Soil Moisture Network Operating Manual

Installation, maintenance, and scaling guide

Digital Resources

Version 1.0
Prepared by Benjamen D. Miller
April 24, 2026

Contents

1 Purpose and Scope

This manual provides instructions for the installation, maintenance, and scaling of the soil moisture sensor network. It covers hardware setup, software configuration, data management, and troubleshooting procedures.

This network currently consists of three field nodes, each equipped with a TEROS 12 soil moisture/temperature sensor, and a base station that collects data for publishing to ThingSpeak. The main items you may edit, replace, expand, or remove are:

- base-station firmware, which controls Wi-Fi access, SD logging, ThingSpeak uploads, and packet handling
- field-node firmware, which controls node identity, transmit timing, sensor reads, and LoRa messaging
- ThingSpeak channels and API keys, which control where data is published and how it is read back (this is the main storage location for data)
- custom data visualizing and downloading webpages, which control how live charts and CSV exports are displayed (benmiller.us is an example of a custom webpage that reads from ThingSpeak channels and renders live charts and CSV exports)

2 System Overview

2.1 High-Level Architecture

Each soil moisture node is a remote field station that measures soil moisture and temperature, then transmits data wirelessly to a base station. The software loaded onto field nodes and base stations is maintained in the public teros12-LoRa GitHub repository (field node: field-node/field-node.ino, base station: base-station/base-station.ino). The base station handles SD logging, Wi-Fi access, ThingSpeak publishing, and UTC timestamping. The public project page renders live ThingSpeak widgets and charts for individual nodes and network-level summary plots.

The website also uses a ThingSpeak API feed download endpoint with per-node read keys in the JavaScript download helper. The manual below describes how to maintain that setup and how to extend it. The source of truth for node and base-station firmware behavior is the public teros12-LoRa GitHub repository, while website behavior is defined by the download helper script and the soil-moisture-network page source. These files can be downloaded directly in raw source text form here: download.js source (txt) and soil-moisture-network.html source (txt). These are helpful for setting up your own custom visualization webpage and serve as the foundation for the benmiller.us site.

3 Base Station Maintenance and Configuration

3.1 Rendering

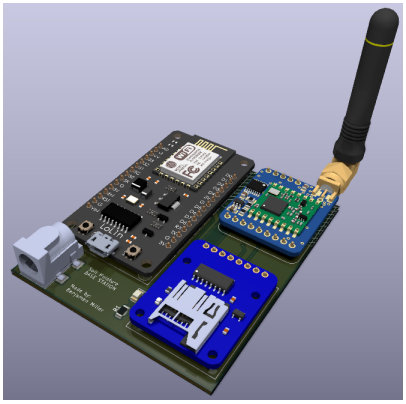

Figure 1: Base station rendering with omnidirectional antenna.

3.2 Base Station Bill of Materials

Subsystem	Suggested part / description	Manufacturer part number	Procurement/Datasheet link
Wireless radio	Adafruit RFM95W LoRa Radio Transceiver Breakout, 915 MHz	Adafruit 3072	DigiKey 1528-1667-ND / datasheet
Antenna	TE Connectivity Linx compact right-angle whip antenna, 915 MHz	ANT-916-CW-RCS-SMA	DigiKey 343-ANT-916-CW-RCS-SMA-ND / datasheet
Base station controller	Aceirmc wireless internet development board, ESP8266 NodeMCU V3 compatible	B0828L4743	Amazon purchase page
microSD adapter	Adafruit MicroSD Card Breakout Board, 5V or 3V logic compatible	Adafruit 254	DigiKey 1528-1462-ND / reference
Barrel jack connector	2.0 mm ID × 5.5 mm OD SMT power jack for 5V input adapter	PJ-002AH-SMT-TR	DigiKey CP-002AHPJCT-ND / datasheet
Wall power adapter	5V AC/DC wall adapter used with the barrel jack power input	16-00014	DigiKey 839-16-00014-ND / datasheet
Status LED	Blue SMD indicator LED for base-station status indication	LTST-C230TBKT	DigiKey 160-1889-1-ND / datasheet
LED current-limit resistor	100 Ω, 1%, 1/4 W, 1206 resistor for the blue status LED	RC1206FR-07100RL	DigiKey 311-100FRCT-ND / datasheet

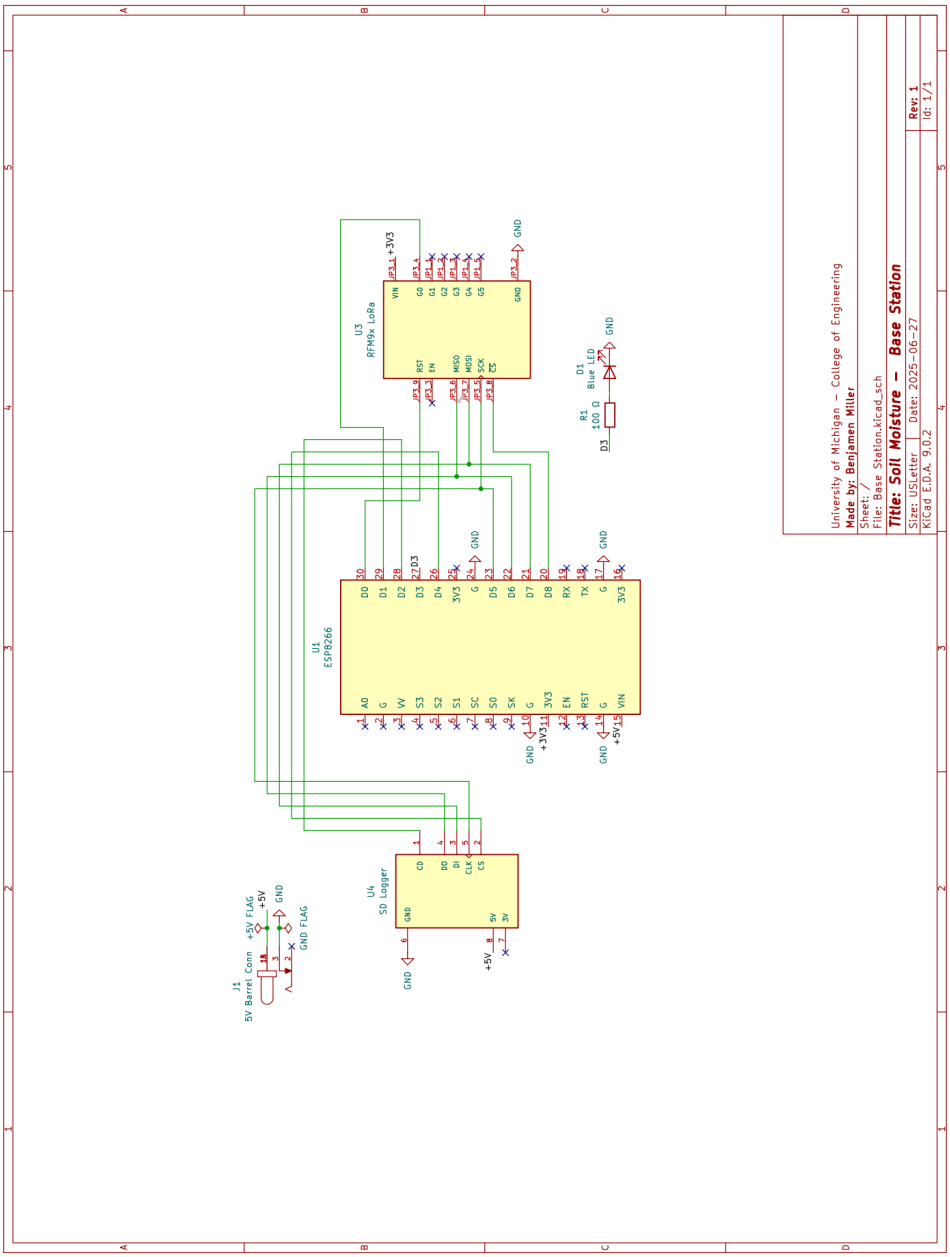

Figure 2: Base station schematic and pin mapping reference.

3.3 Base Station Custom PCB Files

- Base Station Custom PCB: BaseStation_Rev1.zip

3.4 3D Printed Parts

The base station includes an optional 3D printable housing and matching door for users who want to print the enclosure instead of buying a separate box.

Figure 3: Base station 3D printable housing preview.

- Base station housing STL: base_station.stl
- Base station door STL: base_station_door.stl

3.5 Code and Configuration Files

The base-station software source of truth is:

- firmware file: base-station/base-station.ino
- firmware overview: teros12-LoRa README

3.6 How to Edit Base Station Setup in Code

When changing deployment setup, you may (but need not) edit these items in base-station/base-station.ino:

1. Wi-Fi configuration values used to connect to the site network.
2. `THINGSPEAK_API_KEYS[4]` write-key assignments for node IDs 1–4.
3. Node-count guard logic if scaling beyond four nodes.
4. Field mapping in the ThingSpeak upload payload (fields 1–6).
5. SD logging behavior for `/soil.csv` and `/events.csv`.

After changing any field mapping, update website download and visualization code at the same time so plots and CSV exports remain consistent. For ThingSpeak credential handling and key placement rules, see API Key Management. For exact node-scaling code edits (including `NODE_ID`, `THINGSPEAK_API_KEYS`, and node-ID guard changes), see Scaling and Adding More Nodes.

3.7 Wi-Fi Commissioning Procedure

1. Follow the University of Michigan MSetup workflow and register the base station as **Other Device**. For portal details, see UMich MSetup documentation.
2. Enter the base station MAC address so `MSetup` authorizes the controller.
3. Power the base station and confirm it joins `MSetup`.
4. Verify the base station receives an IP address and successfully syncs time with `pool.ntp.org`.
5. Confirm a packet upload appears in ThingSpeak after a node transmission.

The base station controller currently in use has MAC address `08-3A-8D-DO-BE-89`. If you replace the controller, update the MAC address in the MSetup portal and re-register the device. The MAC address can be found by connecting to the base station's serial console and pushing a command to print the MAC address.

3.8 SD Card Verification Procedure

The base station should continuously maintain:

- `/soil.csv`: measurement rows for timestamp, battery, node ID, RSSI, SNR, and payload values.
- `/events.csv`: operational events including boot, time-sync, packet receive, timeout, and no-Wi-Fi events.

During maintenance, pull the card and verify both files are present, non-empty, and appending new rows after test transmissions.

3.9 Base Station Service Checklist

- verify power input and boot sequence
- verify network join and NTP sync
- verify SD writes for both `/soil.csv` and `/events.csv`
- verify successful ThingSpeak upload for at least one live packet

4 Field Node Maintenance and Configuration

4.1 Rendering

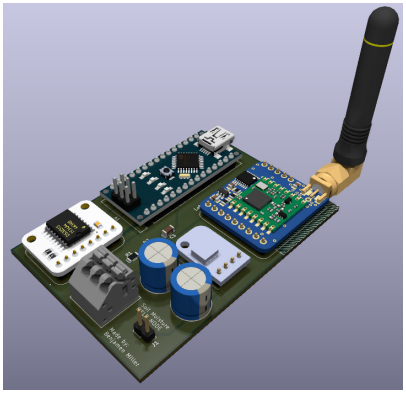

Figure 4: Field node rendering with omnidirectional antenna.

4.2 Field Node Bill of Materials

Subsystem	Suggested part / description	Manufacturer part number	Procurement/Datasheet link
Soil probe	METER Group TEROS 12 soil moisture sensor	TEROS 12	METER Group product page
Node controller	Arduino Nano Classic, field node controller	Arduino A000005	DigiKey search
Wireless radio	Adafruit RFM95W LoRa Radio Transceiver Breakout, 915 MHz	Adafruit 3072	DigiKey 1528-1667-ND / datasheet
Antenna	TE Connectivity Linx compact right-angle whip antenna, 915 MHz	ANT-916-CW-RCS-SMA	DigiKey 343-ANT-916-CW-RCS-SMA-ND / datasheet
Battery pack	Field battery source	D-size ER34615 non-rechargeable	-
Boost converter	Adafruit MiniBoost 5V @ 1A for boosting battery voltage to regulated 5V rail	Adafruit 4654	DigiKey 1528-4654-ND / datasheet
Bulk storage capacitor	Aluminum electrolytic capacitor, 470 µF, 16 V, radial through-hole	EEU-FR1C471	DigiKey P14394-ND / datasheet
Teros terminal block	3-position, 3.5 mm top-entry PCB terminal block for bare-wire TEROS 12 connection	1-2834011-3	DigiKey A123849-ND / datasheet
Battery connector header (Molex option)	2-pin vertical 2.54 mm Molex header listed for battery connection	0022272021	DigiKey WM4111-ND / datasheet

Subsystem	Suggested part / description	Manufacturer part number	Procurement/Datasheet link
RTC module	Adafruit DS3231 Precision RTC breakout	Adafruit 3013	DigiKey 1528-3013-ND / datasheet
RTC backup battery	3 V CR1220 coin cell for RTC module	CR1220	DigiKey SY033-ND / datasheet
Teros series resistor	510 Ω, 1%, 1/4 W, 1206 resistor listed in series with TEROS 12 line	RC1206FR-07510RL	DigiKey 311-510FRCT-ND / datasheet
Teros pull resistor	100 kΩ, 1%, 1/4 W, 1206 resistor listed parallel with Arduino on TEROS 12 line	RC1206FR-07100KL	DigiKey 311-100KFRCT-ND / datasheet
Teros line capacitor	220 pF, 50 V, C0G/NP0, 1206 ceramic capacitor listed in parallel with 100 kΩ resistor	C1206C221J5GACTU	DigiKey 399-C1206C221J5GACTUTR-ND / datasheet

4.3 Field Node Custom PCB Files

- Field Node Custom PCB: FieldNode_Rev2.zip

4.4 3D Printed Parts

The field node includes an optional 3D printable mounting bracket that is designed to fit a PJ864L enclosure.

- Field node mounting bracket STL: mounting_plate.stl

The PJ864L enclosure part number is included here so the bracket can be matched to the correct enclosure before printing or installation.

4.5 Code and Configuration Files

The field-node software source of truth is:

- firmware file: field-node/field-node.ino
- firmware overview: teros12-LoRa README

4.6 How to Edit Code for Each Node

For each physical node deployment:

1. set the node identity by editing `NODE_ID` in field-node/field-node.ino
2. confirm node-specific timing and transmit offset behavior
3. confirm Teros read path and SDI-12 line assumptions match hardware revision
4. compile and flash that firmware image to the intended node only
5. confirm the base station has a matching ThingSpeak write-key slot for that `NODE_ID`

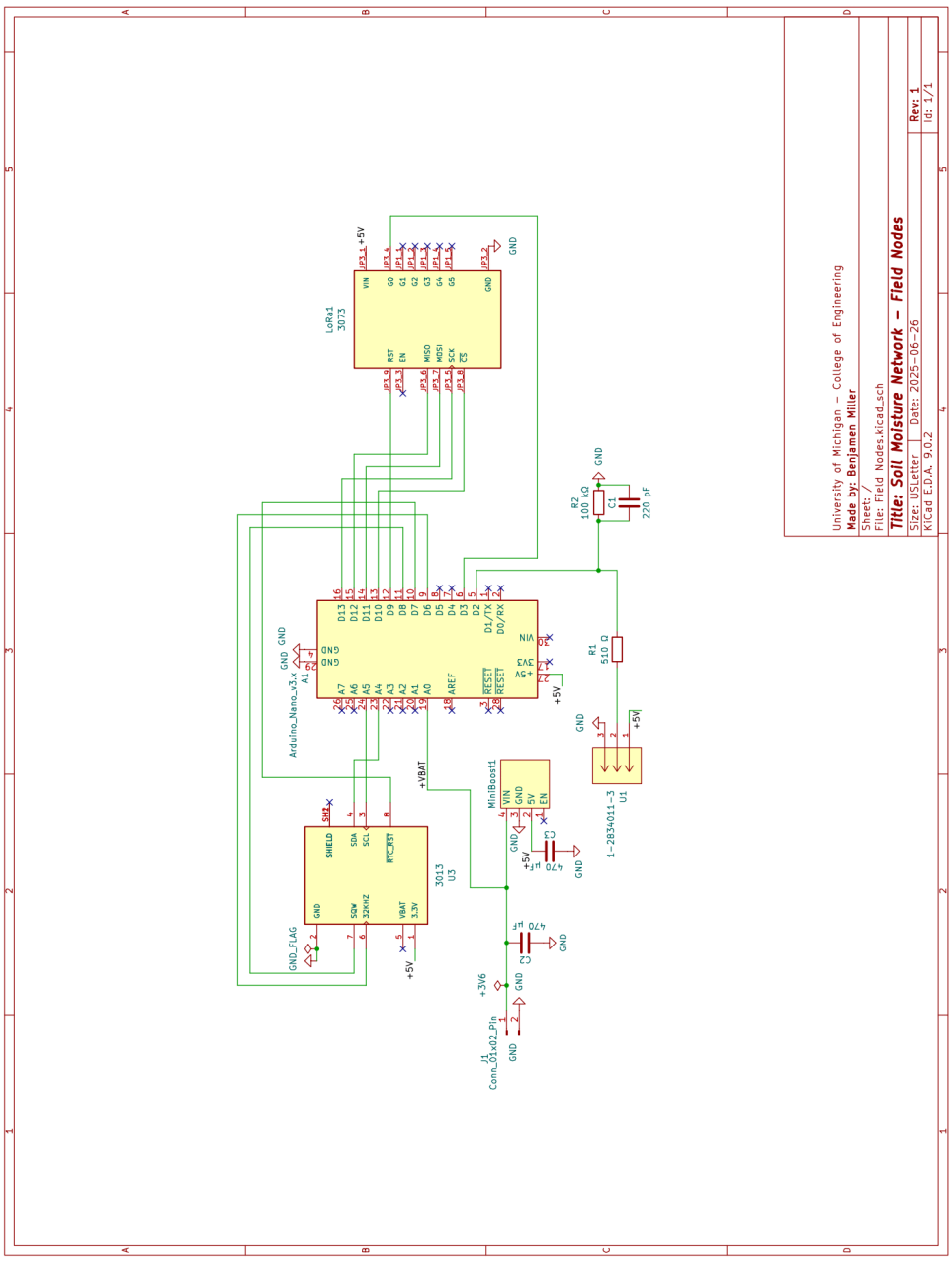

Figure 5: Field node schematic and pin mapping reference.

Figure 6: Field node 3D printable mounting bracket preview.

4.7 TEROS Wiring Order

When terminating the TEROS sensor lead into the field-node terminal block, keep the conductors in the same order shown in the installed reference photo below.

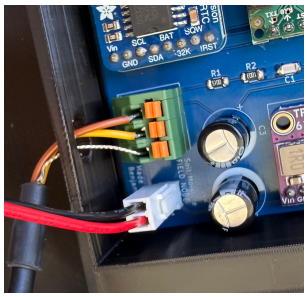

Figure 7: TEROS terminal block wiring reference.

4.8 Packet Format and Transport Behavior

The node sends LoRa records in the form:

- `DATA:<node_id>,<epoch>,<payload values...>`

  The base station confirms receipt with:

- `ACKTIME:<epoch>`

  Payload values are mapped to ThingSpeak field order through the base-station uploader.

4.9 Field Node Service Checklist

- verify battery voltage and polarity
- verify sensor cable condition and connector seating
- verify enclosure dryness, desiccant condition, and seal quality
- verify at least one packet is received and acknowledged by the base station
- verify a corresponding ThingSpeak update appears for that node

5 Code-Driven Device Order of Operations

This section describes runtime sequencing across both devices and the built-in fault handling path.

5.1 Nominal Sequence

1. Field node boots and initializes sensor and radio interfaces.
2. Field node samples TEROS data and battery voltage.
3. Field node transmits `DATA:<node_id>,<epoch>,<payload...>` via LoRa.
4. Base station receives packet, parses values, and logs to SD files.
5. Base station attempts ThingSpeak upload using the write key assigned to that node ID.

6. Base station sends `ACKTIME:<epoch>` back to the node.
7. Field node clears acknowledged backlog records and continues next cycle.

5.2 Error Handling and Fallback Paths

- If Wi-Fi is unavailable, the base station still logs events and data locally on SD for later recovery.
- If ThingSpeak upload fails, SD logs preserve a local operational history.
- If ACKs are missed, the field node retains records in backlog and retries.
- If backlog saturates at 10 records, field-node firmware forces a reboot.
- If two consecutive ACK cycles are missed, field-node firmware forces a reboot.
- If operator service is required, pressing reset triggers a clean runtime restart and a near-immediate post-boot transmit cycle.

5.3 Operational Checks After Any Change

After firmware edits, wiring changes, or channel/key updates:

1. verify node transmission and base-station ACK behavior
2. verify SD logging in both `/soil.csv` and `/events.csv`
3. verify ThingSpeak field mapping (fields 1–6) is correct
4. verify website CSV/download tools still parse the updated channel layout

6 Site and Network Setup

6.1 ThingSpeak Account Setup

1. Create or log in to a ThingSpeak account at https://thingspeak.mathworks.com/.
2. Create one private channel per node. The current firmware and website tooling support four nodes end to end on a free plan.
3. Enable at least six fields in each channel. The current payload writes Teros values to fields 1–3, battery voltage to field 4, RSSI to field 5, and SNR to field 6.
4. Record each channel ID and both write and read API keys.
5. Keep a copy of the channel IDs and key assignments for setup and maintenance reference.

6.2 Field Layout Recommendation

For consistency across nodes, use the following order in ThingSpeak fields:

1. field1: Teros BusID or first payload value
2. field2: soil moisture raw value or second payload value
3. field3: soil temperature in °C or third payload value
4. field4: battery voltage

5. field5: RSSI in dBm
6. field6: SNR in dB

This ordering matches the current upload code in base-station/base-station.ino. The uploader posts JSON to ThingSpeak's `/update.json` endpoint, stamps the sample time in UTC, and maps the Teros values into fields 1–3, with field 2 carrying raw soil moisture, field 3 carrying soil temperature in °C, battery into field 4, RSSI into field 5, and SNR into field 6. If you change the field layout, update the base-station uploader, the website CSV download helper, and any MATLAB visualizations at the same time.

6.3 API Key Management

ThingSpeak requires separate credentials for reading and writing. The operational rule is:

- write key: used by the base station when publishing live data to ThingSpeak
- read key: used by the website CSV download tool and any other analytics scripts

In the current firmware, write keys live in the base-station `THINGSPEAK_API_KEYS[4]` array. The node firmware does not contain ThingSpeak credentials; it only sends LoRa packets to the base station. Website read keys live in the source code for any visualization or download tool that needs to access the channel data.

Store keys only where they are used. Never hard-code a write key in a public repository. If you add, replace, or rotate a node, update both the base-station write-key array and the website read-key map so the channel remains usable from the upload path and the download path.

6.4 Website Download Tool Configuration

The current visualizing and download tools at benmiller.us are set up for 3 nodes. Adding further nodes can be done by adding additional ThingSpeak channels, but to keep the plots and data arrangement looking clean in the custom webpage format, the source code would need to be updated. See the System Overview section for links to the website source code files (download.js and soil-moisture-network.html) if you'd like to make your own site and expand the network. The helper expects each channel to expose the same field ordering described above.

7 Commissioning and First Data Upload

7.1 Initial Bring-Up Procedure

1. Power the base station and confirm it boots cleanly.
2. Power the node and confirm the sensor begins transmitting.
3. Verify the base station receives the packet and forwards it to ThingSpeak.
4. Open the channel dashboard and confirm the first record appears.
5. Validate the raw soil moisture, temperature, voltage, RSSI, and SNR values.

7.2 Interpretation of Live Dashboard Fields

The website charts render the following observables per node:

- raw soil moisture chart

- temperature chart
- battery voltage chart
- received signal strength chart
- SNR chart

The network summary page also shows cross-node visualizations for volumetric soil moisture and ground temperature. The node pages on the website are hard-linked to the current ThingSpeak channels, so update both the embedded channel IDs and the CSV download map when a node changes.

8 Reconnect and Recovery Guidelines

8.1 Normal Reconnect Behavior

Nodes should reconnect automatically after transient power loss or radio interruption. The field node maintains a backlog of up to 10 unsent records. Every transmission includes the entire backlog, and the base station acknowledges a batch with `ACKTIME:<epoch>` after logging the records. If the backlog reaches 10 records, the field node forces a reboot. If two ACKs are missed in a row, the field node also forces a reboot.

8.2 Operator Recovery Steps

If a node stops reporting:

1. verify battery voltage
2. check whether the base station is online and on the correct Wi-Fi network
3. inspect the enclosure for water ingress or a failed seal
4. confirm the antenna is attached and not physically damaged
5. power-cycle the node, or use the reset button for a forced single-point resend
6. if the node resumes, confirm that timestamps continue normally and that no duplicate records were created
7. if the backlog overflowed, expect a forced reboot as part of the recovery path

8.3 Forced Single-Point Resend by Reset Button

The current firmware does not expose a special user command for a one-shot resend. The supported operator action is to press the node's reset button, which reboots the field node, re-synchronizes time, and causes the next transmission cycle to start from a clean runtime state. Because the first slot after boot skips the node-offset delay, this is the nearest supported behavior to an immediate resend. Use this when you need a single current reading immediately after servicing.

Do not use repeated reset presses as a substitute for fixing a persistent connectivity issue.

9 SD Card Contents for the Base Station

The base station uses a microSD card; treat it as the local operational log and offline recovery media. The current firmware writes directly to `/soil.csv` and `/events.csv` on the SD card. The SD card should contain:

- `soil.csv`: sensor rows in the format `timestamp,battery,node_id,rssi,snr,...`
- `events.csv`: boot, NTP, no-Wi-Fi, time-request, time-sync, packet-received, and timeout events

10 Scaling and Adding More Nodes

10.1 Node Addition Workflow

To add a new node to the network:

1. Assign the next available node ID (for example, 4, 5, 6, ...).
2. Create a ThingSpeak channel for that node and collect its Write API key.
3. Create one field-node firmware copy per additional hardware unit and edit the field-node sketch file (`field-node.ino`):
   - At line 22, change `constexpr uint8_t NODE_ID = 1;` to a unique value for that specific node (for example, 2, 3, 4).
   - This `NODE_ID` is embedded in transmitted messages (`REQT:<id>` and `DATA:<id>,...`).
   - This `NODE_ID` also controls transmit staggering: each node waits `45 * NODE_ID` seconds before transmitting to reduce over-the-air collisions.
4. Flash each modified `field-node.ino` build to its corresponding Arduino Nano unit.
5. Edit `base-station/base-station.ino` so the base station accepts and uploads the new node ID(s):
   - `THINGSPEAK_API_KEYS` array (lines 37–42): the current code is sized for 4 nodes. Increase the array size from `[4]` to the new node count, and add one Write API key entry per new node.
   - Node ID guard (line 98): update `if (nodeId == 0 || nodeId > 4) return false;` so the upper bound matches your new maximum node ID.
   - If this guard is not updated, packets from new nodes can still be logged to SD, but their ThingSpeak uploads will be rejected.
6. Update website download helpers and dashboard mappings for the added channel(s).
7. Commission each new node and confirm both SD logging and ThingSpeak uploads are working.

10.2 Dashboard and Website Updates

When a node is added or removed, update:

- the live ThingSpeak channel list
- the CSV download node map
- the network overview charts if they are node-specific

- any summary tables on the project page

The current visualizing tools at benmiller.us are set up for 3 nodes. Adding additional nodes is straightforward at the ThingSpeak level, but if you want the plots and data layout to stay nicely arranged in the custom webpage format, the source code for the site visualizers and downloaders needs to be updated. Feel free to grab the source code from the previous sections, host it on your own GitHub Pages site, and customize the layout as needed for your node count and channel arrangement.

10.3 Code Edit Points

When you need to change the system, these are the files that matter most:

- field-node firmware (public repo): field-node/field-node.ino for `NODE_ID`, timing, backlog logic, LoRa parameters, or the Teros read routine.
- base-station firmware (public repo): base-station/base-station.ino for Wi-Fi SSID, ThingSpeak write keys, SD logging behavior, or packet handling.
- Download helper script source: download.js source (txt) for read keys, node IDs, CSV column naming, or download behavior.
- Embedded dashboards and channel widgets.
- Firmware documentation: teros12-LoRa README for operational description updates.

11 Troubleshooting

11.1 No Data on ThingSpeak

Potential causes include:

- wrong write key
- wrong channel ID
- no Wi-Fi connectivity
- base station not authenticated to the network
- node not powered or not transmitting
- incorrect field order in the payload

11.2 Data Appears But Is Corrupted

Check for:

- swapped sensor leads
- incorrect sensor power voltage
- loose ground connection
- serial framing or field parsing issues
- stale or partial transmissions after a reset

11.3 Low RSSI or High Packet Loss

Inspect antenna placement; for omnidirectional antennas, keep them in the same plane between the base station and field nodes. Also check enclosure shielding, distance to the gateway, and the battery supply.

12 Revision Log Template

Date	Revision	Notes
July 10, 2025	Remote logging system installed	Remote logging system installed for the network.
April 22, 2026	Field node 3 relocated	Field node 3 was moved from its original location to a new location northeast of the original site to accommodate Matthaei Botanical Gardens staff operations.

13 Appendix A: Current Public Website References

The current site uses the following project resources and live endpoints:

- Soil moisture network page: Soil Moisture Network
- ThingSpeak dashboards
- Combined CSV download helper source: download.js source (txt)
- MATLAB visualizations