nRF54L15 BLE & LoRa
Sensor-Node Secure Hub

Encrypted Wireless Sensor Network

Michael Neumüller | April 2026

Embedded Systems | IoT | Wireless Sensor Networks

A secure point-to-point wireless sensor network combining a SeeedStudio XIAO nRF54L15 sensor node with an ESP32-S3 LoRa gateway.

Real-time FFT Audio calculation with onboard PDM Microphone. Visualization of FFT/Audio, Waterfall Diagramm and Browser BLE Connectivity.

Real-time IMU data is encrypted with AES-128-CTR and transmitted via 868 MHz LoRa. Two separate web dashboards provide live visualization: WebBluetooth for BLE mode and an HTTP Cyber-Hub via mDNS (lora.local) for LoRa P2P mode.

BLE Mode

`prj.conf`

- WebBluetooth Dashboard
- Audio FFT + Spectrogram
- IMU Real-time Charts
- UART Debug Console

LoRa P2P Mode

`prj_p2p.conf`

- AES-128-CTR Encryption
- 868.1 MHz / SF9 / BW125
- ESP32-S3 Gateway
- http://lora.local (mDNS)

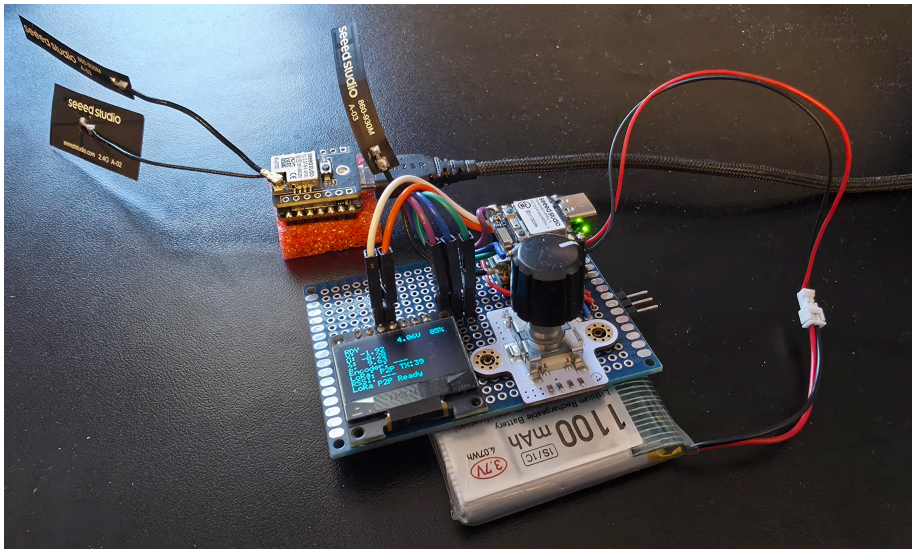
Eco / Battery Mode

`prj_battery.conf`

- UART Disabled (Power)
- BLE Beacon Active
- Industry Sleep/Wake
- 34 KB Flash Savings

LoRa P2P - Key Specifications

Frequency	868.1 MHz (EU868)
Encryption	AES-128-CTR (Mbed TLS / mbedtls)
Payload	[8B Nonce | 12B Ciphertext] = 20 Bytes
Radio	SF9 / BW125 / CR4/5 / +14 dBm
BLE Dashboard	WebBluetooth: Audio FFT, Spectrogram, IMU Charts
P2P Dashboard	http://lora.local (ESP32-S3 mDNS): IMU, Signal, CSV



Technical Highlights

CMSIS-DSP FFT Processing

512-point real-valued FFT at 16 kHz sample rate. 32 log-spaced frequency bands (50 Hz - 7.8 kHz). Hanning window, exponential smoothing, ring buffer with overrun recovery. DC offset removal with fast convergence (alpha=0.5).

Battery Monitoring & Power Gating

12-bit SAADC with 8x oversampling on AIN7 (P1.14). 10k/10k voltage divider, 900mV internal reference, 1/4 gain. TPS22916CYFPR load switch for sensor power gating. Li-Po discharge curve mapping (2.8V-4.2V -> 0-100%).

Dual Operation Modes

Continuous: Real-time sensor streaming at configurable intervals. Industry: Periodic wake, collect IMU samples, run FFT, transmit aggregated data, return to low-power sleep. Reduces average current consumption.

Multi-Protocol Radio Architecture

Same hardware supports BLE (WebBluetooth NUS), LoRa P2P (AES encrypted), and LoRaWAN (OTAA, EU868, TTN). Build-time selection via Kconfig. BLE and SPI00 cannot coexist due to nRF54L15 DMA resource conflict.

SSD1306 OLED via Software I2C

128x64 monochrome display via bit-bang GPIO (P2.09/P2.10). Port 2 GPIO does not support hardware TWIM on nRF54L15, requiring software I2C implementation. Multi-page status view with all telemetry.

Robust Error Handling

PDM overrun recovery (auto-restart after 3 errors), AES decryption validation (NaN + range check), LoRa CRC error counting, watchdog timer (30s timeout). Graceful degradation: OLED and encoder are optional peripherals.

Hardware Setup

Sensor Node: SeeedStudio XIAO nRF54L15 Sense

MCU	Nordic nRF54L15 (Cortex-M33, 128 MHz, 256 KB RRAM, 192 KB RAM)
RTOS	Zephyr RTOS v4.2.99 (nRF Connect SDK v3.2.3)
IMU	LSM6DS3TR-C 6-axis (I2C @ 0x6A, P0.03/P0.04)
Microphone	MSM261DGT006 PDM (16 kHz, 512-pt FFT, CMSIS-DSP)
LoRa	Semtech SX1262 via SPI00 (868.1 MHz, SF9, +14 dBm)
Display	SSD1306 OLED 128x64 (Software I2C on P2.09/P2.10)
Input	EC11 Rotary Encoder (Quadrature on P2.06-P2.08) + Button (D0)
Battery	Li-Po 3.7V / 1100 mAh via SAADC AIN7 (P1.14), 10k/10k divider

Gateway: ESP32-S3 + SX1262 LoRa Module

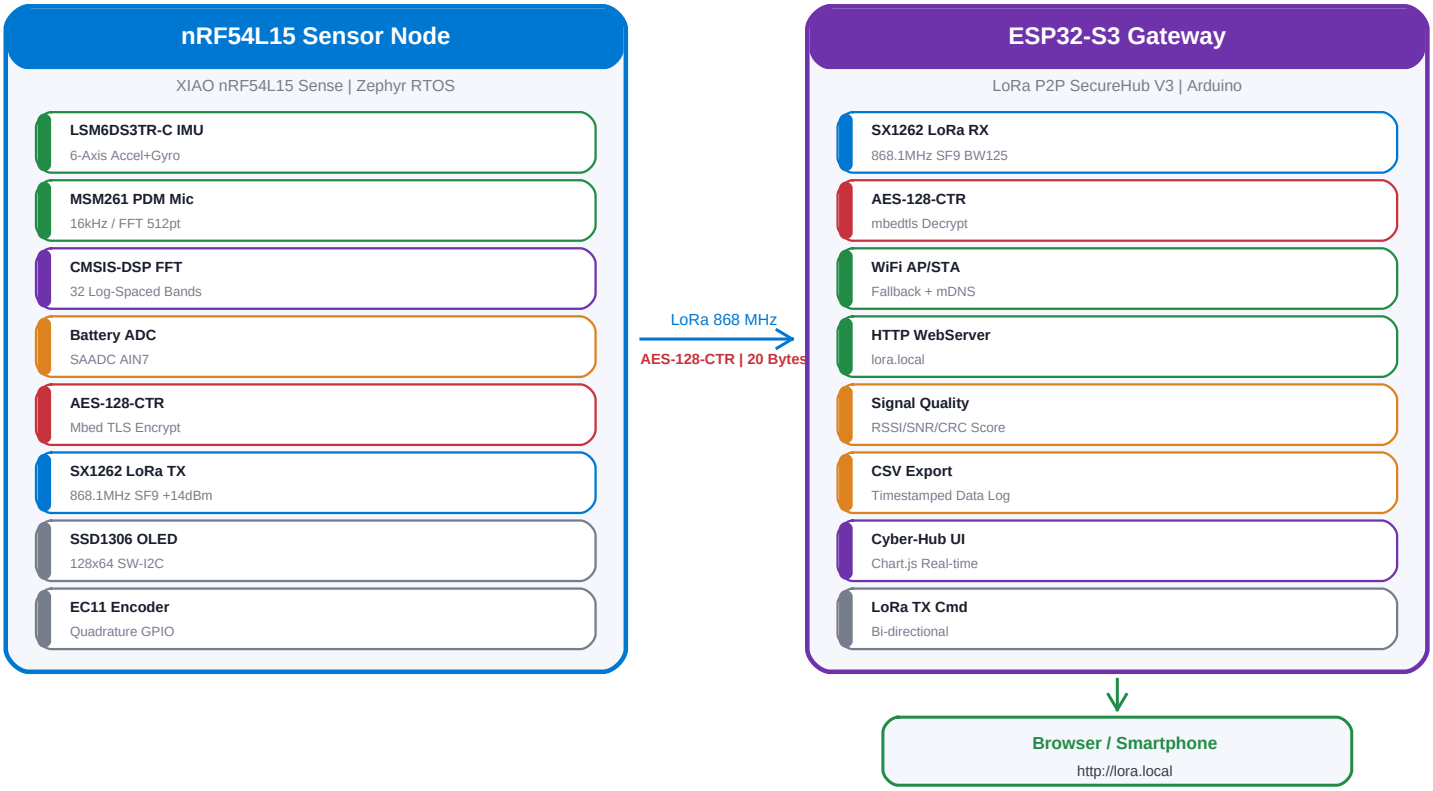
MCU	Espressif ESP32-S3 (Xtensa LX7, Dual-Core 240 MHz, WiFi+BLE)
Framework	Arduino (ESP-IDF based), RadioLib for SX1262 driver
LoRa	Semtech SX1262 via SPI (868.1 MHz, SF9, BW125, TCXO 1.7V)
WiFi	802.11 b/g/n, STA + AP fallback, mDNS: http://lora.local
Encryption	mbedtls AES-128-CTR (ESP-IDF built-in, identical to nRF)
Web UI	Cyber-Hub v3: IMU Charts, Signal Quality, CSV Export, TX Cmds

Pin Mapping (nRF54L15)

Function	Pin	Peripheral	Notes
IMU SDA/SCL	`P0.04/P0.03`	I2C30	LSM6DS3TR-C
OLED SDA/SCL	`P2.09/P2.10`	SW I2C	Bit-bang GPIO
SPI SCK/MOSI/MISO	`P2.01/P2.02/P2.04`	SPI00	LoRa module
LoRa NSS	`P1.10 (D4)`	GPIO	Chip Select
LoRa RST/BUSY/DIO1	`P1.06/P1.07/P1.05`	GPIO	SX1262 ctrl
Battery ADC	`P1.14`	SAADC	AIN7, 10k/10k
Encoder A/B/SW	`P2.08/P2.07/P2.06`	GPIO	EC11 Quadrature
UART TX/RX	`P2.08/P2.07`	UART21	Debug (BLE mode)

System Architecture

The system consists of two nodes communicating via encrypted LoRa P2P. The nRF54L15 sensor node collects IMU data, encrypts it with AES-128-CTR, and transmits 20-byte packets to the ESP32-S3 gateway, which decrypts and serves a real-time web dashboard accessible via WiFi/mDNS.



Data Flow

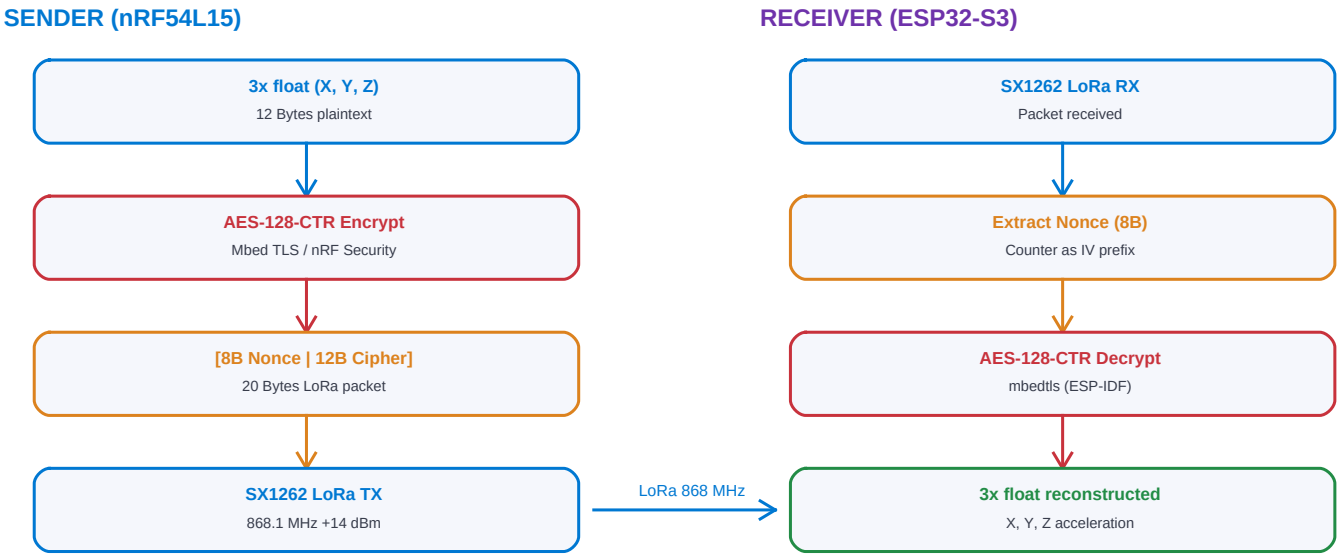
```
IMU Sensor  -->  3x float32 (X, Y, Z)  -->  12 Bytes plaintext
AES-128-CTR  +  8B Counter/Nonce  -->  [8B Nonce | 12B Ciphertext] = 20B
SX1262 TX  -->  868.1 MHz, SF9, BW125, CR4/5, +14 dBm  -->  Air
SX1262 RX  -->  RSSI/SNR measured  -->  AES-128-CTR Decrypt
3x float32 reconstructed  -->  Web Dashboard  -->  Browser (Chart.js)
```

LoRa Radio Parameters

Frequency	**868.1 MHz (EU868)**
Spreading Factor	**SF9**
Bandwidth	**125 kHz**
Coding Rate	**CR 4/5**
TX Power	**+14 dBm**
SyncWord	**0x34**
Payload	**20 bytes (encrypted)**

AES-128-CTR Encryption Pipeline

All sensor data is encrypted end-to-end using AES-128-CTR mode with a pre-shared 128-bit key. A monotonic 64-bit packet counter serves as nonce, ensuring each packet has a unique keystream. The same Mbed TLS library runs on both platforms.



Security Details

Pre-Shared Key (PSK)

Identical 128-bit AES key compiled into both firmware images. Key: 0x2B7E1516...CF4F3C

Counter-Mode (CTR)

Each packet uses a unique 64-bit counter as nonce, preventing replay attacks. Counter increments monotonically per transmission.

Cross-Platform Compatibility

nRF54L15: nRF Security subsystem (Mbed TLS). ESP32-S3: ESP-IDF built-in mbedtls. Byte-level compatible AES-CTR implementation.

Packet Format

[8 Bytes Counter/Nonce][12 Bytes AES-CTR Ciphertext] = 20 Bytes total. Minimal overhead for LoRa air time optimization.

Validation

Decrypted floats are range-checked (-100..+100) and NaN-checked to detect corruption. CRC errors and decrypt failures are counted separately.

Software Architecture (nRF54L15)

Zephyr RTOS Thread Model



Source Code Modules

`src/main.c`	Application entry, thread creation, subsystem init
`src/hardware/`	pdm_handler, imu_manager, power_control
`src/lora_p2p/`	lora_p2p_handler, lora_crypto (AES-128-CTR)
`src/bluetooth/`	ble_service (NUS), ble_bas, ble_dis
`src/lorawan/`	lorawan_manager, region/power overrides
`src/processing/`	dsp_processor (CMSIS-DSP FFT)
`src/system/`	battery_monitor, power_manager, watchdog
`src/display/`	oled_simple (SSD1306 driver)
`src/config/`	app_config.h (compile-time constants)

Build Configurations

	BLE Mode	LoRa P2P	Eco / Battery
Config File	prj.conf	prj_p2p.conf	prj_battery.conf
BLE	Yes	No	Yes
LoRa	No	Yes (P2P)	No
Encryption	No	AES-128-CTR	No
UART/Log	Yes	No	No
Web UI	WebBluetooth	lora.local	N/A
Use Case	Development	Secure P2P	Field Deploy

Web Dashboards

Two separate web interfaces for different modes: WebBluetooth for BLE mode (Audio FFT, Spectrogram, IMU) and an HTTP Cyber-Hub on the ESP32-S3 gateway for LoRa P2P mode, accessible via mDNS at http://lora.local.

1. LoRa P2P: ESP32-S3 Cyber-Hub (lora.local)

Access & Telemetry

WiFi STA + AP fallback (SSID: LoRa-Hub-S3). mDNS: lora.local. CPU temp, SRAM, SNR, CRC, freq error, uptime. Signal quality composite score.

IMU + CSV Export

Chart.js IMU visualization (X/Y/Z, 60 samples). Lock icon for encrypted packets. CSV export with full telemetry (BOM UTF-8).

2. BLE: WebBluetooth Dashboard

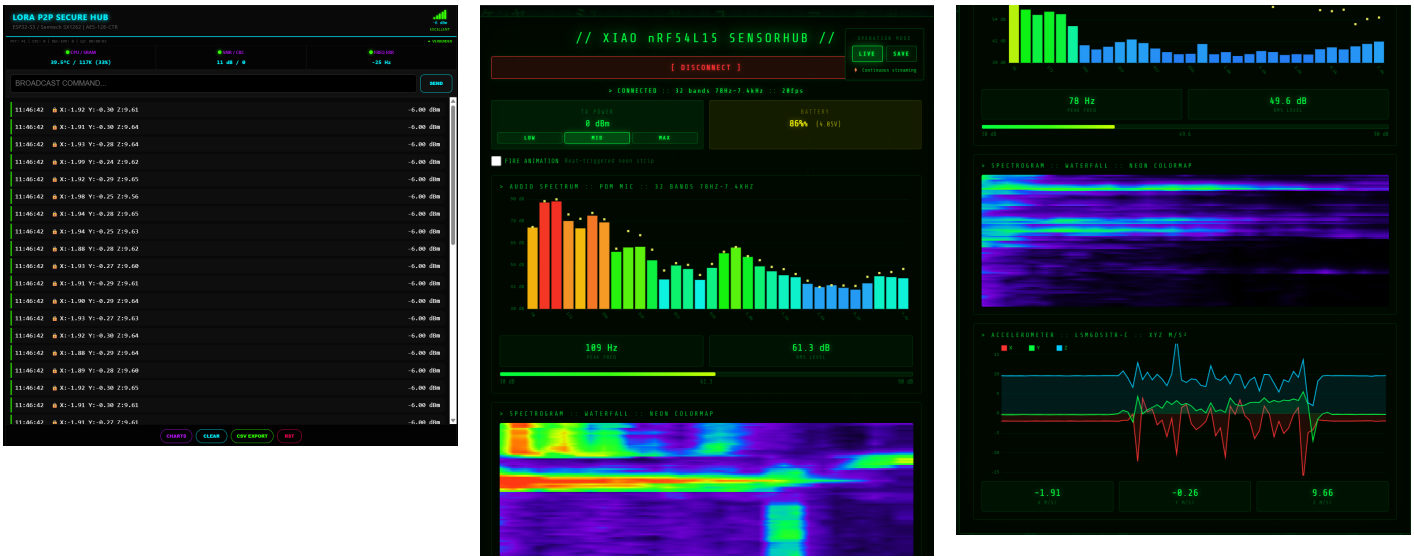
Audio Spectrum + Spectrogram

32 log-spaced bands (50 Hz-7.8 kHz), dBFS axis, peak hold. Neon waterfall with pixel-level ImageData rendering.

IMU + Performance

3-axis accel chart (Float32Array circular buffer, 80 samples). Custom Canvas 2D renderers, ~3.4ms/frame, no Chart.js, 60fps rAF loop.

Screenshots: LoRa P2P Secure Hub (left) | BLE Audio+FFT (center) | BLE Spectrogram+IMU (right)



ESP32-S3 LoRa SecureHub V3

The ESP32-S3 Arduino sketch (LoRa_P2P_SecureHub_V3.ino, 573 lines) implements the complete LoRa P2P gateway with AES decryption, web server, and real-time data visualization. Key implementation details:

AES-128-CTR Decryption

Uses ESP-IDF mbedtls (not Arduino Crypto Library). decryptPayload() extracts 8B nonce, runs AES-CTR, validates float range (-100..+100). Decrypt errors tracked separately from CRC errors.

RadioLib SX1262 Driver

SPI configuration: NSS=41, DIO1=39, RST=42, BUSY=40, RF_SW=38. TCXO 1.7V, DIO2 as RF switch. ISR-based receive with volatile flag. State machine prevents TX during active RX.

WiFi & mDNS

STA mode with 10s timeout, fallback to AP mode ('LoRa-Hub-S3'). MDNS.begin('lora') -> accessible as http://lora.local. WebServer on port 80 with JSON API endpoints.

Embedded Web UI

Full HTML/CSS/JS embedded as raw string literal (R-string). Cyber-Hub dark theme with CSS Grid layout, Chart.js for IMU visualization, responsive design (mobile-optimized via media queries).

Core Decryption Function

```
int decryptPayload(const uint8_t* input, size_t inputLen,
                   uint8_t* output, size_t* outLen) {
    if (inputLen <= 8) return -1;
    uint8_t nonce[16];
    memset(nonce, 0, 16);
    memcpy(nonce, input, 8);       // 8B counter as IV
    size_t nc_off = 0;
    uint8_t stream_block[16];
    mbedtls_aes_setkey_enc(&aesCtx, SHARED_KEY, 128);
    mbedtls_aes_crypt_ctr(&aesCtx, inputLen - 8,
        &nc_off, nonce, stream_block, input + 8, output);
    *outLen = inputLen - 8;
    return 0;
}
```

HTTP API Endpoints

Endpoint	Description
`GET /`	Serve embedded HTML page (Cyber-Hub UI)
`GET /update`	Return latest LoRa messages (payload|RSSI|msgID)
`GET /sysinfo`	JSON: temp, RAM, SNR, RSSI, CRC, freq error, uptime
`GET /csvexport`	Download full CSV log (BOM-prefixed UTF-8)
`GET /sendMsg?msg=`	Transmit text via LoRa TX to sensor node
`GET /reboot`	Restart ESP32-S3
`GET /clear`	Clear message log

nRF54L15 BLE & LoRa

Sensor-Node Secure Hub

Encrypted Wireless Sensor Network

Technology Stack

Zephyr RTOS v4.2.99 | nRF Connect SDK v3.2.3 | nRF54L15 (Cortex-M33)
ESP32-S3 (Arduino) | Semtech SX1262 (RadioLib) | LoRa EU868
AES-128-CTR (Mbed TLS / mbedtls) | CMSIS-DSP FFT | LSM6DS3TR-C IMU
WebBluetooth API | HTTP WebServer + mDNS | Chart.js | SSD1306 OLED

Michael Neumüller | April 2026

Embedded Systems | IoT | Wireless Sensor Networks

linkedin.com/in/michael-neumueller1

https://www.linkedin.com/in/michael-neumueller1/