Audio Processing With An Open Development System

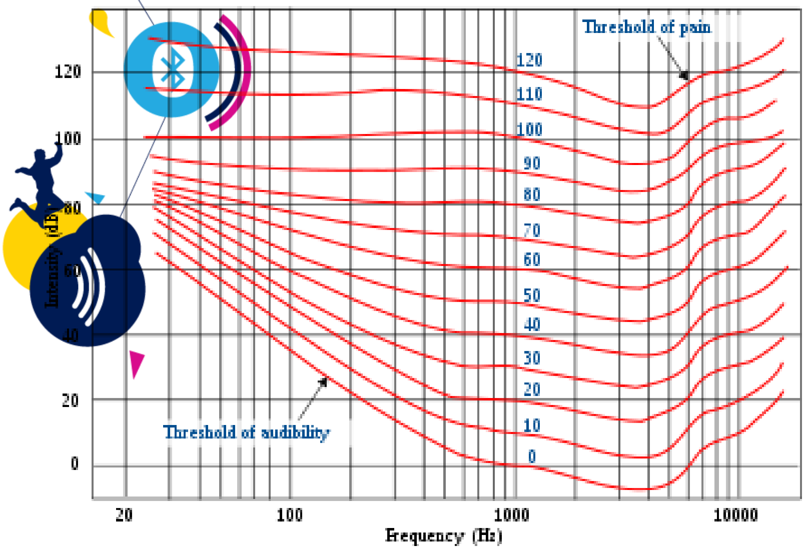
Overview

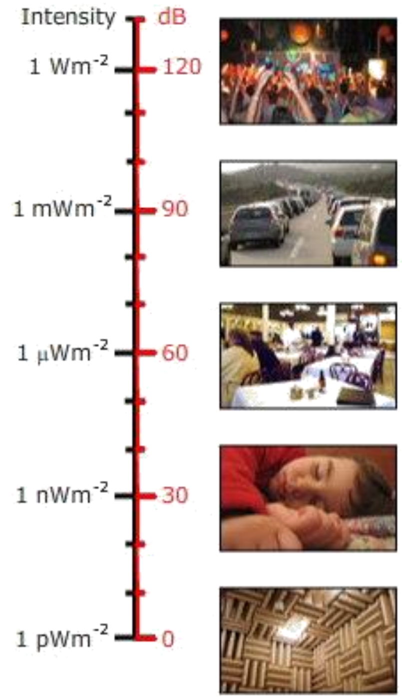
- MEMS MICROPHONES
  - Inside a digital MEMS microphone
  - Interface
  - PDM format

- FROM COMPONENTS TO SYSTEMS
  - MCU based devices and microphones acquisition
  - Microphones Arrays
  - Decimation, filtering and the PCM format

- FROM SYSTEMS TO APPLICATIONS
  - The Open.Audio framework
    - BlueVoice
    - Echo cancellation
    - Source localization
    - Beamforming

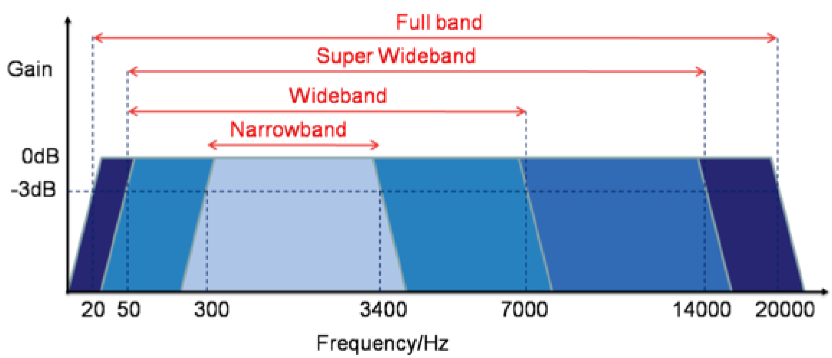
PCM Digital Audio





Dynamic Range

- 16 bit per sample,
- linear quantization
- $20\log_{10}(2^{16}) \cong 96\ dB$



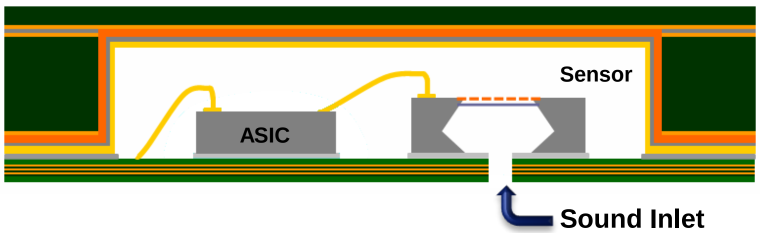
Frequency

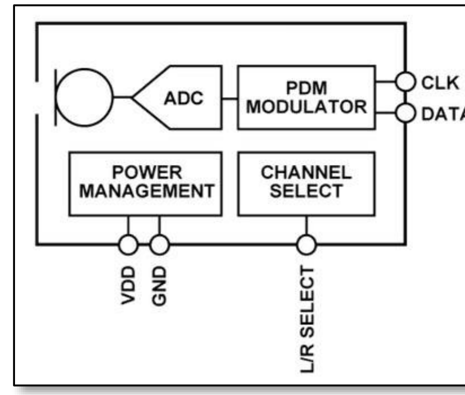
- Bandwidth: 20 Hz – 20 kHz
- Most common sampling rates:
  - 8 kHz, 16 kHz, 44.1 kHz, 48 kHz,
  - High Quality Audio: 88.2 kHz, 96 kHz, 192 kHz.

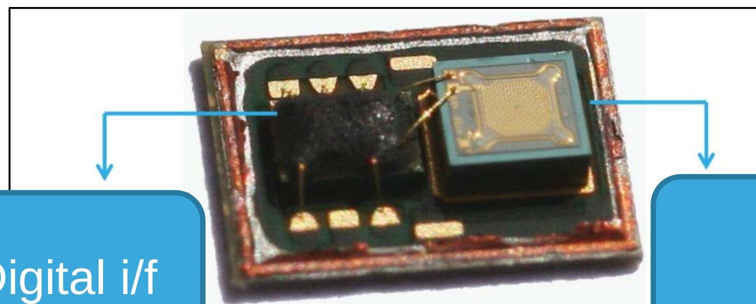
MEMS Microphones

Digital MEMS Microphones: inside view

Digital MEMS microphones:
- ultra-compact, low-power, omnidirectional
- built with a capacitive sensing element and an IC interface







A/D and Digital i/f

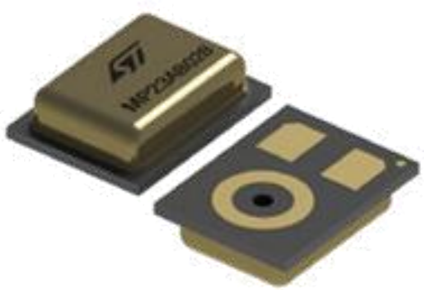
Sensing

Pulse Density Modulation interface:
- 1 to 3 MHz
- 1-bit resolution
- Fully digital

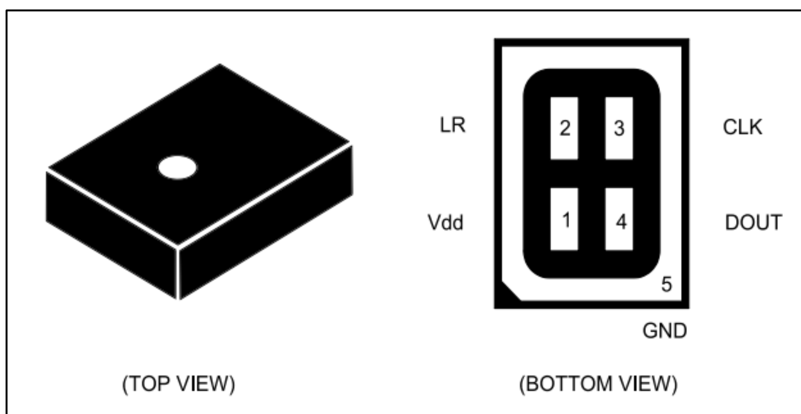
- Capacitive membrane
- Omnidirectional
- Analog

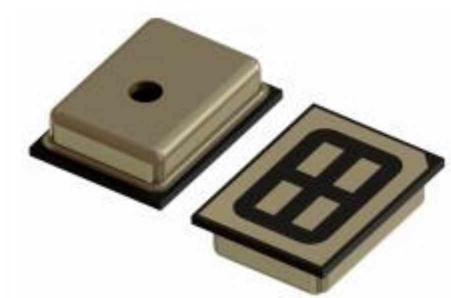
Digital MEMS Microphones: outside view

- Digital MEMS microphone interface: relevant signals
  - Input Clock (CLK) @ 1 to 3 MHz (high frequency)
  - Data Out (DOUT): PDM, 1 bit per sample, @ input CLK frequency
  - Left Right (LR)
    - Enables connection of 2 microphones on the same data line
    - LR selects the CLK edge where data is valid



Bottom port

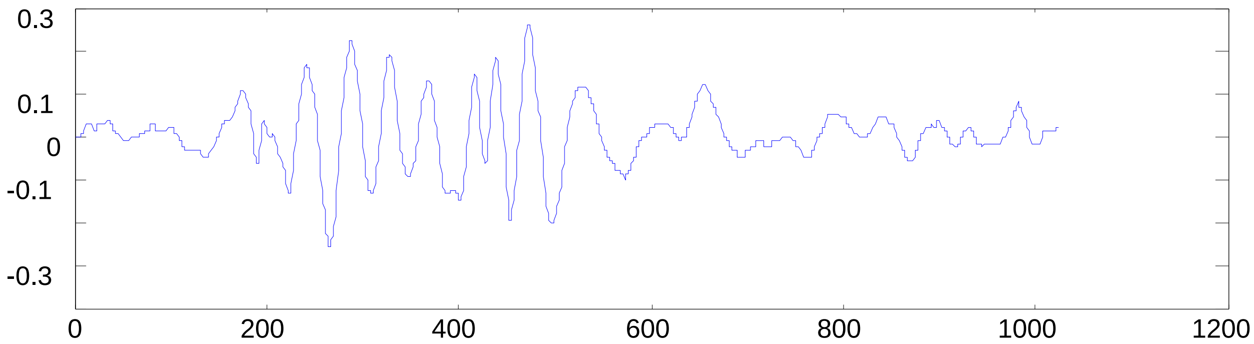




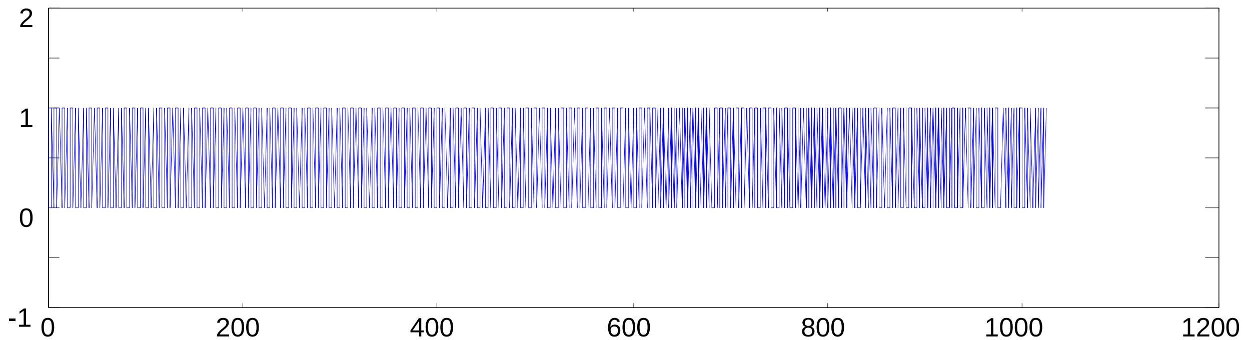
Top port

PCM Vs. PDM data format

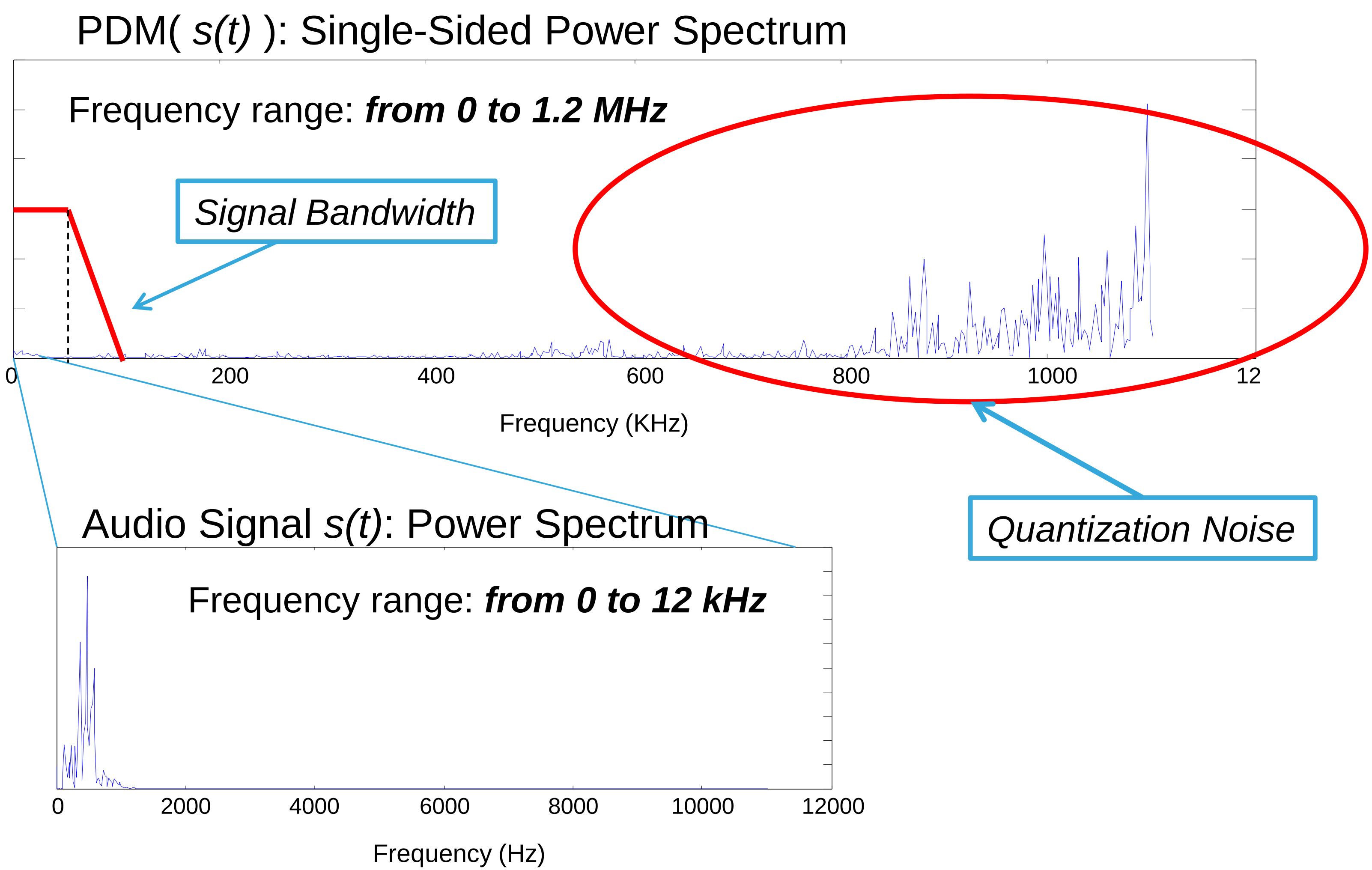
- PCM Audio Signal – 16 bit per sample



- Pulse Density Modulation
  - The relative density of the pulses corresponds to the analog signal's amplitude



Noise Shaping property of PDM



From components to systems

STM32 MCU for audio acquisition and processing

- STM32 Micro Controller Unit
  - Family of 32-bit Flash microcontrollers based on the ARM® Cortex®-M core
  - Very broad family of MCU featuring from very high performance to ultra low-power, supporting system optimization via on-chip peripherals and communication interfaces while maintaining ease of development.

- Example: STM32F446
  System-level performance
  - ARM® Cortex™-M4 32-bit RISC
  - <u>225 DMIPS and 608 CoreMark® @ 180MHz</u>
  - 128 Kbyte RAM, 512k Flash
  - Floating Point Unit, DSP instructions
  - <u>Up to 20 communication interfaces</u>,
    including 4x USARTs 2x UARTs @ 11.25 Mbit/s, 4x SPI @ 45 Mbit/s, 3x I²C, 2x CAN, SDIO, HDMI CEC and camera interface.

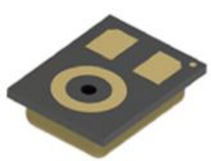
- Example: STM32L4
  Ultra low power, dedicated PDM interface
  - ARM® Cortex™-M4 32-bit RISC
  - 100 DMIPS w. FPU and ST ART Accelerator™ @ 80MHz
  - 128 Kbyte RAM, up to 1024k Flash
  - <u>Ultra Low-power mode, wake-up time 5 µs</u>
  - <u>DFSDM microphone interface with PDM to PCM conversion</u>

life.augmented

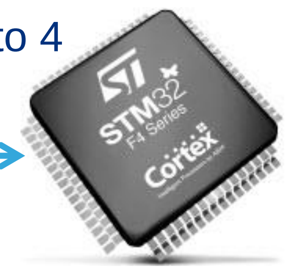
STM32 MCU and audio acquisition

Direct acquisition of digital MEMS microphones via:

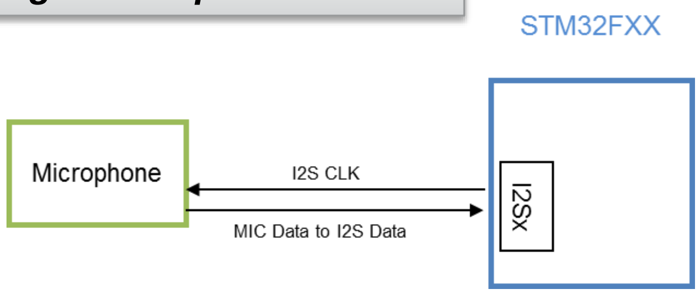
- Serial: SAI/I2S/SPI: 1 or 2 microphones share CLK and data line
- Parallel: GPIO: Up to 16 (or 32) microphones
- DFSDM dedicated interface – only on STM32L4
  - DFSDM = Digital Filter for Sigma Delta Modulator
  - Hardware acceleration for PDM to PCM conversion, for up to 4 microphones



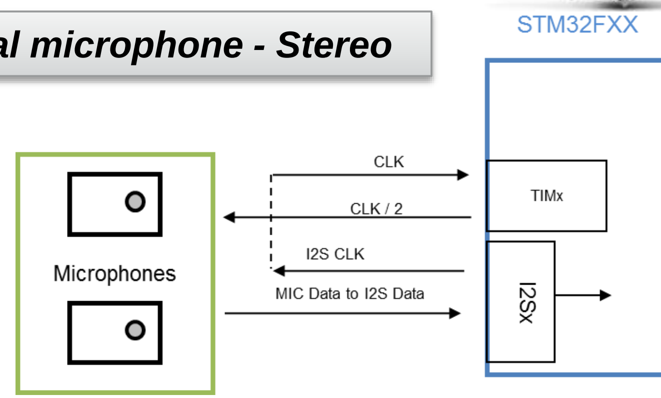
Example of Microphone acquisition via I2S



Single microphone - Mono



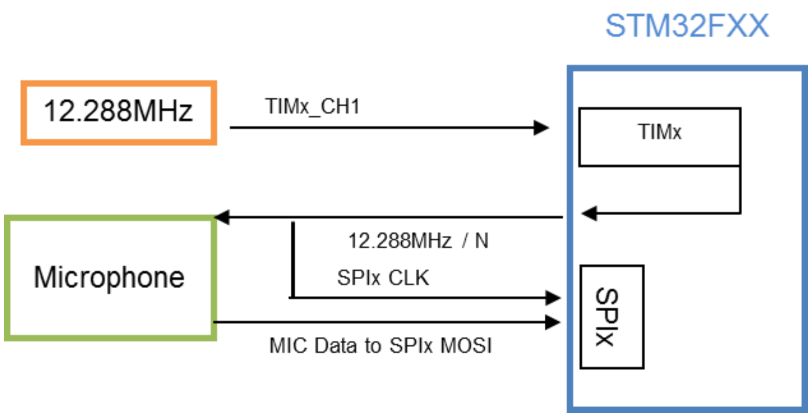
Dual microphone - Stereo



life.augmented

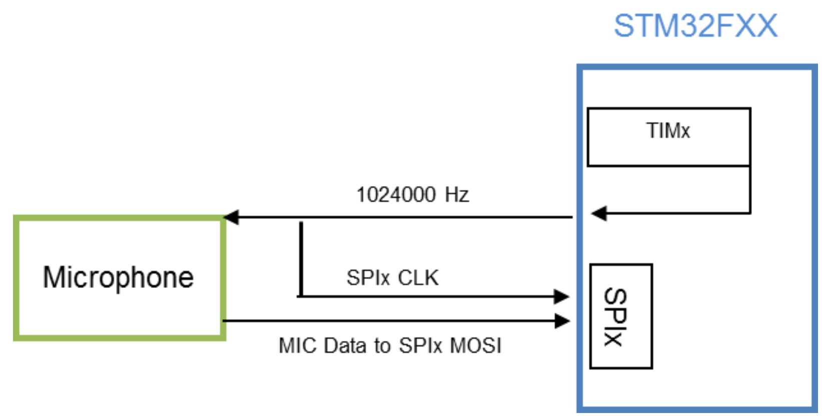
SPI Acquisition of a single microphone

SPI as a slave
external Quartz



- *12.288 MHz quartz is divided by 12 (timer)*
- *Microphone clock = 1024000 Hz*
- *Target Fs = 16 KHz*
- *Decimation factor = 64*

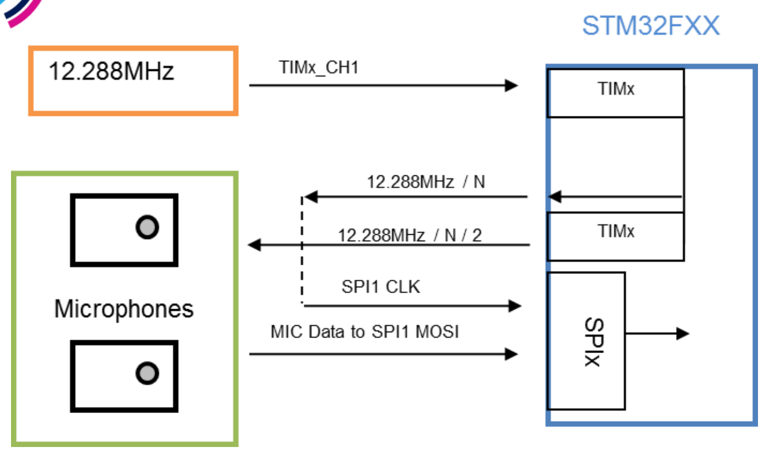
SPI as a slave
Clock generated by the timer



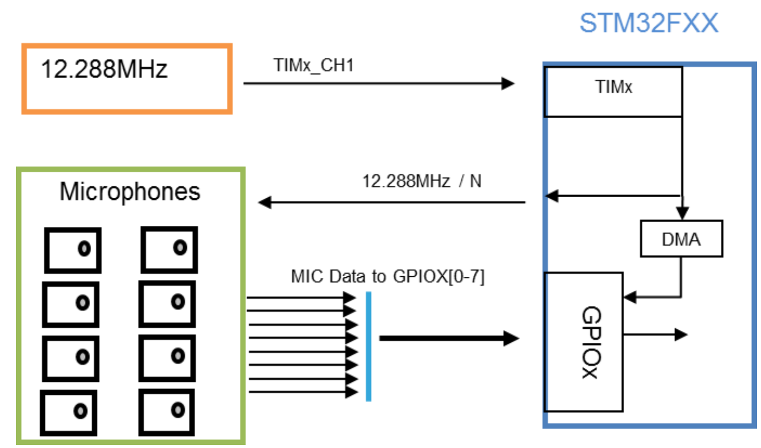
- *Timer clock input = 48000000*
- *Microphone clock = 1024000 Hz*
- *Target Fs = 16 KHz*
- *Decimation factor = 64*

Multi-Microphone Acquisition

SPI – Stereo Mode
With External Quartz



GPIO acquisition of
up to 16 microphones



External oscillator generates PCM Audio Sample rates
De-interleaver stage is required (can be SW):

Before de-interleaver:

M1 1	M2 1	M1 2	M2 2	M1 3	M2 3	M1 4	M2 4	...

After de-interleaver:

M1 [1..8]	M2 [1..8]	...

Before de-interleaver:

M1 1	M2 1	M3 1	M4 1	M5 1	M6 1	M7 1	M8 1	M1 2	M2 2	...

After de-interleaver:

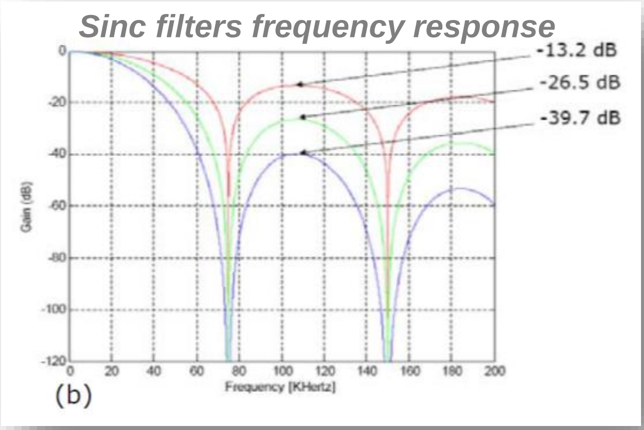
M1 [1..8]	M2 [1..8]	M3 [1..8]	M4 [1..8]
M5 [1..8]	M6 [1..8]	M7 [1..8]	M8 [1..8]

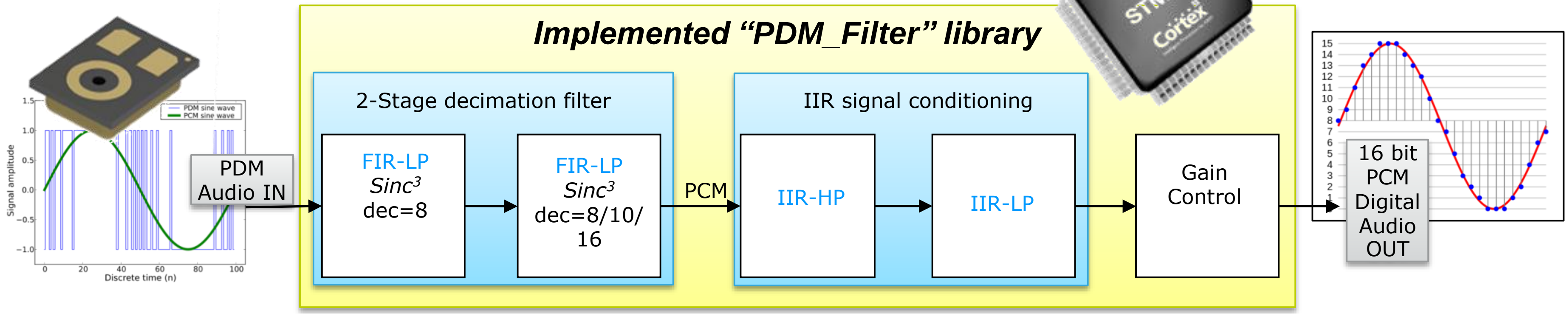
life.augmented

PDM to PCM Conversion library

- Target: 16 bit PCM Format

- PDM to PCM conversion
  - PDM signal: 1 bit, @ 1 to 3 MHz
  - PDM 2 PCM Key processing stages:
    - Sinc3 low-pass filter with decimation
    - IIR signal conditioning
    - Gain control





PDM2PCM library structure

Initialization – once at system startup

- Audio Sampling Frequency
- Audio Channels Input / Output
- LP filter cutoff frequency
- HP filter cutoff frequency

PDMFilter_InitStruct

Runtime – once every 1 ms

- PDM Buffer (in)
- PCM Buffer (out)
- Microphone Gain
- PDMFilter_InitStruct

PDM2PCM library performance

- Library configuration
  - Low pass filter (24000Hz) enabled
  - High pass filter (10Hz) enabled

- Values expressed in MIPS for typical configurations (*)

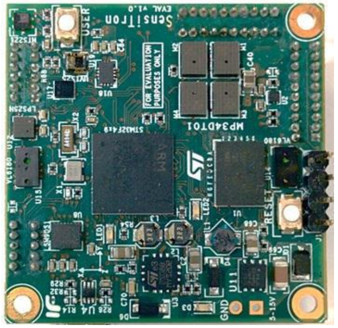
PCM sampling rate	16 KHz	32 KHz	48 KHz
PDM_Filter_64	~5	~10	~15
PDM_Filter_80	~6	~12	N/A
PDM_Filter_128	~9	N/A	N/A

(*) microphone clock = PCM sample rate x decimation factor

- Total amount of RAM = 780 bytes
- Audio frame corrisponds to 1 ms of PDM audio.
- Profiling done using IAR Embedded workbench, version 7.30 in simulation mode; optimization set to High Speed

MEMS Microphone development systems examples

- *Single-board* solutions:

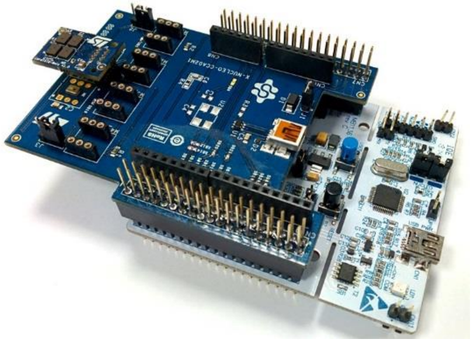


- Modular architectures:

+

+

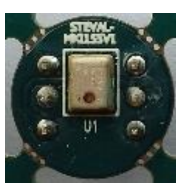
=



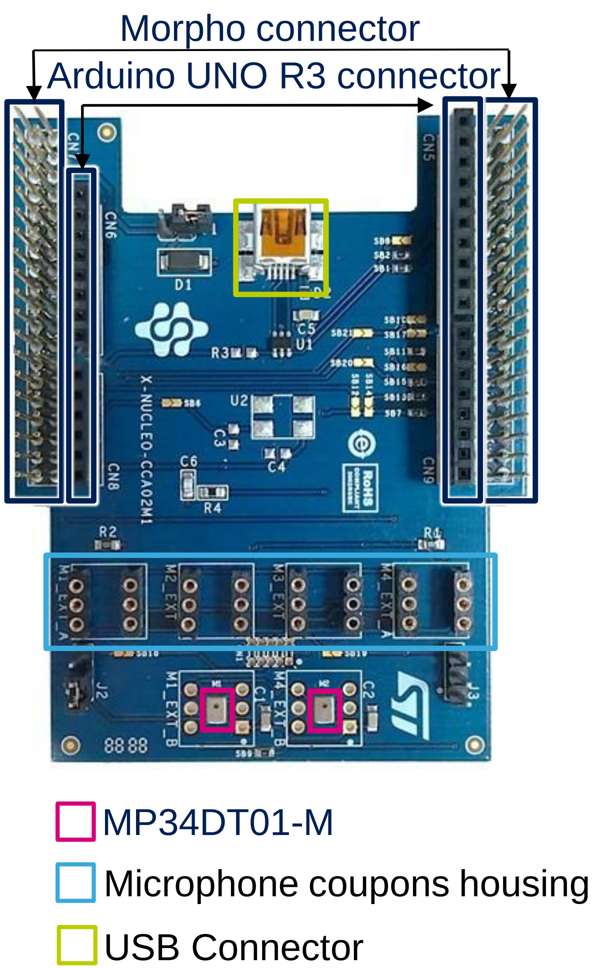
life.augmented

X-NUCLEO-CCA02M1

- MEMS microphone evaluation board
  - STM32Nucleo Expansion, compatible with STM32 ODE
- 2X MP34DT01–M microphones
- 1x miniUSB FS connector:
  - USB audio data streaming
- Up to 4 microphone synchronized acquisition and streaming
- 6X ST MEMS Microphone coupons housing:

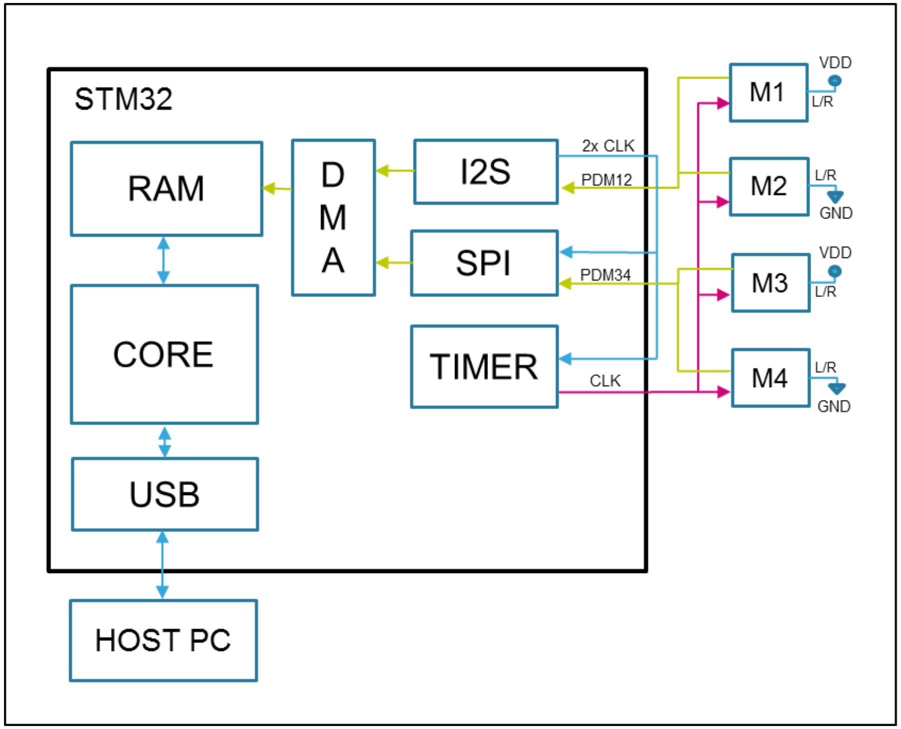


STEVAL-MKI129V1
STEVAL-MKI129V2
STEVAL-MKI129V3



X-NUCLEO-CCA02M1 Acquisition Architecture

- 2 microphones sharing the data line
  - Enabled by L/R PIN configuration
- 1X I2S
  - Master mode: precise timing
  - Acquires data from 2 microphones
- 1X SPI
  - Slave mode: uses I2S clock
  - Acquires data from 2 microphones
- 1X Timer
  - Halves the frequency of the I2S clock to feed the microphones

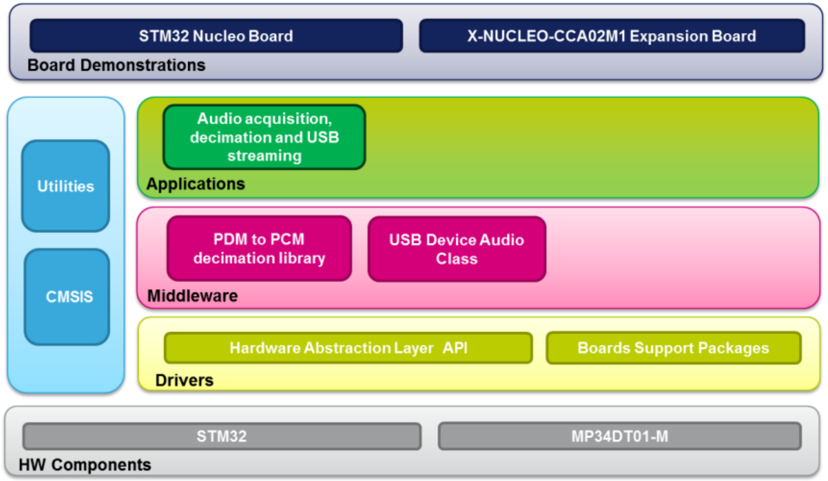


life.augmented

X-NUCLEO-MEMSMIC1

- SDK to build applications using digital MEMS microphones
  - Microphones acquisition source code
- Audio input class USB driver
  - Standard USB microphone
  - Audio streaming to a Host PC

User may just focus on audio processing development




```c
/**
* @brief  User function that is called when 1 ms of PDM data is available.
*         In this application only PDM to PCM conversion and USB streaming
*                  is performed.
*         User can add his own code here to perform some DSP or audio analysis.
* @param  none
* @retval None
*/

void AudioProcess(void)
{
  BSP_AUDIO_IN_PDMToPCM((uint16_t * )PDM_Buffer,PCM_Buffer);
  Send_Audio_to_USB((int16_t *)PCM_Buffer, AUDIO_SAMPLING_FREQUENCY/1000*AUDIO_CHANNELS);

}
```

Single entry point for user's Audio Signal Processing code

life.augmented

From systems to applications

Low power, low resources design

- IoT typical constraints can be met also by embedded Audio designs based on MEMS microphones and Microcontrollers:

- Low Power constraints
  - Battery powered designs
  - Wearable
  - Always on systems

- Low resources constraints
  - Limited MIPS/MHz count
  - Limited RAM and FLASH

- Low cost constraints

A system-level solution should deploy an optimal balance of HW and SW resources

Open Software eXtension free SW licensing program

Open.Audio
free SW licensing program

Open Software Expansion Environment

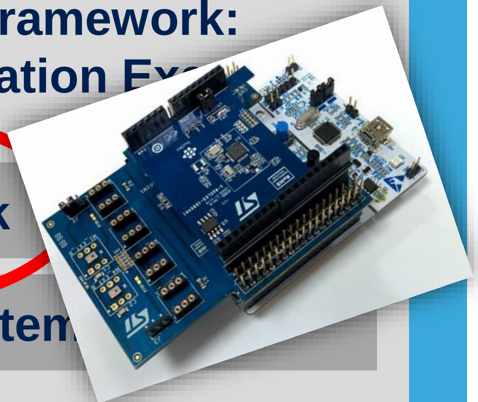
Libraries

Open. MEMS | **Open. RF** | **Open. Audio**

Open.Framework: Implementation Ex

BlueVoiceLink

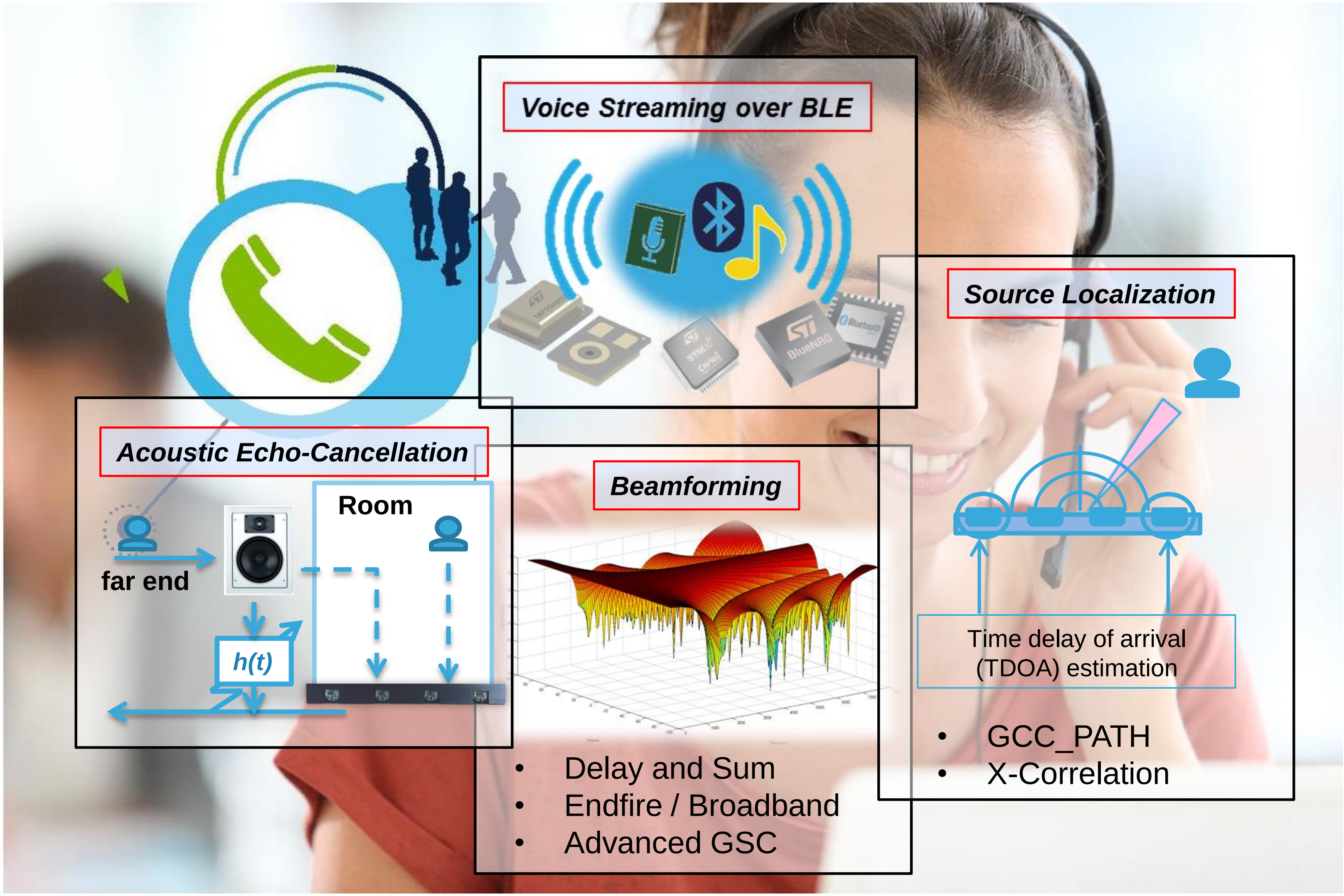
BlueMicroSystem



STM32 ODE

STM32 Nucleo development boards	STM32 Nucleo expansion boards	STM32 Cube software	STM32 Cube expansion software

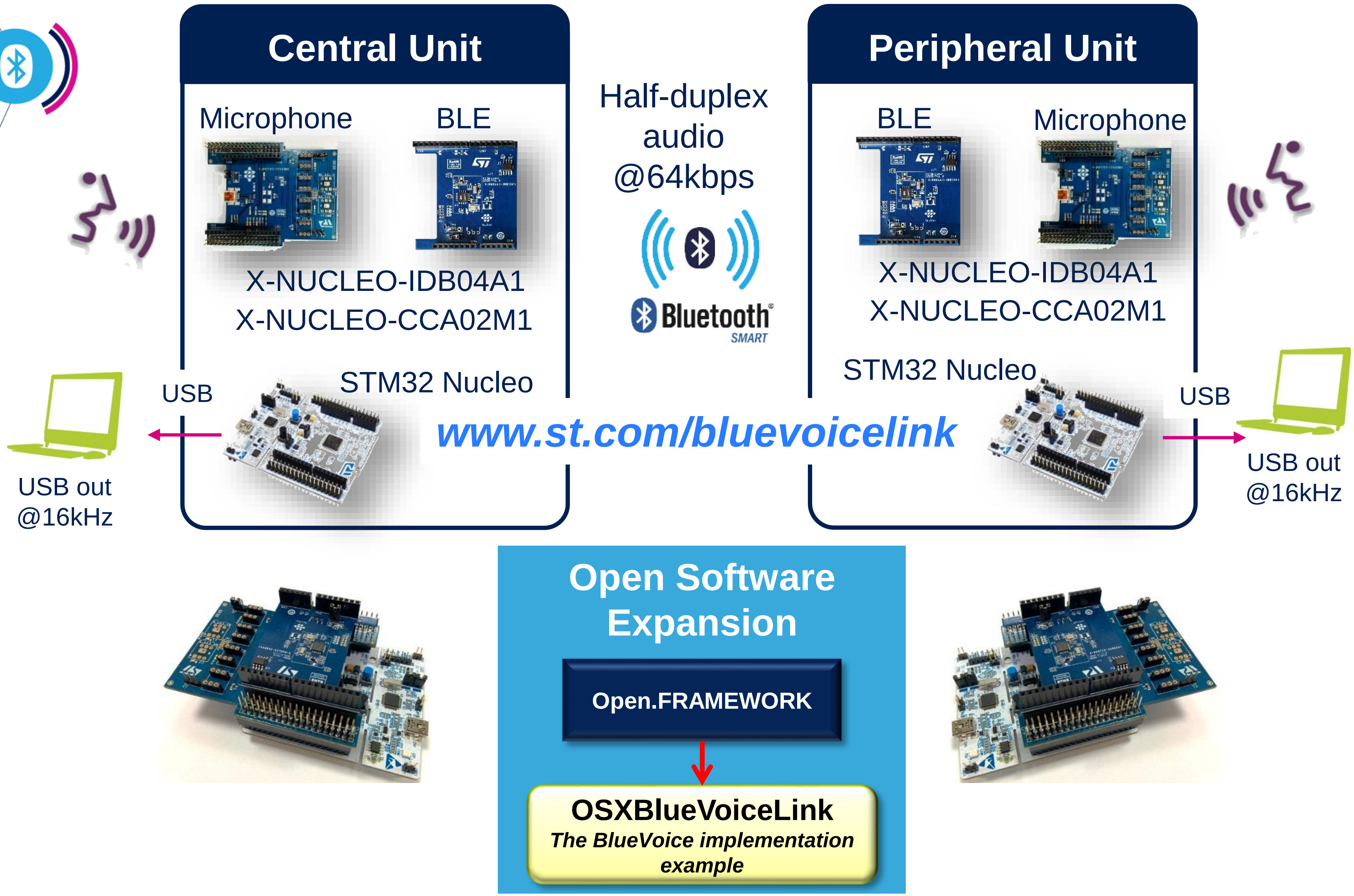
Embedded Audio
Software building blocks



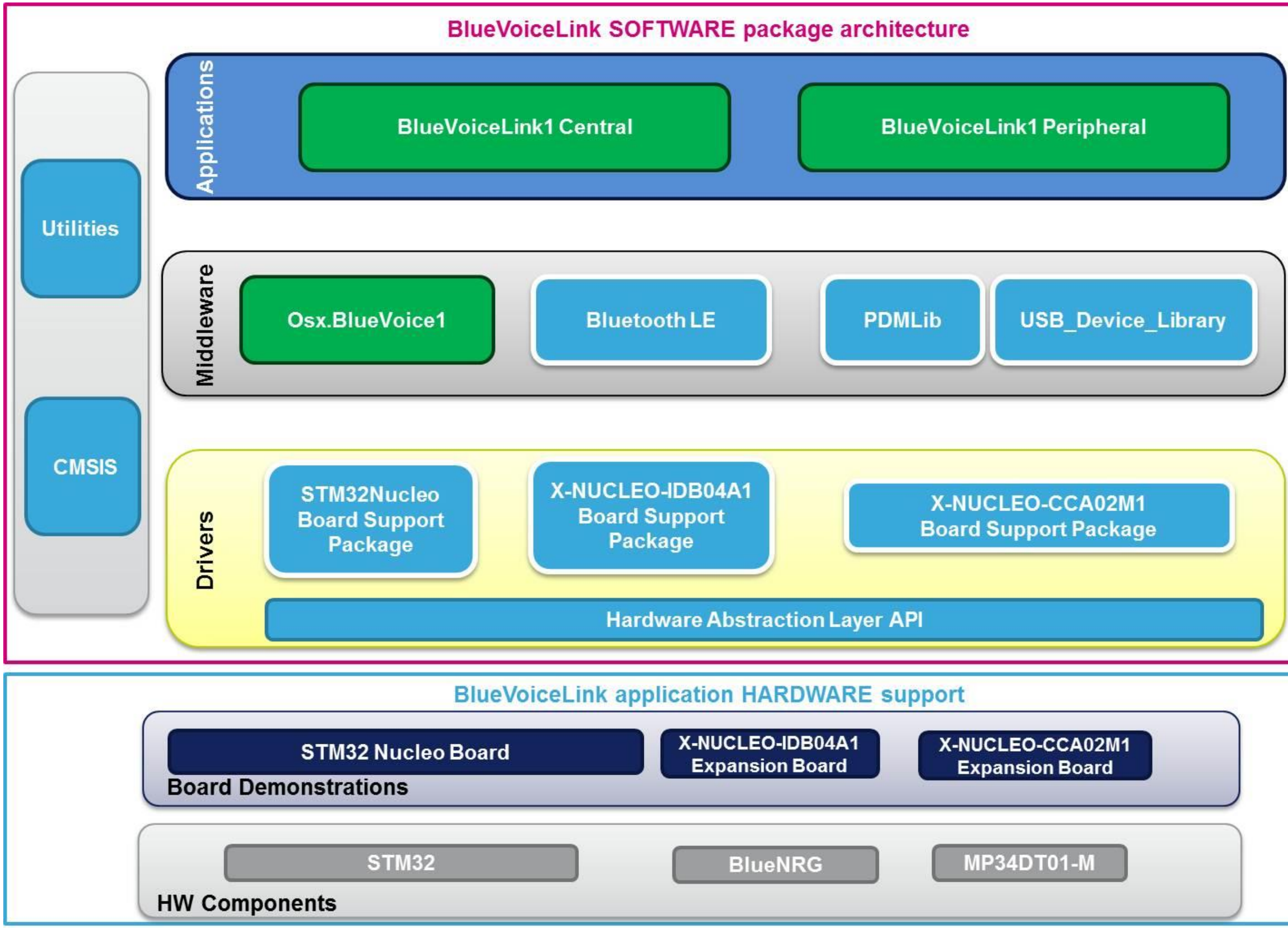
Open.AUDIO BlueVoice:
Voice streaming over BLE

life.augmented

BlueVoiceLink application overview



BlueVoiceLink SW and HW Architecture

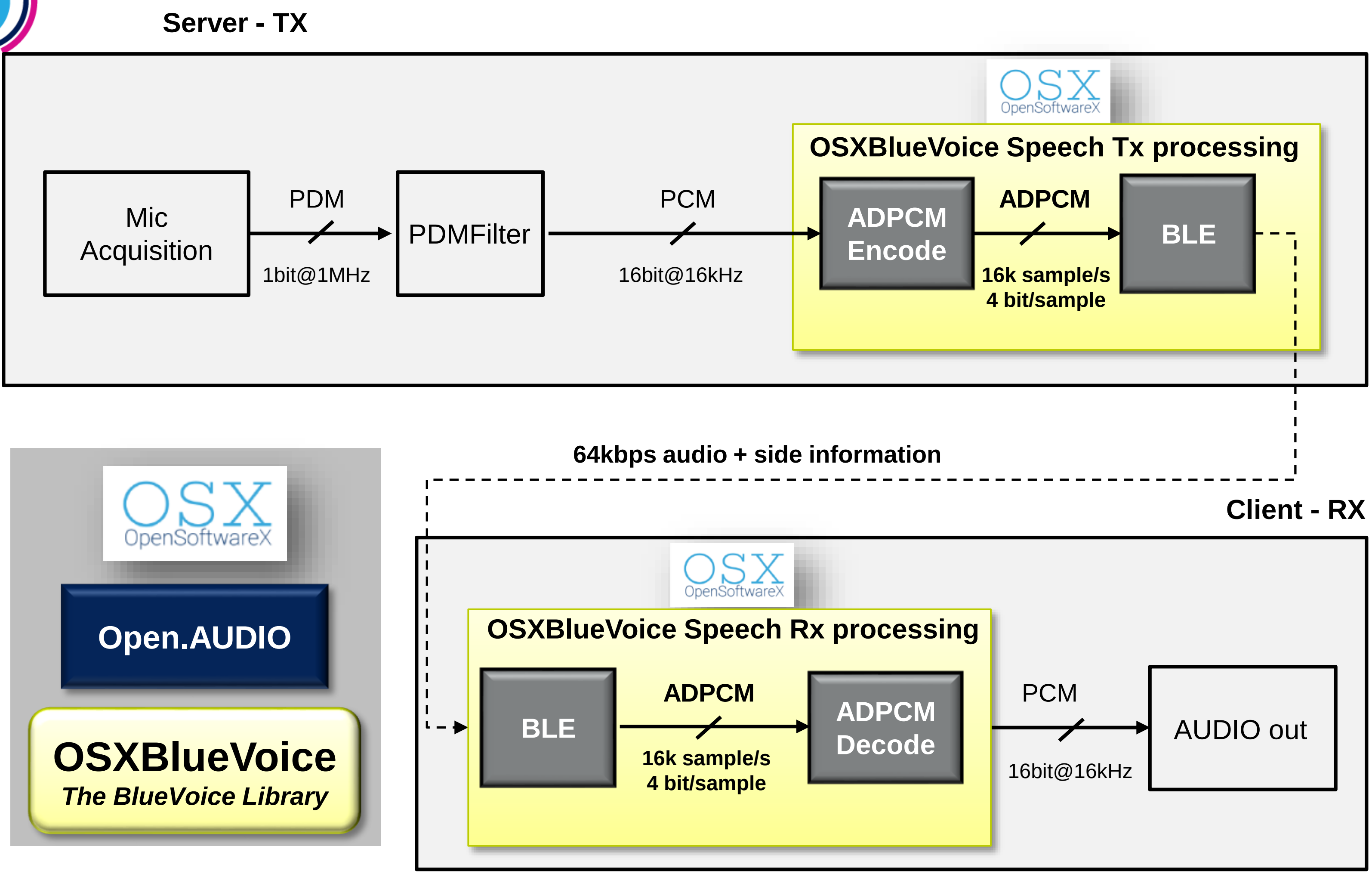


life.augmented

Open.AUDIO OSXBlueVoice library

Voice over BLE Vendor Specific Profile library

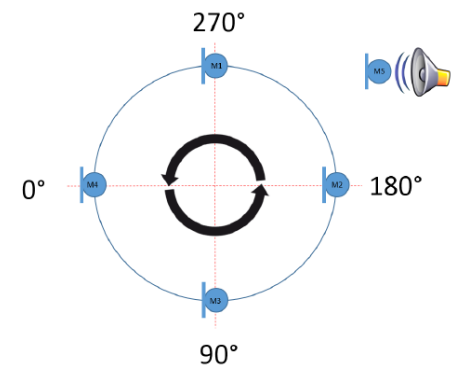
www.st.com/bluevoice



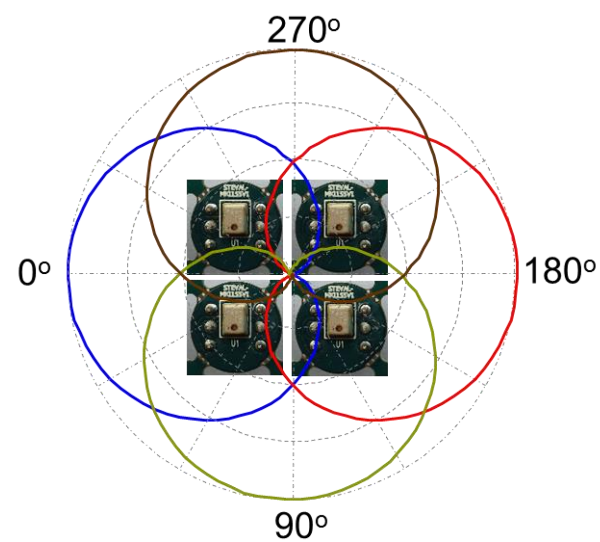
life.augmented

MEMS MICs SW libraries soon to join the Open.AUDIO family

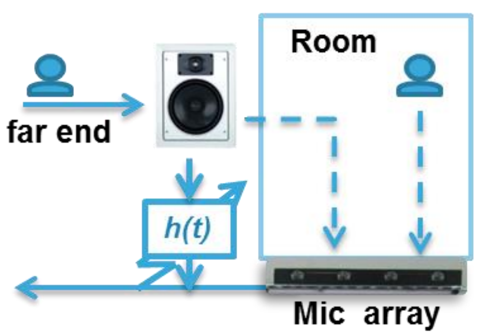
- SOUND SOURCE LOCALIZATION
  - using a MEMS microphone array, it estimates the angle of arrival of audio signal



- BEAMFORMING
  - using a MEMS (omnidirectional) microphone array, it creates a virtual directional microphone



- AEC
  - Acoustic Echo Cancelation (Speex based)

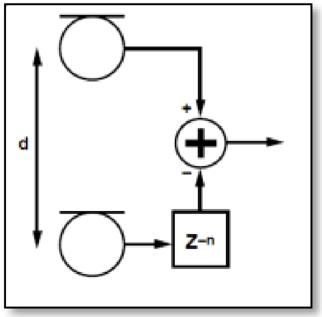


Open.AUDIO
Beamforming

life.augmented

Open.AUDIO Beamforming Library

- Input: 2 PDM signals from ST digital MEMS microphones
  - PDM to PCM conversion integrated for high efficiency
  - PDM domain processing achieves very high temporal resolution (no PCM sample interpolation)



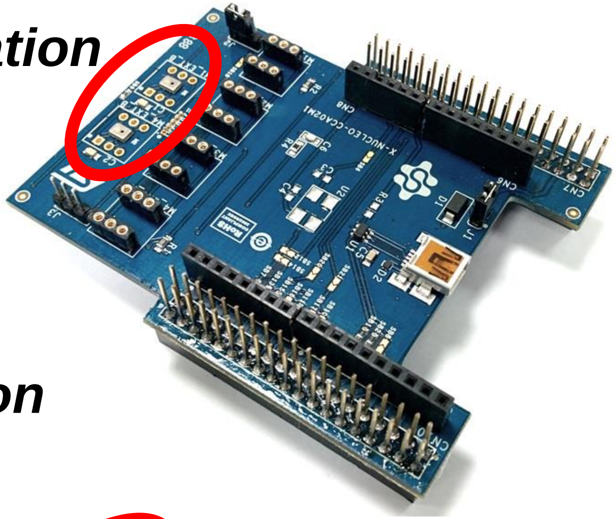
- Based on Differential Microphone Array configuration, with microphones spaced *up to* 21 mm.
  - Compatible 'by default' with the X-NUCLEO-CCA02M1 STM32Nucleo Expansion board
  - Easily customizable using plugin Microphones Coupon boards

- The library supports 4 algorithm options:
  - "Cardioid basic",
  - "Cardioid Denoise"
  - "Strong"
  - "ASR ready"

life.augmented

X-Nucleo based Beamforming: basic configurations

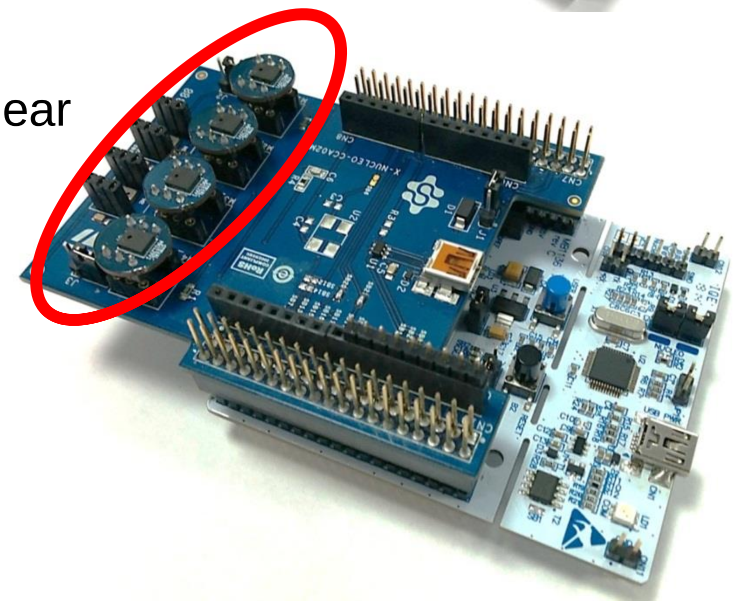
2-microphone configuration based on the onboard MP34DT01-M



4-microphone configuration based on MEMS microphone coupon such as STEVAL-MKI155V1

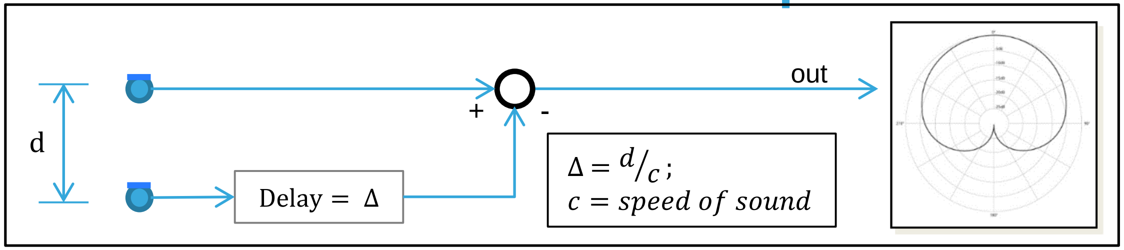
Squared

Linear

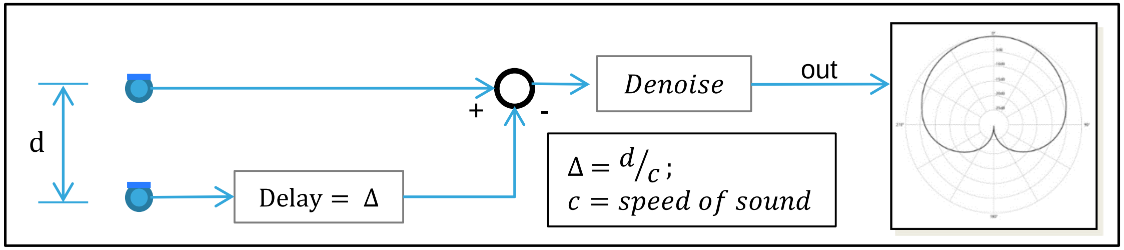


Beamforming Library – Algorithm options

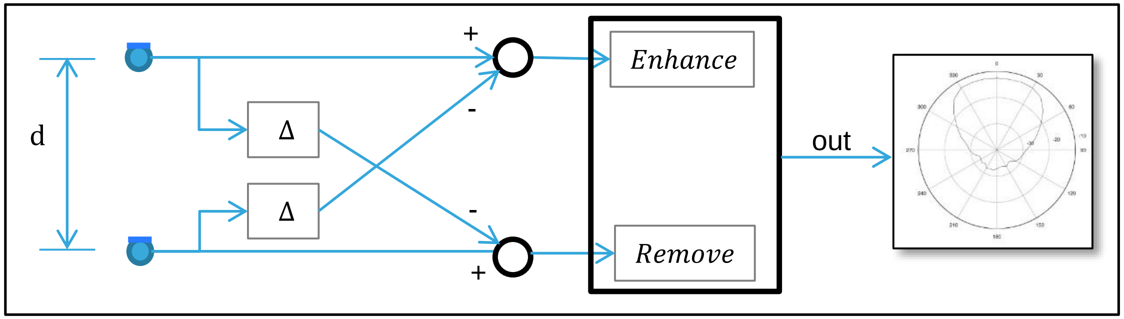
- Cardioid basic: 1st -order Differential Microphone Array (DMA)



- Cardioid denoise: a denoise filter is added to the end fire beam forming output



- Strong: back to back cardioid and adaptive noise removal filter



- ASR ready: conceptually equivalent to the Strong optimization, but specifically optimized for Automatic Speech Recognition applications.

Beamforming Profiling

- MIPS footprint has been computed when input signal is in PDM format. Both generic distance and the special case when d = 2.1 cm are shown.

	d < 2.1 cm	d = 2.1 cm
Level	**CPU (MIPS)**	**CPU (MIPS)**
Basic Cardioid	11.5	9.7
Cardioid Denoise	41.2	39.5
Strong	84.3	73.4
ASR Ready	54.3	43.4

- Audio frame corrisponds to 128 PCM sample, equal to 8 ms.

- RAM footprints varies with the level of optimization, as follows:

Level	RAM (Bytes)
Basic Cardioid	2580
Cardioid Denoise	20892
Strong	41212
ASR Ready	22900

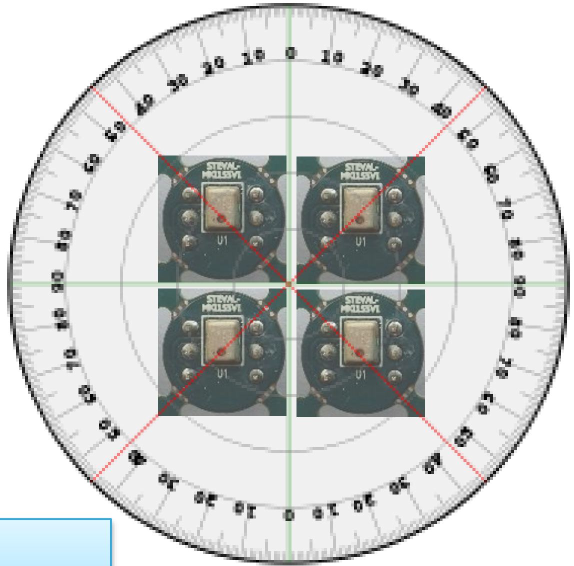
- Profiling done using IAR Embedded workbench, version 7.30; optimization set to High Speed

life.augmented

Open.AUDIO
Sound Source Localization

Sound Localization

- Signals are acquired by one or two *couples* of microphones in order to estimate the sound direction of arrival
  - 360 degrees range with 4 microphones
  - 180 degrees range with 2 microphones



- Up to 1 degree theoretical resolution
  - Selectable resolution
- Two algorithms implemented
  - **XCORR:**

    low-MIPS and low-resolution
    Supports cm-sized microphone arrays
  - **GCC-PHAT:**

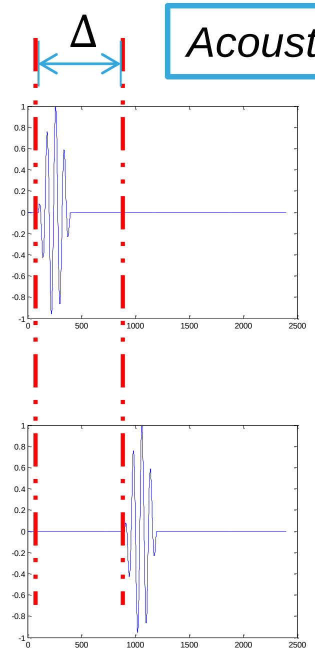
    Supports mm-sized Differential Arrays
- Simple Voice activity detector included, based on energy threshold

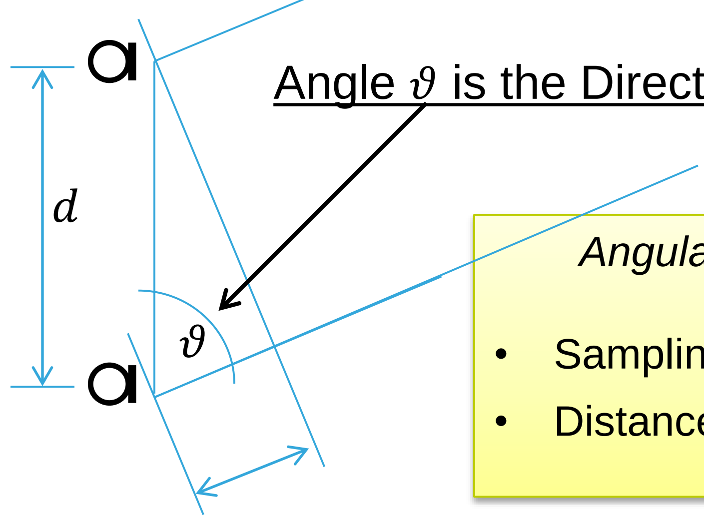
Sound Direction of Arrival

X-CORR: Cross correlation algorithm
- estimates the acoustic delay Δ
- Derives angle ϑ (DoA) from Δ

Far Field Sound Source

Acoustic Delay





<u>Angle ϑ is the Direction of Arrival</u>

$$\Delta = d \cos \vartheta$$

Angular resolution depends on:

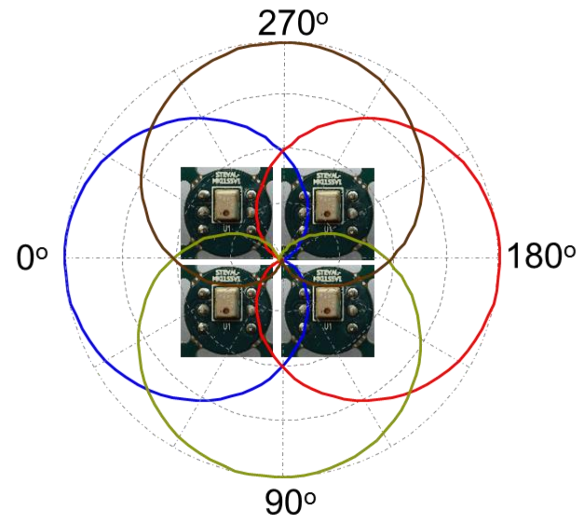
- Sampling frequency Fs
- Distance *d* between microphones

Source Localization Profiling

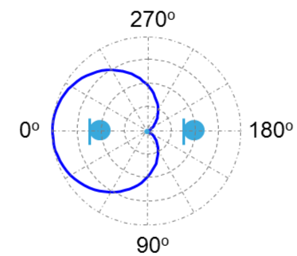
- RAM, FLASH and MIPS footprint has been computed for a continuous execution of the source localization library at every input frame
- In typical applications a simple and effective optimization can be implemented by reducing the frequency of localization update, leading to a lower MIPS count.
- Audio frame corresponds to 512 PCM sample at 16 KHz (32 ms).
- Profiling was done using IAR Embedded Workbench, version 7.30; optimization set to High Speed, based on target STM32F401 @ 84MHz

<table>
<tr><th>Algorithm</th><th>Microphones</th><th>Resolution</th><th>CPU (MIPS)</th><th>FLASH (Bytes)</th><th>RAM (Bytes)</th></tr>
<tr><td>XCORR</td><td>2</td><td>30 degrees (distance = 0.08 m)</td><td>2.2</td><td rowspan="10">3424 + 24714 (*)
(*) for CMSIS DSP software library by ARM, code and tables.</td><td rowspan="4">1161+ 2048*
Channel Number</td></tr>
<tr><td>XCORR</td><td>4</td><td>30 degrees (distance = 0.08 m)</td><td>4.5</td></tr>
<tr><td>XCORR</td><td>2</td><td>12 degrees (distance = 0.16 m)</td><td>4.0</td></tr>
<tr><td>XCORR</td><td>4</td><td>12 degrees (distance = 0.16 m)</td><td>7.9</td></tr>
<tr><td>GCC-PHAT</td><td>2</td><td>1 degree</td><td>58.4</td><td rowspan="6">9236 + 4096*
Channel Number</td></tr>
<tr><td>GCC-PHAT</td><td>4</td><td>1 degree</td><td>116.2</td></tr>
<tr><td>GCC-PHAT</td><td>2</td><td>2 degrees</td><td>31.8</td></tr>
<tr><td>GCC-PHAT</td><td>4</td><td>2 degrees</td><td>63.0</td></tr>
<tr><td>GCC-PHAT</td><td>2</td><td>6 degrees</td><td>14.1</td></tr>
<tr><td>GCC-PHAT</td><td>4</td><td>6 degrees</td><td>27.6</td></tr>
</table>

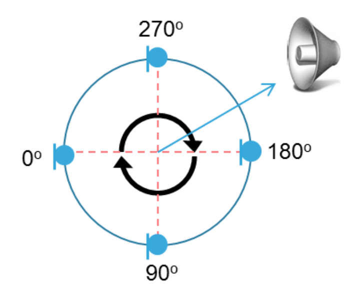
Beamforming and Localization development system



- BEAMFORMING
  - using a MEMS microphone (omnidirectional) array, the library enables to create a virtual directional microphone



- SOUND SOURCE LOCALIZATION
  - using a MEMS microphone array, it can estimate the angle of arrival of the audio signal

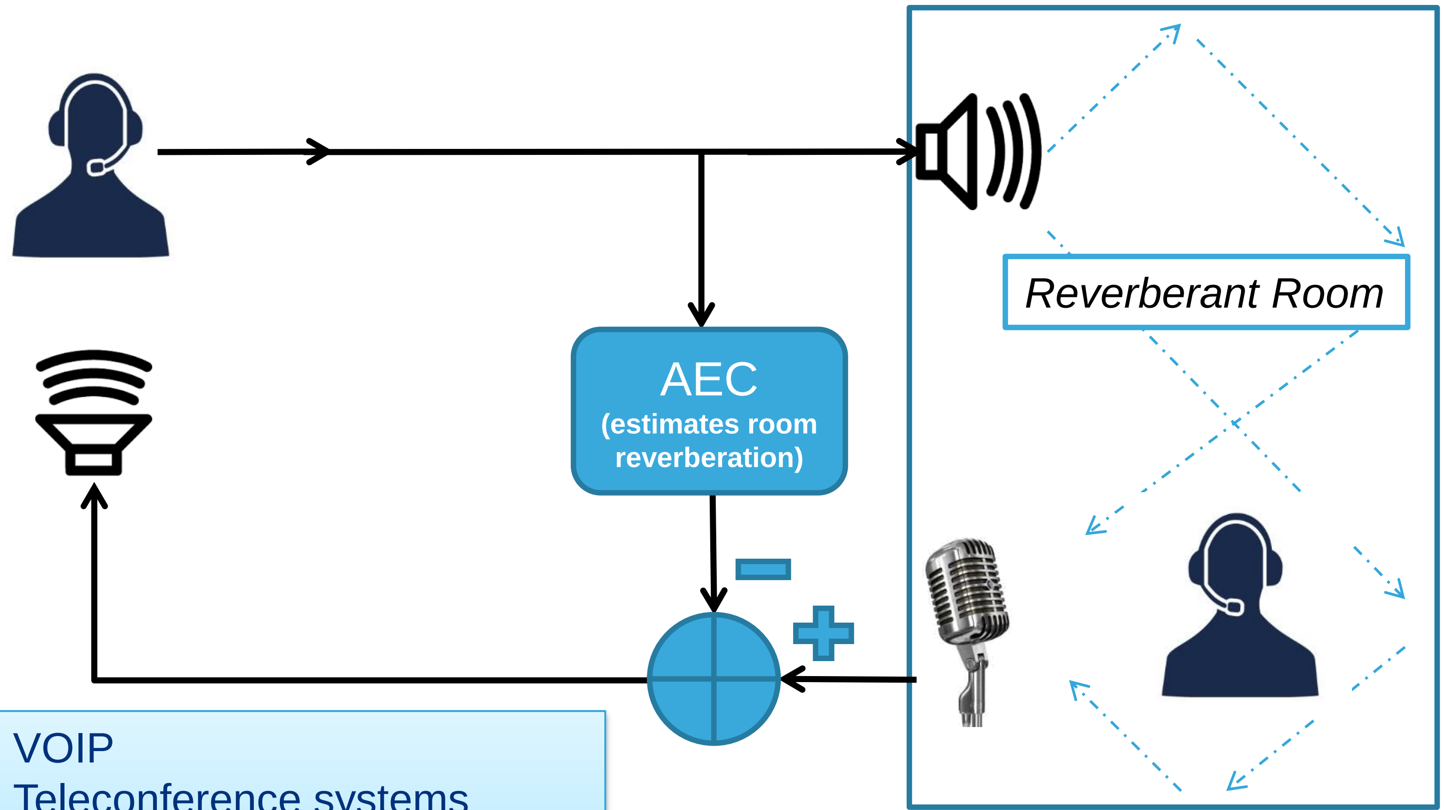


Open.AUDIO
Acoustic Echo Cancellation

life.augmented

Acoustic echo cancellation

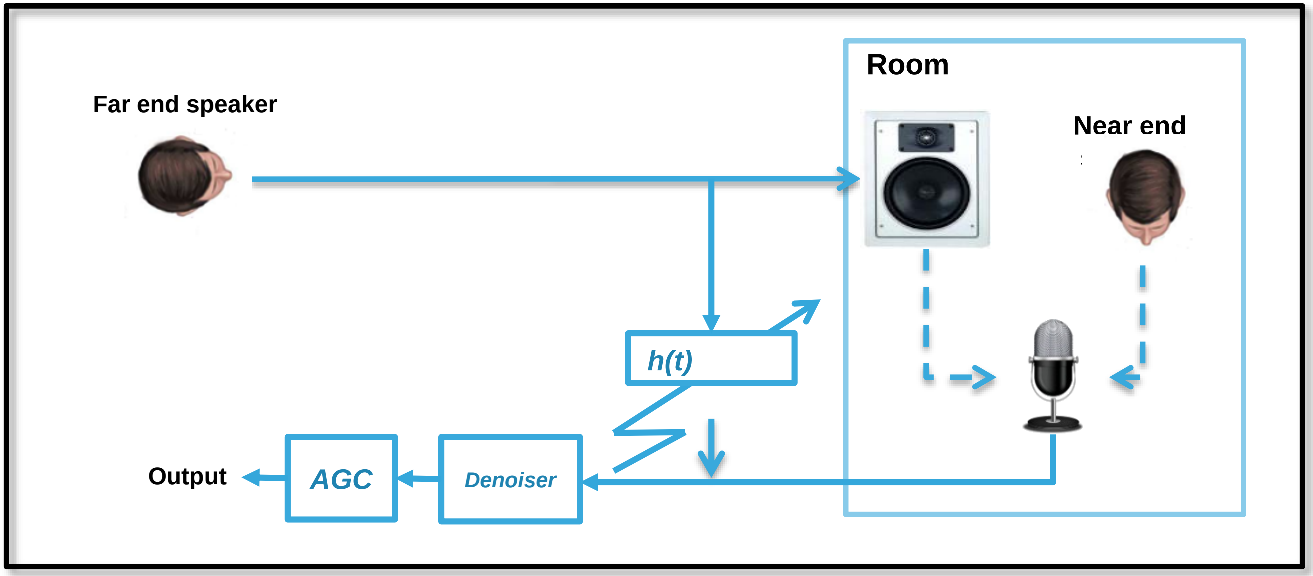
REMOVE FAR END ECHO IN NEAR END TRANSMISSION



- ✓ VOIP
- ✓ Teleconference systems
  - Business
  - Residential
  - Consumer
  - Automotive

Acoustic echo cancellation

- Acoustic Echo Cancellation removes a *far end* signal from the audio acquired by a microphone.
- Typical use: when a microphone and a loudspeaker coexist in the same environment and are driven by the same microcontroller.
- The Open.AUDIO AEC library is based on the Open Source Speex suite and includes:
  - AEC filter
  - Adjustable filter tail length
  - White noise filter
  - Simple automatic gain control



AEC - profiling

- Profiling has been done in order to estimate MIPS, RAM and FLASH consumption.
- The tool chain used for the tests is IAR Embedded Workbench 7.30, with optimization set to High, speed. Target adopted is a STM32F407 running at 168 MHz, with a total amount of 210 MIPS.
- For the test, filter tail length has been set to 128

Options	CPU (MIPS)	FLASH (Bytes)
AEC	54.7	35507
AEC + Denoiser	92.3	
AEC + Denoiser + AGC	97.7	

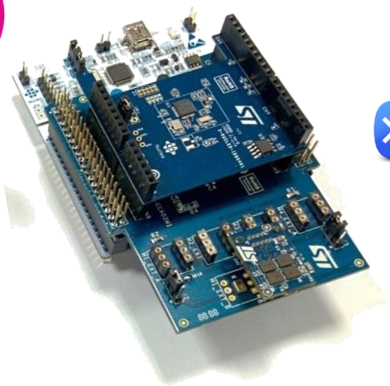
- RAM utilization depends on the chosen tail length and on the activated options. For example, using a tail length of 128 with all the options activated the library needs about 40 KB.

Advanced Demonstrators

BlueVoice + Speech Recognition

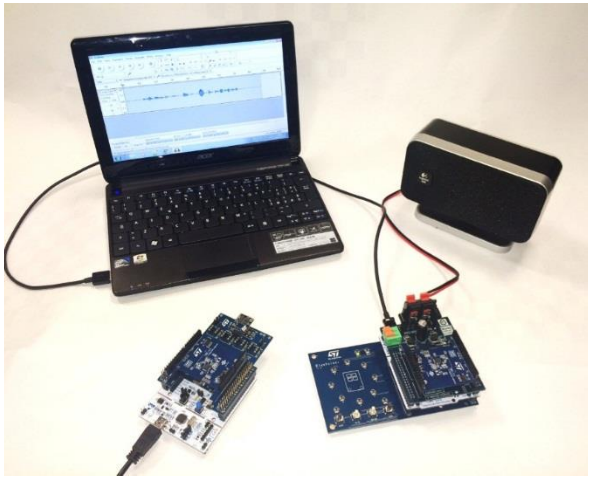
cloud-based ASR

Voice Capture - Tx



Audio
Over
ble

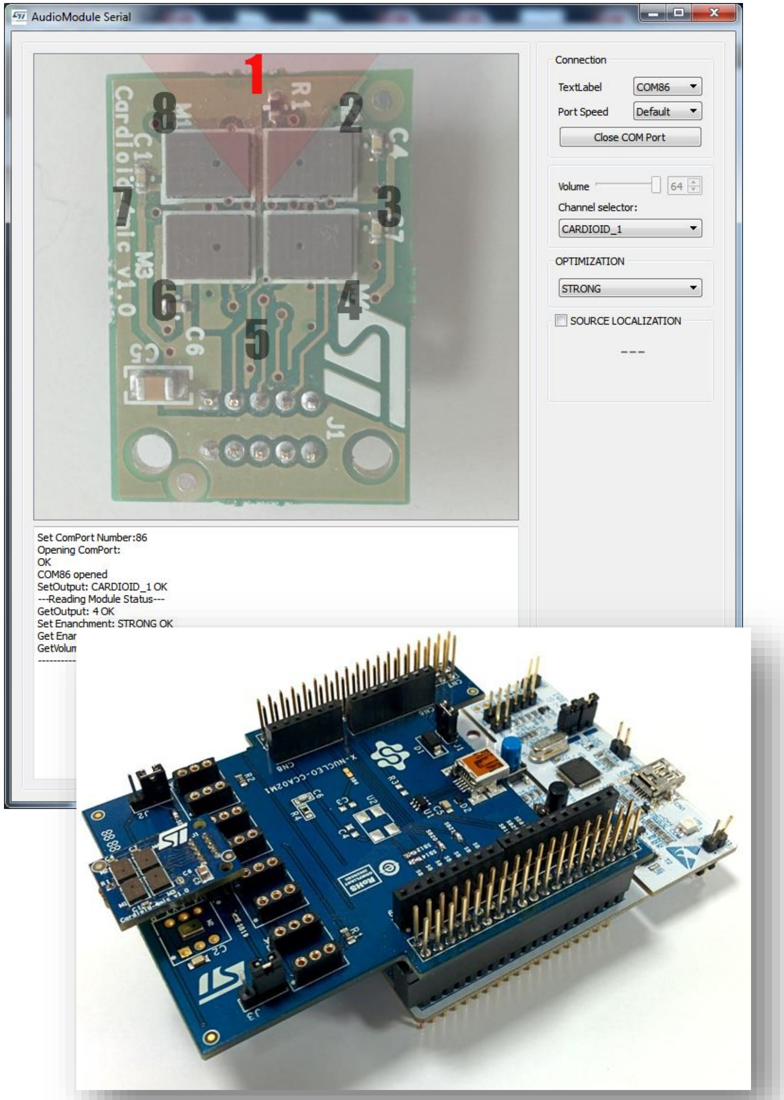
Central Unit - Rx



BlueVoice processing brings no degradation of performance for 3rd party ASR

Beamforming and Localization Demo system

- Hardware
  - Nucleo-F401RE (STM32 board)
  - X-NUCLEO-CCA02M1 (AUDIO IN expansion board)
  - µ-4 array hosting 4 MP34DT01 digital MEMS microphones
- Firmware
  - Microphone acquisition
  - Embedded Beamforming
  - Embedded Source Localization
  - Composite USB output:
    - Audio class (audio streaming)
    - Virtual COM Port (configuration)
- Software
  - PC control software to configure:
    - Optimizations
    - Beam direction
    - Source localization
    - ASR (3rd party embedded solution)



Thank you!