Episode Book

EPISODE #5871

Why Your USB-C Charger Won't Talk to Cheap Devices

PUBLISHED	11 October 2026 at 20:21 UTC
DURATION	19:35
LISTEN	myweirdprompts.com/episode/usb-c-charging-handshake-explained-89404/
HOSTS	Corn, Herman
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.3
SOURCES	10 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:40 UTC

SYNOPSIS

Daniel has a 300W GaN charging brick with ten ports, and one of them refuses to charge his pocket speaker. USB-A works fine. USB-C to USB-C just makes the light flash red. Is the speaker damaged? Did the charger fry it? Not even close. This episode unpacks the two completely different philosophies behind USB-A and USB-C power delivery: one is a dumb pipe that hands out five volts unconditionally, the other waits for a proper introduction over the Configuration Channel before it energizes anything. We walk through the resistor-based current tiers, the Power Delivery menu of voltage and current pairings, what PPS actually means, and why a failed handshake produces darkness rather than damage. Along the way we explain why the "better" cable is the one that exposes the problem, why a rapid red flash is a fault code and not a death sentence, and why a fraction of a cent in missing components is the real culprit.

TRANSCRIPT



Corn

Okay, before we get into this, I want everyone to picture the object. It's a brick. It has ten ports on it. Ten. Four USB-A, six USB-C, three hundred watts total, and it lives on Daniel's desk doing the work of what used to be four separate chargers.



Herman

And one of those ports has met its match. A pocket speaker. Clips on your shirt. Charges fine from the old port, flashes a red light at him from the new one.



Corn

So Daniel sent us the puzzle. He's got this GaN block, and he did the thing most of us never do, which is actually read the printing on the underside of it. Four A ports, sixty watts each. Two C ports at a hundred watts, four C ports at thirty, and the whole C side rated from three point three volts up to twenty.



Herman

Which tells you something already.



Corn

It tells Daniel something, which is that every single port on this thing negotiates. Nothing here hands out a fixed voltage. And that's what's making him nervous. He says the fast charging standards make his head spin a bit, he's always a little uneasy plugging an older device into this thing, and he's heard overvoltage fries electronics.



Herman

Reasonable fear.



Corn

So he plugs the speaker into one of the C ports, C to C cable, nothing happens, except the charge light is doing a rapid red flash. He yanks the cable. Then he digs out an A to C cable, plugs into one of the A ports, and it just charges. No drama. And his read on it is that the A port saw no fast charging support on the other end, dropped to five volts, the speaker drew what it needed, everyone went home happy.



Herman

Which is half right.



Corn

And his actual question is the other half. Why didn't C to C do the same thing? What's different about how those two ports decide what to send, and did he hurt the speaker?



Herman

Two handshakes, Daniel. Same connector on one end, completely different etiquette on the other.



Herman

The thing to hold onto is that USB-A and USB-C aren't two versions of the same port. They're two different philosophies of who's in charge. A USB-A port is a dumb pipe. It is always the provider, it always starts at five volts, and it doesn't ask the device anything before it energizes the rail. It just hands out five volts, and if the device wants to negotiate up to nine or twelve, that's a proprietary trick on top of a connection that already works.



Corn

So the A port never refuses.



Herman

The A port essentially cannot refuse. There's no state in which it sits there holding power back. Power is on, that's the default, and the device takes what it takes. Which is why a cable that's A on one end and C on the other works with almost anything, because the A side has already made the decision, and the cable just presents a sensible face to the C device on the other end.



Corn

And the C port.



Herman

A C port with Power Delivery on it is the opposite. It starts from nothing. It will not put voltage on the rail until it has identified something on the other end as a legitimate sink. That's the whole design. Both sides have to introduce themselves before any current flows.



Corn

Introductions through what?



Herman

Through the Configuration Channel. When you look at a USB-C plug, there are two little pins in there that have nothing to do with power or data. They're called CC1 and CC2, and they carry the entire relationship between the two devices. That's the handshake that Daniel's C to C cable was trying to have and that his A to C cable skipped entirely.



Corn

So on the A side there's no handshake at all.



Herman

On the A side there is no handshake because there's no one to have it with. The A port has no CC pins. It can't ask. So the cable translates. An A to C cable presents the right resistor to the C device to satisfy it, and presents plain five volts to the A port, and the two



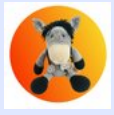
Herman (continued)

ends never speak.



Corn

That's a strange sentence. The cable does the lying.



Herman

The cable does the translating. It's a diplomat. It tells the A port nothing and tells the C device everything it wants to hear.



Corn

So back to the C to C connection. Daniel plugs in the speaker, and the charger goes through this introduction.



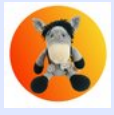
Herman

And step one is physical, before a single digital message. The source, the charger, puts a pull-up resistor on those CC pins. The sink, the device, is expected to put a pull-down resistor to ground. When the charger sees that pull-down, it knows something is plugged in and it knows roughly what size that something is, because the resistance value tells it.



Corn

Tells it what?



Herman

The current tier. There are basically three default levels a device can advertise just with the resistor. Nine hundred milliamps, one and a half amps, or three amps. That's before any digital conversation at all. Just a resistor value saying, I'm a small device, I'm a medium one, I'm a big one.



Corn

That's the entire hardware cost of being a good USB-C citizen.



Herman

It is tiny. For a device that just wants to pull five volts at up to three amps and never wants anything fancier, the whole requirement is two resistors. Five point one kilo-ohms, one on each CC pin, to ground. That's it. No chip, no firmware, no power delivery controller. Two resistors that cost a fraction of a cent.



Corn

And that gets you fifteen watts.



Herman

That gets you fifteen watts, which is more than enough to charge a pocket speaker at a reasonable rate. Which is the part that should sting. The device that's failing here isn't failing because it's too old or too simple. It's failing because of two parts that cost less than the plastic in the connector housing.



Corn

So what happens when the charger looks at the CC pin and sees nothing.



Herman

Then as far as the charger is concerned, there is no device. The pin is sitting there with no pull-down on it. There's no advertisement of a current tier, there's no valid sink state, and the source does exactly what it's designed to do when it can't identify anything. It keeps the rail off.



Corn

No power at all.



Herman

That's the fail-safe, and it's deliberate. The whole system is built so that the failure mode of a failed handshake is darkness, not voltage.



Corn

So let's say the speaker is compliant, and it does present its resistors, and the handshake proceeds. What does the charger do then?



Herman

Then it's a conversation, in a specific order. The charger sends what's called Source Capabilities, which is a list of Power Data Objects. Think of it as a menu. My charger's menu, in Daniel's case, would list five volts at three amps, nine volts, twelve volts, fifteen, twenty volts, plus a couple of programmable entries. Each item on that menu is a pairing of a voltage and a maximum current.



Corn

And the device picks from the menu.



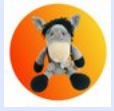
Herman

The device picks, it sends back a Request, the charger sends Accept, then a Ready message, and only after that sequence does the voltage on the rail actually move from five volts to whatever was agreed. The order matters enormously, because the rail is still at the safe default until the very last message. Nothing jumps.



Corn

How fast is this conversation?



Herman

Milliseconds. The sink has about thirty milliseconds to respond to a message before the source starts worrying. The whole negotiation, when it goes well, is effectively instant from a human point of view. You plug in, the light comes on, you never knew any of it



Herman (continued)

happened.



Corn

And the voltages in between. Daniel's charger says three point three to twenty volts, not a list of fixed steps.



Herman

That's PPS, Programmable Power Supply. Instead of the device asking for nine volts, it can ask for nine point one two, or eight point nine four, and the charger will deliver it in twenty millivolt steps. That's what makes some of the fast charging schemes work, because they can track what the battery actually wants in real time and adjust. That's what that printed range means. It's not a random voltage generator. It's a menu with very fine print.



Corn

Okay, so I'm looking at Daniel's speaker, and I want to be fair to it. Is it possible the thing is simply lazy rather than broken?



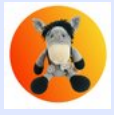
Herman

I think lazy is the right word, and I'd go further. The most likely explanation isn't that the speaker is broken at all. It's that the speaker's USB-C input was never built to be a proper USB-C sink. It's a charging jack that happens to be the right shape.



Corn

Meaning the two resistors aren't there.



Herman

Meaning the two resistors very likely aren't there, or aren't there in the right place, or the port is wired to a simple charging circuit that predates the standard. And this is not rare. A manufacturer building a cheap speaker has a choice: add two resistors and make the



Herman (continued)

port universally compatible, or save the fraction of a cent and assume the customer will use whatever cable is in the box.



Corn

Which manufactures the exact experience Daniel had.



Herman

Except Daniel didn't use the cable in the box. He used a proper C to C cable, which is the honest thing to do, and the honest cable exposed the problem. That's the irony. The better cable made the device quit.



Corn

So to answer his question directly. Why did A work and C to C not?



Herman

The A path never asked the speaker to identify itself. Five volts were waiting before the speaker even arrived on the rail, and the speaker just drew current the way it was built to. Nothing was negotiated because nothing needed to be. Plug in, charge.



Corn

And the C path asked, and got silence, and did the only safe thing it could do.



Herman

Sat there with the rail off. And here's what I'd say to the worry underneath Daniel's question. He thinks something went wrong. Something didn't go wrong. The charger declined to start. That's a completely different event.



Corn

Which is a relief, and also slightly insulting. His charger looked at his speaker and didn't recognize it.



Herman

His charger looked at his speaker and saw an open circuit on the identifying pins. As far as it knows, there's nothing there. It's not judging the speaker. It can't even see it.



Corn

Fine. So now the light. Because the light isn't nothing. If the port were simply dead, the speaker would sit there dark and Daniel would have shrugged and moved on. Instead it's flashing at him.



Herman

And that flash is the speaker's side of the story, which is why it's interesting. A charger that refuses to talk gives you silence. A device that detects a fault gives you a signal. The speaker knew something was wrong.



Corn

So decode it.



Herman

Rapid flashing on a charging indicator is almost universally an error code. Not a status, an error. On just about any device you own, a slow pulsing light means normal work in progress, usually charging a depleted battery. A fast, frantic, irregular flash means the charging logic tried to start the cycle, found something it didn't like, and shut the port down.



Corn

Slow means filling. Fast means fault.



Herman

And the fault we're most likely looking at is the handshake failure itself. The speaker's charging controller goes through its own startup routine, waits for the input to stabilize at a voltage it recognizes, and it doesn't arrive. Depending on how the circuit is built, it may see a rail that never comes up, or it may see power flicker on briefly before the charger cuts it. Either way the controller concludes the cycle failed to initiate, and it signals that the only way it can. Red, fast.



Corn

So the flashing isn't damage.



Herman

The flashing is the opposite of damage. The flashing is a device announcing that it refused to proceed. It's a fault indication, and specifically a fault indication tied to the start of charging, not a health report about anything internal. The speaker was telling Daniel, I couldn't begin.



Corn

He yanked the cable.



Herman

He yanked the cable, which did nothing good and nothing bad. There was no dangerous condition to interrupt. The charger hadn't energized the rail, the speaker had already refused, and the two of them were sitting in a standoff. Pulling the cable ended a standoff.



Corn

I want to be careful here, because I've done exactly what Daniel did. Something flashes at me, I unplug it, and I feel like I've prevented a fire.



Herman

The instinct is right even when the reasoning is wrong. Yanking a cable when something looks wrong is basically always correct behavior. In this case the thing that looked wrong was a device that was already protecting itself.



Corn

Let's take the safety question head-on, because it's the one he actually asked. Is a three hundred watt charger dangerous for an old speaker.



Herman

Here's the part I find elegant. A compliant Power Delivery source is safer for that speaker than a dumb five volt brick. Not equally safe. Safer.



Corn

Explain that, because it sounds backwards.



Herman

The thing the whole protocol is built to prevent is a source forcing high voltage onto an unprepared device. So the source cannot simply put twenty volts on the rail. It can't put nine volts on the rail. It cannot move off the default until a valid sink has requested a specific higher voltage and the request has been accepted in sequence. The negotiation isn't a courtesy layered on top of the power. The negotiation gates the power.



Corn

And the brick.



Herman

The brick has no gate. A cheap fixed supply that puts out five volts and a decent amount of current will hand that straight to whatever is plugged into it, and if the device's input circuitry is bad, the brick will find out the hard way and so will the device. The brick is only safe because five volts happens to be the right answer. It's not safe by design. It's safe by coincidence.



Corn

Whereas the smart charger, which is the one Daniel's nervous about, will verify before it energizes.



Herman

Which is why he should relax about the three hundred watt number. That number describes a ceiling, not a behavior. The charger doesn't have three hundred watts in its hand waiting to dump it into the first thing that shows up. It has a menu and it waits. Two devices that can't talk to each other produce a dead rail. That's the whole outcome.



Corn

It's the least satisfying possible failure. You spent money on a charger that does everything and the result is that nothing happens.



Herman

Nothing happening is the system working.



Corn

Alright. But I want the exception, because there always is one, and I'd rather hear it from you now than find it in a comment.



Herman

The exception is that the negotiation is trust-based. When the charger sends its menu, the device is expected to be honest about what it can accept, and there's no verification in the base protocol that the menu itself is legitimate. Version two and version three of Power Delivery have no mandatory cryptographic check on what the source advertises. If the charger's firmware is compromised or badly written, it can advertise a voltage the device was never built to take and then deliver it.



Corn

BadPower.



Herman

That's the documented class of attack. It requires either a malicious charger or a buggy one. Version three point one added optional authentication, so a device can in principle verify the source before trusting it, but the adoption of that in consumer chips is limited. Most of what's on your desk today doesn't do it.



Corn

The theoretical fear is real but it requires a compromised source.



Herman

It requires a source that's lying. Which takes us back to the same distinction. The danger is never the number printed on the charger. The danger is a source that says one thing and then does another. A compliant charger doing three hundred watts is not the risk. A faulty charger doing sixty watts and lying about it is.



Corn

Which is a useful way to think about it. Judge the charger by whether it keeps its word, not by the size of the number.



Herman

There's a ripple effect here too, which is that the problem Daniel hit is going to get more common, not less. The A port is being retired. Once a charger is USB-C only, there's no bypass cable left. If your device can't say hello, there's nowhere to hide it.



Corn

The A-to-C cable is a dying escape hatch.



Herman

It's a compatibility crutch that only exists while A ports exist. When everything on your desk is C, a non-compliant device just stops charging, and the user has no idea why, because the cable fits and the charger works and nothing is broken. That's the trap. It looks exactly like a dead device.



Corn

What do you do, practically. Because I know the instinct is going to be, test the device with every cable in the drawer.



Herman

Test with A-to-C first if the device is old or cheap, because that tells you whether the device charges at all. If it charges from A and not from C, you've learned something real about the device, and it's the missing resistors. And then decide whether you care. If the speaker lives on your shirt and charges fine from the A port overnight, it doesn't matter.



Corn

If everything you own is C.



Herman

Then a short A-to-C adapter cable in the bag is the least elegant solution that works, and I'd rather have one than argue with a plastic port.



Corn

What about the actual question of damage. Can I say confidently that his speaker is fine.



Herman

You can say the mechanism suggests it, and that's not the same as a documented finding for this exact speaker. The charger withheld power, the speaker refused to start, so there's no obvious path by which energy reached the wrong place. But I can't inspect his circuit board. I'd say this: if the speaker charges normally from the A port now, at its normal rate, and the battery still holds what it used to, there's no story. If it had taken damage, he'd know by now.



Corn

Mark it as inference.



Herman

Mark it as inference, and a fairly confident one, because the design of the failure is friendly. Nothing in the sequence involves a high voltage being applied to a low voltage device. The whole fight was about whether to turn on at all.



Corn

Which is the opposite of what he was afraid of.



Herman

He was afraid the charger would overreach. What actually happened is the charger wouldn't reach at all.



Corn

One more thing before Hilbert weighs in. Daniel says the A port on this charger is also rated from five to twenty volts, sixty watts max. So the A port is doing some negotiation of its own.



Herman

It is, but only in the ways A ports with proprietary fast charging do. That's not Power Delivery. That's a vendor protocol layered on top, and it only engages if the device on the other end speaks it. If the device doesn't, the port sits at five volts and behaves like the dumb pipe we started with. That's why it worked for the speaker. The speaker doesn't speak it, so the port gave up and just handed over five volts.



Corn

The port tried to be clever and then gave up and was kind.



Herman

The A protocol fails open. The C protocol fails closed.

SOURCES

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

1. **USB-IF, USB Power Delivery . [primary]**
   https://www.usb.org/sites/default/files/D2T2-1%20-%20USB%20Power%20Delivery.pdf

2. **rbaron.net, USB Power Delivery for Makers, 2024-06-02 (detailed CC/Rp/Rd, PDO/RDO, PPS, EPR mechanics).**
   https://rbaron.net/blog/2024/06/02/usb-power-delivery-for-makers

3. **kbytechnologies.com, Debugging USB-PD Negotiation Failures (state machine, failure modes, BadPower, timers).**
   https://www.kbytechnologies.com/tech-fundamentals/debugging-usb-pd-negotiation-failures

4. **Cable Chick, Why Won't My USB-C Device Charge?, 2025-02-10 (Type-A vs Type-C, missing pull-down resistors, fail-safe).**
   https://www.cablechick.com.au/blog/whywontmyusbcdevicecharge/

5. **simplysolved.blog, Does Red Blinking Light Mean Charging?, 2026-05-15 (rapid flash = error, handshake failure).**
   https://simplysolved.blog/does-red-blinking-light-mean-charging

6. **Texas Instruments, USB PD Power Negotiations [primary technical].**
   https://www.ti.com/lit/an/slva842/slva842.pdf

7. **Microchip AN1974, Introduction to USB Power Delivery Over the USB Type-C Cable [primary technical].**
   https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ApplicationNotes/ApplicationNotes/00001974A.pdf

8. **itechguides.com, USB-C Power Delivery Explained (default-power state, no forced voltage).**
https://www.itechguides.com/all-about-usb-c-power-delivery-how-usb-pd-works-wattage-chargers-and-cables/

9. **EEVblog forum thread, 2022-08-03 (dead-battery mode, 5.1 kΩ Rd, 56 kΩ Rp).**
https://www.eevblog.com/forum/projects/abusing-usb-c-cc-pin-resistors-to-default-to-power-sink/

10. **Renesas, USB Type-C and Role Swap (CC1/CC2 source/sink detection).**
https://www.renesas.com/en/support/engineer-school/usb-power-delivery-03