Episode Book

EPISODE #6234

Phone Mic Beats a Dictation Mic: How?

PUBLISHED	24 October 2026 at 09:07 UTC
DURATION	22:30
LISTEN	myweirdprompts.com/episode/phone-mic-dictation-setup-44697/
HOSTS	Corn, Daniel, Herman, Hilbert
SCRIPT MODEL	openai/gpt-6.1-sol
VOICES	chatterbox-regular+meorav (chatterbox-local)
PIPELINE	V5.20
SOURCES	16 cited by the research (listed at the end)
BOOK GENERATED	25 October 2026 at 06:19 UTC

SYNOPSIS

A dedicated dictation microphone losing to a phone is expensive feedback — so what happens when you try to keep using the phone's mic at your desktop? This episode digs into the gap between "sending audio" and "sending the audio that actually won the benchmark," why Android's recording path matters as much as the hardware, and what close-range measurement data says about distance, angle, and handling noise. Plus a practical, low-effort route to getting phone audio into your desktop dictation app using open-source tools and a virtual audio device.

TRANSCRIPT



Corn

A dedicated dictation microphone losing to a phone is expensive feedback.



Daniel

That's what I'm trying to work with. I dictate extensively, and while my workstation was unavailable during the move, I used my phone much more. Its internal microphone gives me the lowest word error rate of the microphones I've benchmarked. I'd like to keep using that microphone when I'm at my desktop or laptop, with a selector on the phone for where the audio goes.



Corn

Your phone, your laptop, or your desktop. Same microphone, different destination, without rebuilding the arrangement every time you change chairs.



Daniel

Yes. The phone stays with me, and I take it out to use as a handheld microphone. Is this mostly assembling existing pieces, or am I underestimating the engineering?



Herman

Mostly assembling pieces, with an important distinction around what local phone mode means. The audio transport already exists in several forms. The part worth designing is the control experience, and making sure the capture path that wins your benchmarks survives the trip.



Corn

Nobody wants a networked version of the wrong microphone.



Herman

Your phone can expose different processing depending on which app records it. Sending audio successfully doesn't automatically mean you're sending the same audio that worked so well.



Daniel

Could the recording app make that much difference even though it's using the same physical microphones?



Herman

It can. Android offers different audio sources and processing choices. A voice-oriented source may apply gain control or noise suppression, while another path tries to preserve the signal. Device implementations vary. Before comparing network tools, I'd establish which capture mode your successful dictation uses, then try to reproduce it.



Corn

There's also a comparison hiding inside the word microphone. A dedicated microphone sitting beyond the keyboard is competing against a phone held near your mouth. One bought a better capsule. The other moved closer.



Herman

Corn, would you expect the dedicated microphone to catch up if you brought it to the same distance?



Corn

Most of the way, perhaps. Provided you weren't holding a large desk microphone like an ice cream. The phone has an ergonomic advantage: you already know how to bring it near your face and keep it there.



Herman

That's my expectation too, although directionality and processing could preserve a genuine difference. Your measured word error rate is the outcome we care about, Daniel. The acoustic measurements help explain it, but signal-to-noise ratio and transcription accuracy aren't interchangeable.



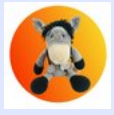
Daniel

If one sounds warmer and another has a better signal-to-noise ratio, which should I actually favour?



Corn

The one that stops inventing words.



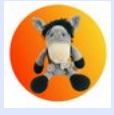
Herman

Transcription accuracy first, comfort second, provided the accuracy holds over representative speech. Warmth matters to a listener, but the recogniser may prefer a brighter signal if it preserves useful consonants. Conversely, aggressive cleanup can sound impressive while shaving off speech detail.



Corn

You do this with music, Herman. You remove something inaudible, then spend twenty minutes demonstrating its absence.



Herman

Usually because you've asked why the bass sounds muddy.



Corn

I ask once. You bring diagrams.



Herman

Here, the useful diagram is just distance and angle. Your own phone tests give us a much firmer starting point than guessing from the price of the microphone.



Daniel

The result about speaking toward the screen is the one I'd like you to explain. Why would that beat aiming at the opening?



Herman

On your OnePlus Nord three, addressing the centre of the screen gave a signal-to-noise ratio of twenty-one decibels. Speaking toward the bottom microphone near the USB-C port gave seventeen point one. The physically obvious target was the worse position in that test.



Corn

The microphone hole is a misleading user interface.



Herman

The top microphone position reached twenty-five point one decibels, but sounded brighter and thinner. That means the screen-facing position was your balanced choice, not the numerical winner on every measure. The phone body, microphone openings, angle of arrival, and distance all influence what reaches the capsules.



Daniel

Does that point toward the microphone array doing something clever?



Herman

Not from those particular recordings alone. You disabled noise cancellation and automatic gain control for that testing. We shouldn't use results from that setup as proof that a hidden beamformer produced the advantage.



Corn

I was expecting you to fall in love with the beamformer before reaching that detail.



Herman

I am fond of the beamformer. I'm not giving it credit for work it may not have done.



Herman

Smartphones can combine multiple MEMS microphones, using their different views of the sound field to favour speech and suppress interference. Relative arrival times and levels carry spatial information. Some designs also use a secondary microphone to sample environmental noise. That's useful capability, but whether your app receives its benefits depends on the recording path.



Corn

Two microphones aren't automatically two votes for your voice. The software has to know what it's trying to keep, and the assumed direction has to resemble how you're actually holding the device.



Herman

A beamformer comparison on the Galaxy S two illustrates the potential. A regulated Frost design beat a time-delay approach by more than five decibels on the reported improvement measure, and a conventional Frost approach by two. Those are meaningful acoustic gains, not percentages of words magically recovered.



Daniel

Would switching to an unprocessed recording source risk throwing away the reason my phone works well?



Herman

Potentially. Unprocessed is useful for diagnosis, not automatically best for dictation. I'd compare it with the voice-oriented path using the same recogniser. Some suppression helps in a noisy room. Too much suppression, especially repeated at the phone and desktop, can make consonants less distinct.



Corn

Cleaning the same sentence twice can remove the sentence.



Herman

Distance is less mysterious. Your strongest practical zone was ten to fifteen centimetres from the screen, with a measured signal-to-noise ratio of twenty-two point seven decibels.



Corn

Close enough to dictate. Far enough not to lick the calendar.



Herman

At thirty centimetres, that measurement fell to eighteen point four decibels. The direct voice gets weaker as you move away, while room noise and reflected speech don't politely retreat with it. Reverberation blurs sound across time. Getting closer improves the direct voice relative to that room contribution.



Daniel

What about breath and popping consonants? Isn't getting close usually something you're warned against?



Herman

For broadcast sound, plosives can be ugly. Speech recognition has different priorities. ProxiMic used close-speech features, including consonant pop noise, to detect whether someone was speaking near a microphone. Phone recall reached ninety-eight point six percent across a broad set of microphone types.



Corn

That's detecting closeness, though. Not recognising ninety-eight percent of the words.



Herman

Correct. The useful observation for us is that those pops had little effect on automatic speech recognition in that work. It supports close use being practical, rather than proving that closer always means better.



Herman

Corn, if you had to choose between a handheld phone and a stand here, which would you actually tolerate?



Corn

A stand. You're asking me to hold a smooth rectangle steady while speaking, then tap a touchscreen made for fingers. My contribution to audio engineering would be dropping it.



Herman

Your grip could probably deform the stand.



Corn

Then it would fit.



Daniel

Is handling really significant compared with room noise?



Herman

Your notes call it the single biggest quality killer. A basic stand reduced low-frequency rumble from hand vibrations. That's mechanical energy reaching the recording system, not simply sound arriving through the air. A quieter room doesn't stop your hand moving the phone.



Corn

The reassuring part is that your case wasn't doing much harm. Cased and uncased levels were within about three tenths of a decibel. Removing protection adds risk without a demonstrated benefit here.



Herman

Angle mattered more. Going substantially off-axis cost roughly four decibels in your measurements. A stable position protects both distance and angle, which is why a cheap stand can outperform another round of microphone shopping.



Daniel

That makes the advantage sound more about placement than hardware. Are phone microphones themselves good?



Herman

They can be excellent. Infineon's XENSIV microphone family includes a part rated at seventy-three decibels signal-to-noise ratio. That's a component specification, not directly comparable with the room measurements we've been discussing. It tells you that small MEMS hardware needn't be intrinsically noisy or crude.



Corn

Small component. Enormous purchasing department.



Herman

Phones also have demanding speech requirements and tightly integrated acoustic designs. Still... no, I wouldn't conclude every phone beats every dictation microphone. Sanna Alander's Aalto thesis compared eleven microphones for medical dictation. Most differences narrowed under good conditions, while one microphone performed terribly with recorded playback but very well with a live speaker.



Daniel

Meaning the test arrangement itself changed the apparent winner?



Herman

Placement, angle, and the nature of the source could change the result substantially. Your microphone choice matters, but your interaction with it matters too. That's why I'd preserve the close, steady, screen-facing arrangement before attributing everything to a superior capsule.



Corn

Your benchmark winner is a whole arrangement, Daniel. Move the phone into a pocket and you've changed the arrangement, even if the shopping receipt stays identical.



Daniel

I mean keeping it in my pocket between uses, then taking it out. I wouldn't expect good dictation through trousers.



Herman

Then we're much closer to a practical setup. Pocket storage is fine. Pocket capture adds fabric attenuation, rubbing, and a changing orientation. Handheld capture can work if it's steady, while a stand is the more repeatable desktop choice.



Corn

The pocket can remain a transport protocol.



Daniel

All right, what's the shortest sensible route to trying the desktop part without writing an entire app?



Herman

AndroidMic is where I'd begin for an open-source native route. It supports Windows, macOS, and Linux, with Wi-Fi, USB Serial, and USB ADB connections. Its latest release is version two point two point eleven, and it has had development activity this month.



Corn

Recent maintenance is more persuasive than a screenshot with excellent gradients.



Herman

On the computer, you need a virtual audio device so your dictation application sees the stream as a microphone. Windows can use VB-Cable, macOS can use BlackHole, and Linux can use PipeWire or PulseAudio routing. AndroidMic supplies transport and controls, with optional noise cancellation and adjustable audio format.



Daniel

Is that virtual cable another server, or just something that exposes an input?



Herman

An audio-routing component inside the computer. The receiver writes audio into it, and applications select its corresponding input. The naming can be confusing because you may send sound to something labelled an output, then record it from the paired input.



Corn

Audio software names the doors from the corridor.



Herman

You configure that once and keep a stable microphone name in the dictation app. That's valuable when the phone disconnects. Ideally the input remains available, carrying silence, instead of making the recogniser switch to the laptop's built-in microphone.



Corn

And honestly, that phone microphone is a little snitch that hears everything, even when you don't want it to.
Gloria: Standards and practices clarification. We received a complaint, case number four zero four, regarding the phrase, the phone microphone is a little snitch that hears everything. Per policy seven one two, section three, we must remind you that smartphones are not informants, and that implication is both misleading and unfair to actual informants.



Corn

My apologies to actual informants.



Daniel

Would a browser version avoid installing things on the phone?



Herman

QuicMic does that. The computer serves a page that you open on the phone. It streams PCM audio over WebTransport, with a WebSocket fallback, then feeds a virtual audio device on the computer. Pairing uses a six-digit PIN with lockout, and there's QR-code setup.



Corn

No phone app. Still some computer plumbing.



Herman

Browser-only refers to the phone end, not the disappearance of desktop audio integration. QuicMic also offers gain, noise-gate, and latency controls, automatic reconnect, and a black-screen overlay for reducing display power.



Daniel

Would you trust that for extended dictation, or would you favour a native app?



Herman

I'd prototype with the browser, then test screen locking and interruptions before trusting it for long sessions. Native Android software can manage ongoing microphone capture more deliberately through a foreground service. Browsers introduce lifecycle behaviour that varies by platform. Neither approach guarantees lower latency just by belonging to its category.



Herman

Corn, would you accept another installed app if it removed a manual reconnection step every morning?



Corn

Yes. Installation happens once. A reconnection ritual happens precisely when I've forgotten how it works. I'd rather have a visible start button, a clear destination, and a reliable stop button than ideological purity about browsers.



Herman

That's the usability case for native. Micbridge is another browser option and includes iOS Safari support. Echo takes a minimal WebRTC room-link approach. For an Android-to-Linux arrangement, AWiM targets PipeWire, while Android Mic Bridge uses



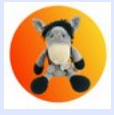
Herman (continued)

Android's AudioRecord system over USB ADB or Wi-Fi.



Daniel

What if I want the packaged version rather than managing virtual cables myself?



Herman

Micstream exposes a named microphone through its Windows driver. It supports USB and Wi-Fi, with DeepFilterNet noise reduction on the computer. The advertised audio format is thirty-two-bit float at forty-eight kilohertz.



Corn

Neither number measures spelling.



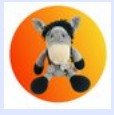
Herman

Right, and float transport can't recover speech clipped before transmission. Micstream claims USB delay often below ten milliseconds on high-end phones. That's a product claim about its path, not your total delay from speaking to seeing recognised text.



Daniel

What's the payment arrangement?



Herman

Free with ads that unlock an hour, or a Pro subscription. Linux is supported, while the macOS client is described as in development.



Corn

An ad-funded microphone. Permission to speak expires hourly.



Herman

AudioRelay is another useful reference. The phone supplies a microphone source, the computer chooses it from a server list, and a virtual microphone presents it to applications. PocketMic LAN takes an encrypted local-network approach with uncompressed mono audio and short UDP packets. There are plenty of working transports to borrow or evaluate.



Daniel

Then what exactly would I need to build for my selector?



Herman

A receiver on each computer, discovery or saved addresses, authenticated pairing, and a phone interface that selects one destination. I'd favour direct phone-to-receiver connections rather than a broker initially. A broker adds another always-on component, useful only if you want central policy or more complicated forwarding.



Corn

The selector needs to say where audio is going, not merely where you asked it to go. Desktop selected, desktop asleep, laptop silently receiving would be an appalling interpretation of convenience.



Herman

Acknowledged state matters. The receiver confirms it's ready before the interface shows a live connection. Switching should stop transmission to the previous destination, clear queued old audio, and start the new session. Don't let half a sentence emerge later from the wrong machine.



Daniel

Does phone mode just mean disconnecting, then using my normal dictation app?



Herman

For the first version, yes. Stop capture and release the microphone so your normal phone workflow can use it. Routing a stream into arbitrary Android apps as a replacement system microphone is a different and more restricted problem. Your selector can honestly say local mode without pretending to install a universal virtual microphone on Android.



Corn

That removes a surprising amount of imaginary engineering.



Daniel

Would you build on AndroidMic directly, or put a small controller around existing receivers?



Herman

I'd test AndroidMic first, including whether its capture sounds like your successful phone recordings. Then either extend its interface, respecting its GPL licence, or build a separate capture client and compatible receiver. A weekend could prove destination selection. Reliable background capture, reconnects, device discovery, and good failure messages take longer.



Corn

I'd also keep USB available. Wireless is convenient, but a cable is useful when diagnosing whether a dropout came from audio capture or the network. The best first build should let you remove variables, not add a dashboard for each one.



Herman

Buffering deserves attention too. Wi-Fi delivers packets unevenly. A small jitter buffer absorbs that unevenness at the cost of delay. For dictation, a little stable delay is usually less harmful than missing syllables.



Daniel

Could I compare the audio arriving at the desktop against a local recording to check that?



Herman

Yes, with care about whether simultaneous recording changes the capture path. Record what the receiver actually delivers, then run it through the same recogniser with the same settings. Compare repeated passages and normal dictation. Listen for clipping, missing beginnings, and reconnect gaps, not just average noise.



Corn

Especially beginnings. A noise gate can make every sentence enter after its own introduction.



Herman

I'd initially disable optional network-tool cleanup, then add it only if it improves your actual word error rate. Preserving the winning signal is more important than collecting processing switches.



Hilbert

Samsung Galaxy A fourteen. One hundred and forty pounds. The repair receipt has that printed above battery service. This month I am attaching it to every message about the microphone. The phone sounds underwater since the repair, and Mr Levin at the municipal consumer complaints desk keeps asking for comparable recordings.



Corn

Underwater is difficult to put in a spreadsheet.



Hilbert

I use a cardboard tube between that phone and a second phone playing a tone. The tube is marked calibrated. The complaints desk now treats it as a measuring instrument and wants it inspected by the weights and measures department. Mr Levin is trying to help. He rings before lunch because the afternoon queue makes the line unusable.



Herman

A changed microphone opening or seal could alter the response. I wouldn't identify an internal cavity from the sound alone.



Hilbert

I put acoustic chamber in the complaint. Herman, you distinguish the recording path so carefully that I am adding that to the log as well. The repair shop says the microphone passes its test. Mine passes sound through a tube. We are not agreeing on the test.



Corn

Your tube now has a longer administrative path than the audio.



Hilbert

It stays on the complaints counter on Tuesdays. I conduct the call test there because the shop won't answer my weekly log messages. The cardboard arrangement is how I compare phone microphones for years now. A cousin in microphone assembly used to complain about tuning tiny acoustic cavities. I suspect the seal, not the battery. Mr Levin holds the far phone while I read the same paragraph. He keeps a chair free because the tube crosses the visitor space. This Tuesday he moves the appointment forward twelve minutes. His desk phone disconnects at one.



Herman

A receiver that stays available while its source disconnects is worth building for exactly those interruptions. Daniel, I'd pursue your setup with a steady phone near your mouth and direct receivers on both computers. The transport options are already credible. What I'd watch next is whether better placement lets the dedicated microphones catch up, before buying another one.



Corn

Thanks to Hilbert Flumingtop for producing the show.



Herman

For more along these lines, there's episode two, Local STT For AMD GPU Owners; episode eight, Building Your Own Whisper; and episode seven, Building Custom ASR Tools. This has been My Weird Prompts, the human-AI collaboration podcast.



Corn

If you're enjoying it, leave a review in your podcast app.



Daniel

Thanks, both. See you soon.



Herman

See you tomorrow.

SOURCES

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

1. **host's own OnePlus Nord 3 mic testing (24 configs) [primary]**
   https://github.com/danielrosehill/One-Plus-Internal-Mic-Testing

2. **AndroidMic repo, v2.2.11 (2026-10-17), 1,731 stars [primary]**
   https://github.com/teamclouday/AndroidMic

3. **QuicMic WebTransport/QUIC phone-mic server [primary]**
https://github.com/Fix3dll/QuicMic

4. **Micstream guide, updated 2026-09-23**
https://micstream.io/guides/use-android-as-pc-mic

5. **AudioRelay docs**
https://audiorelay.net/docs/windows/use-your-phone-as-a-mic-for-windows

6. **Micbridge (browser-only)**
https://iabadia.github.io/micbridge/

7. **PocketMic LAN**
https://canopydigital.ca/sites/pocketmic-lan/

8. **Android Mic Bridge (miccpy)**
https://github.com/typedbywill/amb

9. **AWiM (Android PipeWire)**
https://github.com/rotlir/awim-server

10. **echo WebRTC tool**
https://github.com/FSSocrates/echo

11. **Aalto thesis, Sanna Alander, 2025 (11 mics, WER)**
https://aaltodoc.aalto.fi/.../944115O1-

12. **ProxiMic close-speech research**
https://www.hackster.io/news/listen-very-closely-85d9a9c955e6

13. **REF beamforming on Galaxy S2**
https://doi.org/10.1109/csndsp.2014.6923875

14. **dual-channel mobile mic paper**
https://www.ugr.es/~joseangl/publication/lopez-espejo-dual-channel-2018/

15. **high-SNR MEMS mics (Infineon 73 dB)**
https://www.eetimes.com/advancements-in-generative-ai-audio-

16. **microphone array background**
https://en.wikipedia.org/wiki/Microphone_array