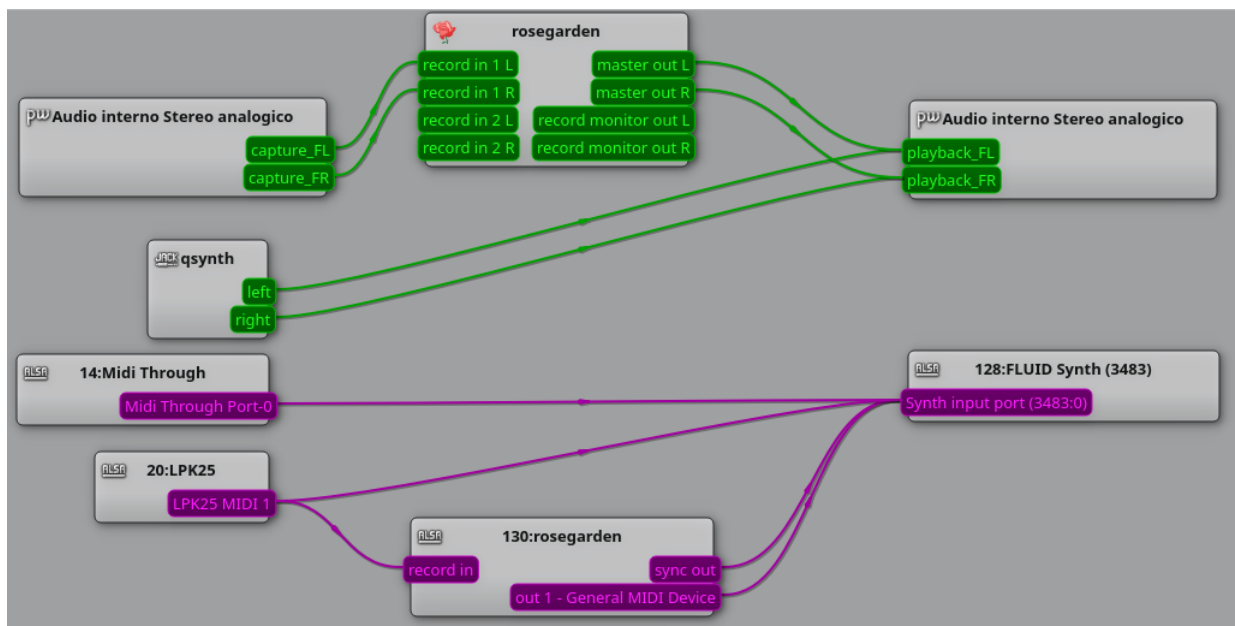
### 3 Uso con Rosegarden

Se avviamo Rosegarden con collegata una tastiera MIDI (nel caso la LPK25) si instaurano automaticamente i seguenti collegamenti



Quanto basta per registrare tracce audio da microfono incorporato e ascoltarle, per creare tracce MIDI e udirne il suono attraverso il dispositivo MODULO DEL SINTETIZZATORE inserendo le note da tastiera o da editor.

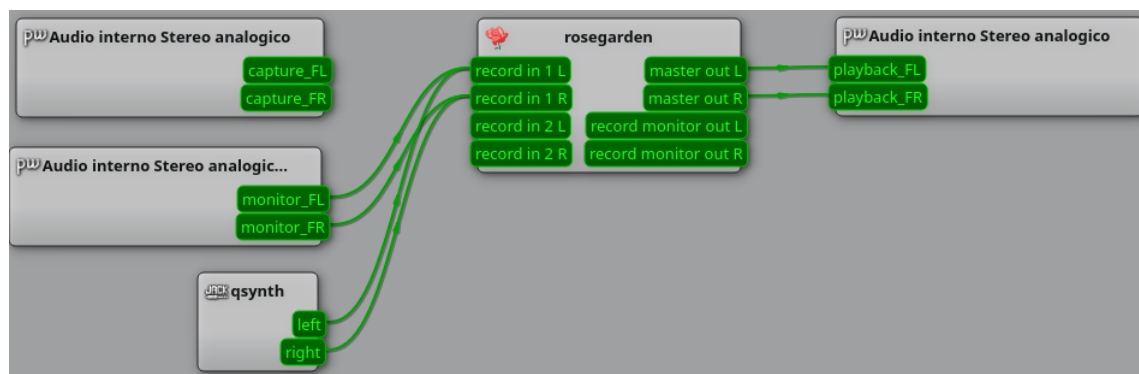
Lanciando Rosegarden, sempre con collegata una tastiera MIDI, con aperto il sintetizzatore QSynth, in modo da poter utilizzare il dispositivo GENERAL MIDI DEVICE, si instaurano questi collegamenti



Quanto basta per registrare tracce audio da microfono incorporato e ascoltarle, per creare tracce MIDI e udirne il suono attraverso il dispositivo GENERAL MIDI DEVICE inserendo le note da tastiera o da editor.

Il nostro progetto viene alimentato con le tracce MIDI e Audio necessarie per generare il file audio contenente il nostro brano musicale: potremmo avere tracce Audio con registrato dal vivo uno strumento musicale, una voce cantante, potremmo avere tracce MIDI importate o generate in Rosegarden, quali utilizzando Soundfont con QSynth quali utilizzando altri sintetizzatori.

Una volta deciso quali tracce debbano essere utilizzate per generare il file audio definitivo, si ammutoliscono quelle che non si intendano utilizzare e si registra il tutto che rimane sulla nuova traccia destinata a contenerlo previa creazione dei seguenti collegamenti



## 4 Regolazione della latenza

La latenza è l'intervallo di tempo che intercorre tra il momento in cui un segnale arriva in input e il momento in cui è disponibile il suo output.

Nel processo di conversione dei suoni da analogico a digitale e viceversa, fino a che ciò che facciamo è semplicemente goderci l'ascolto di un file digitale attraverso altoparlanti o in cuffia (conversione da digitale a analogico) non avvertiamo il problema della latenza in quanto i suoni che udiamo sono belli e sincronizzati indipendentemente dal tempo che ha impiegato il computer a tradurli da digitale (il contenuto del file digitale) a analogico (il suono che udiamo).

Quando invece ci dedichiamo alla produzione di file digitali partendo da suoni (attraverso registrazione dal vivo) o da processi di produzione del suono (attraverso l'elaborazione di file o segnali MIDI) il tempo che impiega il computer a produrre il segnale digitale nelle varie situazioni diventa un fattore di grande disturbo.

Ce ne rendiamo conto semplicemente nel suonare una tastiera MIDI su un computer dotato di un normale sistema operativo, avvertendo il ritardo con cui udiamo il suono prodotto dalla pressione di un tasto.

Già suonare come si deve udendo con ritardo le note che suoniamo è un bel problema, figuriamoci se in questo modo dovessimo registrare in digitale accompagnando un saxofonista i cui suoni siano addirittura convertiti con un ritardo diverso da quello della tastiera MIDI.

Unico modo per risolvere questo tipo di problemi è azzerare la latenza, almeno, visto che la latenza non è azzerabile del tutto, ridurla a dimensioni non avvertibili da orecchio umano.

Tradizionalmente la riduzione della latenza avviene intervenendo sul sistema operativo: o con l'impiego di particolari driver, come in Windows oppure, come in Linux, compilando il kernel con questo obiettivo e producendo le varie distro a bassa latenza (Ubuntu Studio, Fedora Jam, AVLinux).

PipeWire ci offre la possibilità di regolare la latenza in modo che anche lavorando su un normale kernel Linux possiamo ottenere basse latenze.

Dobbiamo prima capire da dove viene la latenza di un computer.

Come avviene per tutti i processi, anche in campo audio i dati da elaborare, nel nostro caso specifico i segnali analogici da digitalizzare, vengono immagazzinati in un buffer per essere elaborati una volta raggiunta una certa quantità.

Più è capiente il buffer, meno lavora la CPU, meno si scalda il computer, ecc.

Ecco perché, normalmente, se il computer non è chiamato a particolari elaborazioni audio (segnatamente registrazioni con garanzia di sincronizzazione) si lavora con buffer capienti, in modo da alleggerire la pressione sulla CPU.

In particolare, nel caso dell'audio, possiamo avere un'idea della latenza, espressa in milisecondi (msec), rapportando la dimensione del buffer (quantum di clock) alla frequenza di campionamento (in kHz) del segnale audio analogico per la sua traduzione in audio digitale.

Per esempio, la latenza in presenza di una frequenza di campionamento da 44,1 kHz (quella del CD Audio) e un buffer da 4096 bytes (dimensione normale) è uguale a  $\frac{4096}{44,1}$ , cioè circa 93 msec.

Sembra poco, ma è tantissimo. Assolutamente inaccettabile per effettuare registrazioni di audio analogico utilizzando una delle varie DAW in circolazione.

I professionisti dell'audio considerano mediamente appena accettabile una latenza di 30 msec e ottimale, soprattutto per registrazioni dal vivo di strumenti musicali che eseguono brani veloci, una latenza di 5 msec.

PipeWire ci offre il modo di gestire la latenza durante la registrazione e la produzione audio, senza dover ricorrere a quanto tradizionalmente si è fatto prima che esistesse PipeWire, utilizzando semplici comandi a terminale.

Abbiamo, innanzi tutto, il comando per verificare come è attualmente configurato PipeWire:

```
pw-metadata -n settings 0
```

Se scriviamo questo comando otteniamo una serie di dati tra cui, in una configurazione di default di PipeWire, i seguenti:

```
clock.rate value 48000
```

```
clock.quantum value 1024
```

Come dire che, di default, PipeWire campiona a 48 kHz (la frequenza ufficiale audio della TV, dei DVD audio, di YouTube, ecc.) con un buffer di 1024 bytes. Pertanto la latenza di default è  $\frac{1024}{48}$ , cioè circa 21 msec., mediamente accettabile senza eccessivo sforzo per la CPU.

Per modificare la latenza possiamo agire su una delle due componenti (frequenza di campionamento o dimensione del buffer) con uno dei seguenti comandi a terminale:

```
pw-metadata -n settings 0 clock.rate <frequenza in Hz>
```

```
pw-metadata -n settings 0 clock.quantum <dimensione in bytes>
```

Per esempio, per la registrazione ottimale di uno strumento musicale su CD Audio potremmo settare il nostro PipeWire così

```
pw-metadata -n settings 0 clock.rate 44100
```

```
pw-metadata -n settings 0 clock.quantum 256
```

ottenendo una latenza di  $\frac{256}{44,1}$ , cioè circa 6 msec.

Sempre ricordando che la qualità del suono che digitalizziamo dipende esclusivamente dalla frequenza di campionamento (con variabilità più che sufficiente tra 44 e 48 kHz) mentre la dimensione del buffer (con variabilità 32, 64, 128, 256, 512, 1024, 2048, 4096 bytes), che è quella che veramente determina la latenza, influisce sulla possibilità di sincronizzare perfettamente suoni provenienti da tracce e origini diverse.

Dimensioni del buffer inferiori a 256 bytes possono comunque rendere instabile il sistema e generare fastidiosi disturbi nell'audio.