

Multichannel Audio on the Raspberry Pi 5: 8x2-Channel I2S Audio HAT with dedicated Quartz Oscillators

Henning Talal Schaar
Computer Sciences and Languages
Hochschule Anhalt
Köthen, Germany
henning.schaar@hs-anhalt.de

Jörg Schambacher
HIFIBerry Deutschland GmbH
Berlin, Germany
joerg@hifiberry.com

Alexander Carôt
Computer Sciences and Languages
Hochschule Anhalt
Köthen, Germany
alexander.carot@hs-anhalt.de

Stefan Weinzierl
Audio Communication
Technische Universität Berlin
Germany
stefan.weinzierl@tu-berlin.de

Abstract—The Raspberry Pi 5 introduces the RP1 I/O controller with a four channel, clock provider capable I²S interface, replacing the legacy PCM block that forced multichannel audio to rely on time-division multiplexing (TDM) or external USB audio. This work leverages that shift to implement a low-cost, 8x2-Channel (8 playback, 2 capture) GPIO audio HAT using discrete DAC/ADC devices and dedicated quartz oscillators. A pair of ALSA ASoC machine drivers (dedicated oscillators vs. RP1 clock tree) enables direct comparison of clocking strategies in terms of latency and jitter. Outlined here are hardware clock distribution, driver architecture and a measurement methodology covering round-trip latency, stability, inter-channel skew, and clock quality. Results show stable operation up to 192 kHz with round-trip latency as low as 0.97 ms at 192 kHz (128 frames buffer size) and inter-channel skew <1 sample. Dedicated oscillators show small improvement on jitter performance. This platform lowers the barrier to affordable, low-latency multichannel prototyping and provides an open reference for upstream Linux audio development.

Index Terms—Raspberry Pi, Multichannel audio, I2S, Low-latency audio, ALSA ASoC.

I. INTRODUCTION

Small form factor, low-cost, multichannel audio solutions are of interest for many applications. These include signal processing for the purposes of communication [1], musical performance [2] and playback [3] as well as networked music performance [4] to name but a few. For all of these use cases real-time capabilities are advantageous or required. Real-time audio depends on the choice of appropriate DSP algorithms and other software optimizations but having a low latency audio device is a prerequisite. Inter-IC Sound (I²S) [5] sound cards like the one proposed in this paper can generally achieve lower latencies than USB sound cards [6].

Single Board Computers (SBCs) like the Raspberry Pi have gained popularity in a variety of fields, including product

development, academic research and the DIY community. However, the capacity for multichannel audio over general-purpose input/output pins (GPIO) was previously constrained due to the I²S block incorporated in the systems-on-chip (SoCs) utilized by all Raspberry Pi SBCs up to the Raspberry Pi 5. This limitation enabled only two native channels of audio input and output [7].

The Raspberry Pi 5 is equipped with the RP1 hardware controller, which includes an I²S interface that supports up to eight stereo audio data lines [8]. This configuration enables eight channels of audio input and eight channels of output to be supported natively for GPIO sound cards.

Understanding the design parameters for multichannel GPIO sound cards, particularly clocking schemes and their effect on perceptual and performance metrics, helps place the RP1's new capabilities in the context of the IoMusT. [9]

The Audio Injector Octo is an example of a commercially available product that worked around the limitations of the earlier SBCs [10]. It allowed multichannel capabilities via I²S through time division multiplexing (TDM). With this approach a higher I²S continuous serial clock (SCK) frequency is needed for achieving the same audio frame rates. This restricts the use of higher frame rates and thus limits low-latency operation.

For the Raspberry Pi 5, HIFIBerry Deutschland GmbH has already developed an eight channel I²S card, that operates without the use of a specific machine driver [11]. That card as well as other SBC audio HATs need more specialized control than a `simple-audio-card.c` [12] driver would grant but still are rather simple in their setup. To avoid having to write a specialized machine driver for each of these audio HATs even though, most of the code would be the same, the Raspberry Pi Linux kernel fork introduced the `rpi-
→ simple-soundcard.c`, where many different simpler

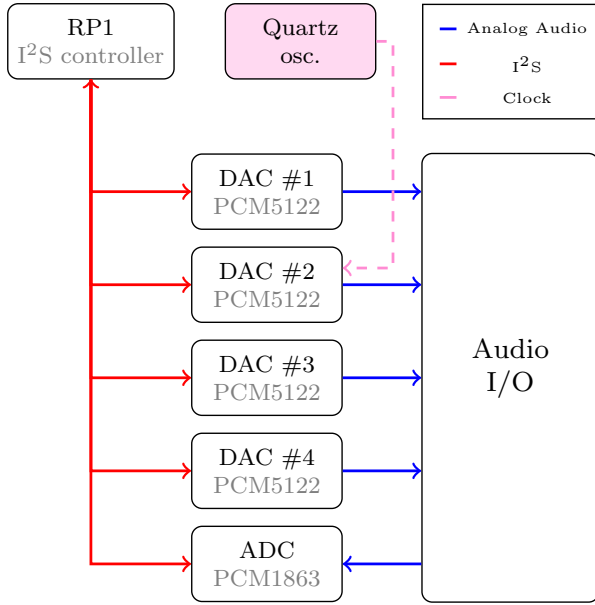


Figure 1: Simplified hardware layout for the proposed sound card with 5 codecs and quartz oscillator unit and the connection to the Raspberry Pi 5 RPi hardware controller. The connection from the quartz oscillators used for clock generation is optional and can be configured with a hardware jumper for comparative evaluation.

HAT drivers are collected in one driver to avoid duplication of code [13]. The proposed device, however warrants the development of a specialized ASoC machine class driver since more specialized clock configuration is required (see Sec. II). ASoC is the Linux kernel framework that aims to enable better ALSA support for SoC processors. It separates Platform Class, Codec Class and Machine Class Drivers [14].

Like the proposed card, many of the sound cards that are available for SBCs are equipped with dedicated quartz oscillators to provide clocking for DACs and ADCs for the purpose of improved audio performance.

The RPi features an internal quartz oscillator and phase locked loop circuits (PLLs) which can be used for Clock generation [8]. The I²S interface consumes this higher frequency system clock (referred to as MCLK) and can then be used as clock provider, supplying SCK and word select (WS). Two versions of the ASoC driver were written and a configuration jumper was included in the hardware design to enable switching between the two clocking schemes.

- **Clock consumer driver** where the RPi clock tree is used and the sound card is the clock consumer (see Fig 4).
- **Clock provider driver** where the dedicated quartz oscillators and the clock dividers of one PCM5122 are used and the sound card is the I²S clock provider (see Fig. 5).

The metric that could differ between these clocking schemes that is relevant for audio performance is clock jitter which has an effect on signal to noise ratio (SNR) as well as distortion

artifacts (see Sec. I-A). Whether dedicated clocking could be advantageous is investigated by comparing these two clocking strategies.

This work builds on previous efforts and develops a multichannel machine class driver, making it possible to use dedicated quartz oscillators for clocking of a multi-codec device and builds a platform for additional features that use the full capabilities of the codecs. A second machine class driver that uses the RPi's clock tree is developed and the performance of the sound card is evaluated with both drivers in terms of clock jitter effects and inter-channel skew. In addition, the proposed card is evaluated in terms of round-trip latency as compared to a USB multichannel sound card.

A. Clock Jitter

Clock sources can differ in quality. The quality of a clock for audio devices is defined by its jitter. Clock jitter is the deviation from periodicity of a clock signal that is presumed to be periodic. For a sampled signal, clock jitter means that the incoming voltage is not sampled at the presumed time and will therefore not sample the expected voltage as seen in Fig. 2 [15].

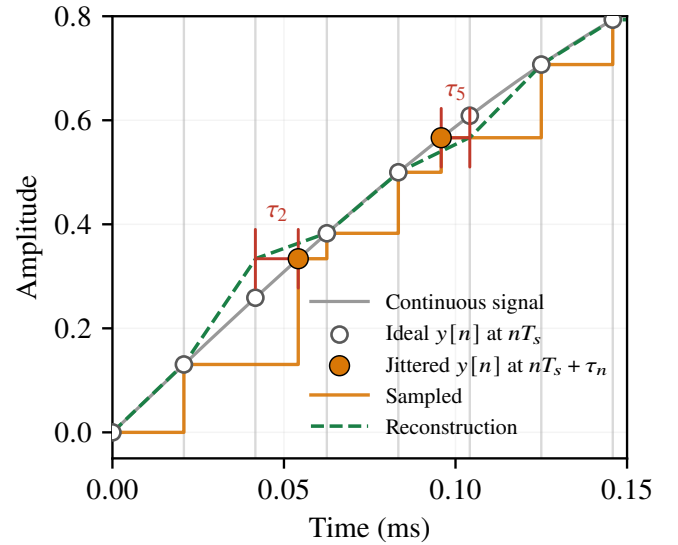


Figure 2: Visualization of the sampling error introduced by clock jitter τ_n .

In general jitter causes the signal to noise ratio (SNR) to decrease (see eq. 1), because the sampled voltage differs from the continuous voltage depending on the slope of the signal and the amount of timing error [16, p. 331].

$$SNR = \frac{P_{sig}}{P_{noise}} \approx \frac{2}{(2\pi f_g \sigma_\tau)^2} \quad (1)$$

where P is signal power, f_g is the limit frequency of a band-limited signal and σ_τ is the standard deviation of the timing error (jitter).

The effect of periodic jitter resembling a sine wave in particular is equivalent to frequency modulation, introducing

sidebands at the signal frequency at distances corresponding to the jitter frequency. (see eq. 2) [17].

$$y(t) = A \sin(2\pi f_c t + I \sin(2\pi f_m t)) \quad (2)$$

where f_c is the signals frequency, f_m is the modulation frequency, and I is the modulation index.

$$I = \frac{d}{f_m} \quad (3)$$

where d is the maximum frequency deviation. This results in spectral sidebands at $f_c \pm n f_m$ for integer n .

Whether clock jitter is audible depends on both the reproduced signal and the type of jitter. If jitter is periodic it will generally become audible at lower σ_τ [18], but jitter signals typically have both random and periodic signal components.

II. HARDWARE AND SOFTWARE DESIGN

The proposed sound card was devised as a proof of concept for future sound card designs. Both the hardware and driver architecture were designed for this purpose and to investigate the performance of the RP1's audio clock tree capabilities.

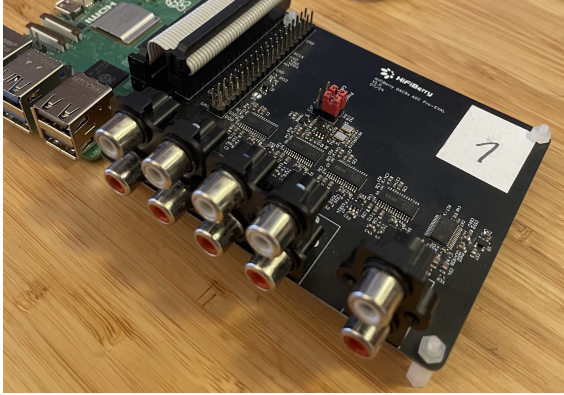


Figure 3: The proposed audio HAT prototype attached to a Raspberry Pi 5.

A. Hardware

The hardware was designed by HIFIBerry Deutschland GmbH (see Fig. 1). The PCB of the proposed sound card is equipped with four PCM5122 DAC chips as well as one PCM1863 ADC from Texas Instruments which are controlled via I²C. Additionally, there are two quartz oscillators (22.5792 MHz / 24.576 MHz) which provide master clocking for the 44.1 kHz and 48 kHz rate families respectively. Rate family is selected by enabling the corresponding clock with the GPIO functionality that the PCM5122 provides. A clock source select jumper (dedicated oscillators vs RP1 clock tree) enables A/B evaluation.

B. Driver Architecture

The machine driver binds one RP1 DesignWare I²S controller with the five codecs. This way, four stereo playback digital audio interfaces (DAIs) are presented as an eight-channel ALSA playback stream while capture exposes 2 channels. In

the clock provider setup one DAC receives the quartz clock input and is bus clock provider (SCK/WS) for all other codecs and the I²S block, which are I²S clock consumers. In clock consumer mode the RP1's clock tree supplies SCK/WS and all codecs are consumers.

The driver consists of a device tree overlay and a machine class ASoC driver. The device tree overlay defines which GPIO pins are used, adds the clocks to the device tree, sets up the codecs and adds the card as a node. It also links to the corresponding driver, enabling the driver's probe function to be called. This function determines whether the device is present at boot time. The ASoC machine driver then defines this probe function, DAI links, stream operations, additional dynamic audio power management controls (DAPM) and usage restrictions¹.

Preliminary testing revealed high inter-channel skew in one DAC of the provider configuration, due to its role as an MCLK consumer, which differs from the other DACs that consume the SCK and WS it provides. To mitigate this, the PCM5122 synchronization register is used to synchronize all codecs via I²C during stream initialisation.

C. Clock tree

The RP1 hardware controller has a fractional-N PLL for audio use that receives a 50 MHz crystal input and generates a 1.536 GHz output. This is then divided into a 12.288 MHz I²S master clock which is then further divided with integer relationships into the 48 kHz rate family for SCK and WS signals. From the DesignWare DW_apb_i2s databook [19] and the RP1 technical documentation [8] it is unclear, how 44.1 kHz family rates are generated, either the Audio PLL generates a different frequency output which is then divided as for the 48 kHz family or there is a second undocumented fractional-N PLL. The DW_apb_i2s WS and SCK signals are then received by all codecs and the SCK signal is used with a fractional-N PLL and dividers to generate all other clocks, like the high frequency clocks for the sigma delta modulator (see Fig. 4).

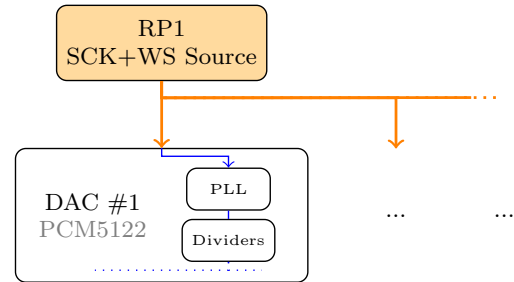


Figure 4: Clock tree for clock consumer mode with RP1 clock generation. All codecs (DACs and ADC) generate internal clocks for sigma delta modulator, filters and other components with a fractional-N PLL and dividers from SCK.

¹Details in ASoC documentation at <https://www.kernel.org/doc/html/v5.0/sound/soc/index.html>

In provider configuration, PLLs are only used with integer relationships by having two crystal oscillators that provide 22.5792 MHz for 44.1 kHz rate family and 24.576 MHz for 48 kHz rate family. This avoids potential additional fractional spurs when using fractional relationships present in the RP1 clock tree [20, p. 196]. The provider mode driver configures one DACs PLL to be disabled and to use dividers to generate all other clocks instead (see Fig. 5). The WS and SCK are then used by the other codecs to create all their other needed clocks with a PLL with integer relationships.

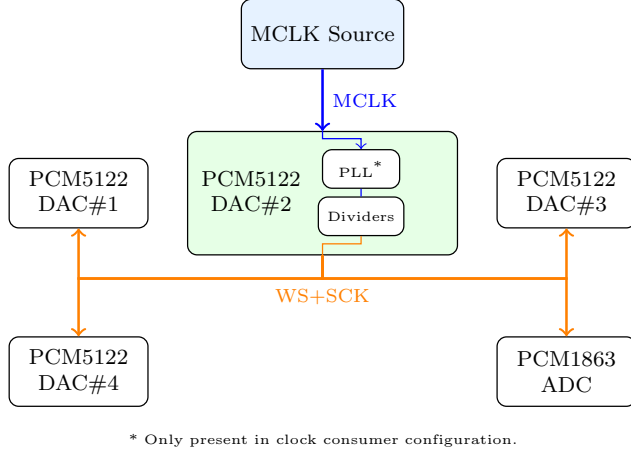


Figure 5: Clock tree for clock provider mode with dedicated oscillators. One DAC (DAC #2) receives crystal oscillator inputs and generates SCK/WS clocks for all other devices which use SCK with integer dividers to create all other clocks.

III. EVALUATION

A. Latency Measurement

To examine how the performance of the I²S interface on the RP1 performs in this example implementation and how well it is suited for multichannel real-time use the round-trip latency of the HAT is measured. Round-trip latency can be measured on device using software like `jack_iodelay` [21], or using an external device to measure the latency of software loop-back.

```
alsaloop -C hw:2,0 -P hw:2,0 -f S32_LE
  ↪ -c 8 -S 0 -r [freq] -l [latency]
```

with `[freq]` being the sampling frequency and `[latency]` the buffer size.

A Fireface UFX II was used at 192 kHz to create a square pulse. An oscilloscope was used to measure directly from the Fireface's analog output which was connected to the HAT. At the same time the output from the HAT was recorded and rising edges of both signals were compared for different period sizes and frame rates. The delay measurement of the Rohde & Schwarz RTB2004 Oscilloscope was used in conjunction with its measurement statistics. The mean of 10 latency measurements was taken and it was assured that the

standard deviation was smaller than 5 μ s which is $>1\%$ of the smallest measured latency.

B. XRUN Measurement

Increasing the frame rate of the system is not the only way to achieve lower latency. Reducing the size of the buffer will also decrease latency.

When choosing small buffer sizes, there is, however, a trade-off where smaller buffer sizes make the system more susceptible to XRUNs. XRUN is the term that is used to describe both buffer under- and overruns. Underruns happen when the application cannot supply bit stream data as quickly as it is read by the audio hardware. Overruns happen when the application cannot process the bit stream data as quickly as it is written by the audio hardware. Some amount of XRUNs may be acceptable in certain use cases, where a low amount of clicks or dropouts are an acceptable trade-off to achieve lower latency. For this reason and to check general stability an XRUN measurement was made.

Although commonly used tools in the linux audio community like `xruncounter.c` [22] exist to evaluate performance via increasing CPU load, preliminary research showed no repeatable results for the tested sound cards. For this reason and to improve validity a minimal example use case was devised. The digital audio workstation software REAPER was used with a single track that took stereo input from the ADC. An algorithmic reverb included in the software (*ReaVerbate*) was inserted as a representative minimal DSP load and the signal was output to all 8 channels. Repeating square pulses and pink noise were sent to the ADC and the output was monitored via headphones and oscilloscope to detect any obvious faults in reproduction. The XRUN counter in `jackd` [23] was used and each latency and frame rate were tested for 5 minutes. Three acceptability criteria were defined: no XRUNs in the test are defined as stable operation, one or less XRUNs per minute were considered acceptable for non critical use, while more than one XRUN per minute or no output was considered unacceptable for all use cases. `jackd` was configured with the standard priority for real time use of -86.

C. Jitter Measurement

Jitter was not measured directly but its effects on the output signal were measured instead. This decision was made due to several factors. Delta sigma modulators as the ones found in the codecs used, have varying tolerances to clock jitter [24] and there are other factors that can influence the final SNR and potential sidebands, including PCB design, power source and coupling, potentially making differences in jitter irrelevant.

The effects of jitter are higher on high frequency signals, since the difference in sampled voltage is higher for the same deviation in sampling time. Taking the derivative of the sine

it is evident that the maximum slope is proportional to f (see eq. 4).

$$\begin{aligned} f(t) &= \sin(2\pi ft) \\ \frac{df(t)}{dt} &= 2\pi f \cos(2\pi ft) \end{aligned} \quad (4)$$

Using a sine wave exactly at nyquist ($\frac{f_s}{2}$) has the added advantage that all harmonic overtones will not be in the analysed signal. This is desirable since the sidebands caused by jitter are very improbable to be harmonics, and harmonic distortion can occur for many other reasons that should not be evaluated in this measurement. By choosing a sine at nyquist, even if the anti-aliasing filter would not be adequate, the harmonics would be aliased down to either DC for odd numbered harmonics or nyquist for even numbered harmonics, and only non harmonic sidebands would remain.

Based on this a measuring scheme was devised to measure the effects of jitter by measuring the output signal for a sine at nyquist and -1 dBFS peak.

1) *General jitter effects measurement:* To evaluate the effects of jitter, the output signal of a DAC, was recorded at 192 kHz 24 bit with an audio interface that has a higher SNR than the PCM5122 [25]. The measurement setup was validated by measuring EIAJ Dynamic Range of the PCM5122 as specified by the manufacturer [26] and comparing it to the data sheet [27].

The nyquist-sine was played for one minute and the recorded signal was then processed with a fast Fourier transform (FFT) and the bin with the sine and everything above it was discarded. Blackman Harris window FFT was used and coherent power gain (CPG), processing gain and equivalent noise bandwidth errors (ENBW) were corrected for [28]. To minimize leakage FFT sizes $> 2^{22}$ were chosen. Output frequency accuracy was measured with a digital oscilloscope at 2.5 GSa/s to ensure accurate frequency reproduction. Scaling losses were then not taken into account because when measuring the same frequency in both cases these losses will be identical, and the effect on measured SNR is too. The power of the resulting signal was measured compared to the wanted signal (dBc). [29]

2) *Periodic jitter effects measurement:* It makes sense to measure sidebands independently, even though they are already included in the SNR measurement, since a sideband can be audible even if it affects SNR only slightly.

To determine whether sidebands are caused by jitter, peaks in the spectrum that are significantly above the average band power (20 dB) are recorded only analyzing peaks that are unique to one setup (± 1 Hz). This ensures that spur sources that are present throughout all measurements but vary in level systematically through the difference in traces that are part of the circuit between the two setups do not influence the measurement [30]. The remaining peaks are summed linearly and the resulting value is compared to the test signal (dBc). An unpaired Welch's t-test is used to compare the mean of the summed spur power for $n = 8$ measurements for each setup. An unpaired test was chosen because there is no way to ensure

that EMI does not vary between any two measurements, since they are linked to random events such as other devices in the environment. FFT-related corrections were omitted here since only relative differences between peaks are of interest.

D. Skew measurement

To evaluate whether there are differences in inter-channel skew when comparing the clocking schemes, the rising edges of a square pulse were compared. Potential differences in inter-channel skew would be expected to be between codecs and not within codecs, since in provider mode one DAC is using a different clocking scheme in which all clocks needed by the codec are created through division of MCLK as opposed to the other three DACs which create all clocks using the SCK and the proposed solution for synchronization needs to be assessed (See Sec. II-B). Since the oscilloscope used for the measurements (Rohde & Schwarz RTB2004) has 4 available channels, the left channel for each DAC was recorded. In a second step the skew within one DAC was measured to test the previous assumption.

The time interval for each codec in relation to DAC1 was taken and the mean absolute was recorded. This was repeated ten times and the mean of the repetitions was calculated. For max absolute skew, the biggest time interval of any two channels for all three repetitions was recorded. Max absolute skew is informative because differences in for example resulting spectrum when adding coherent signals through comb filtering would be defined by this metric. For confidence intervals one sample t-test was used.

IV. RESULTS

The results comparing latency performance with the reference USB sound card are presented followed by the results of comparisons between clocking schemes for inter-channel skew and jitter effects.

A. Round-trip latency and stability

Table I: Measured round-trip latency (ms), Provider mode HAT

Rate (Hz)	Buffer Size			
	32	64	128	256
192k	0.69**	0.63*	0.97	1.63
96k	1.38	1.21	1.89	3.22
88.2k	1.49	1.33	2.05	3.50
48k	2.64	2.39	3.72	6.38
44.1k	2.65	2.60	4.05	6.93

* XRUNs/minute ≤ 1 : acceptable for non critical use

** XRUNs/minute > 1 or no output in stress test: unacceptable

For the proposed HAT, round-trip latency is between 0.63 ms for 192 kHz and a buffer size of 64 frames and 6.93 ms for 44.1 kHz and a buffer size of 256 frames. The lowest round-trip latency that is deemed usable for some applications with few XRUNs is 0.63 ms and the lowest round-trip latency that passed stability testing is 0.97 ms (see Table I). For the

reference USB sound card (*Fireface UFX II*) round-trip latency is between 4.71 ms for a buffer size of 128 frames at 96 kHz and 7.27 ms for a buffer size of 256 frames at 44.1 kHz. The lowest round-trip latency configuration that passed stability testing was 256 frames of buffer at 88.2 kHz frame rate at 5.14 ms (see Table II).

There were no significant differences between provider and consumer drivers ($\Delta = 0.00$ ms, 95 % CI: $[-0.050 - 0.046$ ms]).

Table II: Measured round-trip latency (ms) USB Sound Card

Rate (Hz)	Buffer Size			
	32	64	128	256
192k	—	—	—	5.83**
96k	—	—	4.71*	6.05*
88.2k	—	—	5.14**	5.14
48k	—	5.33**	6.65*	6.65
44.1k	—	5.82	6.69	7.27

* XRUNs/minute ≤ 1 : acceptable for non critical use

** XRUNs/minute > 1 or no output in stress test: unacceptable

— Measurement not possible due to instability

B. General jitter effects

General jitter effects measurements gave higher SNR values in provider setup for all comparisons. For DAC1 at 44.1 kHz and 48 kHz by 0.05 dB and 0.04 dB respectively. For DAC2 they differ by 0.06 dB and 0.05 dB. Measurements between DACs were 0.2 dB lower for DAC1 compared to DAC2 at 44.1 kHz in provider mode and 0.2 dB in consumer mode. For 48 kHz the difference was 0.2 dB for both setups as well.

C. Periodic jitter effects

Mean difference of periodic jitter effects measurements at 44.1 kHz and 48 kHz were significant ($p < 0.05$) for comparisons of combined spur power between setups for both DAC types (SCK or MCLK as clock source in provider setup), with the provider setup showing less combined spur power. SCK consumer (DAC1) at 44.1 kHz with a mean difference of 4.34 dB (95 % CI [3.7064–4.9767 dB]) and a mean difference of 3.65 dB for 48 kHz (95 % CI [2.5172–4.7748 dB]). MCLK consumer (DAC2) at 44.1 kHz with a mean difference of 3.82 dB (95 % CI [2.8510–4.7920 dB]) and a mean difference of 3.96 dB for 48 kHz (95 % CI [2.6646–5.2528 dB]). Mean differences of combined spur power at 44.1 kHz and 48 kHz comparing SCK and MCLK consumer DACs (DAC1 and DAC2) within one setup were not significant for either setup.

D. Inter-channel skew

Measurement of inter-codec skew — absolute mean skew in reference to DAC1 and absolute maximum skew between any two channels — shows sub-sample skew for all available frame rates for both setups (see fig 6). Maximum absolute skew of 0.93 samples (5.28 μ s) was observed at 176.4 kHz in provider setup. Skew for the provider setup with dedicated

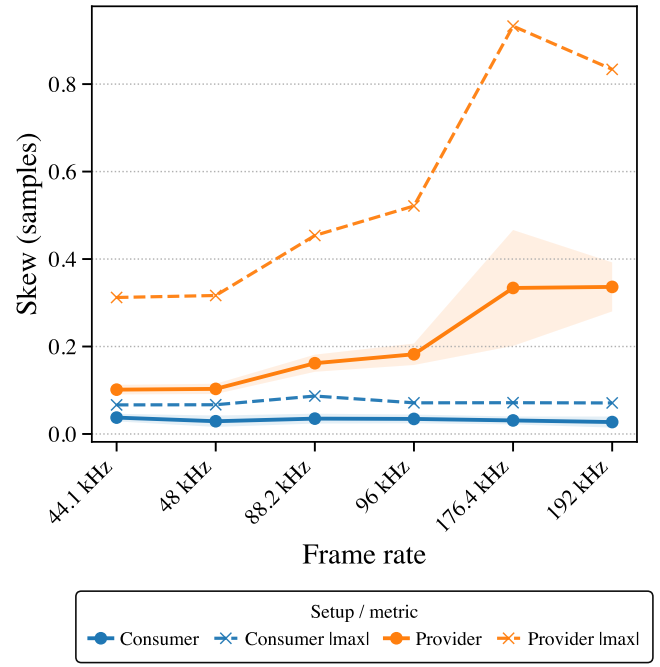


Figure 6: Mean absolute skew between codecs in samples difference to reference DAC1 with shaded area representing 95 % CI for $n=10$. Dotted line represents maximum absolute skew measured between any two codecs.

quartz clocking is higher both in maximum and mean for all frame rates. Mean skew within one DAC between the two channels is 8–12 ns (0.002 samples at 192 kHz $n=30$ for each DAC).

V. DISCUSSION

The measured 4.17 ms improvement in round-trip latency (see Sec. IV-A) for stable configurations when comparing the proposed I²S HAT in provider mode to a typical USB sound card is valuable for real-time applications.

In a networked music performance situation the ensemble delay acceptance limit (EDAL) is highly dependent on the musicians in the ensemble, but a realistic example is 25 ms [4]. The 4.17 ms latency improvements compared with a multichannel USB sound card mean that the geographic distance with a theoretical perfect fibre connection ($v \approx c \cdot 0.7$) would increase by roughly 850 km [31] when assuming perfectly straight fibres and perfect QoS.

For musicians who use FX processing, which is so integral to their playing that it could be understood as part of their instrument, a 4.17 ms improvement in a signal chain that has > 14 ms of round-trip delay (RTD) might also improve the perceived quality of their instrument [32].

The measured differences in SNR between clocking schemes are one order of magnitude smaller than the differences between DACs. Since these differences between DACs do not change with clocking scheme they also do not appear to be related to a difference in operation mode (MCLK

consumer or SCK consumer). They might stem from circuit design or variations in part quality. This suggests that the difference in SNR at nyquist reproduction between clocking schemes is negligible compared to other factors influencing SNR in typical audio applications. In addition to this, all of the described differences in the SNR for a sine at nyquist were below 1 dB for all measured signals. This makes any differences unlikely to be audible if it is assumed to be a difference that is equivalent to band limited noise [33, p. 194] and that the observed difference should be the same or lower for other signals that contain less high frequency information.

The results for periodic jitter show that there is a systematic difference in spurs between the two setups which implies a difference in periodic jitter between the two clocking schemes.

To quantify the worst-case jitter implied by the measured spur differences, it can be considered that each spur at frequency offset f_m from the carrier represents sinusoidal clock jitter at that frequency. For small modulation index (eq. 2) the first order Bessel approximation $J_1(I) \approx \frac{I}{2}$ [34, p. 185] can be used to relate spur power to rms jitter for a single first order spur. A spur at level L_{spur} (dBc) corresponds to RMS timing jitter of

$$\sigma_{rms} = \frac{r \cdot f_m}{\pi f_c \sqrt{2}} \quad (5)$$

where $r = 10^{L_{spur}/20}$ and f_c is the frequency of the test signal tone. In the extreme case where the entire 4.34 dB difference originated from a single spur, the RMS jitter difference would scale with the spur frequency—ranging from 20 ps for low-frequency jitter components (20 Hz) to 22 ns for higher frequencies (20 kHz). Since measurements show a median of 16 spurs distributed across the spectrum, the per-spur jitter contribution is correspondingly smaller. This worst-case analysis places any differences at the lower end of the 10–300 ns perceptibility thresholds reported by Benjamin and Ganon [18], indicating that the observed differences might be audible for some listeners with high frequency sine playback, but are unlikely to be audible in typical audio applications.

Inter-channel skew measurements show that the consumer configuration has low skew across configurations, suitable for most multichannel applications. The provider, however, has a higher mean and maximum skew value. The maximum skew is close to 1 sample at 176.4 kHz and 192 kHz. For coherent summation of signals of channels at 20 kHz the maximum skew of 5.28 μ s using equation 6 [35] shows a maximum reduction due to comb filtering of –0.49 dB which is suitable for most audio applications.

$$\bar{I} \propto p_{total}^2 = 2p^2 \left[\cos \frac{\varphi}{2} \right]^2 \quad (6)$$

For applications that rely on timing coherence between channels it could potentially pose a challenge. For example, a simulation with maximum observed jitter values up to 5 μ s (see Appendix) using a reference beam forming implementation [36] shows that for a linear array of 8 speakers with 2 cm spacing reproducing a 10 kHz tone the main lobe steered towards 45° shifts by approximately 3°.

VI. LIMITATIONS & FUTURE WORK

A comparison in performance with the Audio Injector Octo [10] could not be carried out since it is no longer commercially available. Since it is limited to 96 kHz, an improvement of minimal viable round-trip latency around 0.5 ms seems plausible based on the observed difference between 96 kHz and 192 kHz frame rates in the proposed HAT, but this remains speculative.

The differences in measured combined spur power between setups are systematic, but they might not be fully explained by periodic jitter. Mitigation steps to reduce the influence of other sources like AC coupling by excluding spurs that are not unique were taken but that does not exclude all potential factors of the chosen measurement method. Part of the observed difference could stem from a difference in the amount of coupling or sensitivity to EMI that is caused by the difference in traces that are part of the circuit in either configuration [30]. Spurs could cross the detection threshold for this reason and the exclusion of non-unique peaks would not be able to mitigate this. To minimize this factor, another series of measurements could be done in an anechoic RF chamber with high shielding. Redesigning the PCB so that the differences in traces between the setups are minimal could reduce potential effects related to trace length.

Despite the fact that jitter effect measurements demonstrate a systematic difference between the two clocking schemes, the measured differences are minimal and are unlikely to be audible in typical audio applications. However, definitive statements regarding audibility would necessitate the conduction of listening tests with a representative group of listeners.

The higher skew in the provider clock configuration is a fundamental limitation of the design, since in provider mode DAC2 is the only codec that creates all needed clocks through division of MCLK, while all other codecs create all clocks using SCK. This creates latency in DAC2 compared to the other codecs in provider mode. Future designs could mitigate this by using other codecs that do not exhibit this behaviour or have the option to make timing adjustments. Multiple codecs that create all clocks through division of MCLK or using an external clock divider could also be potential solutions.

For inter-channel skew and jitter effects DAC performance was analysed exclusively. For inter-channel skew a design with multiple ADC codecs would be needed for equivalent testing. In terms of jitter performance the difference would most likely also appear in ADC performance since it uses the same clocking scheme as DAC1, DAC3 and DAC4 in provider mode or the same as all other codecs in consumer mode. The amount of difference could be smaller or greater since the sensitivity of the sigma delta modulator in the PCM1863 could be different. Based on the measurement scheme for random and periodic jitter effects, a similar scheme could be devised for ADCs in future works.

VII. CONCLUSION

This work presents a proof-of-concept 8x2-channel I²S audio HAT with dedicated clocking for the Raspberry Pi 5

that leverages the RP1's enhanced I²S capabilities to achieve multichannel audio without TDM. The system demonstrates stable operation up to 192 kHz with round-trip latency as low as 0.97 ms (see Sec. IV-A), representing a 4.17 ms improvement over typical USB multichannel interfaces in stable configurations.

Two ASoC machine drivers — one utilizing the RP1's internal clock tree (consumer mode) and another employing dedicated quartz oscillators (provider mode) — enable direct comparison of clocking strategies. Measurements reveal that dedicated oscillators provide small improvements in periodic jitter-related spurs (3.65–4.34 dB reduction in combined spur power, see Sec. IV-C), though these differences remain below typical audibility thresholds for most audio applications. Inter-channel skew remains sub-sample for consumer mode across all frame rates, while provider mode exhibits slightly higher skew approaching 1 sample at 192 kHz (see Fig. 6) which is suitable for most use cases.

The 4.17 ms latency advantage extends the viable geographic range for networked music performance applications by approximately 850 km under ideal network conditions and may improve perceived quality in real-time FX processing chains exceeding 14 ms total RTD. This platform demonstrates that the RP1's I²S interface can support professional multichannel audio applications while maintaining the accessibility and low-cost characteristic of the Raspberry Pi ecosystem.

The open-source driver implementation and documented hardware design provide a reference for future GPIO audio HAT development and contribute to upstream Linux audio support for the RP1 platform.

VIII. ACKNOWLEDGMENTS

I would like to thank HIFiBerry Deutschland GmbH for designing and producing the hardware used in this work as well as providing support during driver development. Especially Jörg Schambacher's expertise was invaluable. I would also like to thank Prof. Dr. Alexander Carôt for his continued support and advice during the development of this work and Emanuele Porcinai for proofreading earlier drafts. Finally, I would like to thank Julia Dietz for her endless support and patience during the writing of this work.

REFERENCES

- [1] F. Drakopoulos, D. Baby, and S. Verhulst, "Real-time audio processing on a raspberry pi using deep neural networks", in *Proceedings of the 23rd International Congress on Acoustics*, Aachen: Deutsche Gesellschaft für Akustik. DOI: 10.18154/RWTH-CONV-239335
- [2] R. S. Pereira and R. V. Andreão, "Electric guitar distortion effects unit using a raspberry pi", in *Simpósio Brasileiro de Computação Musical (SBCM)*, ISSN: 0000-0000, SBC, pp. 121–128. DOI: 10.5753/sbcm.2021.19436
- [3] F. Schlieper, L. Aspöck, and M. Vorländer, "A bin-aural demonstrator on a single board computer", in *Fortschritte der Akustik - DAGA 2016*, Aachen: Deutsche Gesellschaft für Akustik e.V., pp. 826–828, ISBN: 978-3-939296-10-2.
- [4] A. Carôt, "Musical telepresence - a comprehensive analysis towards new cognitive and technical approaches", Ph.D. dissertation, Universität zu Lübeck, Lübeck, Germany.
- [5] Philips Semiconductors, *I²S bus specification*. [Online]. Available: http://www.nxp.com/acrobat_download/various/I2SBUS.pdf
- [6] A. P. McPherson, R. H. Jack, and G. Moro, "Action-sound latency: Are our tools fast enough?", in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Brisbane. [Online]. Available: <http://qmro.qmul.ac.uk/xmlui/handle/123456789/12479>
- [7] F. Meier, *Bcm2835 i2s ALSA SoC driver*, version rpi-6.12.y, commit 9fec535. [Online]. Available: <https://github.com/raspberrypi/linux/blob/9fec535/sound/soc/bcm/bcm2835-i2s.c>
- [8] "RP1 peripherals. register-level documentation for the RP1 i/o controller", Raspberry Pi Ltd, manual. [Online]. Available: <https://datasheets.raspberrypi.com/rp1/rp1-peripherals.pdf>
- [9] L. Turchet et al., "The internet of sounds: Convergent trends, insights, and future directions", *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11 264–11 292, ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2023.3253602 [Online]. Available: <https://ieeexplore.ieee.org/document/10061604/>
- [10] M. Flax. "Audio injector". [Online]. Available: <https://www.audioinjector.net/rpi-octo-hat>
- [11] D. Matuschek. "More than just stereo – the DAC8x — HiFiBerry". [Online]. Available: <https://www.hifiberry.com/blog/more-than-just-stereo-the-dac8x/>
- [12] K. Morimoto, *ASoC: Add generic simple-card support*. [Online]. Available: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/sound/soc/generic/simple-card.c?h=v6.17-rc7&id=f2390880ec0264a0ed26b32c23bc23435b4297da>
- [13] T. Gover, *ASoC: Add generic RPI driver for simple soundcards*. [Online]. Available: <https://github.com/raspberrypi/linux/commit/28c42d5b035b0e99a66bc42aec3d8b3731b5e340>
- [14] L. Torvalds, *ALSA SoC layer overview. advanced linux sound architecture system-on-chip audio*, Version: Linux Kernel 4.10. [Online]. Available: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/Documentation/sound/soc/overview.rst?h=v6.18-rc3&id=dcb6fa37fd7bc9c3d2b066329b0d27dedf8becaa>
- [15] M. Werwein, "A/D- und D/A-Wandler für Audiosignale", in *Handbuch der Audiotechnik*, S. Weinzierl, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, ISBN: 978-3-662-60368-0 978-3-662-60369-7. DOI: 10.1007/978-3-662-60369-7

- [16] F. Puente León, *Messtechnik*. Berlin, Heidelberg: Springer Berlin Heidelberg, ISBN: 978-3-662-44820-5 978-3-662-44821-2. DOI: 10.1007/978-3-662-44821-2
- [17] A. Bartetzki, “Klangsynthese”, in *Handbuch der Audiotechnik*, S. Weinzierl, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, ISBN: 978-3-662-60368-0 978-3-662-60369-7. DOI: 10.1007/978-3-662-60369-7
- [18] E. Benjamin and B. Ganon, “Theoretical and audible effects of jitter on digital audio quality”, in *105th AES Convention*, San Francisco, California: AES.
- [19] Synopsys Inc., “DesignWare DW_apb_i2s databook”, Synopsys Inc., Mountain View, CA, manual, Edition: 1.11a.
- [20] R. E. Best, *Phase locked loops, sixth edition: Design, simulation, and applications*. McGraw Hill LLC, ISBN: 978-0-07-159521-6.
- [21] F. Adriaensen, *Jack_iodelay*. [Online]. Available: <https://github.com/jackaudio/jack-example-tools/tree/main>
- [22] H. Meyer, *Gimmeapill/xruncounter*. [Online]. Available: <https://github.com/Gimmeapill/xruncounter>
- [23] P. Davis et al., *JACK audio connection kit*. [Online]. Available: <https://jackaudio.org/>
- [24] R. Saad, S. Hoyos, and S. Palermo, “Analysis and modeling of clock-jitter effects in delta-sigma modulators”, in *MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1*, V. Katsikis, Ed., InTech, ISBN: 978-953-51-0750-7. DOI: 10.5772/46453
- [25] M. Carstens, “Bedienungsanleitung RME Fireface UFX”. [Online]. Available: https://rme-audio.de/downloads/fface_ufx_d.pdf
- [26] L. Gaddy and H. Kawai, “Dynamic performance testing of digital audio d/a converters”, BURR BROWN. [Online]. Available: <https://www.ti.com/lit/an/sbaa055/sbaa055.pdf?ts=1761981879931>
- [27] Texas Instruments, “PCM512x 2.1-VRMS DirectPath™ 106/112/106-dB audio stereo DAC with 32-bit, 384-kHz PCM interface”, Dallas, TX, USA, manual. [Online]. Available: <https://www.ti.com/lit/ds/symlink/pcm5122.pdf>
- [28] F. Harris, “On the use of windows for harmonic analysis with the discrete fourier transform”, *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, ISSN: 0018-9219. DOI: 10.1109/PROC.1978.10837
- [29] S. Scholl, *Exact signal measurements using FFT analysis*, Pages: 10 Type: coursematerial. [Online]. Available: <https://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-42930>
- [30] M. Mehri and N. Masoumi, “Estimation of EMI-induced phase noise and jitter on oscillator/clock signals”, *Electronics Letters*, vol. 53, no. 10, pp. 652–653, ISSN: 1350-911X. DOI: 10.1049/el.2017.0322
- [31] A. Carôt, A. B. Renaud, and P. Rebelo, “Networked music performance: State of the art”, pp. 1–7.
- [32] R. H. Jack, T. Stockman, and A. McPherson, “Effect of latency on performer interaction and subjective quality assessment of a digital musical instrument”, in *Proceedings of the Audio Mostly 2016*, Norrköping Sweden: ACM, pp. 116–123, ISBN: 978-1-4503-4822-5. DOI: 10.1145/2986416.2986428
- [33] E. Zwicker and H. Fastl, *Psychoacoustics* (Springer Series in Information Sciences), red. by T. S. Huang, T. Kohonen, M. R. Schroeder, and H. K. V. Lotsch. Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 22, ISBN: 978-3-540-65063-8 978-3-662-09562-1. DOI: 10.1007/978-3-662-09562-1
- [34] M. Meyer, *Kommunikationstechnik*. Wiesbaden: Vieweg + Teubner Verlag, ISBN: 978-3-322-93951-7 978-3-322-93950-0. DOI: 10.1007/978-3-322-93950-0
- [35] S. Weinzierl, “Akustische Grundlagen der Audiotechnik”, in *Handbuch der Audiotechnik*, S. Weinzierl, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, ISBN: 978-3-662-60368-0 978-3-662-60369-7. DOI: 10.1007/978-3-662-60369-7
- [36] R. Scheibler, E. Bezzam, and I. Dokmanić, “Pyroomacoustics: A python package for audio room simulations and array processing algorithms”, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355. DOI: 10.1109/ICASSP.2018.8461310

APPENDIX

All tests were done with one Raspberry Pi 5B with 4 GB of memory running the Raspberry Pi Linux kernel fork version 6.12.47+rpt-rpi-2712 using the 50 Hz 5 V 5 A Raspberry Pi power supply with EN 50075 plug.

The Oscilloscope used for inter-channel skew and latency measurements was a Rohde & Schwarz RTB2004. The audio interface used for jitter effects measurements was a RME Fireface UFX II with Firmware version 1.63 and recordings were made with Audacity 3.7.5 for macOS version 26.0.1. For round-trip and stability comparison the Fireface UFX II was used as reference USB sound card in class compliant mode on the Raspberry Pi running REAPER version 7.49.

Exact values from 176.4 kHz inter-channel skew measurement used in beam forming simulation were 0 μ s, 1.84 μ s, 3.28 μ s and 5.28 μ s, each value applied sequentially to two channels.

The source code for the driver and device tree overlays as well as scripts used for measurements are available at <https://git.tu-berlin.de/henningschaar/hifiberry-dac8xadcpro.git>.

The plots for Figures 2 and 6 were created using Python code that was generated in part by Microsoft’s GitHub Copilot using Claude Sonnet 4 and Claude Sonnet 4.5 by Anthropic as underlying models. The code was then manually adapted to fit the data and desired plot styles. A tool for parsing CSV files from the oscilloscope and for automatically populating manually created latex tables from the output of analysis scripts was created in a similar manner. The generated code was reviewed and adapted to ensure correctness and suitability for the specific use case. DeepL Write by DeepL was used for

proofreading sections of the text, and output from the software was used to improve grammar and clarity. All generated text was reviewed and adapted to ensure it accurately reflected the intended meaning.