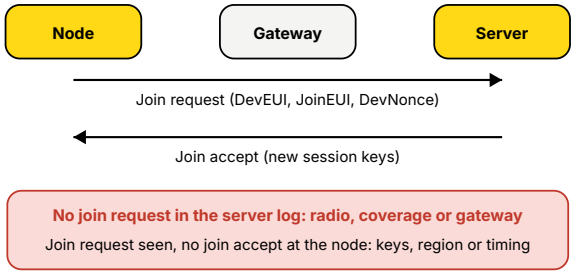
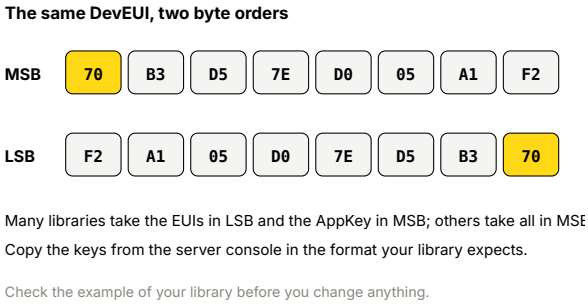
LoRaWAN node won't join? 6 checks, in order

The node sends join requests and never gets a join accept. Before you change the code, check these six points in order. Valid for any LoRaWAN node or PLC and any network server (The Things Stack, ChirpStack or others). If the radio link itself is weak, check range first: see the LoRa lesson on range in the field.

How a join works



OTAA join: the node asks, the server checks the keys and answers with new session keys. Where the chain breaks tells you where to look.



Byte order: a key with the right digits in the wrong order is a wrong key for the server.

Prefer OTAA

With ABP the session keys are fixed in the device and the frame counters must never go back. OTAA renews the keys at every join.

Read the server log first

The events of the gateway and the device in the network server tell you which check to start with.

Six checks, in this order

Check	What to do	Good result
1. Activation	Use OTAA when you can. Register the device in the server with the same activation mode as the node.	Same mode in node and server.
2. Keys	DevEUI, JoinEUI (AppEUI) and AppKey identical in node and server, in the byte order your library expects.	Join request with no MIC error.
3. Region	Same frequency plan in node, gateway and server (for example EU868 or US915). In US915, the same sub-band.	Same plan and sub-band everywhere.
4. Gateway	Gateway connected to the server, with traffic from your node in its events.	Join request visible in the server.
5. Counters	1.0.4 and 1.1: DevNonce keeps increasing, also after a reset (non-volatile memory). ABP: never reset frame counters.	No DevNonce errors, no dropped uplinks.
6. Timing	Random, growing delay between join retries; respect the duty cycle. Far gateway: start with a higher SF. ADR after the join.	Join accepted within the duty cycle.

What the server log tells you

Message or symptom	Possible causes	Next step
No join request in the log	No coverage, gateway offline or not forwarding, wrong frequency plan.	Checks 3 and 4.
MIC mismatch	AppKey different in node and server, or keys in the wrong byte order.	Check 2.
DevNonce has already been used	DevNonce not stored after a reset or reflash.	Check 5.
DevNonce is too small	DevNonce not increasing, counter restarted.	Check 5.
Join accept sent, node keeps joining	Accept arrives late (slow gateway backhaul) or the node does not hear it (coverage).	Checks 4 and 6.
Joins, then uplinks dropped (ABP)	Frame counters reset in the device.	Check 5; prefer OTAA.
Joins sometimes, after many retries	SF too low for the distance, retries too fast for the duty cycle.	Check 6.

From the specification

- LoRaWAN 1.0.4 and 1.1: DevNonce is a counter; if the device can be power-cycled, keep it in non-volatile memory.
- LoRaWAN 1.0.x limits the aggregated transmit time of join retries, on top of local regulations.
- JoinEUI is the newer name of AppEUI.

US915 sub-bands

US915 has 64 uplink channels of 125 kHz in 8 sub-bands of 8. An 8-channel gateway usually listens to one of them.

Configure the node on the same sub-band, or let the server tell it which channels to use after the join, if it supports it.

If it has never joined

- Run the join example of your library without changes, only with your keys.
- Print the keys in the node and compare them byte by byte with the server console.
- Check the gateway events while the node tries to join.

Keep the keys safe

The AppKey gives access to the session keys of the device.

Do not publish it in code repositories, screenshots or support tickets.

References: LoRa Alliance TS001-1.0.4 LoRaWAN L2 1.0.4; LoRa Alliance RP002 Regional Parameters; LoRa Alliance, LoRaWAN 1.0.x Join Synch Issues Remedies v1.0.0; The Things Stack documentation: troubleshooting devices, ABP vs OTAA, best practices. Device manuals and local regulations take precedence over this general guide.