



1. Process Chain Localization: Where Are We?

We are on the **Tx side**, right after IQ signal generation (typically float32, complex baseband). Here's the precise localized stage:

[Analog or Digital Source] → [Baseband DSP] → [I(t), Q(t) as float32] →
⚠ [NEED TO SCALE] → [Convert to int8] → [Save to .iq File or Send to DAC]

So your design decision is in this red ⚠ block:

Do you apply scaling? If so, how much? Where is the logic embedded — in code, in metadata, in spec?



2. What's the Real Problem?

Float32 IQ data (e.g., 1.2, -1.1, 0.85, ...) **does not fit** directly into int8 (-128 to +127).

If not scaled:

- High-amplitude samples may **clip or wrap**
- Low-amplitude samples may **get truncated to 0**
- File becomes **non-standard and unplayable**

Yet:

- Over-scaling reduces SNR (loss of detail)
- Scaling applied in different tools (e.g., GNU Radio vs custom script) can cause **inconsistent playback**



What is *not* decided yet?

You have **not yet decided**:

Question	Design Decision You Must Make
? Should all IQ data be scaled before writing?	Yes / No / Optional per config
? Should the scaling factor be dynamic (per signal) or fixed?	Fixed at 127? Per signal peak?
? Should metadata store the scaling factor?	Include "scaling_factor" in JSON or header?
? If scaling is not applied, should the format explicitly say so?	"normalized": false or skip the field?
? Should your standard enforce clipping protection?	e.g., no value > 127 allowed at all?

3. What You're Designing

This is **not just about code** — it's about defining **behavior and traceability** in your format spec.

Your **standard must answer**:

"When someone writes or reads an IQ file using this format, how is scaling handled, enforced, or declared?"

Updated Problem Statement (Final)

Given float32 I/Q samples at the Tx side, determine if and how scaling must be applied before converting to int8 for file storage, in a way that prevents distortion, enables file consistency, and supports standard-conformant reading.

You Must Decide (today's design decision):

Pick one of the following:

Option	Design Intent	Metadata?	Code Simplicity	Signal Integrity
A Always apply dynamic peak-based scaling	Normalize every signal	✔ Yes	✔ Easy	✔ Clean
B Use fixed scaling (e.g., $\times 127$), and require user to pre-normalize	More manual, faster	⚠ Optional	✔ Fast	⚠ May clip
C Let writer choose, but enforce "scaling_applied" flag	Flexible	✔ Yes	⚠ Needs handling	✔ Traced

Your task

Reply with:

- The option you prefer (A , B , or C)
- Any constraints you'd like to enforce (e.g., “no clipping ever,” or “must have metadata always”)

Then I'll:

- Finalize the **spec rule** for this scaling behavior
- Give you the code that follows that design
- Close `issue_tkt_1` with both how-to and working module

You're the standard designer — I'm just the tool.

Ready to decide?