Unfolding the mystery of automotive audio applications **for not** using Zephyr and Linux PREEMPT_RT

Khasim Syed Mohammed
khasim@ti.com

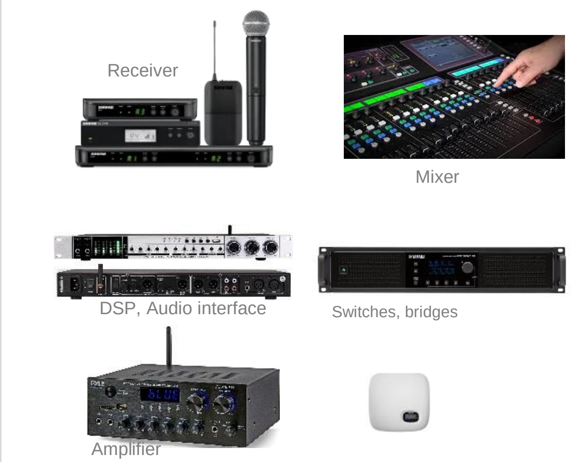
Engineering Lead – Texas Instruments

20 Sept 2024

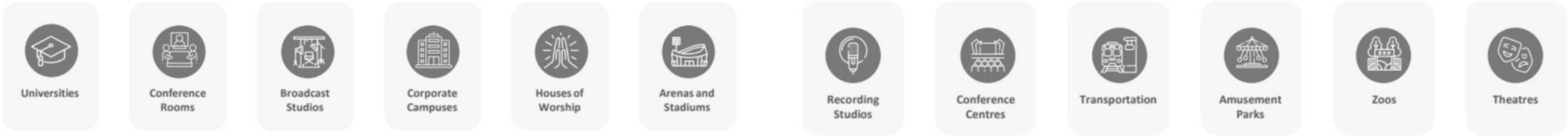
Audio Industry – An overview of what we are talking about ?

Microphones / Speakers

Receivers, DSP, Mixers, Switches, bridges, Amp



Car Audio and Amplifiers

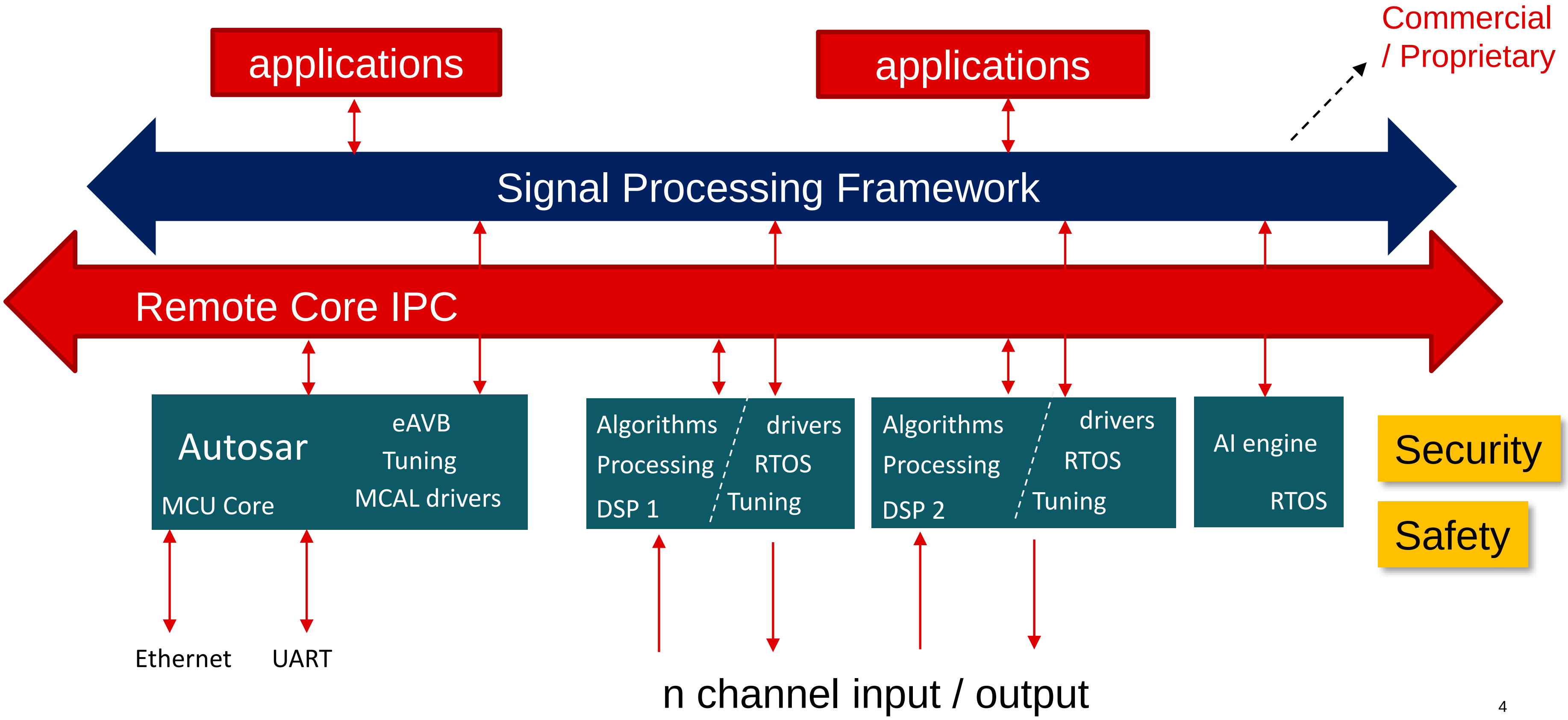


Texas Instruments

Scope of the session

- Share with you the Software stack required for the key audio centric solutions.
- Tools that are required to validate, standardize, configure and benchmark.
- What are latency requirements and other expectations – can RT Linux or Zephyr meet those.
- Can we build the required software stack on Zephyr or Linux.
- What are the safety constraints imposed for qualifying for the safety certifications (if any).
- Security, over the air upgrade, streaming media over the network – do they have to be proprietary ?.
- How do we make the software scalable to cater to wide range of audio equipment ?

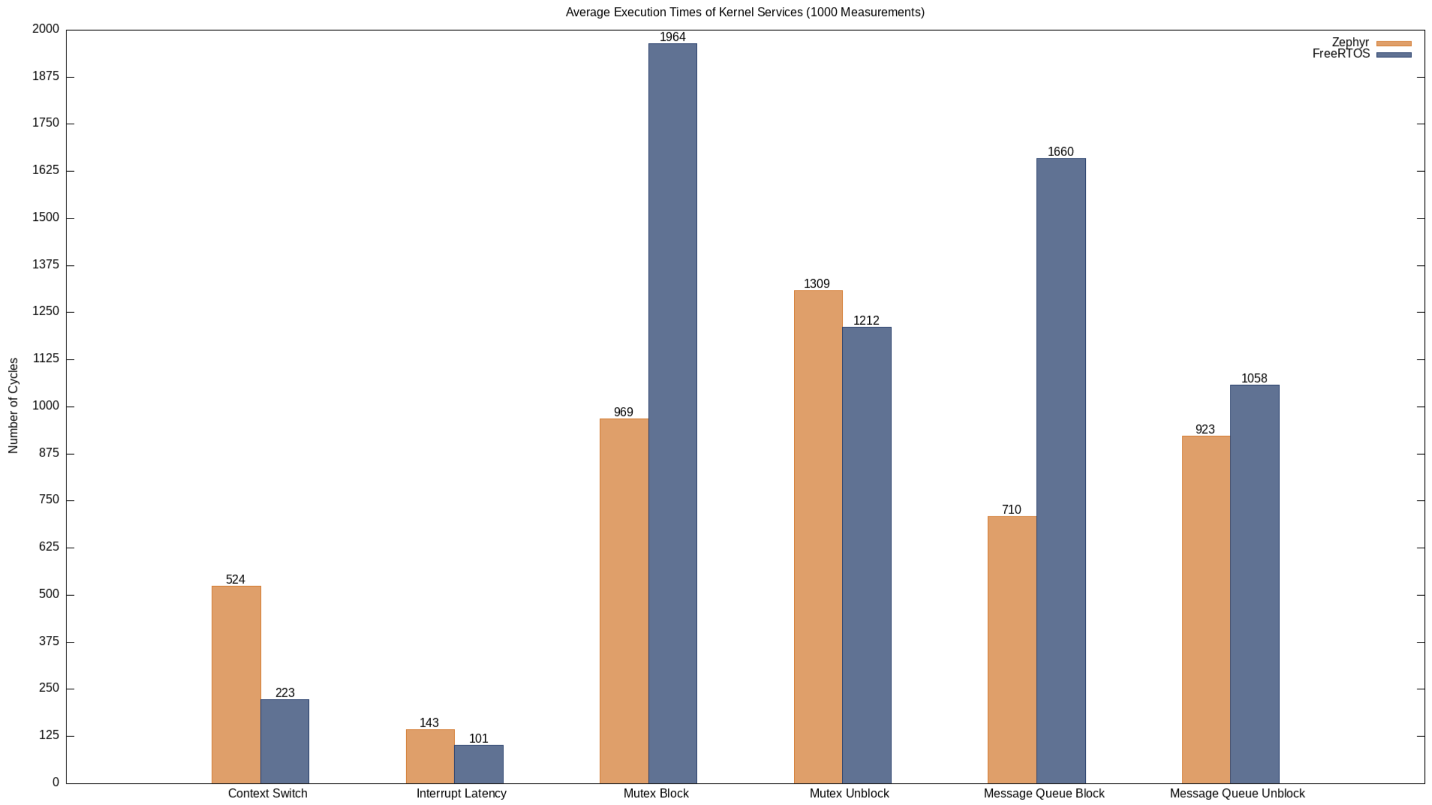
Typical Software Stack (Auto Audio)



Key care about

- **Low Latency :**
  - Typical Jack to Jack expected Latency numbers are 750 to 900msec for 32 channels at 48 kHz.
- **Real time processing**
  - Handling of live data from network interfaces and pre and post processing media streams. Runtime Tuning of parameters.
- **High memory Throughput.**
  - High capacity buffers using shared memory are passed to and from AVB cores to DSPs.
- **Scalable Media processing**
  - One core to multi-core.
  - ARM based processing to DSP based processing.
- **AI enabled interfaces.**
  - Both bring your data and bring your model support – easy interface layers.
- **UI based Tools to customize and optimize the processing.**
  - Auto generated code and optimized signal chain framework.

Zephyr for Audio use cases

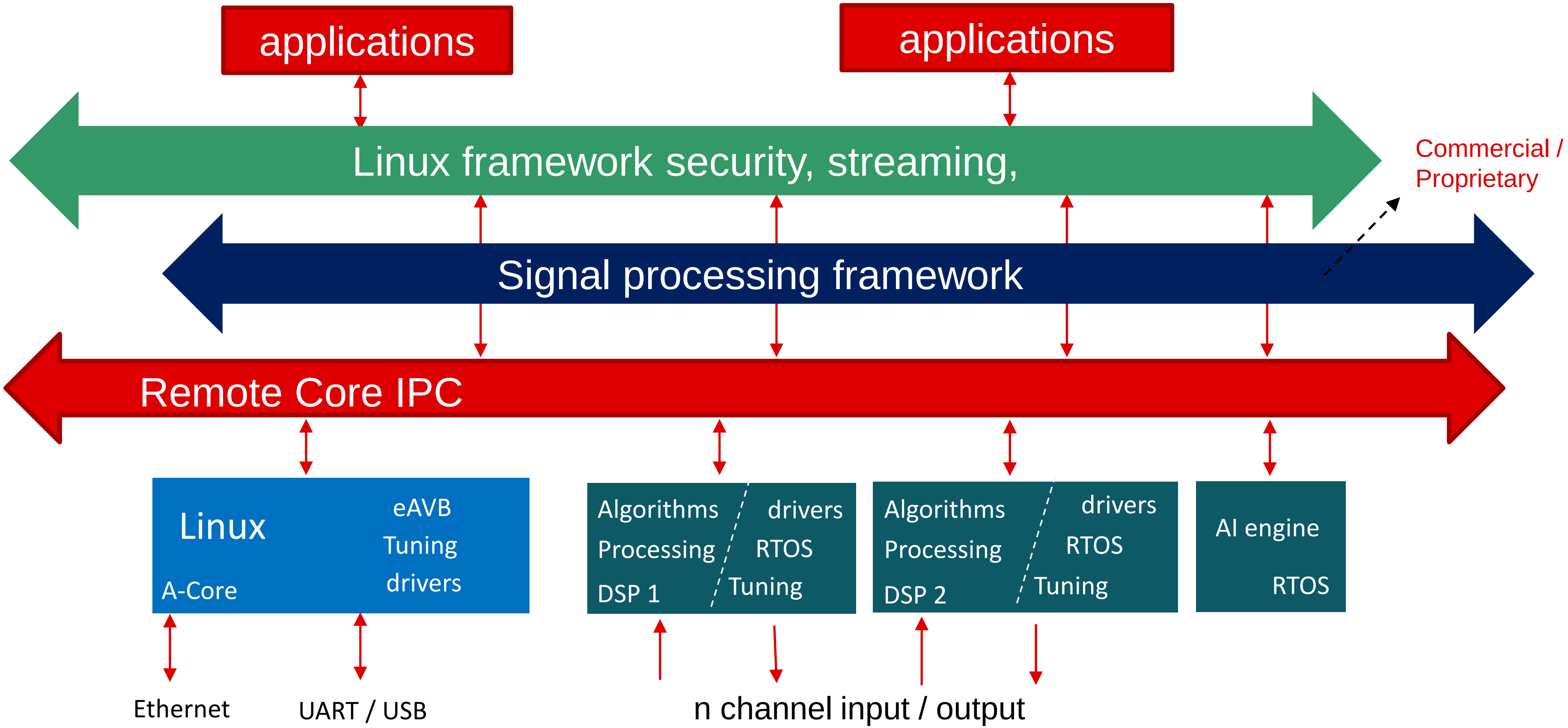


- Safety certified RTOS
- High Latency numbers
- Framework for IPC and Signal chain
- DSP port of Zephyr is quite expensive.
- Productized AVB stack for Zephyr
- Replacement for AUTOSAR ?

Courtesy : Measuring Real-Time Operating System Performance – Part II: Comparing FreeRTOS vs. Zephyr | Method Park by UL

General purpose - professional audio

Typical Software Stack



Key care about

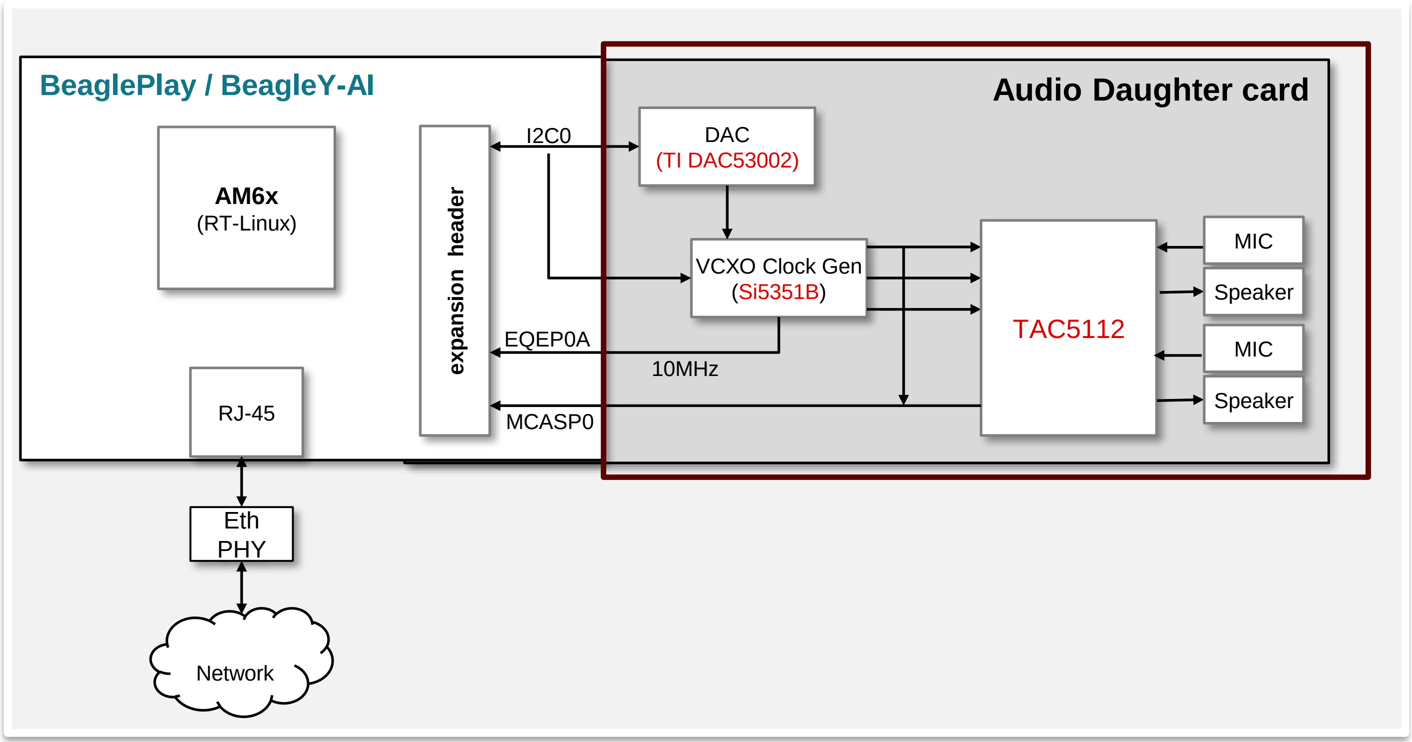
- **Low Latency :**
  - Typical Jack to Jack expected Latency numbers are 0.25 to 2 milli seconds.
- **Key processing elements in public stack**
  - Beamforming
  - Automatic mixers
  - Gain controls
  - Acoustic Echo cancelers (AEC)
  - Nosie reduction (NR)
  - Equalizer (EQ)
  - Audio conversion
  - PoE or PoE+ speaker w/ integrated amp
- **UI based Tools to measure performance and optimize the processing.**
  - Standardized performance measurement techniques and tools
- **Scalable Standardized opens source software stack**
  - One stack fits all applications – security, AI enabled, network streaming, low power
  - Flexibility to leverage commercially purchased/acquired tools, frameworks where and when required.

Linux for Audio use cases

- **Productize any of the Audio servers**: *PulseAudio*, ***JACK*** and *PipeWire*.
  - *JACK* has been favorable to provide real-time, low-latency connections for both audio and MIDI. Will this be shortly replaced with PipeWire ?
- **PREEMPT_RT and cyclic tests** :
  - a slight addition of user space service or kernel config update results in drastic change in cyclic test numbers.
- **PREEMPT_RT and Power Management** :
  - Power plays an important role when the devices are battery operated – but – the latency and deterministic response is important as well. How can they co-exist ?
- **Ease of use and Standardization**
  - Selection of audio server, frameworks and fine tuning isn't standardized. A simple distro that fits all audio requirements could have been easy.
- **Missing standardized signal processing framework**
  - Runtime attach of accelerators and DSPs to enable remote core processing of media streams.
- **Public versions of AVB stacks aren't community maintained**
  - OpenAVB last patch submitted 5 years ago.
- **Security while media streaming**
  - Multiple options available, no clarity on audio use cases

Let’s collaborate

Audio HAT for AM62x/AM67x based BeagleBoard platforms



- In design phase, to be launched by BeagleBoard foundation by end of December 2024.
- Enables open source community to collaborate on audio software stack development.

Common Audio card that works across BeagleBoard and header on TI starter kits

Collaborate & develop an Audio Stack with Beagle Audio Board :

- **Optimize latency and meet the performance requirements**
  - Maintain Preempt-RT patches with kernel & userspace configs for audio configurations.
- Identify and **optimize the open source audio stacks**
  - Jack, NetJack2, pipewire
- Introduce **OpenVx based signal chain framework for audio application**
  - Remote core management and offload of processing media stream to DSPs.
- Revisit **Opensource OpenAVB project**
  - Enable community to revisit open source version of AVB stack.
- **Collaborate to port open source audio solutions to beagle board platforms**.
  - RAVENNA : is a solution for real-time distribution of audio and other media content in IP-based network environments.
  - AES67 : port and optimize the stack for low latency applications.
  - Identify the appropriate protocol stack to support security enabled streaming of real-time audio over network interfaces.
  - Explore sound open firmware : https://www.sofproject.org

References

- AWE Core OS 8.B.19 Documentation: AWE Core OS Integration Guide (dspconcepts.com)
- An Introduction to AVB Networking | PreSonus
- RAVENNA IP-based Networking Technology - RAVENNA Network (ravenna-network.com)

Credits

Thanks to the following team members for helping in populating this info :

- Manisha Agrawal <manisha.agrawal@ti.com>
- Asheesh Bhardwaj <asheeshb@ti. com>
- Rahul Prabhu <r-prabhu@ti.com>
- HN, PRADEEP <p-hn@ti.com>
- Venkatesan Krishnamoorthy < v-krishnamoorthy@ti.com>
- Dhruva Gole <d-gole@ti.com>
- Rutvij <info@siliconsignals.io>

Community Partners