Introduction

The Qualcomm QCS6490 is a multi-core microprocessor featuring multiple powerful ARM Cortex-A cores, which is targeted towards autonomous robotics, smart vision, and industrial automation.

This application note describes how the QCS6490 can be used to test the Soundskrit direction of arrival (DOA) algorithm with a Butterfly demo board.

Direction of Arrival

The Soundskrit direction of arrival algorithm uses multiple microphones to determine the origin of sound waves.

Specifically, its two-dimensional version, illustrated in Figure 1, relies on two orthogonal dipole microphones and an omnidirectional microphone to carry out DOA estimations.

Figure 1: Direction of arrival running on a Beetle demo kit microphone.

The algorithm uses phase differences and relative microphone intensities to determine the angle of arrival, as shown in Figure 2.

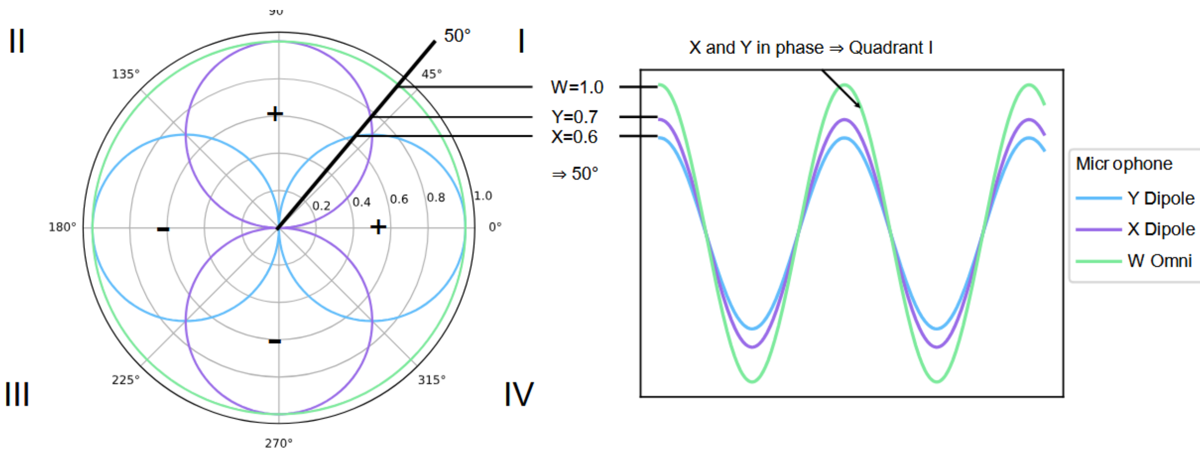


Figure 2: DOA algorithm estimating the angle of a 50-degree incident sound.

Test Hardware

Development Kit

This application note uses the [Thundercomm TurboX C6490 development kit] to evaluate the Soundskrit DOA algorithm on the Qualcomm QCS649 microprocessor running the Linux operating system.

Its Qualcomm Kryo™ 670 CPU contains multiple cores operating at different performance levels:

- Kryo Gold Plus: 1x high-performance core up to 2.7 GHz,
- Kryo Gold: 3x high-performance cores up to 2.4 GHz,
- Kryo Silver: 4x low-power cores up to 1.9 GHz.

This development kit, which features 8GB of RAM, is composed of the following boards:

- Audio board: hosts the audio codecs and audio connectors,
- Carrier board: connects all boards together,
- Interposer board: contains the microprocessor and RAM module.

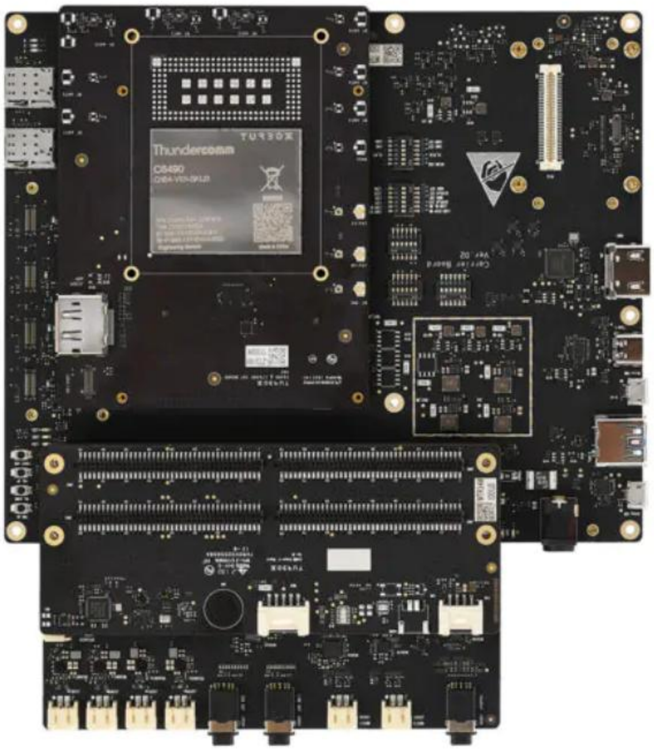


Figure 3: Thundercomm TurboX C6490 development kit.

Microphones

The QCS6490 is paired with a Soundskrit Butterfly demo board. This module is specifically designed to run a DOA algorithm, and features an array of microphone:

- 2x Soundskrit SKR0710 directional microphones for XY DOA,
- 1x additional Soundskrit SKR0610 directional microphone for XYZ DOA,
- 1x Knowles SPH0690LM4H-1 omnidirectional microphone acting as a reference.

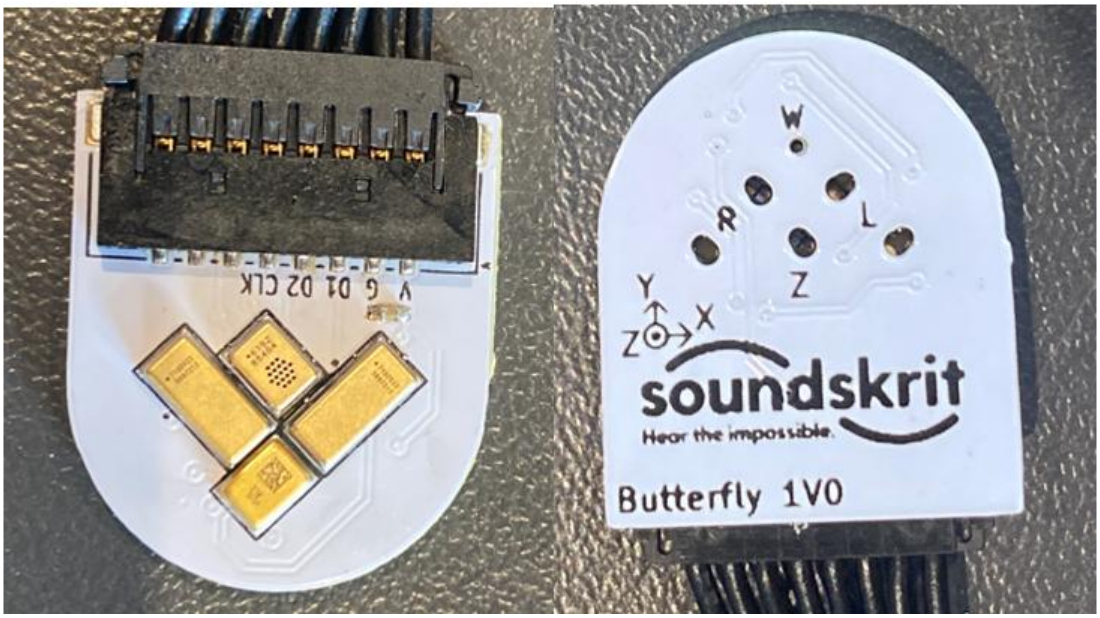


Figure 4: Soundskrit Butterfly demo board.

These microphones are arranged as two PDM pairs sharing the same VCC, GND and clock:

- Pair D1: L and W microphones,
- Pair D2: Z and R microphones.

Board-to-Board Connections

In order to connect the Butterfly demo board to the TurboX C6490 development kit, some soldering is needed, as the board does not directly expose the necessary signals. The solder points are described in Table 1, and can be seen in Figure 5.

A [Molex 151320806] cable is recommended to connect both boards.

Location	Audio Board Connector Designator	Connector Pin	Butterfly Signal
Audio board	J901	5	G
Audio board	J901	45	V
Audio board	SW3400	1	CLK
Audio board	SW3400	2	D1
Audio board	SW3401	6	D2

Table 1: TurboX C6490 development kit solder points.

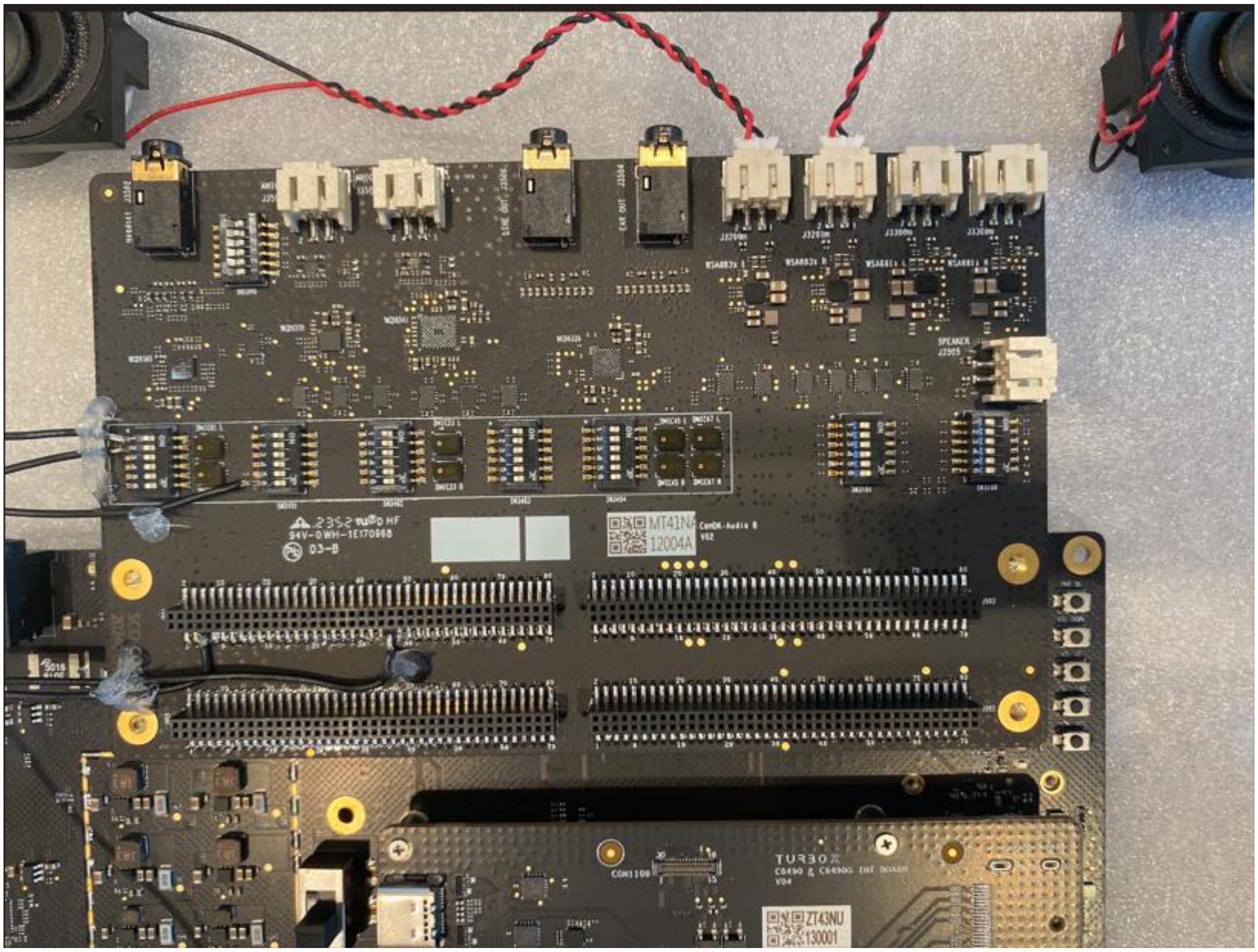


Figure 5: Soundskrit Butterfly demo board connected to the TurboX development board.

Audio Board Switch Configuration

The audio board switches of the development kit must also be configured to support the externally-soldered microphones. The differences over the configuration recommended by Thundercomm are highlighted **in bold** in Table 2.

Location	Switch Designator	Configuration
Carrier board	SW1301	OFF-OFF-OFF-OFF-OFF-OFF
Carrier board	SW2200	OFF-ON-ON-OFF-OFF-OFF
Audio board	SW3100	OFF-OFF-OFF-OFF-OFF-OFF
Audio board	SW3101	OFF-OFF-OFF-ON-OFF-ON
Audio board	SW3400	**OFF**-**OFF**-OFF-OFF-OFF-OFF
Audio board	SW3401	OFF-OFF-OFF-OFF-**OFF**-**OFF**
Audio board	SW3402	OFF-OFF-OFF-OFF-OFF-OFF
Audio board	SW3403	OFF-OFF-ON-ON-OFF-OFF
Audio board	SW3404	OFF-OFF-OFF-OFF-OFF-OFF
Audio board	SW3500	OFF-OFF-OFF-OFF-ON-OFF

Table 2: Thundercomm TurboX C6490 development kit switch configuration.

Operating System

This application note uses the Ubuntu 20.04.3 Linux image provided by Thundercomm (FlatBuild_TurboX-C6490_xx.xx_LU1.0.R.debug.CS.r002002.zip).

Linux relies on the Advanced Linux Sound Architecture (ALSA) to interface at a low level with audio hardware, for configuration, recording and playback. The code and commands contained in this section make use of this framework.

CLI Recording using ALSA

This section describes how the QCS6490 can be configured to record 4 microphones simultaneously on the command line, using the `amixer` and `arecord` ALSA tools.

Configuring ALSA to record 4 microphone channels simultaneously

```
# Configure the decimator inputs to digital microphones
$ amixer -c 0 cset name='TX DEC0 MUX' 'MSM_DMIC'
$ amixer -c 0 cset name='TX DEC1 MUX' 'MSM_DMIC'
$ amixer -c 0 cset name='TX DEC2 MUX' 'MSM_DMIC'
$ amixer -c 0 cset name='TX DEC3 MUX' 'MSM_DMIC'

# Configure the digital microphone multiplexers with the right inputs
$ amixer -c 0 cset name='TX DMIC MUX0' 'DMIC0'
$ amixer -c 0 cset name='TX DMIC MUX1' 'DMIC1'
$ amixer -c 0 cset name='TX DMIC MUX2' 'DMIC2'
$ amixer -c 0 cset name='TX DMIC MUX3' 'DMIC3'

# Configure input as 4-channel signed 16-bit samples at 48kHz
$ amixer -c 0 cset name='TX_CDC_DMA_TX_3 Channels' 'Four'
$ amixer -c 0 cset name='TX_CDC_DMA_TX_3 SampleRate' 'KHZ_48'
$ amixer -c 0 cset name='TX_CDC_DMA_TX_3 Format' 'S16_LE'

# Enable the 4 decimators connected to the 4 digital microphones
$ amixer -c 0 cset name='TX_AIF1_CAP Mixer DEC0' 1
$ amixer -c 0 cset name='TX_AIF1_CAP Mixer DEC1' 1
$ amixer -c 0 cset name='TX_AIF1_CAP Mixer DEC2' 1
$ amixer -c 0 cset name='TX_AIF1_CAP Mixer DEC3' 1

# Set microphone input volume
$ amixer -c 0 cset name='TX_DEC0 Volume' 84
$ amixer -c 0 cset name='TX_DEC1 Volume' 84
$ amixer -c 0 cset name='TX_DEC2 Volume' 84
$ amixer -c 0 cset name='TX_DEC3 Volume' 84

# Enable recording of a 4-channel stream
$ amixer -c 0 cset name='MultiMedia1 Mixer TX_CDC_DMA_TX_3' 1
```

Recording 4-channel audio

```
# Record 4-channel signed 16-bit audio at 48kHz to a WAV file
$ arecord -c 4 -r 48000 -f S16_LE dmics.wav
```

Real-Time DOA in C using Libasound (ALSA)

This section relies on a Soundskrit test program and signal processing library, which need to be obtained from Soundskrit separately.

You will need to install Make and the ALSA library onto the development board, before compiling and running the test program:

```
$ apt-get install make libasound2-dev

$ unzip doa_butterfly_qcs6490_demo.zip
$ cd doa_butterfly_qcs6490_demo/

$ make

$ ./doa_butterfly_qcs6490_demo
```

You should see the following output:

```
Audio device opened successfully
Library initialized
Mixer configuration complete
Hardware parameters applied to device:
  Access type: ACCESS_RW_INTERLEAVED
  Format: FORMAT_S16_LE
  Sample Rate: 48000
  Channels: 4
  Period size: 6000
  Buffer size: 24000
Recording for 20 seconds...
Angle: 0 deg
Angle: 200 deg
Angle: 209 deg
Angle: 321 deg
Angle: 350 deg
Angle: 342 deg
Angle: 347 deg
Angle: 353 deg
Angle: 349 deg
Angle: 25 deg
Angle: 65 deg
Angle: 86 deg

<...>

Finished recording
Microphone data written to WAV file
Beamformer data written to WAV file
Library clean-up complete
Cleanup complete
```


Performance Evaluation

This section provides information about the Soundskrit DOA algorithm's performance, in terms of CPU time and memory.

Methodology

Block processing time is defined as the time taken inside the `skblock_doa_butterfly_process()` function. It is estimated by averaging the function's runtime over 400,000 blocks of random data, executing on CPU7 (Kryo Gold Plus core) running at 2.7 GHz using the "performance" governor.

The MPCS is estimated using:

```
BLOCK_SIZE = 256

SAMPLING_RATE = 48000

BLOCK_PERIOD = BLOCK_SIZE / SAMPLING_RATE = 0.005333333

MCPS = CPU_FREQ_MHZ * BLOCK_PROCESSING_TIME / BLOCK_PERIOD
```

The heap usage is evaluated using `mallinfo()` before and after calling the `skblock_doa_butterfly_create()` function. The static memory usage is evaluated by calling `sizeof()` on the internal algorithm data structure.

Results

Configuration	**Block Processing Time** **(us)**	**Million Cycles per Second** **(MCPS)**	**Memory Usage** **(kB)**
Butterfly DOA + cardioid beamforming	50.7	25.7	Heap: 16.0 Static: 56.5

Table 3: Performance metrics for the DOA algorithm.

Due to the block size, the algorithmic latency (`BLOCK_SIZE / SAMPLING_RATE`) of the Soundskrit DOA and beamforming module is 5.3 ms.

Troubleshooting

If you notice that only two channels are populated in the WAV recordings, you can play a WAV file and try again:

```
$ paplay short_silence.wav
```

Additional Support

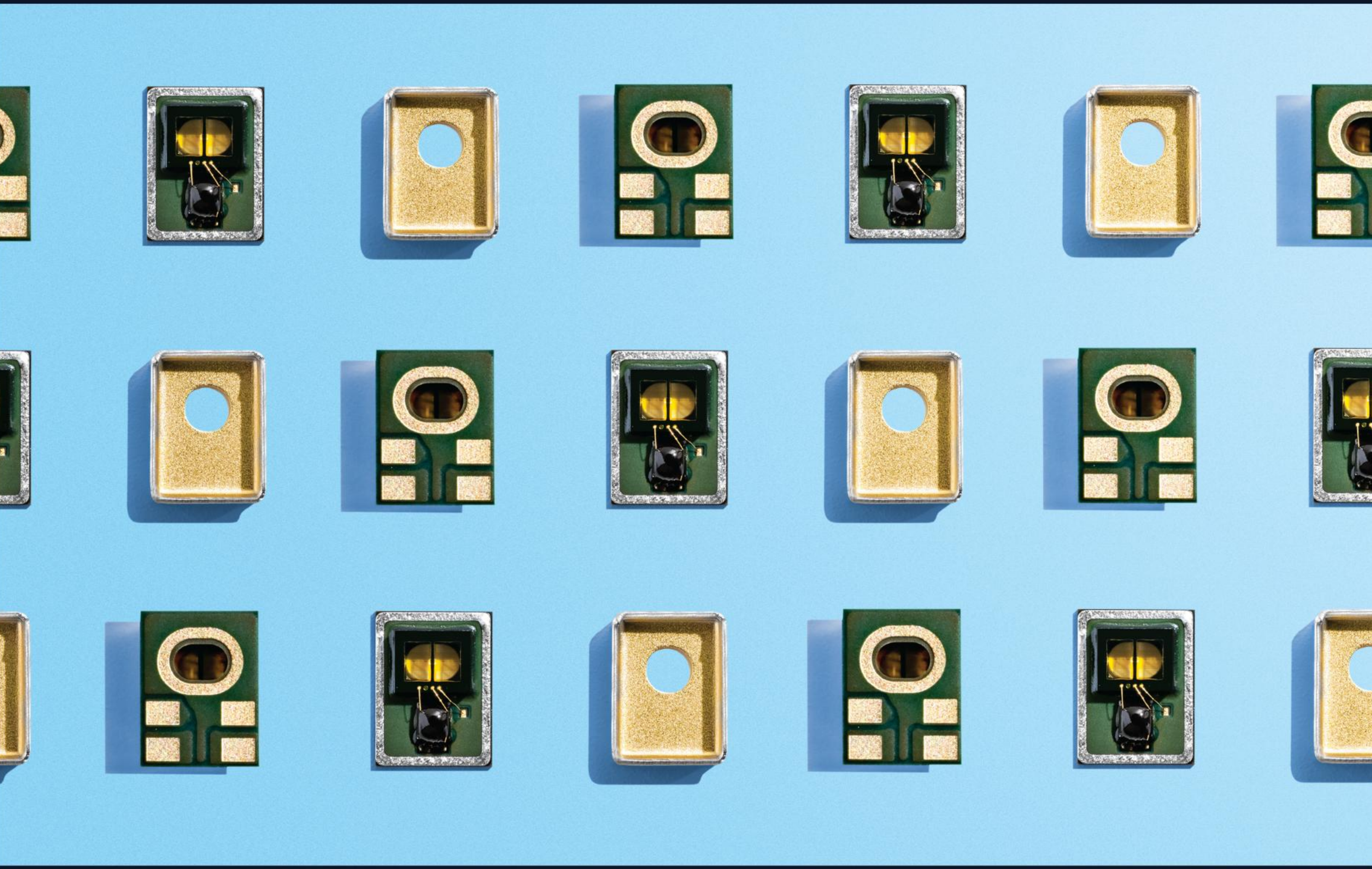
For additional design and applications support, please reach out to applications@soundskrit.ca.

Revision History

Revision Label	Revision Date	Sections Revised
-	September 2025	Initial release

Soundskrit developed the first high-performance directional MEMS microphone on the market, leveraging years of research in bio-inspired MEMS based on how spiders and other insects in nature hear. In combination with Soundskrit’s in-house audio processing algorithms, directional microphones can be used to capture and isolate any sound in an environment with a fraction of the size, power, and computation of traditional omnidirectional-based microphone arrays.

Soundskrit was founded in 2019 and is headquartered in Montreal, Quebec with an R&D facility in Ann Arbor, Michigan.



soundskrit

Hear the impossible.