IoT Applications

- Introduction to LoRa Mesh for IoT
- Remote Sensor Deployment Guide

Introduction to LoRa Mesh for IoT

LoRa mesh networks provide a compelling platform for IoT sensor deployments, especially where WiFi doesn't reach, cellular is too expensive, and wired connections are impractical.

When LoRa mesh is the right choice for IoT

Scenario	LoRa mesh advantage
Remote sensors (field, barn, remote cabin)	No cellular or WiFi needed; solar-powered nodes transmit data back to base
Large properties (farms, ranches, campuses)	Single gateway + relay nodes covers miles; WiFi would require many access points
Emergency/event temporary deployment	No infrastructure setup; nodes self-organize; deploy in minutes
Low-bandwidth telemetry (weather, soil, water)	LoRa's low data rate matches sensor data volumes perfectly
Deep sleep battery operation	Sensors sleep between readings; nRF52 nodes draw only microamps in deep sleep, so the radio's duty cycle dominates run time

LoRa mesh vs. standalone LoRaWAN for IoT

LoRaWAN (The Things Network, Helium) requires fixed gateways with internet uplinks. LoRa mesh (MeshCore, Meshtastic) self-organizes and works in areas with no internet or gateway infrastructure. Tradeoffs:

	LoRaWAN	LoRa Mesh (MeshCore)
Infrastructure required	Yes - gateway needed	No - self-organizing
Range extension	Gateway-only (no repeating)	Multi-hop relay through mesh
Data rate	Higher (ADR)	Lower (fixed preset)
Cloud integration	Built-in (TTN, Helium)	Manual (MQTT bridge)
Best for	Fixed sensor fields near gateways	Remote, no-infrastructure, or mobile IoT

Typical IoT payload sizes

LoRa mesh is suitable for low-bandwidth sensor data. Typical packet sizes:

- Temperature + humidity: ~10 - 20 bytes
- GPS position: ~20 - 30 bytes
- Multi-sensor (temp + humidity + pressure + battery): ~40 bytes
- Short text alert: ~50 - 100 bytes

Long Fast is the faster of the common presets (higher data rate, shorter airtime); Medium Slow and Long Slow are progressively slower (lower data rate, longer airtime) in exchange for more range and sensitivity. A 40-byte sensor reading transmits in a fraction of a second on Long Fast and takes longer on the slower, longer-range presets. Either way, IoT use cases are generally not limited by data rate.

Battery life for IoT sensor nodes

With the Heltec T096 (nRF52840, ~13 µA deep sleep on the bare board, around $30 as of 2026-06-08) and a 1000 mAh LiFePO4 cell:

```
Bare-board deep-sleep current: ~13 µA
Wake + measure + transmit: ~25 mA for ~0.5 seconds every 15 minutes

Average current ≈ 13 µA + (25,000 µA × 0.5s / 900s) ≈ 27 µA average

Battery life (radio/MCU only) ≈ 1000 mAh / 0.027 mA ≈ ~37,000 hours ≈ ~4 years (theoretical)
```

This is a best-case theoretical figure for the bare board: a real enclosed node adds sensor and quiescent draws, and LiFePO4 calendar aging plus self-discharge mean the cell will not actually deliver four full years of capacity. A small solar cell can keep the battery topped up across most of

the year, but size for your worst-month sun and expect seasonal limits - "maintenance-free for 5+ years in any climate" is an over-promise. Never charge LiFePO4 below 0 °C (32 °F); use a charger/controller that blocks charging below freezing.

Remote Sensor Deployment Guide

A practical guide for deploying LoRa mesh sensor nodes in the field for environmental monitoring, agriculture, and infrastructure monitoring.

Example use cases from the community

Weather station network

Multiple BME280-equipped nodes reporting temperature, humidity, and pressure at regular intervals. A gateway node (with internet uplink) receives all reports and logs them to a database. The mesh provides coverage even when nodes are miles apart across a farm or forest.

Soil moisture monitoring

Capacitive soil moisture sensors (e.g. STEMMA Soil Sensor) connected to a RAK4631 WisBlock. Node wakes every 30 minutes (this interval is user-configurable), reads moisture level, transmits if below threshold (irrigation alert), returns to sleep. With a typical pack (e.g. a 3000 mAh cell at well under 1 mA average) the battery can last many months on a single charge with occasional solar topping.

Water level monitoring

Ultrasonic or pressure transducer water level sensors for creek gauging, tank monitoring, or flood early warning. The mesh allows data to reach a gateway even when the sensor is in a remote canyon without cellular coverage. Note: a hobbyist mesh is a supplementary indicator only - official NWS/USGS warnings remain the authoritative flood-warning source.

Gate and door alerts

Reed switch or Hall effect sensor triggers a mesh message when a gate, shed door, or access point is opened. Uses a latching trigger architecture (interrupt-driven, not polling) for maximum battery life.

Choosing hardware for IoT sensor nodes

Board	MCU	Sleep current	Sensor interfaces	Best for
Heltec T096	nRF52840	~13 µA (per Heltec spec, bare board)	I²C, SPI, UART, ADC	Ultra-low power remote sensors
RAK4631 WisBlock	nRF52840	~2 µA (MCU only)	WisBlock sensor slot ecosystem	Modular sensor builds with WisBlock sensors
Heltec V4	ESP32-S3	~10 µA (bare-MCU deep-sleep figure; verify against Heltec V4 spec - typical whole-board deep-sleep current is higher)	I²C, SPI, UART, ADC, WiFi	Gateway nodes that also serve WiFi
XIAO nRF52840	nRF52840	~5 µA (see Seeed XIAO nRF52840 power spec)	I²C, SPI, UART	Compact DIY sensor builds

The RAK WisBlock sensor ecosystem

RAKwireless produces a modular ecosystem (WisBlock) where sensor modules snap directly onto the RAK4631 base board without soldering (see each module's RAKwireless datasheet page):

- RAK1901: Temperature + humidity (SHTC3)
- RAK1902: Barometric pressure (LPS22HB)
- RAK1904: 3-axis accelerometer (LIS3DH) - vibration / motion detection
- RAK12010: Ambient light sensor (see RAK12010 datasheet)
- RAK12019: UV index sensor (see RAK12019 datasheet)
- RAK12500: GPS module (u-blox ZOE-M8Q)
- RAK13010: SDI-12 interface for agricultural soil/water sensors (see RAK13010 datasheet)

This ecosystem significantly reduces build complexity - no custom PCB or wiring required for common sensor types.

Deployment checklist

- Enclosure rated IP65+ with UV-resistant housing
- Cable glands on all penetrations (sensor cable, antenna, power)
- Desiccant pack inside enclosure; replace annually
- Solar panel and MPPT charge controller for remote sites
- **In-line fuse (or PTC/polyfuse) on the battery positive lead**, sized just above peak load, to protect against a short-circuit fire if outdoor wiring chafes, floods, or is damaged. A lithium pack can source tens of amps into a fault.
- LiFePO4 battery (not LiPo) for outdoor/temperature-variable deployments - but note that LiFePO4, like all lithium chemistries, must **not be charged below 0 °C (32 °F)**; cold-charging causes lithium plating and is hazardous. Use a charge controller with a low-temperature charge cutoff. (LiFePO4 tolerates cold *discharge* better than LiPo, which is why it is preferred outdoors.)
- Antenna mounted outside enclosure (even 1 meter above the enclosure adds range)
- GPS coordinates recorded and logged - for the network map and maintenance records
- Label the enclosure with node name and maintainer contact

Getting data off the mesh

Sensor data on the mesh is useful only if someone can read it. Options for data collection:

1. **Manual retrieval:** A person with a phone and the app periodically reads nodes. Simple but not real-time.
2. **Room server (MeshCore):** A MeshCore Room Server (running on dedicated nRF52840 or ESP32 hardware) is a store-and-forward BBS for room chat history, not an MQTT/telemetry gateway. Verify the Room Server's role against the MeshCore Room Server documentation; MQTT bridging is a separate gateway component, not a built-in Room Server feature.
3. **Python API script:** For MeshCore, the meshcore-py library is scoped to message/contact logging - use a script on a server-connected node to log incoming messages/contacts to a database (see the MeshCore Python API page). For logging actual structured *sensor telemetry*, use the Meshtastic Telemetry + MQTT path instead.
4. **Internet-bridged gateway:** A MeshCore node with WiFi (Heltec V4) connected to a home network; the Python library reads data and forwards to any cloud service.