# AVB Workshop and Discussion



## Topics and priorities

1 ) Networked audio devices - AVB

2 ) cross ECU interactions (remote control, including comparison with remote servers interaction)

3 ) Audio effects

## AVB Network Audio Synchronization

### workshop 26/11/2020

- ECUs spread across a network have different clocks (CPU, network, audio,...)
- Clock drift, oscillators accuracy (1/day => 10 us/sec => for 44.1KHz one sample loss each 2 sec)
- Solution run SW on synchronized time
- how to synchronize time ?
- asynchronous resampling needed as the speakers DAC always need to be synchronous to their own clock
- PTP => Wall clock (e.g. +-500 ns to master clock)
  - media clock recovery, second layer (IEEE 1722 frame protocol, presentation time stamp taken as reference)
  - each takes as reference and reproduces a local clock that is in synch with the wall clock
- PTP does not solve jitter problem => how to compensate jitter ? => increase buffer size => increase latency
- Clock recovery. Or how to generate a clock when all we get from ptp is sporadic info about delay status + or -
- real-time accuracy vs latency
- TNS = Sync + Latency + Reliability, ... (SRP Stream Reservation Protocol)
- Qav (IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams)
  - issues of packet loss / drop
  - constraints for each source
  - spread out of stream (traffic shaping)
- What if we cannot synchronize the time ?
  - Mixing of different streaming sources (can't follow multiple clocks)
  - Asynchronous domain crossing => latency
  - Synchronous resampling (format conversion only for already synchronized clocks 48KHz => 32 KHz)
- AVB : Routing only => cable connection for a continuous audio stream => No media control, no start, no pause, no skip,...
  - separate control protocol (someip,... although TNS has some but automotive using someip)
- AVB Software
  - Kernel module (Steve => RENESAS_RCH3M3M3NE3_EAVB_AVBStreaming_UME_v2.00 page 5)
  - (RENESAS_RCH3M3M3NE3_EAVB_LauncherApplication_ANE_v2.00 page 10)
- ALSA to AVB (RENESAS_RCH3M3M3NE3_EAVB_MSE_UME_v2.00 page 4)
- Existing open source AVB linux integrations (HW dependent topic)

#### References

| | | |
|---|---|---|
| **Clock drift** | clock drift : Interaction of systems running on different clocks | • https://en.wikipedia.org/wiki/Clock_drift<br>• ppm : oscillator-design-guide-for-stm8afals-stm32-mcus-and-mpus-stmicroelectronics.pdf |
| **Clock recovery** | PLL vs synchronization (ramping) | • https://en.wikipedia.org/wiki/Clock_recovery<br>• https://en.wikipedia.org/wiki/Clock_skew |
| **Resampling** | signal processing and resampling | • https://en.wikipedia.org/wiki/Sample-rate_conversion<br>• http://ldesoras.free.fr/doc/articles/resampler-en.pdf |

| | | |
|---|---|---|
| **Avnu** | low-latency, time-synchronized, highly reliable synchronized networked devices<br><br>Avnu = AVB + TNS<br><br>AVB = AV + PTP | • https://avnu.org/<br>• https://en.wikipedia.org/wiki/Avnu_Alliance |
| **PTP** | Precision Time Protocol | • https://en.wikipedia.org/wiki/Precision_Time_Protocol (profile 802.1 AS)<br>• as-kbstanton-8021AS-overview-for-dot11aa-1108.pdf<br>• Tutorial on time synchronization for AAA2C based on IEEE Std 802.1AS Kevin-Stanton.pdf<br>• tutorial-Automotive-Ethernet-0717-v02.pdf |
| **Automotive AVB** | Focus on subset of standards (features) | • avnu.org Auto-Ethernet-AVB-Func-Interop-Spec_v1.6.pdf |
| **802.1 TNS** | Time Sensitive Networking | • ieee802.org tsn-farkas-intro-0517-v01.pdf |
| **AVB on linux** | Existing open source integrations | • https://tsn.readthedocs.io/avb.html |
| **AVB Renesas** | Open source AVB integrations | • renesas-rcar/avb-mch, avb-applications, avb-streaming, avb-mse |

## FAQ:

- Q : Is is possible to use AVB over a standard network adapter (without HW support)
  - A : No, not for high fidelity Audio automotive or pro equipment.
  - Although it is possible to apply the same algorithms with as reference a timer for SW timestamp, but that might induce one or more orders of magnitude of precision loss, that results in a different concepts of buffer dimensioning and so on.
- Q : ptp target precision (~ 5 ns)
  - 5 ns ? 5 us ?
    48 KHz => ~ 20 us
    1, 2 samples ? => 50 us
  - produce a sample (3 ms), touch the screen (interrupt) => loop 2ms, 2 ms buffer (100 samples)
- Latency (transmitter + network propagation + receiver)
  - network propagation depends on number of switches e.g. 2 ms latency over 7 hops
  - end to end latency (ADC  net  DAC) hop ~ 250 us
  - 6 samples packets processing (48 KHz) every 125 us
  - buffers go critical down to 1 sample...
  - e.g. 5 us => 125 that means 4% more latency for 5 us precision in stead of 5 ns
- what are the latency requirements and which equipment needs which latency
  - ANC Active Noise Cancelling (< 1ms) (Automotive - should not involve netwrok)
  - musical instrument(end to end ~ 2 ms) (not Automotive- should not involve netwrok)
    - Gaming ?
  - playing smartphone instrument (10 ms) (Automotive infotainment - might involve network)
  - streaming musing (1 s) (Automotive Infotainment - can involve network)
    - Lip sync not problematic because none of the audio/video is real time, you can delay both stream with the same amount => sync 20~40 ms.
    - cold start latency : is a different KPI than the sync delta of video audio => (~ 500 ms).
  - Phase manipulation in case of audio zones
    - problem starts if the different zones are played from different network devices
- Network stability, reduction of jitter
  - => SRP Stream reservation protocol
    AVB over RTP IP without HW
- Renesas example on references, usage of OpenAvenu
  - Avenu as open source but more reliable implementation than other existing ones
- Q : Steps to be taken for starting the SW
- Q : clock recovery algorithm, might recover smoother by catching up

# Brainstorming

## Use cases

1) external Amplifier, controlled by Android only (cross ECU interface used by headunit only)
2) interact of an Android Audio ECU with the rest of the audio system other ECUs (e.g. remote control - e.g. non Android system)
3) interact with the Audio System from a server

## Relations to existing work in other SIG

CVII : Common Vehicle Interface Initiative
VSS : Vehicle Signal specification
VSC : Vehicle Service Catalogue (list of interfaces)

Audio and VSS, data interfaces,...

common model for data (call functions with params, services, interfaces, franca,...)

-no dealing with streaming data, rather simply connection points urls exchange
-can we define similar things to audio e.g. currently played track, volume,... ?
-requests to add media in VSS

-standardized interface for audio/media streaming, renewed interest by Hyundai
- complex question, OEM => standardize / Netflix => App custom standard

- if possible to add info in VSS
- different scope from cross ECU interfaces

-prevent fragmentation by collecting a survey about state of the art
- MPRIS dbus standard

19 Nov 2020

1) Bluetooth : the AVRCP allows to control

2) Project Mode : / Audroid Auto - has virtual key input generator (volume up/down,...)

3) the same infrastructure from Project mode : can be used to inject the same as if it was a projected mode

4) Automotive : specific vehicle HAL service vehicle bus and converts in Android key codes
Android input system

Q: if volume key up/down limited to current volume ?
A: in automotive concept of volume groups which has the focus
Q: If the same interface include the audio settings
A: no intended in that interface
A: done through the audio effects HAL which is manufacturer specific

---

mention as info : MQTT Pub/Sub mechanism

Remote topic split in two :
- play pause,...
- effects, calibration,...

---

Q: What to develop next in the project

- Organization of the Android build
=> scripting the build ongoing, testing,...

Q: interest in AVB networked devices ?
- control / routing
- available AVB in automotive / professional audio
- https://github.com/Avnu/OpenAvnu
- Android (no many echos about AOSP/AVB)
- NXP/Renesas/ : board has AVB support
- Hikey 960 does not support AVB
- HW mandatory, not only low cost but automotive HW
- Android 11 support for Dragonboard

=> TODO organise an AVB introduction next week 26/11/2020
(TSN wider scope : https://en.wikipedia.org/wiki/Time-Sensitive_Networking)

Q: Any close to Android transport protocol or if it's part of the applications
A: usually part of the applications
A: approach interfacing Android system from outside like any Android app would do

=> CVII common standard initiative

# A2B Network Audio

A²B

I²C
I²S

Android A2B/ALSA driver
Android => (I2C,I2S_1,I2S_2) => (A2B devKit) <=> (A2B remote1),(A2B remote2)