Episode Book

EPISODE #6683

How Android Phones Fake a Microphone Array

PUBLISHED	19 November 2026 at 14:42 UTC
DURATION	28:58
LISTEN	myweirdprompts.com/episode/android-phone-microphone-array-14495/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.36
SOURCES	25 cited by the research (listed at the end)
BOOK GENERATED	19 November 2026 at 22:18 UTC

SYNOPSIS

A mid-range Android phone ships with just two to four microphone capsules, yet its pickup pattern shifts dramatically between a call, a voice memo, and a video. Rather than true steerable beamforming, phones compare level differences and coherence between a primary bottom mic and a secondary reference mic. By looking at hardware teardowns, Android's AudioSource menu, and research from a Hannover group's sixteen-capsule rig, this deep dive explains how apps dictate what your phone hears.

TRANSCRIPT



Corn

The microphone in your phone. Everyone assumes it's an afterthought — a pinhole, a bit of tin, whatever was left in the budget after the cameras. Daniel went digging and discovered otherwise, and his prompt this week is essentially: I thought my phone mic was mediocre, I now suspect it's a small stealth microphone array, and I want the deep dive. How beamforming actually works in a typical Android, how many mics are in there and where, how they get combined, and why the pickup pattern apparently changes completely between a call, a voice memo, and a video. His closing line is that he suspects "you can do a lot more with a phone for capturing audio than most people realise" — and I think he's right, in a way that surprised me too.



Herman

The starting point is a number that sounds too small to be true. A mid-range-or-better Android ships with two to four microphone capsules. That's it. Google's own hardware diagram for the Pixel nine Pro lists three entry points — one at the bottom near the charging port, one at the top edge, one on the rear in the camera bar. The Pixel six a has two. The OnePlus fifteen actually dropped from four to three compared to the thirteen.



Corn

A flagship losing a microphone between generations. Progress now means subtraction.



Herman

In this case, possibly, yes — and we'll come back to why, because it's more interesting than it sounds. The key layout, though, is the geometry. The primary mic sits at the bottom edge, near your mouth when you're on a call. The secondary sits at the top edge, maybe fifteen centimetres away, far from your mouth. And that long baseline along the phone's body is the entire trick. The top mic hears your voice much quieter than the bottom mic does, and it hears the traffic noise roughly the same. That difference is information.



Corn

So the second microphone isn't a spare. It's a reference — it exists to tell the phone what your voice isn't.



Herman

That's exactly what it is. It's worth pausing on how counterintuitive that is, because it inverts the way people talk about hardware. Normally, more components means more capability doing the same job — a second camera gives you a second focal length. A second microphone doesn't give you a second recording. It gives you a measurement of the environment, and the recording is then computed from the difference between two measurements. The capsule isn't capturing your voice so much as it's capturing the world your voice has to be extracted from.



Corn

It's less like a second lens and more like a control group.



Herman

A control group is a lovely way to put it, yes. The bottom mic is the experiment, the top mic is the control, and the noise suppression is the statistics. And the other thing to get straight immediately, because it reframes everything Daniel asked: the pickup pattern is not a property of the phone. The phone's own guidance for Pixels says the three microphones are identical, that they're not all used at once, and — this is Google's wording — "the quality of the recording depends on the app you use." That's a remarkably candid admission from a manufacturer. Same hardware, and the app decides what you get.



Corn

Which is a strange thing for a spec sheet. Imagine a camera where the sensor depended on which app opened it.



Herman

Well — that is actually what computational photography does too, but the audio version is far more dramatic, because Android doesn't just hand apps a microphone. It hands them a menu of AudioSources, and each one routes the capsules through a completely different processing chain. And unlike the camera, where the differences are mostly in tone mapping and sharpening, the audio differences are structural — different capsules, different gain, different suppression, sometimes a different theory of where the sound is coming from. Before we get into that menu, though, the count question deserves its own moment, because there's a reason nobody can just tell you "an average Android has three mics."



Corn

The compatibility rules.



Herman

The compatibility document requires that a handheld device "MUST include a microphone." Singular. There is no minimum count anywhere, and no requirement for noise suppression at all. So the count is a pure per-manufacturer decision — which is why teardowns are the only real source. The Oppo Find X9 Pro's subboard carries two mics under little red rubber gaskets for water protection. Some Motorolas have three. The range across the market is two to four, and nothing in the spec constrains it.



Corn

And the placement is equally unconstrained? Or is there a de facto standard — bottom, top, rear?



Herman

The bottom-primary placement is near-universal, because the call use case dictates it — the mic has to be near your mouth, and the mouth end of a phone is the bottom by long convention. Everything beyond that is up to the vendor. The rear placement in the camera bar is a Pixel signature, and it makes sense the moment you think about it: when you're shooting video, the sound you want is in front of the camera, and the camera faces away from you. A mic back there is pointed at the subject by geometry alone, no processing required. It's the one placement decision that's actually about acoustics rather than ergonomics.



Corn

Red rubber gaskets. The microphones get better waterproofing than most listeners.



Herman

Better than my old DJ headphones, certainly. Now — the mechanism. When an app opens the microphone, it names an AudioSource, and the audio policy manager maps that source to a capture path — which capsule, what gain, which preprocessing — according to rules the vendor writes. Those rules live in a config file in the vendor partition, and the defaults are documented: the voice communication source gets echo cancellation and noise suppression, the camcorder source gets automatic gain control. Even better, the



Herman (continued)

tuning intent is written down. The noise suppressor is, quote, "tuned for wind noise" for camcorder but "tuned for stationary noise" for voice communication. The gain control is "tuned for close-talk" on a call and "tuned for far-talk" for video.



Corn

One suppressor for wind, one for the fridge. The phone has different enemies depending on the app.



Herman

It really does. Your voice memo and your video call are fighting different battles, and the vendor knows it. And to make that concrete: think about what actually happens when you record video of your kid's recital. You're holding the phone at arm's length, the HVAC is roaring, someone three rows back is coughing, and the phone has to decide, moment to moment, whether the sound it's hearing is the recital or the room. The far-talk gain and the wind-tuned suppression are the vendor's pre-made answer to that exact scenario. The same phone on a call is answering a completely different question — is this the caller's voice or the bus going past the window — and the close-talk tuning is the pre-made answer to that one. Same capsules, different questions.



Corn

So the phone isn't just processing differently. It's asking different questions of the same signal.



Herman

Now here's where Daniel's beamforming question gets its honest answer, because the mental model most people have — the phone steering a beam around like radar — is wrong, and the literature is refreshingly blunt about it. A Granada group's paper on dual-microphone phones opens with the line that beamforming performance "can be insufficient if a small number of microphones is considered, as in the case of dual-microphone smartphones."



Corn

Two mics is not an array. It's a disagreement.



Herman

Ha — well, it's an array in the technical sense, but not a steerable one. What it actually does is this family of algorithms that's been standard for years. The close-talk version uses the power level difference between the two mics: your voice drops off fast with distance, so a sound that's loud at the bottom mic and quiet at the top mic is almost certainly you. The far-talk version uses a speech-presence-probability estimator plus the coherence between the two channels — your voice is coherent across both mics, diffuse room noise is less so. Then both approaches do the same final thing: they apply a gain to the main mic's signal, basically a Wiener filter, attenuating the moments where the statistics say "this is noise, not speech."



Corn

So it's not pointing a beam at my mouth. It's the bottom mic saying "I hear him," the top mic saying "barely," and the arithmetic between those two statements doing the filtering.



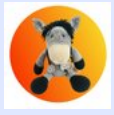
Herman

That's a fair rendering of it, yes. A level-difference and coherence discriminator with a single-channel cleanup on top. It's clever — clever, given the constraints — but it's worth knowing what a real beamforming array requires. A Hannover group built one as a research rig: sixteen MEMS capsules mounted on the back of a phone-sized device, an aperture of about eleven by six centimetres, evaluated on directivity index and side-lobe attenuation, then thinned with a genetic algorithm. That's what directional hearing costs. Your phone has a sixth of the capsules and a tenth of the aperture, if that.



Corn

We did cover this in the conference-speaker episode — the delay-and-sum maths behind beamforming proper, and how many elements you need before the pattern means anything. A phone fails that bar by a wide margin.



Herman

Comfortably. And there's a negative result I find delicious: a Delft group measured an actual phone's directional response, fed that measured pattern into a beamformer, and got no improvement. An earlier study measured nine smartphones and found the capsules' own unmodelled directivity degrades MVDR beamforming. The mics' directionality is a nuisance to be worked around, not a free win.



Corn

The one part of the phone that's accidentally directional, and it's the wrong kind.



Herman

So the pickup pattern isn't a property of the phone at all, it's a property of the app and the AudioSource it requests. Which is exactly the kind of thing a certain announcer would now try to sell you a fix for.

L

Larry

Are you tired of your phone's microphone array failing to steer a proper beam toward your mouth? Of course you are. Introducing the BeamFormer Buddy, the world's first clip-on acoustic baseline extender. It clips to your phone's long axis and adds two extra capsules, so your handheld device finally satisfies the pickup pattern you deserve. In a call, it suppresses wind. In a voice memo, it suppresses stationary noise. In a video, it suppresses your relatives. One hundred percent of users report a source-dependent pickup pattern they can feel. And now, the small print. Results vary by AudioSource, by vendor audio effects file, and by whether your phone actually has a third microphone. BeamFormer Buddy does not add microphones, does not contain microphones, and is a small piece of plastic. Not available in any store, not endorsed by any standards document, and not a substitute for a proper teardown. Ask for it by a name you cannot pronounce. BeamFormer Buddy.



Herman

Hold on, "one hundred percent of users report a pickup pattern they can feel" — the pickup pattern is by definition inaudible as a pattern, it's a...



Corn

Herman. It's a small piece of plastic. He said so himself, it's the one honest sentence in the read.



Herman

...Fine. Moving on. Where were we — the AudioSource menu, because that's where the pickup-pattern question gets answered, and Android is more explicit about it than most people realise. Since API twenty-nine there's a whole directionality interface. An app can tell the microphone to point toward the user — the screen side, "as in a selfie," that's the



Herman (continued)

documentation's own example. Away from the user, which the docs describe as "the taking a video of something else case."



Corn

The documentation writes "the taking a video of something else case," and you can hear the engineer who wrote it deciding not to explain further.



Herman

It's one of my favourite doc strings, honestly. There's also an external setting for off-device mics, and an unspecified setting that says don't do any directionality processing at all. And then — this is the control nobody talks about — a beam-width control. There's a method that takes a zoom value from minus one to plus one. Minus one is wide angle, zero is no zoom, plus one is maximum zoom. And the documented intent is that the audio focus follows the camera zoom — you zoom the lens in, the audio beam narrows onto what you're looking at.



Corn

Wait. That's the clearest evidence yet that there's real spatial filtering in there — the phone narrows its audio pickup as you zoom the lens — and it's just sitting in the API, undocumented by anyone outside the developer docs?



Herman

Sitting there since Android ten, as far as I can tell barely reported anywhere. And it confirms something: the vendor DSP is doing genuine spatial processing, not just gating between loud and quiet. And that gets us to the use-case walkthrough, which is really the heart of what Daniel's asking — what does the pattern actually do differently in a call, a memo, a video.



Corn

Go source by source. I want to hear how different they are.



Herman

A call uses voice communication. That's the most processed path on the device — echo cancellation, noise suppression tuned for stationary noise, gain tuned for close-talk, because the phone knows the mic is an inch from your mouth and it knows its own speaker is blasting the other caller at your ear at the same time. That echo canceller is the hardest-working block in the whole phone; Android made it a hard requirement on the voice path back in version ten.



Corn

And it's worth spelling out why it's so hard, right? The speaker and the mic are centimetres apart on the same slab of glass.



Herman

The phone's own speaker is, acoustically speaking, the loudest thing in the room as far as that bottom mic is concerned. And it's not just the direct sound — it's the sound bouncing off your cheek, off your hand, off the car window, all of it arriving a few milliseconds late and threatening to go back up the line so the other person hears themselves as an echo. The canceller has to model all of that, continuously, while the "room" changes every time you shift your grip. It's a solved-enough problem that nobody thinks about it, which is itself a triumph.



Corn

And a video?



Herman

Camcorder. Same hardware, opposite assumptions. Noise suppression tuned for wind — because you're outside, moving, the wind is hitting the top mic. Gain tuned for far-talk, because the subject is across the room, not at your face. And the capture is oriented "with the same orientation as the camera if available" — meaning if you flip to the front camera, the audio emphasis can flip with it.



Corn

So the same phone, in the same room, in the same minute — phone call and video — is running two different theories of where the sound is.



Herman

Two different theories of who you are, almost. And then there's the third one, the one that surprises people: voice recognition. That source is deliberately clean. AOSP specifies that effects and preprocessing are disabled by default on it, and the requirements are the strictest in the stack — flat frequency response within three decibels from a hundred hertz to four kilohertz, a defined sensitivity target, linear level tracking, distortion under one percent. The whole point is that an ASR model wants a predictable, boring signal, not a helpful one.



Corn

The speech model doesn't want the phone's help. It wants the phone to get out of the way.



Herman

And that's the interesting inversion. The "smart" path for a human listener, with all the suppression, would be terrible input for a model, because suppression removes information the model could have used. It's the same reason you don't pre-sharpen a photo you're about to feed to a machine-vision system — the model has its own ideas about what counts as signal. And then there's the fourth path, unprocessed, which is the raw one — a source added in Android seven that bypasses the DSP entirely, where supported. You get the capsules, whatever bit depth the hardware gives you, all available channels, and a level multiplier and nothing else.



Corn

Which is the one people get excited about — the "turn your phone into a measurement microphone" path.



Herman

And that's where I'll pump the brakes, because the enthusiasm outruns the spec. The compatibility requirement for unprocessed is a tolerance of plus or minus ten decibels from a hundred hertz to seven kilohertz, looser outside that band.



Corn

Ten decibels. That's not flat, that's a landscape.



Herman

It's a prairie. Fine for "raw signal," useless for calibrated acoustic measurement — a real measurement mic is specified in fractions of a decibel. Some blog posts do imply you can do sound-level metering off it, and the tolerance just doesn't support that. To give you a sense of the scale: plus or minus ten decibels means at some frequencies the phone could read a sound as roughly a tenth or ten times the actual pressure level and still be within spec. That's the difference between "quiet library" and "lawnmower" as far as a meter is concerned. But as a raw capture path for experimenting — recording a room, grabbing multichannel, feeding an analyser — it's legitimately powerful. The API even supports channel index masks beyond stereo, up to four channels in the sample code, though the positional channel mask is capped at two.



Corn

Multichannel in principle, stereo in practice. Which raises the messy question — when an app asks for a source, which physical capsule does it actually get? Is that guaranteed anywhere?



Herman

It is not, and this is where the tidy story falls apart. The accepted answer from people who've fought this in production is blunt: there's no reliable way of selecting whether you use the primary or secondary mic for a mono recording. The plain mic source usually means the bottom capsule, the camcorder source might mean a different one — but there's no guarantee, because it depends on the vendor's routing rules. And stereo is worse: there are well-documented phones where there was simply no way to activate a real stereo recording through the built-in capsules, the manufacturer's own camera app excepted. HTC One M7, the Nexus five, several others.



Corn

The API promises a request. The hardware delivers a shrug.



Herman

On a lot of devices, yes. Some phones could be coaxed into stereo with the camcorder source at forty-eight kilohertz, which is the kind of folklore that grows up around any API this under-specified. And notice what that folklore actually is — developers reverse-engineering the vendor's private routing table by trial and error, one phone at a time, and posting the results to forum threads that then outlive the phones they describe. The documentation tells you what you may ask for. The forum tells you what you'll get. Corn, let me turn it around on you, because you've recorded voice memos into this show's workflow for years — what Daniel's really asking is whether the phone in his pocket is a



Herman (continued)

serious capture device. You use one daily. Is it?



Corn

It depends which app I open, which is the whole point, but I'll say it differently: the phone is a serious capture device the way a Swiss Army knife is a serious knife. Nothing about the phone tells you what you're holding. The app opens, asks for a source by name, and the phone decides — bottom mic, top mic, both, wind suppression, close-talk gain — all of it invisible. My guess is that most people have never once thought about which source their memo app requests, and that single line of code is doing more for how they sound than the hardware.



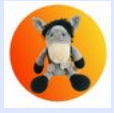
Herman

Do you know which source your memo app requests?



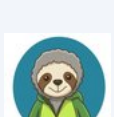
Corn

No. I'm going to find out, and that's the honest answer. But here's my follow-up question back at you, because you're the one who reads the teardowns: if the DSP is where the sophistication lives, and it's all vendor-proprietary, then who actually knows what any given phone is doing? Is there any public documentation of the actual beamforming algorithm in a shipping phone?



Herman

No. And that's a genuine gap, not a lazy answer. Android documents that the preprocessing is configured per source and what the tuning targets are, but the algorithm itself — whether it's the level-difference family or the coherence family or something proprietary on top — no vendor publishes. The academic literature describes the algorithm families, and we know phones do something in that territory, but which phone does which is unstated. We're inferring the engine from the exhaust.



Corn

Which loops back to the OnePlus dropping a mic. If the sophistication is software, the capsule count is a cost line, not a capability line.



Herman

That's how I read it, and I'd wager the removed capsule was the one contributing least to the DSP's model. Four to three isn't a retreat from quality; it's a bet that the software has caught up. And there's a precedent for that bet, actually — this is exactly what happened in computational photography. Sensors stopped being the differentiator once the software got good enough to compensate, and the marketing quietly shifted from megapixels to "computational" everything. Audio is running the same play a few years behind.



Corn

Fewer microphones, more maths. Every year the phone gets a little more imaginary.



Herman

Careful, that's how you end up with a clip-on plastic extender.



Corn

I'd have bought one in the nineties. Don't test me.



Herman

You'd have bought two, one to clip on and one to keep in the tin.



Hilbert

Which one of you said the pinhole was for water protection? That was you, Herman. I want to ask you something. Do either of you know what the third hole on a Pixel is for? Not the mic, not the ejector pin. The third one.



Corn

There's a third one?



Hilbert

There is on mine, and there wasn't supposed to be. I had a Pixel four a. Sold it last year. Its recordings went muffled in March — I filed a noise complaint with the council about the building works on Rehov Agrippas, and the noise officer, a Mr Dagan, kept rejecting my clips as "inaudible." Four submissions. So I did what anyone would do: I recorded a one-kilohertz test tone from the kitchen radio, forty seconds at a time, while probing each port with a wooden skewer wrapped in a single ply of tissue. When the tone jumped up by what the level meter said was eleven dB, I had my port. Turned out I'd been cleaning the wrong hole for five months — the ejector pinhole, diligently, weekly, with a pin. The mic port I'd never touched. It was packed solid.



Herman

Eleven decibels of skewer. That tracks, actually — a blocked port acts as an acoustic low-pass, and the level difference between occluded and clear runs right around that magnitude. The skewer was functioning as an occlusion probe. Technically sound method, Hilbert.



Hilbert

It was a kebab skewer, and it was the flat kind, which is why it took me three evenings. I kept the notes. Each test on its own page, the port probed, the reading, the date. Mr Dagan eventually accepted the fifth clip. He only asked whether my tone tests could be submitted as evidence, and I said no, they were for my file. Anyway, Herman, you spent the whole first half saying the app decides everything — the app decides nothing if the hole is blocked. The spec sheet talks about capsules. Nobody talks about the pinhole, and that's where it all lives or dies. These claws are useless for it, incidentally — a pinhole is built for a fingernail, and I have neither.



Corn

No, wait — the flat skewer is the detail I can't get past. Three evenings, and you didn't upgrade the tool.



Hilbert

The skewer was already in the drawer. Herman, one thing — you said the gaskets were red on the Oppo. On my Pixel they were black. So it's not even standard across the line.



Herman

It wouldn't be — the gasket colour's a supplier choice, not a spec. But you're right that the port geometry matters more than anyone documents. The pinhole is part of the acoustic path, and it's the one component with no substitute and no spec at all.



Corn

And it's the one failure mode that looks like a software problem. Muffled recordings — you'd blame the app, the source, the DSP, you'd factory reset the thing before you'd think to look at the hole.



Herman

That's the cruel part. Every layer we've talked about today — sources, routing, suppression — gives you somewhere else to point the finger. The pinhole gives you nowhere. And no teardown ever mentions whether it clogs, only how many millimetres deep it sits.



Corn

Which is the gap Daniel's question runs into at the physical layer. The one thing to take from all this, if you take one thing: the pickup pattern is not a property of your phone — it's a property of the app, the source it requests, and, as we've now established beyond argument, the state of the pinhole. Same hardware, four different microphones depending on who asked for it.



Herman

And a fifth one depending on whether the hole is clear.



Corn

And the sharpest single proof is that zoom control — the audio beam narrowing with the camera lens. If that exists in a shipping API, the phone's mic is a software-defined instrument, full stop.



Herman

So here's where I'm left. If the array is this small, the routing this vendor-specific, and the algorithm unpublished, then reproducible acoustic work on a phone is nearly impossible — every result is a result about one firmware on one model. And if flagships are shedding capsules while the DSP grows, the hardware is being optimised away on purpose. The question I'd leave with: what else is sitting in the audio stack that nobody's documenting? The zoom control hid in plain sight for years. There will be more of those.



Corn

Producer credit where it's due — Hilbert Flumingtop, who has apparently been running a one-man acoustics lab with kebab equipment. If you want more of this, try episode two thirty-three, How Math Gives Microphones Directional Ears; episode ten, How ASR Went From Frustration To … Whisper Magic; and episode eighteen sixty-one, Emergency Prep You Can Sing To. This has been My Weird Prompts — no question too obscure, no rabbit hole too deep. If today's episode told you something your phone was hiding from you, leave us a review — it's the fastest way other people find the show. See you next time, Herman.



Herman

See you, Corn. Watch your pinholes, everyone.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Android Developers, `MediaRecorder.AudioSource` API reference, retrieved 2026-11-19. [primary]**
https://developer.android.com/reference/android/media/MediaRecorder.AudioSource

2. **Android Developers, `MicrophoneDirection` API reference (API 29+), retrieved 2026-11-19. [primary]**
https://developer.android.com/reference/android/media/MicrophoneDirection

3. **AOSP, Configure preprocessing effects, retrieved 2026-11-19. [primary]**
https://source.android.com/docs/core/audio/implement-pre-processing

4. **AOSP, Microphone input, last updated 2026-09-24. [primary]**
https://source.android.com/docs/automotive/audio/microphone-input

5. **Android CDD section 7.8 Audio / 7.8.1 Microphone, retrieved 2026-11-19. [primary]**
https://android.googlesource.com/platform/compatibility/cdd/+/master/7_hardware-compatibility/7_8_audio.md

6. **Android CDD section 5.11 Capture for Unprocessed, retrieved 2026-11-19. [primary]**
https://android.googlesource.com/platform/compatibility/cdd/+/ad3d038/5_multimedia/5_11_unprocessed-audio.md

7. **Android 17 Compatibility Definition, retrieved 2026-11-19. [primary]**
   https://source.android.com/docs/compatibility/17/android-17-cdd

8. **Google, Pixel phone hardware diagram, retrieved 2026-11-19. [primary]**
   https://support.google.com/pixelphone/answer/7157629?hl=en

9. **Google, Pixel 9 Pro hardware tour, retrieved 2026-11-19. [primary]**
   https://pixel.withgoogle.com/Pixel_9_Pro/get-to-know-your-hardware

10. **Android Authority, OnePlus 15 teardown confirms the cameras are a spec downgrade, 2025-10-29.**
    https://www.androidauthority.com/oneplus-15-teardown-3611210/

11. **iFixit community answer, 2025-08-11.**
    https://www.ifixit.com/Answers/View/911414/Can+anyone+show+me+where+the+microphones+are+on+the+google+pixel+6a

12. **Phone Repair Guru, Oppo Find X9 Pro teardown, dated Nov 26 in page text.**
    https://www.phonerepairguru.com/news/i-tore-down-the-oppo-find-x9-pro-in-china-and-it-completely-blew-my-mind

13. **Poco F8 Ultra teardown video, retrieved 2026-11-19.**
    https://www.youtube.com/watch?v=I5rzpNmAasw

14. **Motorola Razr Ultra 2025 teardown video, retrieved 2026-11-19.**
    https://www.youtube.com/watch?v=m-cCTeQG814

15. **Martín-Doñas et al., A postfiltering approach for dual-microphone smartphones, IberSPEECH 2018, Barcelona, 21**
    23 Nov 2018.
    https://www.isca-archive.org/iberspeech_2018/martindonas18_iberspeech.pdf

16. **On near-field beamforming with smartphone-based ad-hoc microphone arrays, IWAENC 2014 (abstract).**
    https://doi.org/10.1109/iwaenc.2014.6953345

17. **TU Delft, On Determining Smartphone Microphone Directivity with Application to Beamforming.**
    http://resolver.tudelft.nl/uuid:fffbf440-6a2a-4b5c-b41e-d97a8770f766

18. **Poets, Schlieper, Preihs, Peissig, On the Design of a MEMS Microphone Array for a Mobile Beamforming Application, DAGA 2020, Hannover.**
    https://www.pub.dega-akustik.de/DAGA_2020/data/articles/000405.pdf

19. **Sensors 18(5):1467, Speech Enhancement of Mobile Devices Based on the Integration of a Dual Microphone Array and a Background Noise Elimination Algorithm.**
    https://www.mdpi.com/1424-8220/18/5/1467

20. **Wikipedia, last edited 2026-10-25.**
    https://en.wikipedia.org/wiki/Beamforming

21. **Wikipedia, last edited 2024-11-06.**
    https://en.wikipedia.org/wiki/Microphone_array

22. **Stack Overflow, accepted answer on selecting primary vs secondary mic.**
    https://stackoverflow.com/questions/15418267/how-to-access-the-second-mic-android-such-as-galaxy-3

23. **Stack Overflow, answer by Hartmut Pfitzinger, 2015-01-20, on device-dependent stereo capture.**
https://stackoverflow.com/questions/25982814/recording-stereo-with-audiorecord-in-android

24. **Stack Overflow, answer by Jerome on mic selection via AudioSource.**
https://stackoverflow.com/questions/68444421/how-to-select-physical-microphone-and-audio-recording-path-in-android

25. **Vuzix, Audio Subsystem, updated 2023-12-19.**
https://support.vuzix.com/docs/audio-subsystem-2