

# Slackware Live DAW: Connecting MIDI Devices

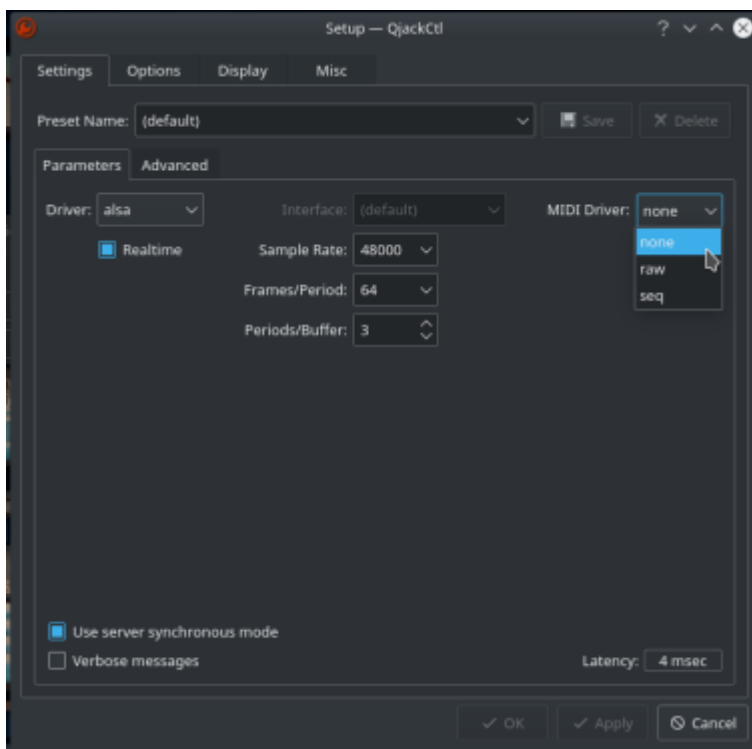
## Introduction

MIDI devices can be connected using a MIDI cable into a MIDI port of an audio interface, or with a USB cable directly to a computer. In either case the MIDI connection will be handled by ALSA so the MIDI connections may not be present in JACK/QJackCtl's connection graph. There are several methods of bridging MIDI from ALSA into JACK.

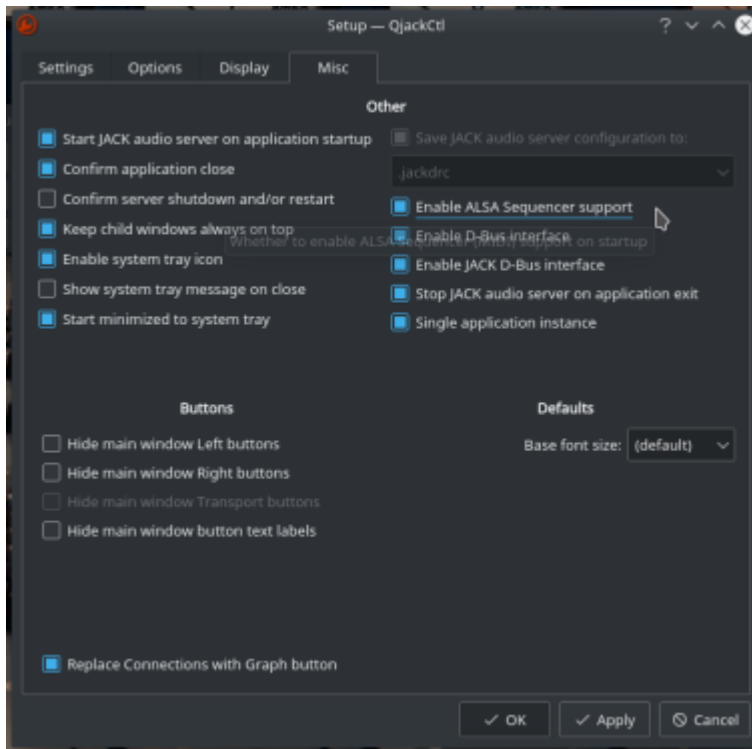
## Bridging ALSA to JACK

The Slackware Live DAW defaults to running a JACK server that is configured with QJackCtl. Bridging ALSA MIDI to JACK can be done within QJackCtl using the built in “raw” or “seq” modes. The other option is to run a2jmidid, which is a daemon that bridges ALSA MIDI connections to JACK.

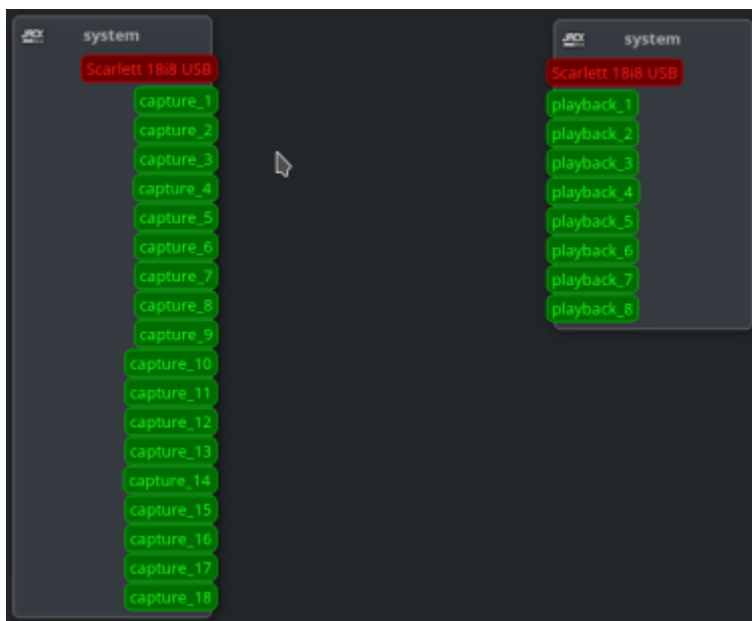
Using raw or seq can be selected from the JACK server settings in QJackCtl:



Note that there is also a checkbox for enabling the ALSA sequencer under the “misc” tab of QJackCtl.

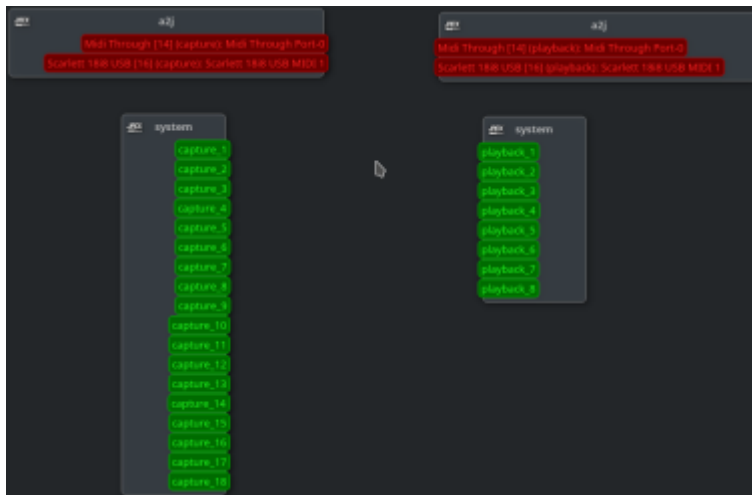


Once the JACK server is started with raw or seq enabled, MIDI ports will be exposed for the detected hardware devices.



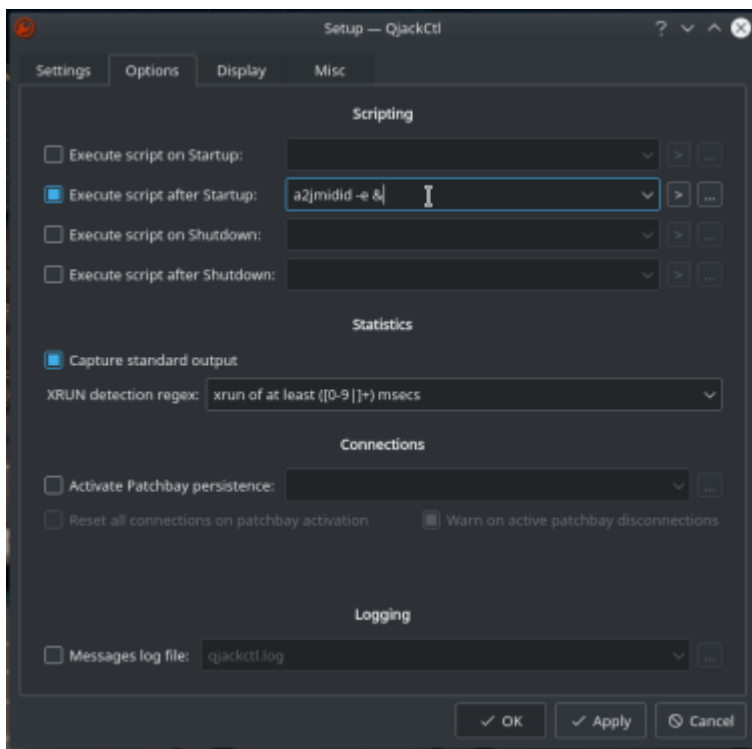
*JACK running in raw mode*

However, the built-in modes of JACK are worse in terms of timing error and jitter (according to the developers of Ardour, see references for details). An alternative with better performance is "a2jmidid" which is a program that runs as a daemon and exposes ALSA MIDI devices to JACK.



a2jmidid Connections

The “a2jmidid” daemon is shipped with the Slackware Live DAW and can be started from the command line with “a2jmidid -e” (the -e option exposes hardware interfaces from ALSA). Alternatively, the a2jmidid daemon can be started automatically when the JACK server is started. This is achieved by adding “a2jmidid -e &” to the startup scripts section of QJackCtl.

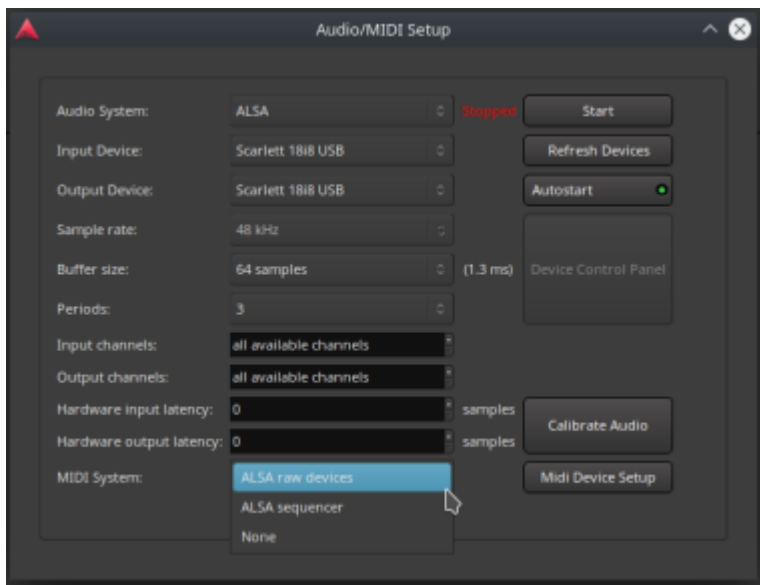


Note that if using the a2jmidid daemon, the MIDI setting in QJackCtl should be set to “none”.

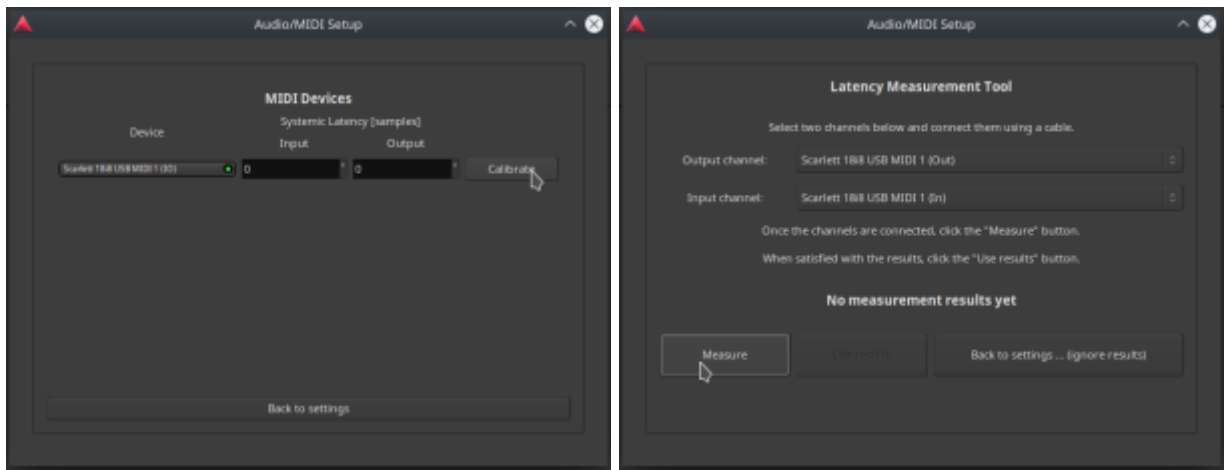
## Compensating MIDI Latency (Ardour Only)

MIDI devices connected through hardware will also incur latency due to hardware delays. This latency can be measured and compensated for when using the built in audio setup menu in Ardour. Presently this type of compensation can't be done for a JACK server.

To compensate latency in Ardour, stop the JACK server from QJackCtl (and exit the program if desired). Then use the “Audio/MIDI Setup” menu of Ardour to set up ALSA directly.



Selecting an ALSA option for the MIDI system will enable the “MIDI Device Setup” option which includes features for measuring and compensating latency in Ardent.



Note that to calibrate MIDI latency, your interface must have a MIDI input and output port. The two ports are bridged with a MIDI cable. Then a MIDI signal can be sent through the hardware ports and looped back, measuring the latency of the signal upon return. Also note that the ALSA audio server must be started to make the MIDI calibration utility available.

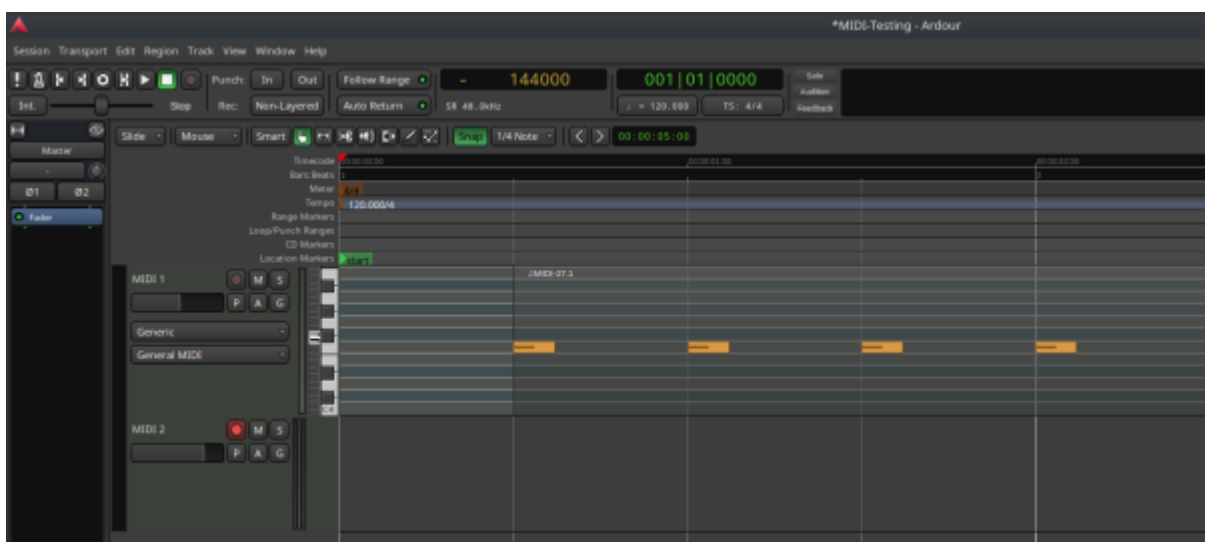


Once latency is measured, the utility will split the measurement into equal parts for input and output port latency, and fill the values into the fields.

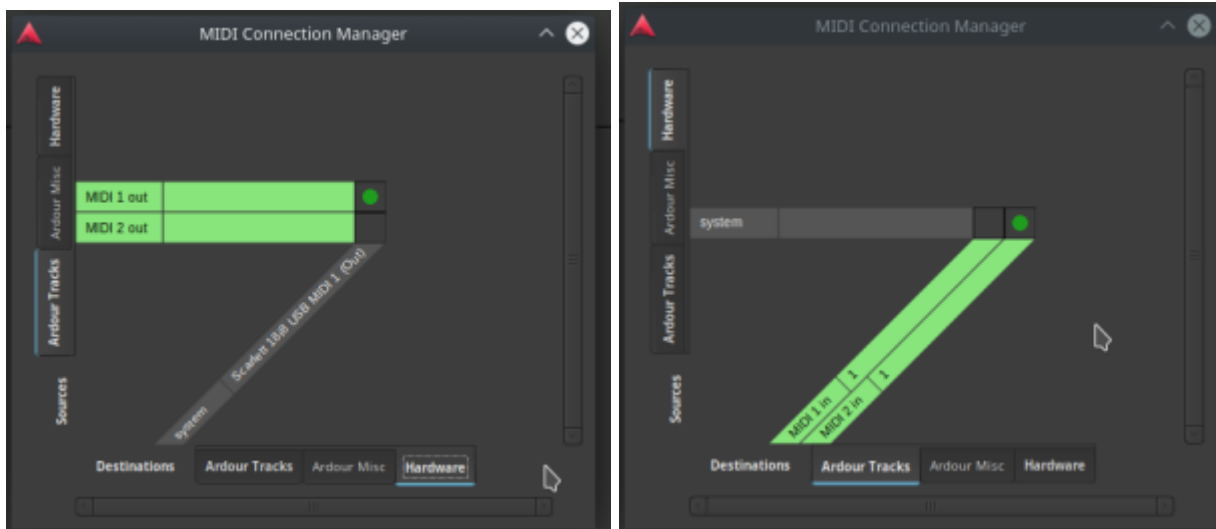
The MIDI latency can also be measured with the program “jack\_midi\_latency” (noted in references) when running a JACK server. However, there is no way to compensate MIDI port latency within JACK at this time.

## A Visual Measurement of MIDI Latency

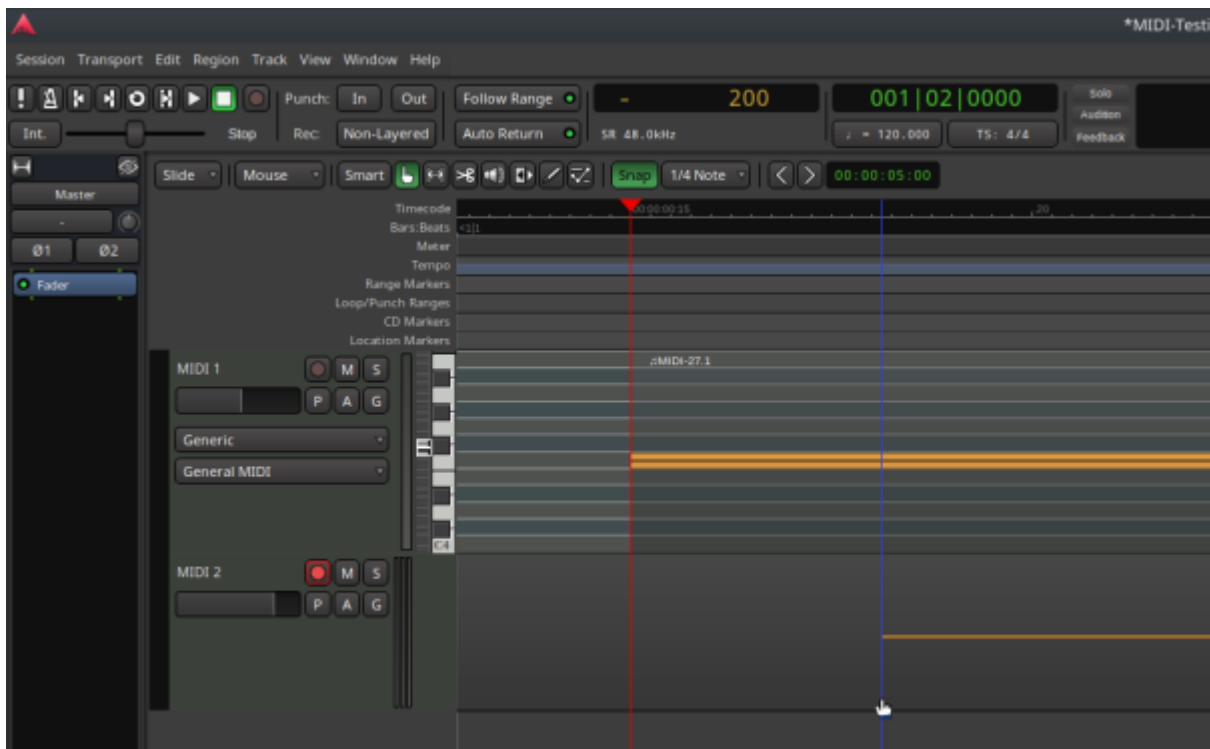
To get an idea of how MIDI latency can affect a MIDI signal, a MIDI track can be routed through the MIDI cable loop and recorded to another track. The original track has some test MIDI notes, put down on the beat markers.



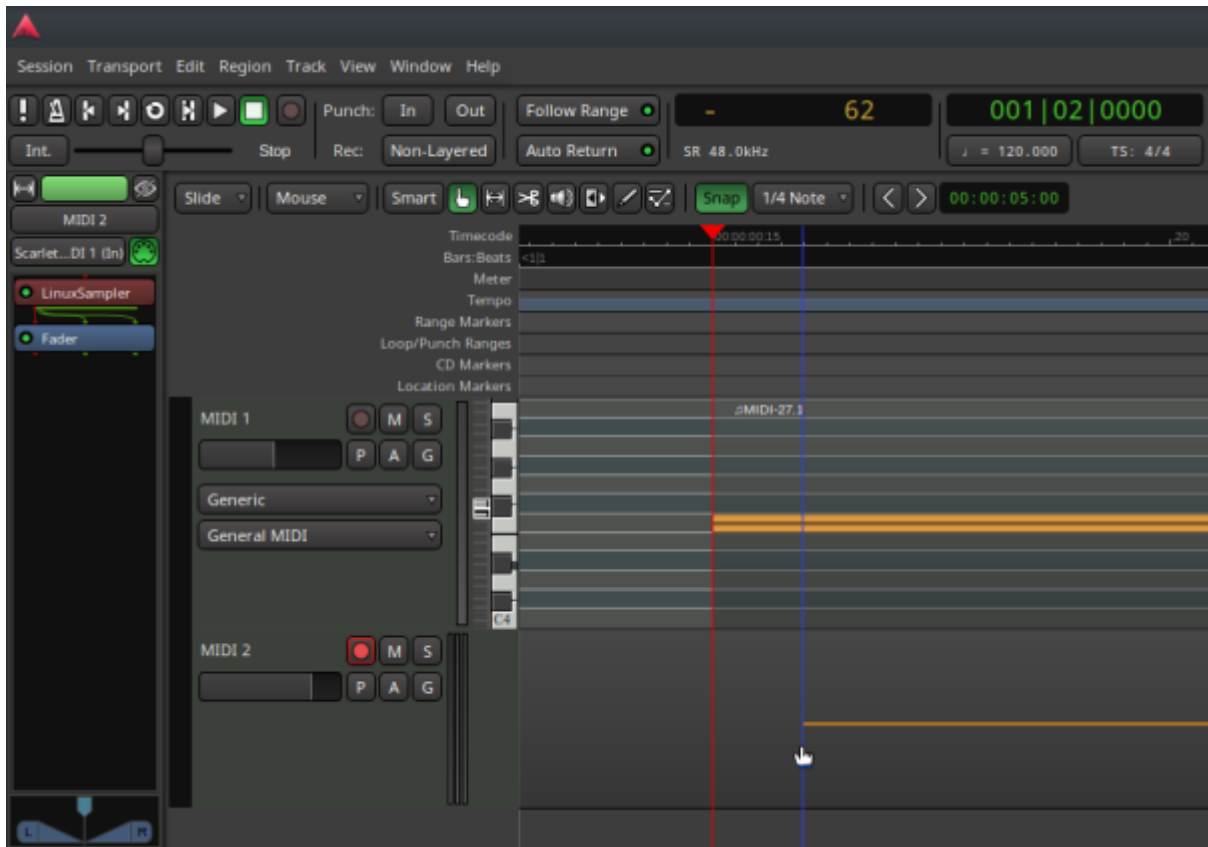
In this example the MIDI 1 track in Ardour is routed to output its MIDI signal through the hardware MIDI output port, around the looped back cable to the MIDI input port, and into MIDI 2. By routing the signal through the hardware ports, their latency will added. Note that this test is only to “see” how much latency is present. In normal circumstances you would route the MIDI for two tracks like this in software MIDI connections, which would add no latency at all.



When no latency compensation is in place, the MIDI notes will be noticeably delayed on the recorded track:



Once the systemic latency of the MIDI hardware is compensated, the MIDI recording will be better aligned with the original track:



Note that the latency pictured includes both output and input port latency. If the MIDI signal is coming from a keyboard to the interface, only the input latency has an effect. This is why the compensation settings are split for input and output latency.

If using Ardour as your primary DAW program, it may be beneficial to use Ardour's audio setup utility with ALSA, instead of QjackCtl running a jackd server. The advantage being the ability to compensate both audio and MIDI latency easily with automated test tools.

## References

<https://manual.ardour.org/setting-up-your-system/setting-up-midi/midi-on-linux/>  
<http://linux-audio.4202.n7.nabble.com/ls-Xalsarawidi-the-same-as-Xraw-or-Xseq-td106370.html>  
<https://discourse.ardour.org/t/midi-latency-measurement/87829>  
[https://github.com/x42/jack\\_midi\\_latency](https://github.com/x42/jack_midi_latency)

## Sources

- Originally written by [Bob Funk](#)

[howtos](#), [daw](#), [multimedia](#), [midi](#), [author](#) [0xbf](#)

From: <https://docs.slackware.com/> - **SlackDocs**

Permanent link: [https://docs.slackware.com/howtos:multimedia:digital\\_audio\\_workstation:connecting\\_midi\\_devices](https://docs.slackware.com/howtos:multimedia:digital_audio_workstation:connecting_midi_devices)

Last update: **2020/09/20 14:41 (UTC)**

