Power Configuration Settings Reference

Meshtastic's power management settings control how your node balances battery life against responsiveness. Understanding these settings is essential for field deployments and battery-powered infrastructure.

Key Power Settings

Setting	Default	Description
power.is_power_saving	false	Enable aggressive power saving (see below)
power.adc_multiplier_override	0	Override ADC calibration for voltage reading (float, typically ~2-6; 0 = use firmware default)
power.wait_bluetooth_secs	60	Keep BLE active for N seconds after last connection
power.ls_secs	0	Light sleep interval (ESP32 only). A default of 0 means the firmware uses its built-in 5-minute light-sleep interval; it does NOT mean "no light sleep." No effect on nRF52/RP2040.
power.min_wake_secs	10	Minimum time to stay awake after waking from sleep

Note: the older `power.sds_secs` (super deep sleep) and `power.mesh_sds_timeout_secs` keys, and `power.on_battery_shutdown_after_secs`, are no longer part of the current power config and should not be set. Verify any power key against the current firmware config protobuf before using it.

is_power_saving Mode

When enabled, the node aggressively reduces power consumption:

- Turns off WiFi when not actively connecting. (Caution: on a WiFi/MQTT gateway this can disrupt the connection — leave power saving off for always-on gateways.)

- Turns off BLE after `wait_bluetooth_secs` with no activity
- Reduces CPU clock speed when idle
- Turns off the display backlight more aggressively

Recommended for: Battery-operated client nodes, portable nodes used intermittently

NOT recommended for: Infrastructure repeaters, always-on gateway nodes

Light Sleep vs Deep Sleep

Meshtastic supports power-saving sleep behavior on ESP32 hardware:

- **Light sleep (ls_secs)** - Node sleeps briefly between LoRa receive windows. LoRa radio stays partially active and can wake on a received packet. Moderate power reduction at the cost of slight receive latency. (ESP32 only.)
- **Deep sleep** - On ESP32 the node can drop into a much deeper low-power state during long idle periods, drawing very little power but missing incoming messages until it wakes. This is most useful for sensor-style nodes that transmit periodically and don't need to receive continuously.

Battery Voltage Monitoring

Meshtastic reports battery voltage and calculated charge percentage via the telemetry system. The voltage reading depends on the hardware's ADC (Analog-to-Digital Converter) and a calibration factor:

```
# If battery percentage reads incorrectly, adjust ADC multiplier:
# First, measure actual battery voltage with a multimeter
# Then compare to what meshtastic reports:
meshtastic --info   # reported battery voltage appears under the node's deviceMetrics

# Adjust if needed (value is a float multiplier, typically ~2-6):
meshtastic --set power.adc_multiplier_override 1.05
```

ESP32 vs nRF52 Power Comparison

The figures below are **approximate community measurements** intended for rough comparison only. Actual current draw varies with board, firmware, peripherals (GPS, display), and LoRa preset.

For authoritative numbers consult the relevant chip datasheets (Espressif ESP32/ESP32-S3, Nordic nRF52840) and the Semtech SX126x receive-current spec.

Chip	Idle Current	LoRa RX	LoRa TX @22 dBm	WiFi Active
ESP32 (T-Beam)	~80 mA	~50 mA	~120 mA	+100-200 mA
ESP32-S3 (T-Beam Supreme)	~30 mA	~25 mA	~90 mA	+80-150 mA
nRF52840 (RAK4631)	~0.5 mA	~4 mA	~80 mA	N/A (no WiFi)

Note: 22 dBm here is a typical board hardware output level used for this benchmark, which is distinct from the 30 dBm (1 W) conducted FCC legal maximum for 902-928 MHz.

The nRF52840 platform has dramatically lower idle and receive current, which is why RAK4631-based nodes generally achieve significantly longer battery life (often several times) than ESP32 nodes in typical deployment scenarios.

Recommended Configurations by Use Case

```
# Portable client node (maximize battery life):
meshtastic --set power.is_power_saving true
meshtastic --set power.wait_bluetooth_secs 30
meshtastic --set position.position_broadcast_secs 300 # 5 min GPS broadcast

# Fixed indoor node (always accessible):
meshtastic --set power.is_power_saving false
meshtastic --set power.wait_bluetooth_secs 0 # always keep BLE on

# Solar outdoor node (balance receive and power):
meshtastic --set power.is_power_saving true
meshtastic --set power.ls_secs 30 # light sleep 30s intervals (ESP32)
meshtastic --set power.min_wake_secs 15
```

Revision #3
Created 2026-05-03 06:38:50 UTC by Mesh America Admin
Updated 2026-06-09 12:07:55 UTC by Mesh America Admin