The Seventeenth Annual Interactive Audio Conference
PROJECT BAR-B-Q 2012

Group Report: The Speakers are Talking, are the Microphones Listening? Connecting Smart Speakers & Microphones

Participants: *A.K.A. "Sounds Like Chicken"*
Devon Worrell, Intel
Mike Spence, TI
James Skov, Connexant
Mikko Suvanto, Akustica
Chu-Ting Su, RealTek
David Roach, Optimal Sound
Shawn Beus, Knowles
Wayne W Chien, DTS
Ryan Roberts, Wolfson Microelectronics
Josh Atkins, Beats Electronics
Evan Wang, RealTek
Facilitator: Linda Law, Project Bar-B-Q

Brief statement of the problem(s) on which the group worked

In order to get enhanced audio features, multiple speakers & microphones are a necessity. At this time there is no standardized protocol to govern and define communication exchange between these multiple endpoints (speakers/microphones). There is a proliferation of a variety of interfaces that have no commonality. Without guidance, then we run the risk of having to support multiple interfaces. There will be smart speakers & smart microphones, and they will need to be inter-connected to the rest of the system.

Four dominant interfaces between Microphones and Speakers exist today: PDM, HDA, I2S, & 3.5mm jack (data only). None of these interfaces supports all the requisite capabilities; nor is there a standardized format that can manage this system. Additionally within these existing interfaces, a low-latency up-path also does not exist.

It is clear that there is a need to expand one of these existing formats, or we need to create a new I/O format. Interfaces such as I2S & PDM are not applicable in this situation as they assume all inbound information is data only and are done out of band. As you will see in the outlined requirements, there is more information than just the data element that needs to be transmitted, and it is believed that this must be done in band.

Sounds Like Chicken addresses the Issues and Answers the Questions

The group decided that the best course of action would be to recommend extending the existing standards to support the added functionality of smart microphones and speakers. There is a new physical interface that leverages the logical interfaces of exiting standards and provides an unified solution.

We will be providing guidance and support to the other organizations/standards bodies that would be affected.

We addressed the following areas:

- A method for speakers/Mics to have local processing
- An outline for a physical interface with backwards compatibility with existing solutions
- Ability to piggy back on 3.5mm jack
- Provides for “always listening”
- Definition of the link layer protocol that can be used (with longevity in mind)
- Exploration of the challenges and potential solution spaces for processing capability, power reductions, physical interfaces, device to device interactions,
- Ability to fit into existing logical frameworks on PC

The action item list

	Who’s Responsible	**Due Date**	**Description**
1	Wayne W Chien	11/21/2012	Complete report for publication
2	Devon Worrell	01/12/2013	Update “Sounds like Chicken” to HD Audio Spec update team “Screaming Monkeys”
3	Josh Atkins	10/31/2012	Research on current efforts to support this interface on the 3.5mm jack with backwards compatibility

Expanded problem statement

Change is happening/coming and we must guide this change in order to:

- Control the I/O design process – make sure that we are designing what is needed, not what is easy
- Assist platform designers so they can cleanly design platforms for the future
- Make sure that this change has longevity
- Guide low-power requirements & other design features
- We need to support Ultrasonics
- We must support MIDI (musical instrument digital interface)
- Control and maintain an affordable cost structure
- Introduce inter-operability by simplifying the I/O choices

Microphones are already going digital; but the standards work already being done is not broad enough. This group would like to provide a sense of direction to how this work should be done.

Smart devices need DSPs. However, we must define the limits of how this would be created and utilized. Example “always listening” must be supported and the power requirements may be too high. Therefore, intelligence should be pushed out to the peripheral device.

In order to allow for system/ecosystem longevity, the interface will be defined with a long term vision in mind (7+ years and beyond).

Handsets solution providers use SoC solutions with centralized DSP partitioning to perform functions delegated to mics/speakers, and they may not wish to decentralize, therefore the solution must support both centralized and decentralized partitioning.

Providers of thin mics/speakers will only venture to produce intelligent versions when adoption is shown by handset and tablet solution providers.

IO must have a level of certainty to design towards.

How do we communicate back to digital devices. Will intelligence in the transducers assist output and communication with the receiver? We need to manage the power consumption. Behave well and nicely with the algorithms…

OEMs want to move MB between different acoustic environments (plastic vs. magnesium chassis) system should be able to adapt at the implementation level to accommodate this

More work at a single point vs. multiple points.

Control of the vertical.

Eliminate multiple tuning activities…

Low latency up-path does not exist.

Each of the current interfaces have their own “way of doing things”. None of them communicate well between each other. A new I/O that simplifies and addresses

Wearable devices are a burgeoning market space

Expanded solution description

We have created the definition for a new IO platform, darn it! This solution is capable of creating world peace, curing hunger, rendering nuclear by-products safe for consumption, solving education in the United States, eliminate taxes, addressed the issues of global warming, lead paint, BHPs in plastics, autism caused by vaccinations, making the blind lame, allow us all to walk on water, turn lead to gold, live forever, and revive Elvis.

Or we will just patent this idea and sell it to Apple and Intel for trillions of dollars.

Furthermore:

Sounds Like Chicken has defined the following mandatory requirements:

- Affordability: overall incremental cost must be balanced with the value add brought on by The Solution.
- Efficient power utilization: Longer operational battery life Device should be able to operate within existing power transport specifications
- A standardized Framework
- Both centralized and distributed intelligence
- Single channel echo cancelation
- Noise Reduction, Speech clarity, etc.
- Smart – Programmable memory
- Always listening microphones (implies that you have a “smart” microphone)
- What is the Ultra-sonic data do we propose to support?
    - Support both raw (192 KHz sample / 96 KHz band width) and base band (spec TBD)
- Transport of Multiple sets of information depending upon the implementation
    - Capture environmental information (room size, proximity, temperatures, etc…)
    - Ultra-sonic/audio/voice
- Smart Microphone action pathway:

    - Assumes microphone is already running in a low power state
    - Some event occurs that causes the microphone to sit-up and take notice (the smartness)
    - Microphone then “pokes” the system in order to execute some higher function upstream. (wake, speaker ID, etc…)
- Must be simple and have ease of implementation or self tuning

Other Algorithms and Capabilities:

- Voice recognition
- AGC
- Key click suppression
- Capture classification
- Snap, crackle and pop removal
- Clipping detection
    - Bottoming out
    - Digital clipping
- Create an aggregate microphone using the speaker, say, if mic input is too loud
- Thermal management of speaker transducer
- Media classification
- Speaker excursion maximization
- Energy redistribution – consider the spectral content of the room, adjust the output to the speaker accordingly
- Feedback elimination
- Speaker linearization
- Dynamic Range Compression
- Spectral Shaping
- Voice vs. music ID and speaker tuning
- Upmix / downmix / virtualization
- Beamforming
- Smart EQ
    - Speaker tuning – slow or fast
    - Realtime, over long term informs a state
    - Can use the voltage and current in the speaker to inform how to EQ the microphone

1 - Physical Layer (option to define a new physical interface)

- Electrical (not optical)
- 2 pin (Data & Control in/out)
    - Option for a broadcast clock
- Capable of ring structure (daisy chain)
    - Option for point to point
- CMOS voltages
    - Power from external source (micro watts)
- 1.8 volts
- I/O Bit Rate: 6 & 6.144 / 12 & 12.288 / 24 & 24.576
    - (potential for others TBD – wide support to prevent fragmentation of the market) Rate determined by assumption of a environment bounded by physical parameters defined to be: 8 devices with a combined minimum of 4 microphones and 2 speakers.
- Tolerates Simple Connector & non-twisted cable
- Wake/signaling mechanism to “poke” the system

Suggested exploration:

- Can this be layered upon the exiting 3.5mm jack?
- Control & data travelling on the same pathway
- Phase 1 – implement Microphone Input (digital)
- Phase 2 – implement Speaker output

2 - Link Layer (Protocol) – should allow for:

- Streaming at low latency (10ms?) for the features outlined above.
- Firmware Extensibility / Apps upstream allowed for collaboration, customization and upgrade in a standardized fashion
- I/O introduction & hand-shake
- Flow control
    - Master clock control
    - Start/stop flags
    - Action/reaction
    - Buffering? Efficiency of data transport (bucket bursty/pulse-esq transmission)
        Note: Link cannot (should not) be running all the time.

3 - Logic Layer
The 2014 Mobile HD-Audio spec update will address this interface

Potential alternative for introducing Phy: Op 1: MIPI Op 2: AES.
Potential Alternative: SLIMbus (phy/link combo)
Further in depth evaluation of both would be required before either can be fully discounted.

Appendix - raw notes

Possible Use Cases (not limited to)?
Conference-call using several smart-phones/devices in the same area. (human to human)
Controlling devices remotely – TV (human to machine)
Skype (VOIP)
Video conferencing
Music playback
basic content creation
movie playback
speaker phone
headsets (wired or wireless)

Facilitate connections between:

Where do Speakers & Mics coexist currently?
Tablets
ULX (big tablets with detachable keyboard)
Handsets
Headsets
AIO

TV
Bluetooth headsets
Telematics/Infotainment (Automotive Convertibles)
Xbox Kinect (Gaming Consoles)
Conference Speaker Phones
Baby Monitors
Soda Machines
Walkie-talkies

What is currently difficult to do on IC(DSP/CPU)
DC Compensation
Dynamic range control
Frequency response

Where should the implementation go?
All of the above and then more

What needs to be investigated further?
Clock synchronization / adjust local clock - must have in order to control the interpretation and usage of the different data types coming into/leaving the system. If you want the microphone to synch with the speaker otherwise is very difficult/impossible to perform EC/ANC functions. With multiple segments, it may be necessary to explore the necessity of clock synchronization.
Secure connections
Solve the tuning function, simplification, ease of implementation; self tuning

What other data types must be supported?
Control

How do we support multiple channels with different data types running at different rates is it really needed?
Expandable framework (future-ability)

What is the definition of smart? (not dumb… see Doppler Chicken report)
Intelligent enough to do Echo Cancellation alone
Implementation Intelligence (interchanging speakers/transducers, creating a good/better/best environ.)
Operational Control/adaptability (ambient temp access, etc…)
Adaptive to the environment and use case

Why do we want this / What is the necessity?

Change will happen, we must guide this change in order to:
Control the I/O design process
Assist platform designers so they can cleanly design platforms for the future
Make sure that this change has longevity
Guide low-power requirements & other design features
We need to support Ultra-sonics
We must support MIDI (musical instrument digital interface)
Control and maintain an affordable cost structure
Inter-operability, simplify the I/O choices

Strong desire stem from Mics already going digital, there is appearance of no sense of direction of how this should be controlled. The work is not broad enough and current interface is not clearly defined.
Intelligence should happen since we are taking SW and pushing it into centralized DSP. Concern with the limits of the DSP in the CPU or on the primary system platform only.

Example “always listening” power requirements may be too power hungry, so process should be pushed out to the peripheral device / “on the node”.
Need to identify the ecosystem. DSPs in the speakers and microphones.
In order to allow for system/ecosystem longevity, the interface needs to be defined with a longer term vision in mind

4 dominant interfaces between Mic and Speaker interfaces: PDM, HDA, I2S, & 3.5mm jack(data only)
Currently low-latency up-path does not exist. We either need to extend one of the existing or we need to create a new I/O.

I2S & PDM assume all inbound info is data and are done out of band. It is believed that it needs to be done in band.

Structure of HDA will change.

What about Handsets where SoC providers use the centralized DSP portioning to perform functions delegated to mics/speakers in PCs? Why would providers of thin mics/speakers venture to produce intelligent versions without adoption at least in part by handset and tablets.

In order to get enhances audio features, multiple speakers/microphones are a necessity. Currently no clean way to control and communication between these multiple endpoints exists. There is a proliferation of a variety of interfaces that have no commonality. Without guidance, then we run the risk of having to support multiple interfaces.

IO must have a level of certainty to design towards.

How do we communicate back to digital devices. Will intelligence in the transducers assist output and communication with the receiver? We need to manage the power consumption. Behave well and nicely with the algorithms…
OEMs want to move MB between diff acoustic environments (plastic vs. magnesium chassis) system should be able to adapt at the implementation level to accommodate this

More work at a single point vs. multiple points.
Control of the vertical

Eliminate multiple tuning activities…

Low latency up-path does not exist.

Each of the current interfaces have their own “way of doing things”. None of them communicate well between each other. A new I/O that simplifies and addresses

Wearable devices are a burgeoning market space

Defining the Requirements

Affordability: overall incremental cost as well as value add
Efficient power utilization

- Longer operational battery life
- Device should be able to operate within existing power transport specifications

<u>Framework allowing support of</u>

- Both centralized and distributed intelligence
- Single channel echo cancelation
- Noise Reduction
- Smart – Programmable memory

Always listening microphones (implies that you have a “smart” microphone)

Smart Mic action pathway:

1. Assumes mic is already running in a low power state
2. Some event occurs that causes the mic to sit up and take notice (the smartness)
3. Mic then “pokes” the system in order to execute some higher function upstream. (wake, speaker ID, etc…)

What is the Ultra-sonic data do we propose to support?

Support both raw (192 KHz sample / 96 KHz band width) and base band (spec TBD)

Tuning function: must be simple and have ease of implementation or self tuning

Transport of Multiple sets of information depending upon the implementation

Capture environmental information (room size, proximity, temperatures, etc…)

Ultra-sonic/audio/voice

Voice recognition

Talker ID

Talker verification

Speech clarity

AGC

Key click suppression

Capture classification

Snap, crackle and pop removal

Clipping detection

Bottoming out

Digital clipping

Create an **aggregate microphone** using the speaker, say, if mic input is too loud

Render

Thermal management of speaker transducer

Media classification

Render, Requires Capture

Speaker excursion maximization

Energy redistribution – consider the spectral content of the room, adjust the output to the speaker accordingly

Feedback elimination

Speaker linearization

Both Render & Capture

EQ

Dynamic Range Compression

Spectral Shaping

Voice vs. music ID and speaker tuning

Upmix / downmix / virtualization

Beamforming

Smart Volume

Encode / decode / voice / music

Smart EQ

Speaker tuning – slow or fast

Realtime, over long term informs a state

Can use the voltage and current in the speaker to inform how to EQ the microphone

Expandable framework (future-ability)

1 - Physical Layer (option 1)

Electrical (not optical)

2 pin (Data & Control in/out)
Option for a broadcast clock
Capable of ring structure (daisy chain)
Option for point to point
CMOS voltages
Power from external source (micro watts)
1.8 volts
I/O Bit Rate: 6 & 6.144 / 12 & 12.288 / 24 & 24.576
(potential for others TBD – wide support to prevent fragmentation of the market)
Rate determined by assumption of a environment bounded by physical parameters defined to be: 8 devices with a combined minimum of 4 microphones and 2 speakers.

Tolerates Simple Connector & non-twisted cable
Wake/signaling mechanism to “poke” the system

Physical Layer (option 2)
Can this be layered upon the 3.5mm jack?
Control & data travelling on the same pathway

Implementation path (on 3.5mm jack):
Phase 1 – Microphone Input (digital)
Phase 2 – Speaker output

2 - Link Layer (Protocol) – should allow for:
Streaming at low latency (10ms?) for the features outlined above.
Firmware Extensibility / Apps upstream allowed for collaboration, customization and upgrade in a standardized fashion
I/O introduction & hand-shake
Flow control
Master clock control
Start/stop flags
Action/reaction
Buffering? Efficiency of data transport (bucket bursty/pulse-esq transmission on Worrel-time?)
Link cannot (should not) be running all the time.

3 - Logic Layer
The 2014 Mobile HD-Audio spec update will address this interface

Potential Alternatives-Op 1: MIPI Op 2: SLIM BUS (Phy/Logic combo). Further in depth evaluation of both would be required before either can be fully discounted.

Next Step Actions:

1. How do I patent this?
2. This interface needs to be adopted as a part of the HD Audio spec update.
3. Merged group report: “S&M Sounds Like Chicken”

S&M ideas that we may need to consider

What needs to be communicated across devices:

Speaker:

- Frequency roll-off
- Power handling
- SPL, given digital signal level
- Real time speaker displacement
    - Can be used to calculate acoustical displacement and assist with echo cancellation
- Speaker degradation
- Component variation
- Acoustical leaks
- Voice coil temperature
    - or air temp, if voice coil temp n/a
- System and device capability constraints
- Configuration
    - # of channels
    - Geometry
    - Orientation
- Ambient sound levels & spectral content
- Rest of system telemetry
- Inter-channel communication
- Where is the end user relative to the device
- Power budget
- Real-time amp output voltage or acoustic pressure
- When additional transducers are added to the system in real-time

Microphones

- How many
- Configuration
    - Orientation
    - Locations
- Phase & amplitude
- Tolerance
- Power budget
- Dynamic range & Acoustic Overload
- Noise floor
- SPL, given digital signal level
- Long term degradation
    - Can be measured and inferred via processing
- Frequency response
- Pattern
    - As shipped
    - In system is more important

Places for Processing

- Dedicated local DSP
    - In speaker amp, feedforward or feedback
    - In mic(s)
- General purpose DSP

    - In Southbridge (Intel/Tensilica)
    - In HD Codec
    - In Apps processor
    - I2S or Slimbus mobile codec
- Host / Driver / OS / App
- Cloud
    - Potential for power saving
- External peripheral hardware
- GPU
- Analog