

Meshtastic Power Settings Reference

The Power configuration section in Meshtastic controls sleep modes, charge management, and power-related behavior. These settings are critical for battery and solar-powered deployments.

Accessing Power Settings

In the app: **Radio Config** → **Power**

Via CLI: `meshtastic --get power`

Is Power Saving Enabled

```
meshtastic --set power.is_power_saving true
```

When enabled, the node reduces power consumption by putting the CPU into light sleep between receive windows. In light sleep the CPU is suspended but the LoRa radio stays on, so the node can still hear and respond to traffic. This is appropriate for CLIENT role nodes that send and receive messages but want longer battery life.

For the **ROUTER** role, `power.is_power_saving` is force-enabled automatically (on ESP32) and cannot be turned off — you do not set it manually. Even with power saving on, the LoRa radio remains in standby and wakes on incoming packets, so the node continues to relay; light sleep (radio standby) does not prevent relaying. **REPEATER is deprecated as of firmware 2.7.11**; for infrastructure prefer **ROUTER** (or ROUTER_LATE).

On Battery Discharge Values

These settings affect how the node reports battery state-of-charge from measured voltage. The defaults work for LiPo batteries; the ADC multiplier may need adjustment for LiFePO4 or for boards whose voltage divider reads inaccurately:

Setting	Default	Notes
adc_multiplier_override	0 (auto)	May need tuning per board for accurate voltage reading, including LiFePO4 packs

Meshtastic does **not** provide a generic “shut down after N seconds at a critical-voltage threshold” key. (The firmware option `power.on_battery_shutdown_after_secs` shuts the device down after *external power is lost* for N seconds — it is a power-loss timer, not a low-voltage cutoff — and defaults to `0` = disabled.) For low-battery protection rely on the firmware’s built-in low-battery handling rather than a fabricated voltage-cutoff timer.

Sleep Configuration

For ESP32-based nodes, Meshtastic uses light sleep between activity to reduce power consumption. Note that the older super-deep-sleep keys (`power.sds_secs` and `power.mesh_sds_timeout_secs`) have been removed/deprecated and are no longer part of the current, documented power config — do not rely on them.

```
# Enable power saving (CLIENT nodes; ROUTER force-enables it automatically)
meshtastic --set power.is_power_saving true

# Light sleep interval. A value of 0 means the firmware default (5 minutes).
meshtastic --set power.ls_secs 0

# Minimum wake time - stay awake at least this long after waking
meshtastic --set power.min_wake_secs 10
```

Screen and BT Power

```
# Screen timeout (0 = always off)
meshtastic --set display.screen_on_secs 30

# Bluetooth enable/disable
meshtastic --set bluetooth.enabled true

# BLE pairing mode (RANDOM_PIN, FIXED_PIN, or NO_PIN)
meshtastic --set bluetooth.mode RANDOM_PIN
```

Use `RANDOM_PIN` (the secure default) unless you have a specific reason not to. **Setting** `NO_PIN` **disables BLE pairing security** and lets any nearby device connect without a PIN — only use it on physically secured nodes. (These Bluetooth options live under `bluetooth.*`, not the Power config, but are included here because they affect power draw.)

Power Consumption by Role

Meshtastic does not publish per-board current-draw figures; actual milliamp draw is highly board-, display-, and firmware-dependent. The official device-role table describes power qualitatively (Regular / Low / High):

Role	Power saving	Relative power level
CLIENT	Optional (user-set)	Regular
CLIENT_MUTE	Optional (user-set)	Low
ROUTER	Force-enabled on ESP32 (LoRa radio in standby, wakes on packets)	High
REPEATER (deprecated)	Yes	High

Peripherals add to whatever the role draws: an OLED/TFT display and WiFi both materially increase ESP32 power (WiFi is ESP32-only and is not available on nRF52). For exact numbers, measure your specific board or consult its datasheet rather than relying on generic figures.

Practical Power Optimization Checklist

- Set role correctly: CLIENT for personal nodes (enable power saving for longer battery life); ROUTER for fixed infrastructure (power saving is automatic)
- Set screen timeout to 30 seconds or disable (`screen_on_secs 0`)
- On ESP32, note that enabling WiFi disables Bluetooth (the WiFi setting takes precedence)
- Set position broadcast to 30+ minutes for fixed nodes
- Set telemetry broadcast interval up (the device telemetry default is 1800 s / 30 min)
- Reduce TX power to the minimum needed for coverage. For the legal transmit-power ceiling (US/Canada: 1 W / 30 dBm conducted, with EIRP limits for high-gain antennas), see the LoRa Settings Reference.
- Use nRF52840 hardware over ESP32 for substantially better battery life on the same battery

