A Sensor Data Gathering and Cloud-Based Transmission Prototype System

Ebuka Philip Oguchi • Junxiao Zhang • Md Didarul Islam

Department of Computer Science and Engineering
University of Nebraska–Lincoln, Lincoln, Nebraska, USA

Technical report • December 17, 2021

ABSTRACT

Low-cost sensing systems often require a dedicated interface for each sensor type, which increases configuration effort and limits field replacement. This paper presents a prototype Internet of Things architecture that identifies an attached analog sensor, samples its output, sends a structured telemetry packet over a LoRa link, and forwards validated data through a gateway to Microsoft Azure. A passive resistor divider provides a sensor-specific identification voltage. The transmitter compares the measured voltage with predefined windows to distinguish a TMP36 temperature sensor from a SEN0193 capacitive soil-moisture sensor. A SparkFun SAMD21 Pro RF transmitter and an ESP32-based gateway implement the wireless path, while a watchdog timer and a 16-bit status field support recovery and diagnosis. In the reported prototype tests, all 84 engineering trials and all 70 user-oriented trials passed. Adding the identification divider changed the measured 3.3 V supply to 3.299 V, and the recorded sensor-output traces showed no visible degradation. The prototype demonstrates a practical path toward plug-and-play sensing, while highlighting the need for larger sensor libraries, energy measurements, long-duration wireless tests, stronger device authentication, and weather-resistant packaging.

Keywords— Internet of Things, LoRa, sensor identification, cloud telemetry, plug-and-play sensing, TMP36, SEN0193.

1. Introduction

Environmental and agricultural monitoring systems must often combine sensors with different transfer functions and connector conventions. Replacing a sensor may therefore require rewiring, code changes, or manual reconfiguration. The prototype described here explores a simple alternative: each supported sensor carries a passive identification network, allowing a common transmitter to infer the sensor type before sampling and transmitting its data.

The system integrates three operations in one pipeline: automatic sensor-type identification, local measurement with diagnostic status, and long-range wireless forwarding to a cloud service. Its main contribution is the end-to-end implementation of this pipeline using inexpensive development hardware and two representative analog sensors. The work is a proof of concept; it does not claim production readiness or comprehensive performance characterization.

The paper reorganizes and clarifies a team project completed for CSCE 838 in December 2021. Administrative records, schedules, source-code listings, user-manual material, and cloud credentials from the course report are intentionally excluded from this public manuscript.

2. Design Goals and System Requirements

The prototype was designed around five functional goals:

- Accept either of two analog sensors through a common transmitter interface and identify the attached sensor automatically.

- Sample temperature or soil-moisture data and encode the result with node, packet, timestamp, and diagnostic information.
- Transmit telemetry from the sensor node to a gateway through a LoRa radio link.
- Reject packets that do not carry the configured system identification code before cloud forwarding.
- Operate as a compact, battery-capable prototype with watchdog recovery and visible diagnostic output.

Both nodes operate from a 3.3 V design rail. The prototype used the 915 MHz-capable RFM95W radio on the SparkFun SAMD21 Pro RF platform; the project configuration selected 903.9 MHz for point-to-point communication. The sampling loop acquired an analog reading every four seconds.

3. End-to-End Architecture

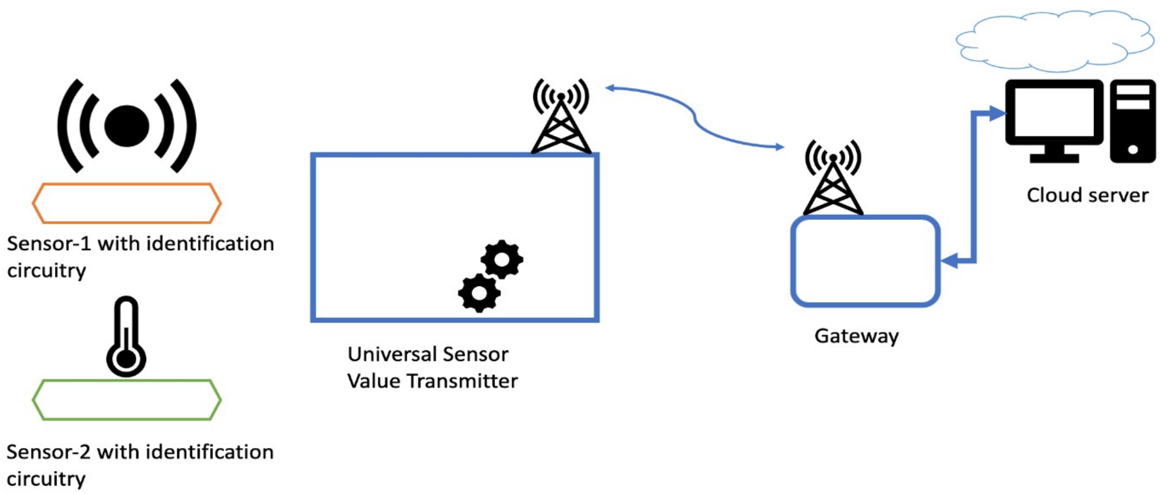


Figure 1. End-to-end prototype architecture: sensors connect to a universal transmitter, LoRa carries the packet to a gateway, and the gateway forwards accepted telemetry to the cloud.

The transmitter contains two analog inputs: one for the sensor measurement and one for the identification voltage. At start-up, firmware maps the identification voltage to a supported sensor type. It then applies the appropriate conversion function, constructs a telemetry packet, and transmits the packet over LoRa. The gateway receives the packet, checks the system identification field, and forwards accepted data to Microsoft Azure in JSON form.

4. Automatic Sensor Identification

4.1 Passive identification network

A resistor divider generates a sensor-specific voltage from the shared 3.3 V supply. With R_1 connected to the supply and R_2 connected to ground, the transmitter observes

$$V_{ID} = V_{CC} \times R_2 / (R_1 + R_2).$$

The prototype assigned approximately 2.30 V to the TMP36 path using $R_1 = 517\ \Omega$ and $R_2 = 1200\ \Omega$. Reversing the resistor values produced approximately 0.99 V for the SEN0193 path. Firmware used a ±0.5 V acceptance window around each nominal identification voltage. Values outside both windows were labeled unidentified.

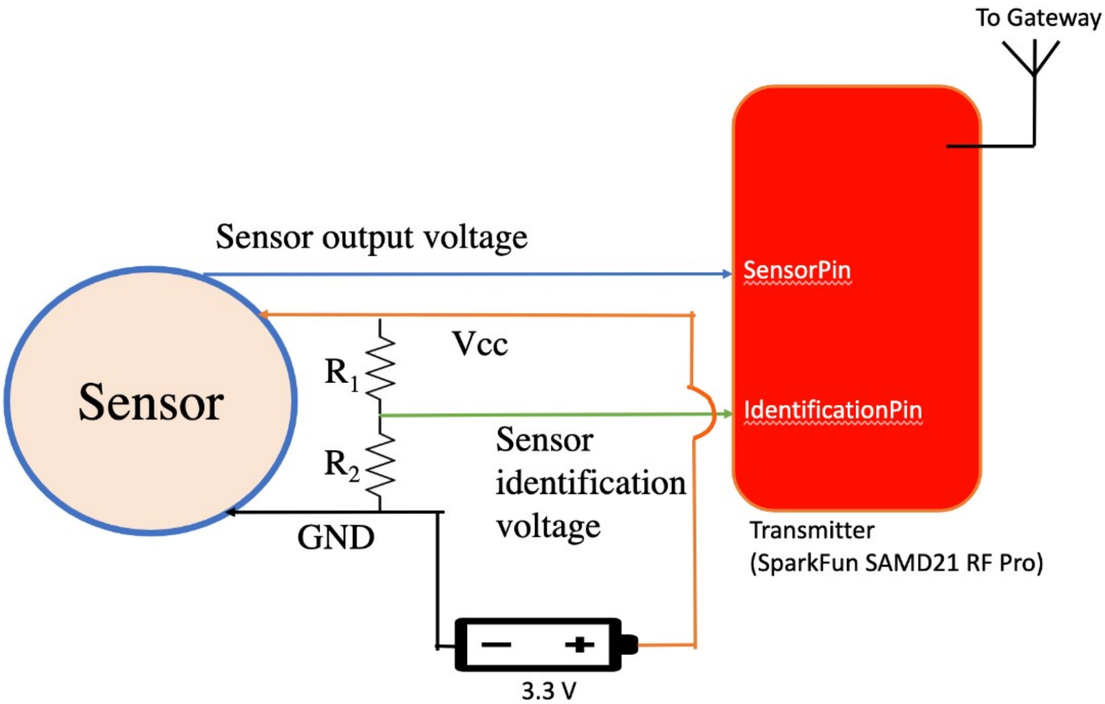


Figure 2. Sensor-to-transmitter interface. Separate analog lines carry the sensor output and the resistor-divider identification voltage.

4.2 Sensor conversion

The TMP36 produces a voltage proportional to temperature with a nominal 10 mV/℃ scale factor and a 750 mV output at 25℃ [2]. The prototype converted its analog voltage to degrees Celsius using the device transfer relation. The SEN0193 produces an analog response that varies with soil moisture; the project used an empirical conversion supplied with the original implementation. Because soil composition and probe placement affect capacitive measurements, application-specific calibration is required before quantitative field deployment [3].

5. Telemetry, Gateway, and Cloud Path

The transmitter represented each reading as an integer array. The original implementation reserved space for growth and used integers because floating-point transmission was not completed during the project. Table 1 clarifies the fields described in the source report.

Field	Purpose
System identification code	Allows the gateway to accept packets from the configured prototype.
Timestamp	Elapsed time since program start.
Node ID	Identifies the originating sensor node.
Sensor type	Encodes temperature, soil moisture, or unidentified.
Packet ID	Monotonically increasing packet counter.
Sensor value	Converted measurement represented as an integer.
Error/status code	Sixteen-bit diagnostic field.
Reserved	Space for future packet fields.

The gateway compared the incoming system identification code with the configured value. Matching packets were serialized and uploaded to Azure; nonmatching packets were rejected. Azure IoT services can receive streaming device telemetry and provide dashboards, queries, and data export for later analysis [5]. Public release of this manuscript omits all device connection strings and keys. A deployed system should store credentials outside source code, rotate any previously exposed key, and use cryptographic device identity rather than a plaintext packet identifier.

6. Reliability and Diagnostic Design

A watchdog timer monitored program execution and supported recovery after a software stall. A 16-bit diagnostic word recorded reset causes, communication failures, invalid measurements, and debugger state. The packet counter and elapsed-time field provided additional context for detecting missing packets.

Bit	Reported status
0	Power-on reset
1	Watchdog reset
2	External reset
3	Protection error
4	Device service unit failure
5	Bus error
6	CPU reset phase
7	Hot-plugging enabled
8–9	Debug communication flags
10	Debugger present
11	Missing packet
12	Reception failure
13	Transmit failure
14	Measurement above 100 or below 0
15	Reserved

7. Prototype Implementation

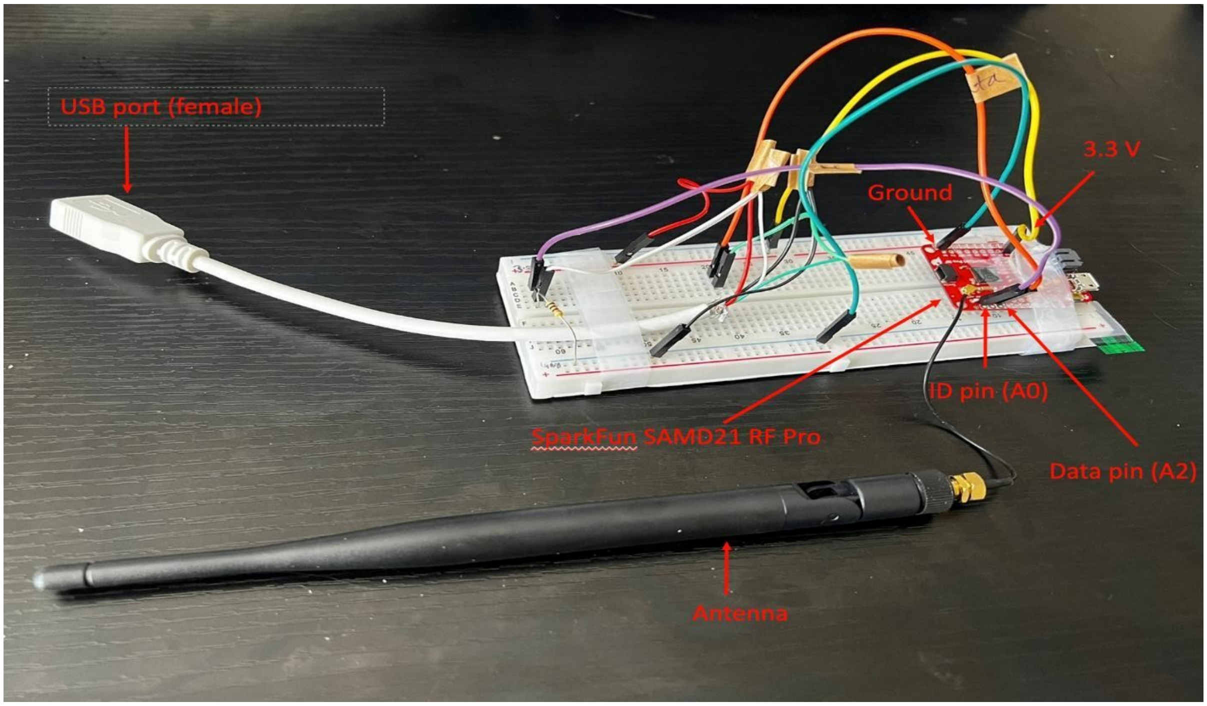


Figure 3. Plug-and-play transmitter prototype with the sensor and identification connections exposed for bench testing.

The transmitter combined a SparkFun SAMD21 Pro RF board, an RFM95W LoRa radio, an antenna, and the analog sensor interface. The gateway used an ESP32-class board with a LoRa radio and Wi-Fi connectivity. The SparkFun platform integrates a SAMD21 microcontroller with an RFM95W radio for operation in the 915 MHz ISM band [1]. The final enclosure separated transmitter and gateway functions while retaining external antenna and sensor connections.

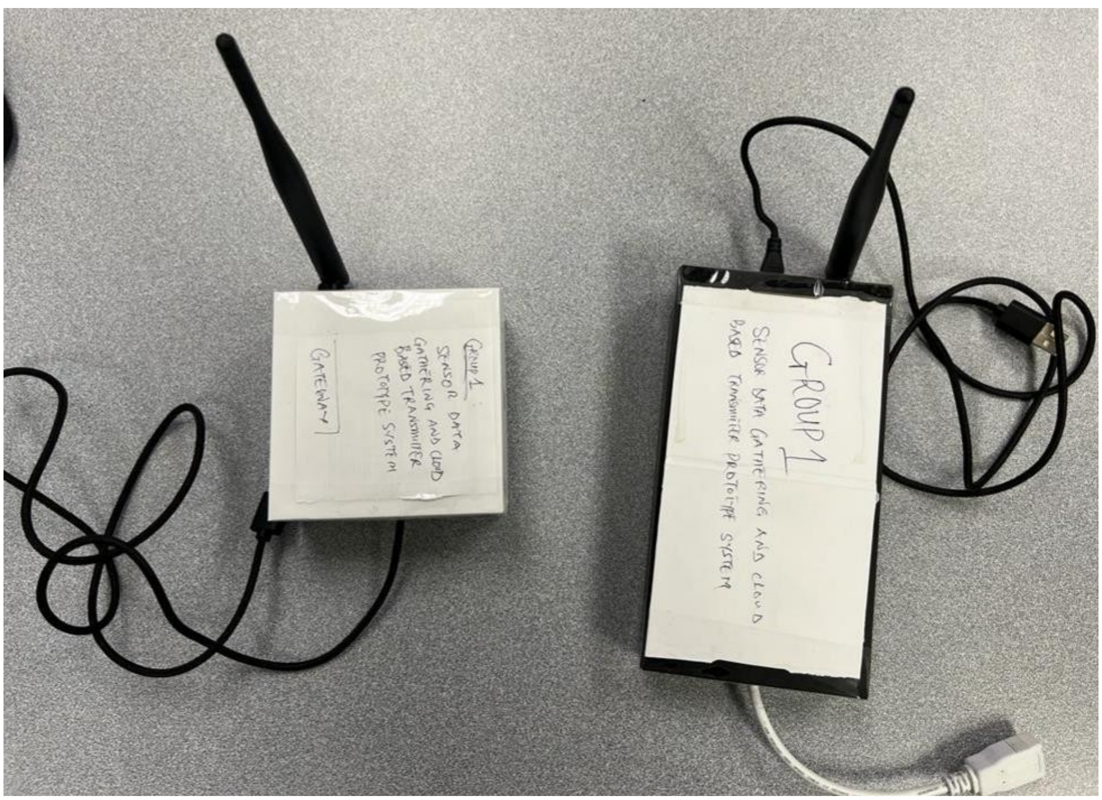


Figure 4. Packaged transmitter and gateway used for the final prototype demonstration.

8. Evaluation

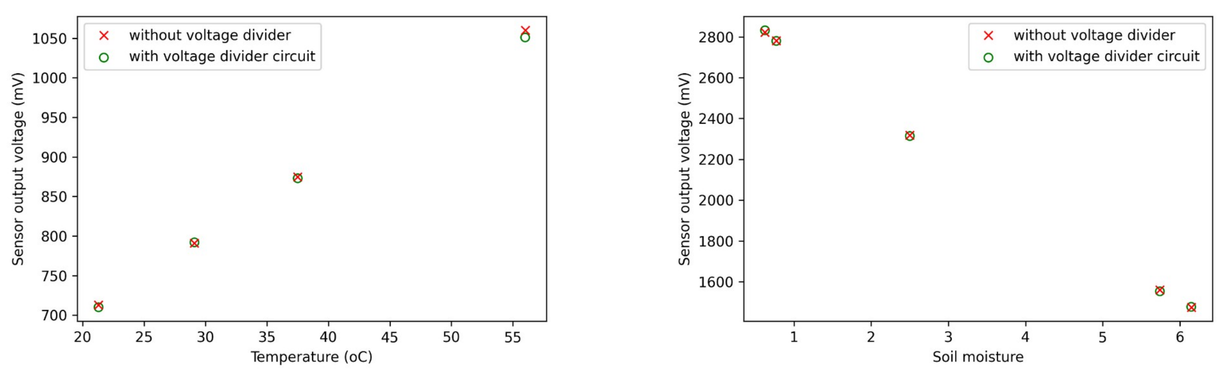
8.1 Functional testing

The source report records 84 engineering trials and 70 user-oriented trials. Every listed trial passed. These counts demonstrate repeatable operation under the project test plan, but the report does not provide raw observations, trial duration, confidence intervals, packet-loss traces, or environmental conditions. The results should therefore be interpreted as functional verification of the prototype.

Test	Trials	Passes	Reported result
Power-on behavior	20	20	Passed
Error logging	4	4	Passed
LoRa communication	30	30	Passed
Universal sensor interface	20	20	Passed
Sensor identification	10	10	Passed
Indoor/outdoor operation	10	10	Passed
Ease of use	20	20	Passed
Durability and reliability	20	20	Passed
Plug-and-play operation	20	20	Passed

8.2 Effect of the identification circuit

The measured supply changed from 3.300 V to 3.299 V after the identification divider was added. The recorded TMP36 and SEN0193 output traces with and without the divider were visually similar, and no operational degradation was reported. The project did not complete current measurements, so the divider's energy cost and its effect on battery life remain unknown.



(a) TMP36 output *(b) SEN0193 output*

Figure 5. Recorded sensor-output voltages with and without the identification circuit, as reported in the original prototype study.

9. Limitations and Security Considerations

- The identification library contains only two sensors. Wider tolerance analysis is needed as the number of voltage-coded sensors grows.
- The ±0.5 V windows are broad; resistor tolerance, ADC error, supply variation, connector resistance, and temperature drift were not characterized.
- Communication testing reports pass counts but no distance-dependent packet-delivery ratio, latency distribution, interference test, or confirmed one-mile field result.
- Current draw, battery life, divider losses, and sleep-mode behavior were not measured.

- The integer-only payload reduces measurement resolution, and the packet-size limit constrains expansion.
- The enclosure was not assigned an ingress-protection rating and was not validated for sustained outdoor exposure.
- The system identification code filters unrelated packets but does not provide authentication, confidentiality, replay protection, or message integrity.

A production revision should use authenticated encryption on the wireless link, unique per-device credentials, secure key provisioning, cloud-side access control, and a defined credential-rotation process. These controls are future design requirements rather than features of the evaluated prototype.

10. Future Work

The most useful next step is a controlled field evaluation with timestamped raw data. Measurements should include packet-delivery ratio and latency versus range, current draw in active and sleep modes, battery life, identification accuracy across supply and temperature variation, and calibrated sensor error. Hardware development should add switched sensor power, disconnect the identification divider after startup, local SD-card buffering, a real-time clock, and an IP67-class enclosure. Firmware should support floating-point or fixed-point encoding, a versioned packet schema, secure device authentication, and queueing during cloud outages. Finally, cloud software should expose sensor history, quality flags, and exportable data while preserving the original acquisition timestamp.

11. Conclusion

This work demonstrates an end-to-end sensor telemetry prototype in which a passive voltage code identifies an attached analog sensor, a common transmitter samples the appropriate quantity, LoRa carries a structured packet to a gateway, and the gateway forwards validated data to a cloud platform. The design successfully integrated two sensor types, watchdog recovery, diagnostic reporting, and cloud delivery under the project test plan. Its principal value is the architectural proof of concept: sensor identification can be added with minimal hardware. A publishable next-generation study should quantify identification margins, energy use, communication reliability, security, and long-term field performance.

Author Contributions

Ebuka Philip Oguchi: watchdog timer, error logging, and physical packaging. Junxiao Zhang: wireless transmission, cloud service, and packet identification. Md Didarul Islam: hardware design and implementation, sensor data gathering, and evaluation of the identification circuit. All authors contributed to integration, testing, and the original report.

References

[1] SparkFun Electronics, “SparkFun SAMD21 Pro RF Hookup Guide,” SparkFun Learn. https://learn.sparkfun.com/tutorials/sparkfun-samd21-pro-rf-hookup-guide.

[2] Analog Devices, “TMP35/TMP36/TMP37: Low Voltage Temperature Sensors,” Rev. H data sheet. https://www.analog.com/media/en/technical-documentation/data-sheets/tmp35_36_37.pdf.

[3] DFRobot, “Gravity: Corrosion-Resistant Capacitive Soil Moisture Sensor, SKU SEN0193,” DFRobot Wiki. https://wiki.dfrobot.com/sen0193.

[4] Semtech, “What Is LoRa?” https://www.semtech.com/lora/what-is-lora.

[5] Microsoft, “What is Azure IoT Central?” Microsoft Learn. https://learn.microsoft.com/en-us/azure/iot-central/core/overview-iot-central.

[6] Microsoft, “How to use data explorer to analyze device data,” Microsoft Learn. https://learn.microsoft.com/en-us/azure/iot-central/core/howto-create-analytics.

Technical Appendices

The following appendices preserve the detailed engineering content of the original 41-page course report while removing schedules, revision histories, grading material, personal instructions, and cloud secrets. They document requirements, hardware, firmware logic, tests, deployment procedures, and proposed extensions.

A.1 Development context

The prototype was developed as a three-person Internet of Things project. The system was expected to demonstrate a complete data path rather than a single isolated circuit: sensor connection, automatic identification, analog conversion, packet creation, LoRa transmission, gateway validation, and cloud ingestion. The implementation emphasized modular replacement of supported sensors and observable failure status.

A.2 Functional scope

Subsystem	Required behavior	Evidence retained in this report
Sensor interface	Power an analog sensor and expose measurement and identification voltages.	Schematics, circuit photographs, and voltage traces.
Transmitter	Identify the sensor, sample every four seconds, convert the reading, and create a packet.	Firmware flowchart and packet dictionary.
LoRa link	Deliver the packet to the gateway on the configured radio channel.	Communication trials and gateway serial output.
Gateway	Reject packets with an incorrect system identifier and forward accepted packets.	Gateway flowchart and accepted/rejected examples.
Cloud path	Upload accepted measurements as structured telemetry.	Azure JSON capture with credentials omitted.
Reliability	Record diagnostic state and recover from stalled execution.	Watchdog and 16-bit status definitions.

A.3 Boundaries

The system is a laboratory prototype. It does not establish regulatory compliance, production radio range, long-term enclosure durability, battery lifetime, calibrated agronomic accuracy, or cryptographic security. Those items are treated as validation targets for a subsequent design.

Appendix B. System Decomposition and Hardware Inventory

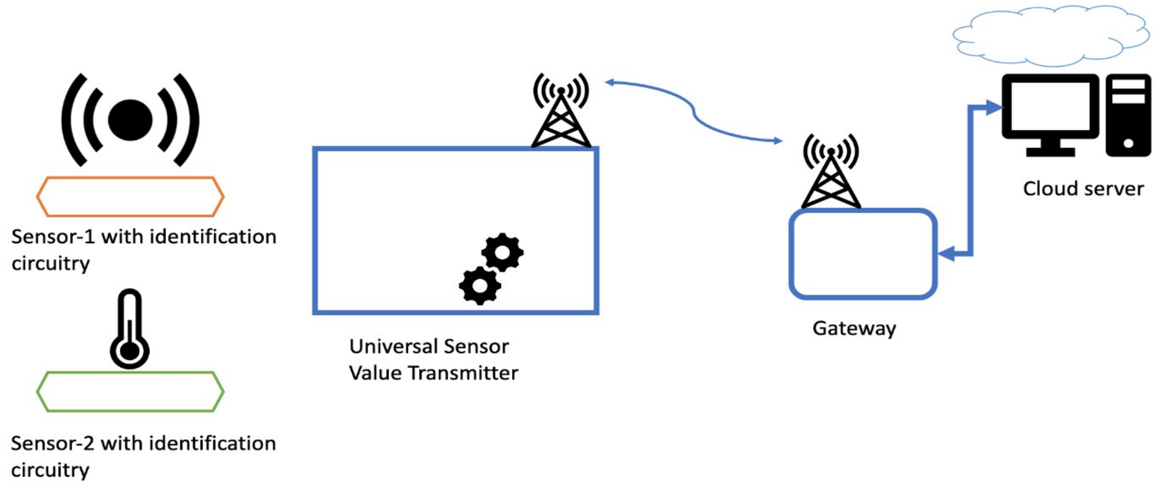
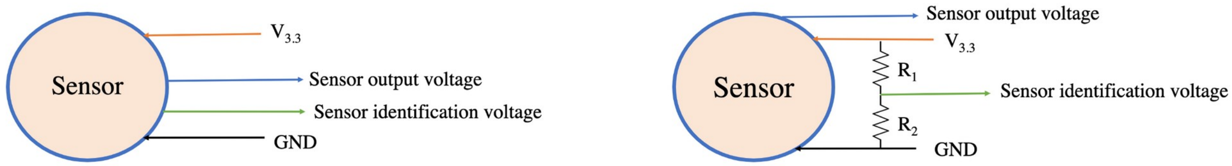


Figure B.1. Functional decomposition from sensor to transmitter, gateway, and cloud service.

The design contains four logical layers. The sensing layer converts a physical quantity to an analog voltage and presents a second voltage that identifies the sensor. The edge layer reads both inputs and packages the result. The gateway layer validates the source and bridges LoRa to an Internet-connected service. The cloud layer stores and exposes the resulting telemetry.

Element	Role	Prototype implementation
Temperature sensor	Measures ambient temperature.	Analog Devices TMP36.
Soil-moisture sensor	Measures a moisture-dependent capacitive response.	DFRobot SEN0193.
Identification network	Generates a type-specific voltage.	Two-resistor divider attached to the sensor interface.
Transmitter/controller	Samples inputs, converts values, logs status, and transmits.	SparkFun SAMD21 Pro RF with RFM95W radio.
Gateway	Receives LoRa packets and forwards accepted telemetry.	ESP32-class LoRa/Wi-Fi development platform.
Cloud service	Receives JSON telemetry for display and analysis.	Microsoft Azure IoT service used in the prototype.

Appendix C. Sensor Connector and Identification Circuit



(a) Sensor connector signals *(b) Voltage-divider concept*

Figure C.1. Measurement and identification signals share the transmitter supply and ground.

The connector exposes power, ground, a sensor-output line, and an identification line. A separate identification conductor is necessary because the measurement voltage changes with the physical quantity. The passive divider produces a stable nominal level that can be checked once at startup or periodically for hot-plug detection.

Sensor path	R_1	R_2	Nominal V_ID at 3.3 V	Firmware window
TMP36	517 Ω	1200 Ω	≈2.30 V	1.80–2.80 V
SEN0193	1200 Ω	517 Ω	≈0.99 V	0.49–1.49 V

The two acceptance windows do not overlap, leaving a gap between 1.49 V and 1.80 V. The firmware treats measurements outside the configured windows as unidentified. A larger sensor catalog would require narrower windows, more precise resistor selection, ADC characterization, or a digital identification mechanism.

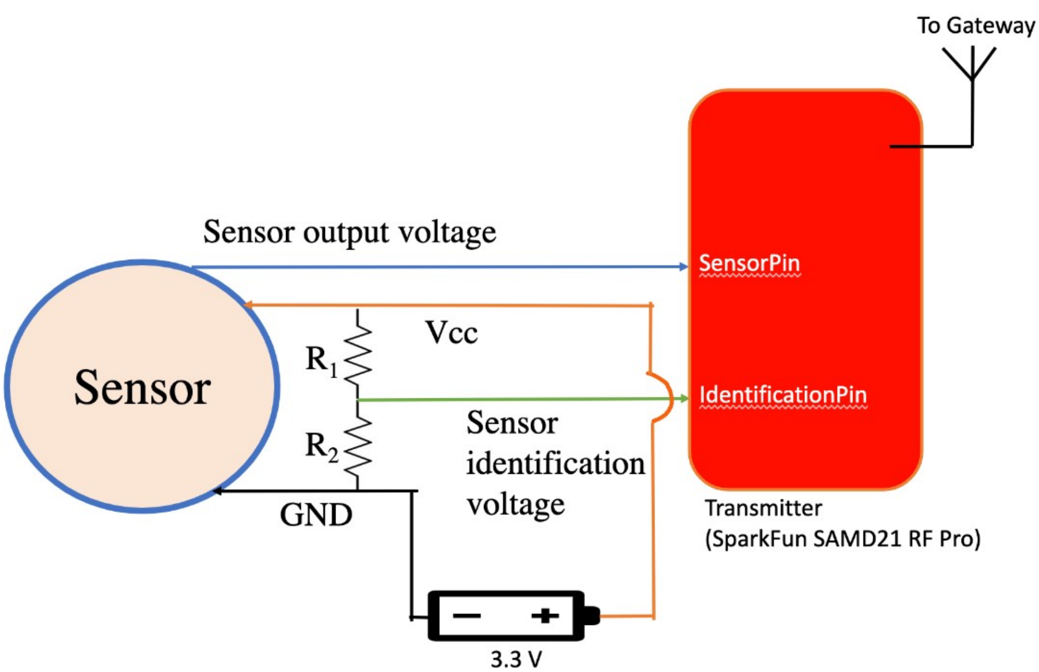
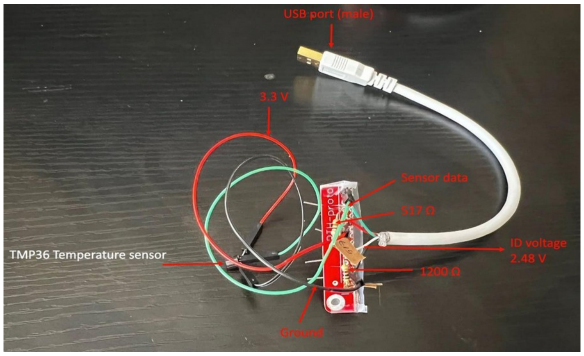
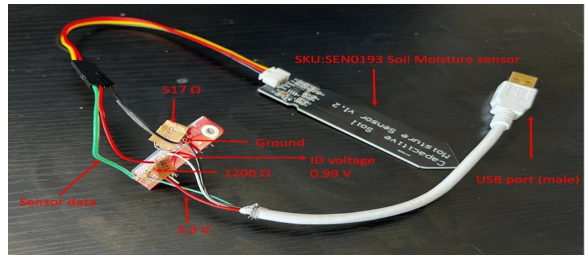


Figure C.2. Complete connection from the sensor and identification divider to the transmitter analog inputs.

Appendix D. Sensor-Specific Hardware Implementation



(a) TMP36 identification circuit



(b) SEN0193 identification circuit

Figure D.1. Bench implementations of the two supported sensor interfaces.

D.1 Temperature channel

The TMP36 operates from the 3.3 V rail and provides an analog output with a nominal scale factor of 10 mV/℃. The prototype used the sensor transfer relation to convert the sampled voltage to degrees Celsius. Representative data-sheet points are 0.10 V at −40℃, 0.75 V at 25℃, and 2.00 V at 150℃.

D.2 Soil-moisture channel

The SEN0193 produces an analog voltage that changes with the dielectric environment around the probe. The project converted this voltage using an empirical relationship included in the implementation. For field measurement, the probe should be calibrated against the target soil, packing density, salinity, temperature, and gravimetric water content.

D.3 Plug-and-play wiring

Both sensor assemblies present the same logical interface to the transmitter. Once power is applied, the transmitter identifies the attached assembly, chooses the corresponding conversion path, and begins periodic sampling without a firmware upload or manual sensor-selection command.

Appendix E. Identification-Circuit Loading Analysis

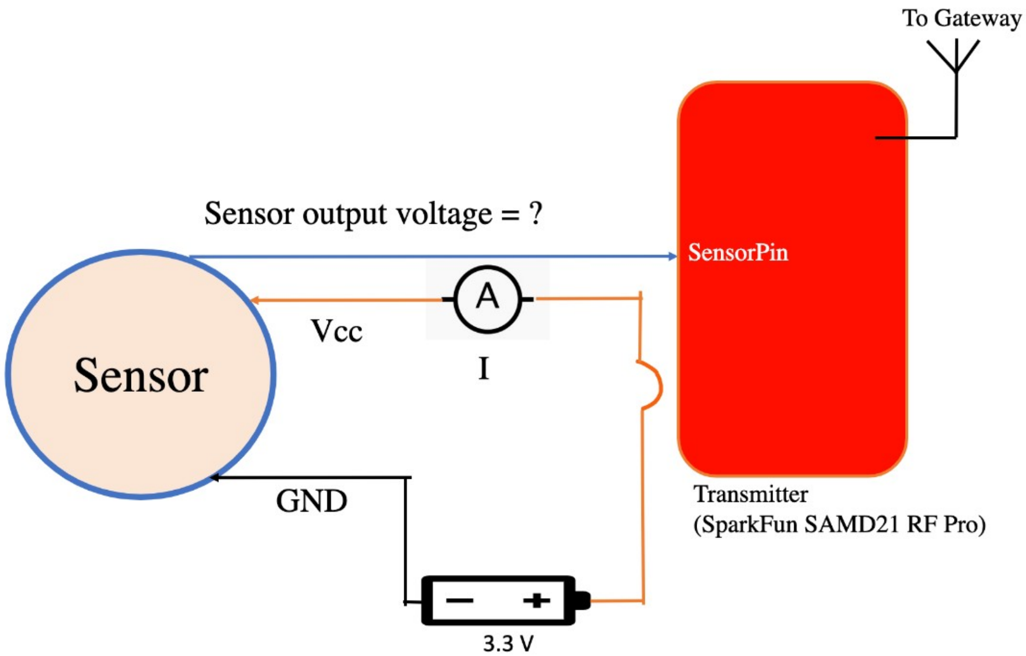


Figure E.1. Proposed current-measurement arrangement before adding the identification divider.

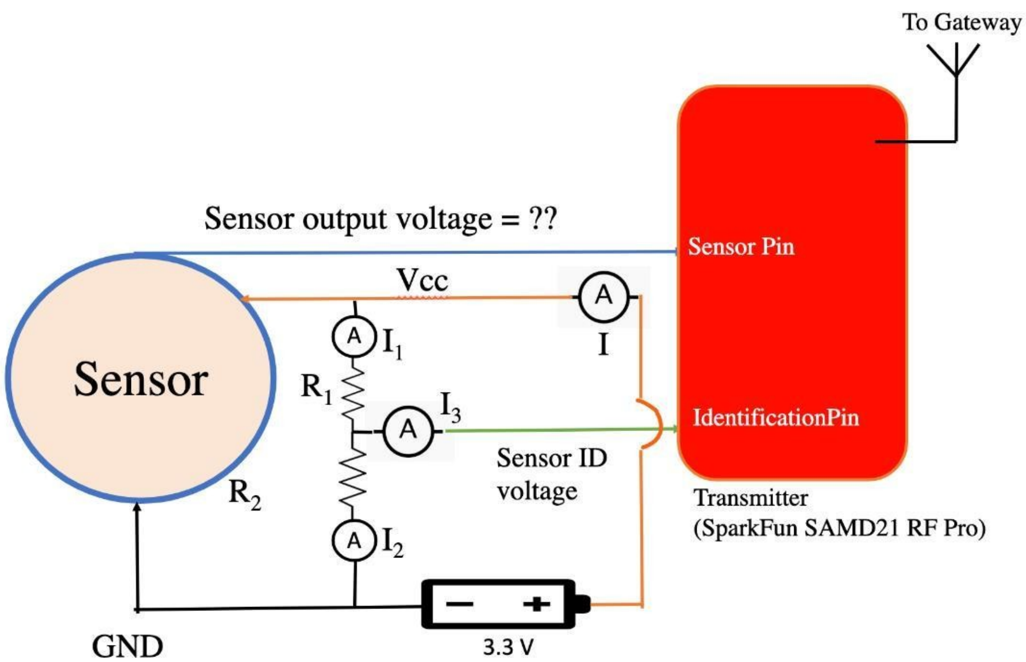
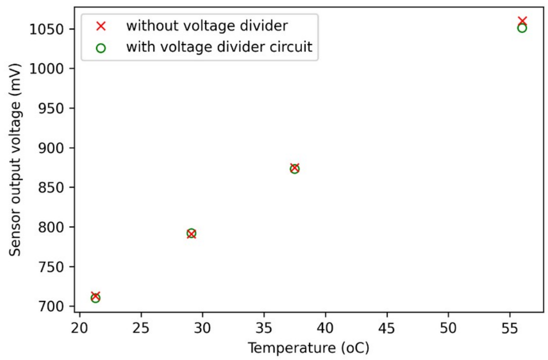


Figure E.2. Proposed current-measurement arrangement with the identification divider attached.

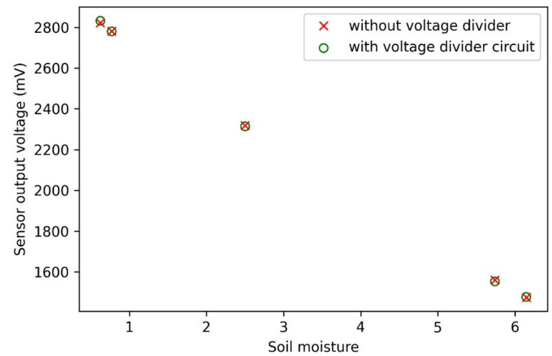
The measured rail changed from 3.300 V to 3.299 V when the identification network was present. This 1 mV observation supports the narrow claim that the prototype continued to operate without a visible supply disturbance. It does not quantify divider current, regulator efficiency, sensor self-heating, or battery-life impact. The two measurement arrangements above were proposed to separate sensor current from identification-network current, but the original project did not complete that experiment.

A follow-up test should log supply voltage and current at a controlled sampling rate for each sensor, with and without the divider. It should report mean, peak, and sleep-state current, energy per transmitted sample, and estimated battery life under a stated battery capacity and duty cycle.

Appendix F. Sensor Output Comparison



(a) TMP36 output



(b) SEN0193 output

Figure F.1. Reported output-voltage samples with and without the identification network.

The plots contain a small set of reported observations rather than a full calibration dataset. For both sensors, the paired values are visually close. This supports the prototype-level observation that adding the divider did not produce an obvious change in the measurement output under the tested conditions.

Question	Current evidence	Needed for a stronger claim
Does the sensor still operate?	Successful readings and visually similar paired outputs.	Repeat across temperature, moisture, supply, and sensor units.
Does the divider bias the measurement?	No obvious shift in the reported samples.	Paired error analysis with raw readings and uncertainty.
What is the power penalty?	Not measured.	Current and energy profiling with stated instruments.
Are IDs robust?	10/10 identification trials passed.	Tolerance and Monte Carlo analysis plus environmental testing.

Appendix G. Transmitter Firmware Workflow

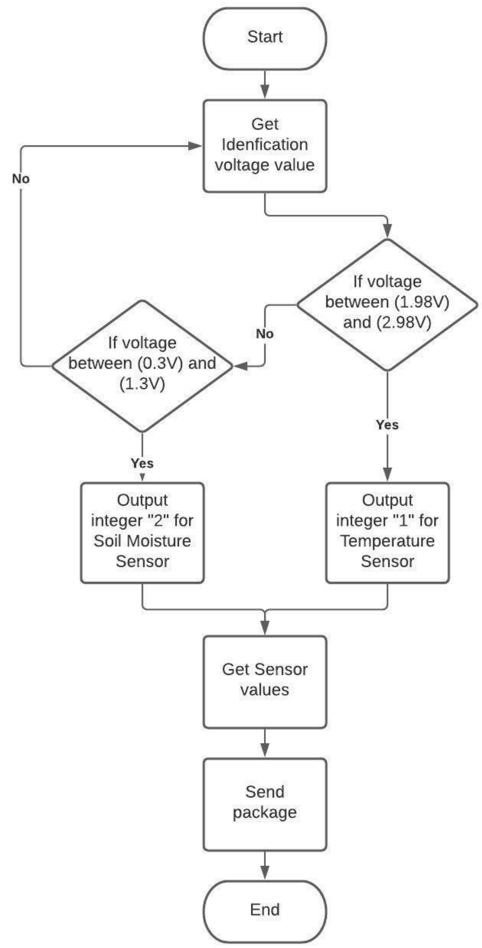


Figure G.1. Transmitter firmware flowchart from startup through sensor identification, conversion, packet assembly, and LoRa transmission.

At power-up, the firmware initializes the serial interface, analog inputs, radio, status storage, and watchdog. It reads the identification channel and selects the TMP36, SEN0193, or unidentified path. During the main loop it acquires the sensor voltage, converts the value, updates the packet counter and timestamp, attaches the diagnostic word, transmits the packet, services the watchdog, and waits for the next four-second interval.

G.1 Clarified pseudocode

```
read V_ID
if 1.80 V ≤ V_ID ≤ 2.80 V:
    sensor_type = TMP36
elif 0.49 V ≤ V_ID ≤ 1.49 V:
    sensor_type = SEN0193
else:
    sensor_type = UNIDENTIFIED

while active:
    raw = read_sensor_adc()
    value = convert(raw, sensor_type)
    status = collect_diagnostics()
    packet = encode(system_id, time, node_id, sensor_type, packet_id, value, status)
    lora_send(packet)
    service_watchdog()
    wait(4 seconds)
```

Appendix H. Gateway and Cloud Workflow

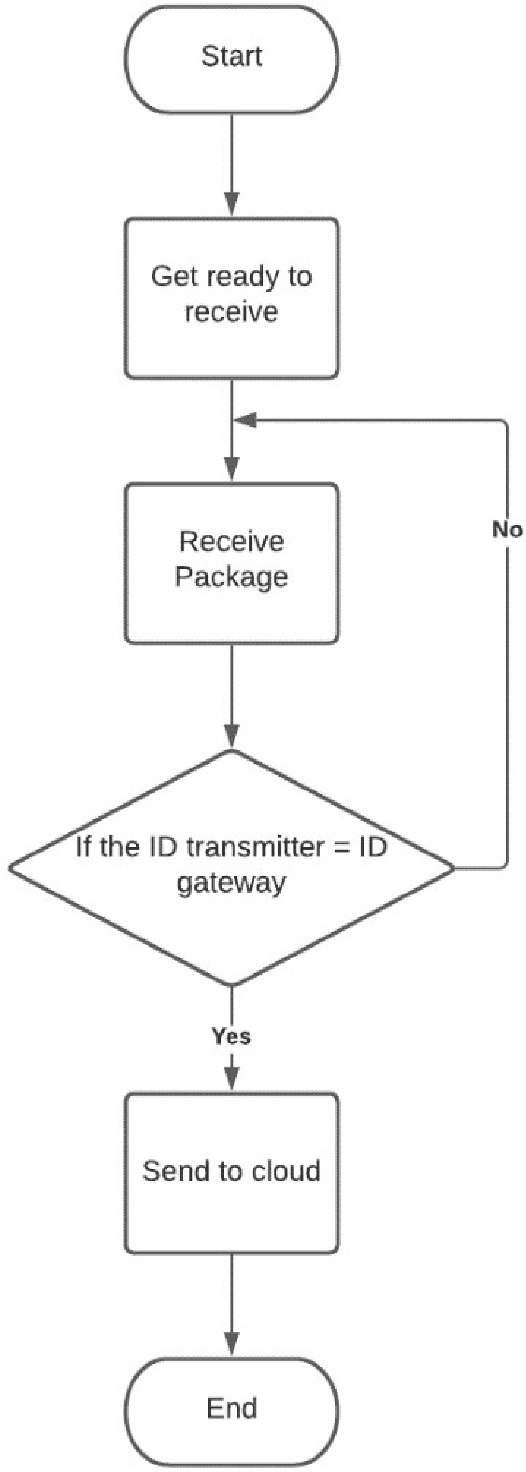
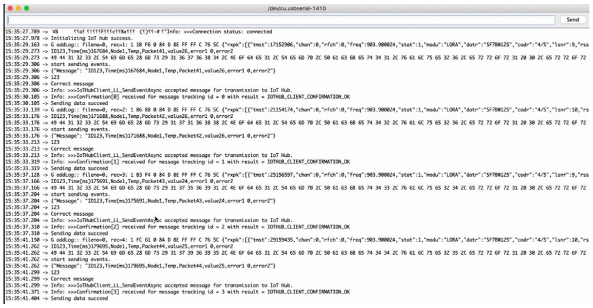
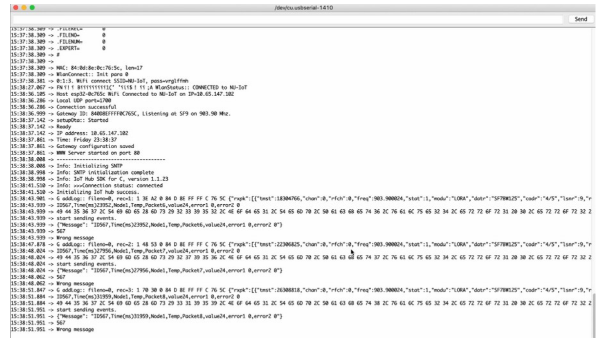


Figure H.1. Gateway decision flow: receive a packet, validate its system identifier, and forward accepted telemetry.

The gateway initializes the LoRa receiver and cloud connection, then waits for an incoming packet. It extracts the configured identification field before performing cloud transmission. Correctly identified packets are printed for local observation and sent to the Azure service; other packets are reported as incorrect and are not forwarded.



(a) Accepted packet



(b) Rejected packet

Figure H.2. Gateway serial-monitor examples for matching and nonmatching packet identifiers.

This filtering prevents unrelated laboratory packets from entering the selected data stream, but it is not a security control. A sender can copy a plaintext identifier. A deployment should authenticate devices cryptographically, protect message integrity, reject replays, and encrypt sensitive telemetry.

Appendix I. Cloud Record and Data Interpretation

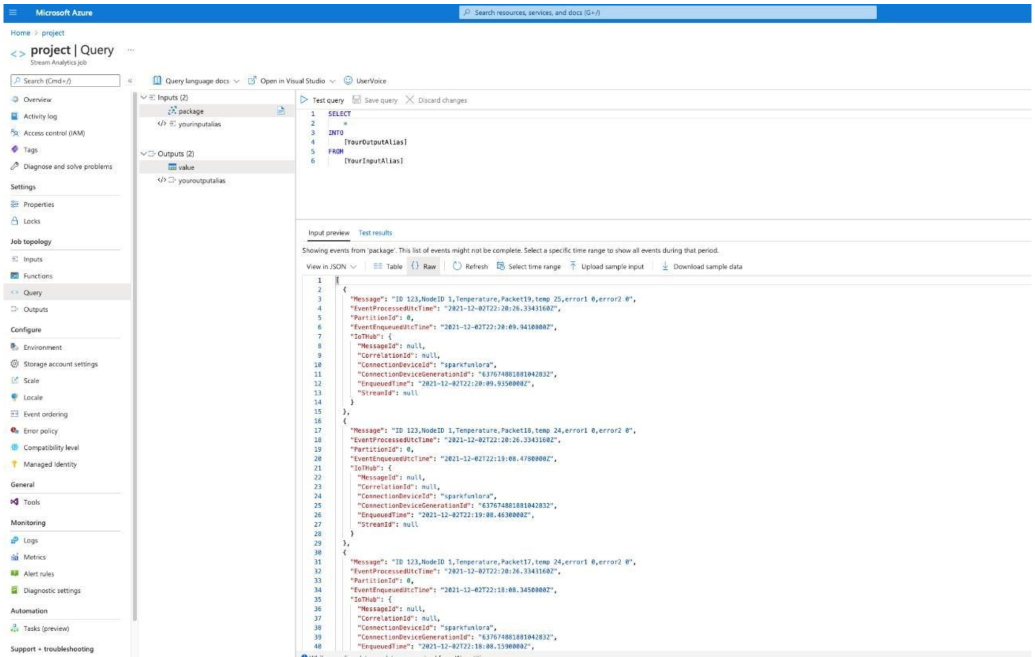


Figure I.1. Example structured telemetry displayed in the Azure environment. Authentication material has been excluded from the public report.

The cloud record demonstrates that the gateway completed the conversion from the LoRa payload to a structured Internet message. The retained fields allow a viewer to associate a value with its node, sensor type, packet sequence, local timestamp, and diagnostic status. This structure is sufficient for a prototype dashboard and for identifying missing packet numbers.

I.1 Data-quality requirements for future deployments

- Use an absolute real-time timestamp synchronized at the node or gateway rather than only elapsed milliseconds.
- Preserve the unconverted ADC value together with the engineering-unit result to support recalibration.
- Include firmware version, packet-schema version, units, calibration identifier, and measurement-quality flags.
- Buffer unsent data locally during cloud outages and retain the original acquisition time when retransmitting.
- Validate ranges at the edge and again in the cloud without silently discarding anomalous measurements.

Connection strings, shared keys, and provisioning secrets must never appear in screenshots, papers, repositories, or source-code examples. The credential embedded in the source course report is not reproduced here.

Appendix J. Verification Plan and Traceability

The project used repeated functional trials to verify individual requirements. The table below connects each test to the behavior it was intended to exercise and identifies the evidence that a stronger follow-up study should preserve.

Test	Trials / passes	Verified behavior	Recommended retained evidence
Power-on	20 / 20	Initialization and start of normal operation.	Boot log, reset cause, voltage trace, startup duration.
Error logging	4 / 4	Diagnostic bits and stored error state.	Injected fault, expected bit, observed bit, recovery action.
Communication	30 / 30	Transmitter-to-gateway packet reception.	Distance, RSSI, SNR, latency, retries, packet IDs.
Universal sensor	20 / 20	Common interface supports both sensors.	Sensor serial number, raw ADC, converted value.
Identification	10 / 10	Voltage window selects the expected sensor type.	Measured V_ID, ADC count, expected and detected type.
Indoor/outdoor	10 / 10	Operation in two broad environments.	Location, weather, obstructions, separation distance.
Ease of use	20 / 20	Connection and operation workflow completed.	Participant, task, completion time, observed errors.
Durability/reliability	20 / 20	Repeated operation under the project procedure.	Duration, cycles, failure definition, enclosure condition.
Plug-and-play	20 / 20	Sensor replacement without reflashing firmware.	Swap sequence, detection time, first valid reading.

All 154 listed trials passed. Because the original paper reports aggregate pass counts rather than trial-level records, the results are best described as successful functional verification. Statistical reliability, environmental qualification, and comparative performance remain open.

Appendix K. Demonstration and User Workflow

Figure K.1. Laboratory demonstration arrangement with transmitter, gateway, antennas, and host computer.

A demonstration begins by connecting a supported sensor assembly to the transmitter and applying power. The identification voltage is read during initialization. The serial output can be used to confirm the selected sensor type and observe outgoing packet fields. The gateway is then powered and connected to the configured network. Accepted packets appear in the gateway serial output and subsequently in the cloud data view.

Step	Operator action	Expected observation
1	Inspect wiring and antenna connections before power-up.	No loose conductors or shorted supply pins.
2	Attach either the TMP36 or SEN0193 assembly.	Sensor measurement and identification lines are present.
3	Power the transmitter.	Startup completes and a supported sensor type is selected.
4	Power and connect the gateway.	LoRa receiver and cloud client initialize.
5	Observe at least two sampling intervals.	Packet IDs increase and readings arrive approximately every four seconds.
6	Swap to the other supported sensor after removing power.	On restart, the new identification voltage selects the other conversion path.
7	Review cloud telemetry.	Node, sensor, value, timestamp, packet ID, and status fields are visible.

Appendix L. Packaging and Deployment Constraints

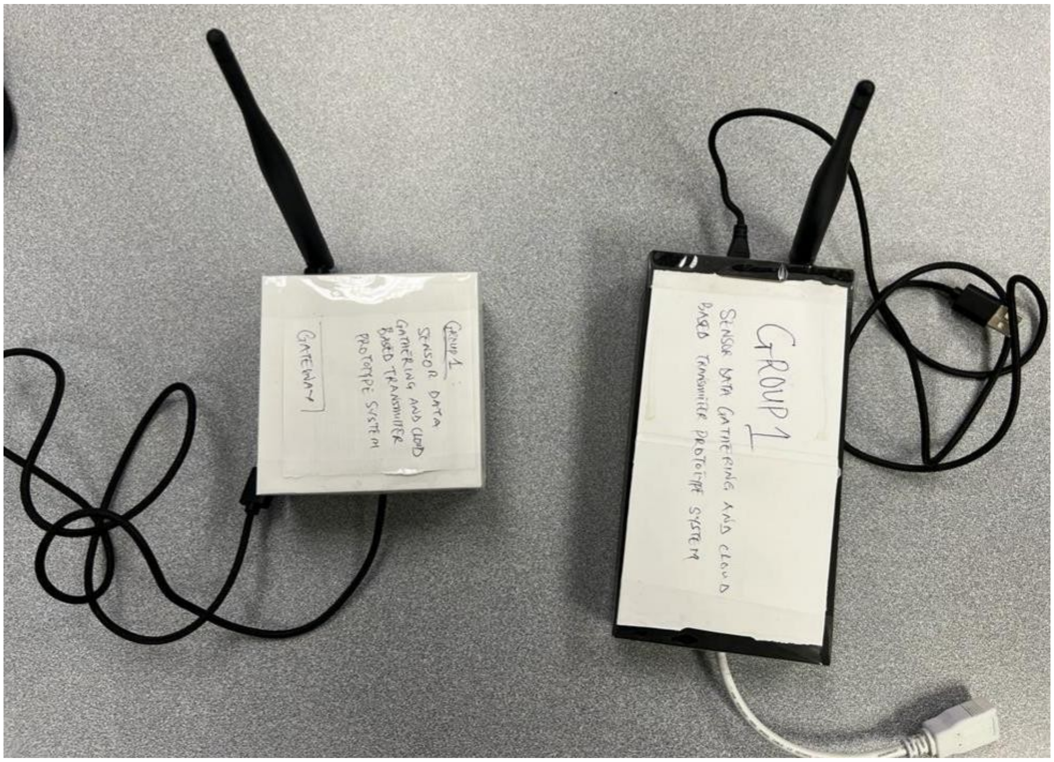


Figure L.1. Final packaged transmitter and gateway prototypes.

The final packaging improved portability and protected the boards during demonstration. External antenna and sensor leads remained accessible. The enclosure was not tested or certified for dust or water ingress, impact, ultraviolet exposure, condensation, temperature cycling, or cable strain. It should therefore be treated as indoor or supervised prototype packaging.

L.1 Deployment checklist for a revised enclosure

- Choose an enclosure and cable glands with a documented ingress-protection rating appropriate to the site.
- Provide strain relief, keyed sensor connectors, reverse-polarity protection, and an accessible power disconnect.
- Maintain antenna clearance and verify that the enclosure material does not materially degrade radio performance.
- Separate wet-soil probe wiring from radio and digital lines, and add transient and electrostatic-discharge protection.
- Label the node ID, supported supply range, sensor connector pinout, radio region, and service procedure.
- Validate thermal rise, condensation control, corrosion resistance, and battery safety before unattended operation.

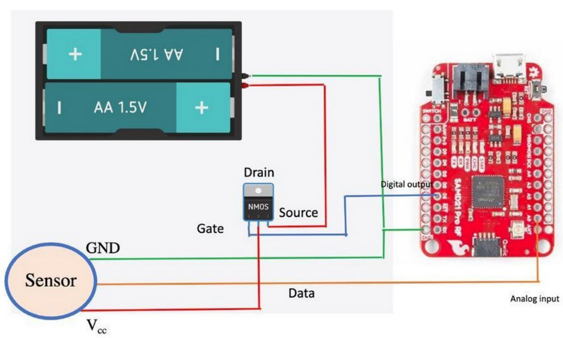
Appendix M. Proposed Extensions

(a) Local removable storage

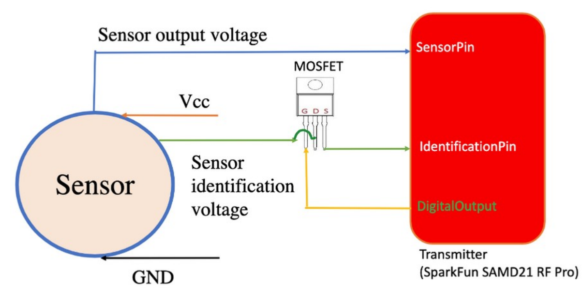
(b) Real-time clock module

Figure M.1. Proposed buffering and timekeeping additions.

Local storage would preserve readings when the gateway or cloud connection is unavailable. A real-time clock would allow the node to retain acquisition time across retransmission and restart. Together, they support a store-and-forward design with a recoverable audit trail.



(a) Switched sensor power



(b) Switched identification divider

Figure M.2. Proposed MOSFET switching to reduce energy use after measurement or identification.

The proposed switching circuits would disconnect loads that are unnecessary between samples. Their benefit must be confirmed through current profiling because switching devices, pull resistors, regulator quiescent current, radio standby modes, and wake-up time all contribute to total energy. Firmware should guarantee a stabilization delay after enabling each sensor before accepting a reading.

Appendix N. Reproducibility and Next-Study Checklist

A publishable follow-up should preserve enough information for another team to reproduce the system and evaluate the central hypothesis that passive voltage identification enables reliable plug-and-play sensing.

Area	Minimum information to report
Hardware	Schematics, bill of materials, board revisions, connector pinout, resistor tolerances, antenna model, enclosure.
Firmware	Versioned source, compiler and libraries, radio parameters, sampling schedule, packet schema, watchdog configuration.
Calibration	Reference instruments, calibration points, conversion equations, units, uncertainty, soil preparation procedure.
Identification	ADC reference, decision thresholds, supply range, temperature range, repeated trials across units.
Wireless	Distance, environment, obstructions, transmit power, spreading factor, bandwidth, coding rate, RSSI, SNR, loss, latency.
Energy	Active/sleep current, sensor warm-up, transmit energy, duty cycle, battery capacity and measured lifetime.
Cloud	Gateway software, time synchronization, buffering, schema validation, data retention, credential handling.
Security	Threat model, mutual authentication, encryption, integrity, replay protection, provisioning and key rotation.
Dataset	Trial-level raw data, timestamps, packet IDs, analysis scripts, missing-data handling, license and repository DOI.

N.1 Recommended experimental sequence

- Bench-calibrate each sensor and characterize the ADC and identification thresholds.
- Measure energy for identification, sampling, packet creation, transmission, reception, and cloud forwarding.
- Run controlled packet-delivery experiments across distance, obstruction, and interference conditions.
- Operate multiple nodes concurrently to test identifier uniqueness, gateway queueing, and cloud throughput.
- Conduct a multi-day outdoor pilot with weather-resistant packaging and independent reference measurements.
- Publish the anonymized raw data, firmware version, schematics, and analysis scripts with the resulting paper.

Appendix O. Original Requirements and Verification Status

The original project defined ambitious customer and engineering targets. The table below preserves those targets while separating verified prototype behavior from goals that were not demonstrated quantitatively. This distinction is important because a successful bench demonstration does not by itself establish field reliability, lifetime, environmental protection, or cost compliance.

Original target	Source requirement	Status in the reported prototype
Plug-and-play temperature and soil-moisture sensing with cloud upload.	CR-1, CR-4, CR-5; ER-2	Demonstrated for the TMP36 and SEN0193 through identification, conversion, LoRa forwarding, and Azure display.
Notify or diagnose missing messages.	CR-1; ER-8	A missing-packet diagnostic bit and packet counter were defined; end-user notification behavior was not evaluated.
Replaceable, rechargeable battery with long life.	CR-2; ER-4	Battery-capable operation was demonstrated. The five-year lifetime stated in the source was not measured or substantiated.
Dust- and splash-resistant enclosure with an IP67 target.	CR-3; ER-9	The prototype was packaged, but the enclosure had no IP rating and was not environmentally qualified.
At least 95% of measurements reach the cloud.	CR-3	The listed communication trials passed, but no long-duration delivery-ratio dataset was reported.
Indoor and outdoor operation.	CR-6; ER-3	Ten listed indoor/outdoor trials passed; distance, weather, and obstruction conditions were not recorded.
Finished-goods cost of $100 or less.	CR-7; ER-1; SC-3	A bill-of-materials target was stated, but a complete verified cost table was not included.
Communication distance greater than 50 m.	SC-4	The radio platform was selected for long range, but the original report did not provide a documented range dataset.
Read and upload in less than five minutes.	ER-6	The transmitter sampled every four seconds; end-to-end latency measurements were not reported.
Accept sensor voltages from 0 to 5 V.	Constraint C2	The implemented controller operated at 3.3 V. Direct 5 V analog compatibility was not demonstrated and would require electrical protection or scaling.
Microsoft Azure as the prototype cloud destination.	Constraint C3	Demonstrated through a structured telemetry record in Azure.

The source also described weekly field sampling while the prototype sampled every four seconds. These are different operating modes. A revised specification should define a configurable sampling interval and separately state the desired cloud-upload interval.

Appendix P. Documented Bugs, Challenges, and Resolutions

The development record identified implementation problems that help explain the final design and establish priorities for a revised system. They are retained here as engineering evidence rather than presented as completed capabilities.

Issue	Disposition in the original project	Remaining implication
Floating-point sensor values could not be transmitted.	The payload was changed to integer values.	Use a documented fixed-point scale or a versioned binary/JSON schema to preserve resolution.
Packets from unrelated transmitters were received.	A system identification code was added at the gateway.	Filtering was improved, but cryptographic authentication and replay protection are still required.
The gateway-to-cloud message had a size limit.	Packet length was reduced and reserved positions were retained.	Define exact field widths, bounds, schema versioning, and fragmentation behavior.
Cloud messages were sometimes queued.	The team adjusted the interval between transmissions and observed improvement.	Measure queue delay and end-to-end latency; buffer readings locally during outages.
The enclosure form factor was difficult to finalize.	A modular, reopenable package was produced.	The enclosure remains unrated for water, dust, impact, and long-term outdoor exposure.
Only two sensors were supported.	Voltage windows were defined for one temperature and one soil-moisture sensor.	Scaling requires tolerance analysis, calibration records, and a more extensible identification method.

P.1 Operations and troubleshooting retained from the source

- Before startup, confirm sensor, antenna, battery, and gateway connections; then verify network connectivity and cloud-device configuration.
- If the node is unresponsive, remove power before reseating the sensor or antenna and confirm that the battery remains within specification.
- If telemetry does not appear, check gateway reception first, followed by Wi-Fi connectivity, cloud configuration, packet identifier, and the most recent diagnostic word.
- Power down and disconnect batteries and sensors before storage; keep the prototype dry and clean because the enclosure is not environmentally rated.

Bibliography

[1] SparkFun Electronics, “SparkFun SAMD21 Pro RF Hookup Guide,” SparkFun Learn. https://learn.sparkfun.com/tutorials/sparkfun-samd21-pro-rf-hookup-guide.

[2] Analog Devices, “TMP35/TMP36/TMP37: Low Voltage Temperature Sensors,” Rev. H data sheet. https://www.analog.com/media/en/technical-documentation/data-sheets/tmp35_36_37.pdf.

[3] DFRobot, “Gravity: Corrosion-Resistant Capacitive Soil Moisture Sensor, SKU SEN0193,” DFRobot Wiki. https://wiki.dfrobot.com/sen0193.

[4] Semtech, “What Is LoRa?” https://www.semtech.com/lora/what-is-lora.

[5] Microsoft, “What is Azure IoT Central?” Microsoft Learn. https://learn.microsoft.com/en-us/azure/iot-central/core/overview-iot-central.

[6] Microsoft, “How to use data explorer to analyze device data,” Microsoft Learn. https://learn.microsoft.com/en-us/azure/iot-central/core/howto-create-analytics.

Source note: Technical details, figures, and test counts in this report were reorganized from the authors’ CSCE 838 final project report dated December 17, 2021. This public version excludes the original cloud connection string and course-administration content.