# FLIGHT LOGGER (v2)

*Hardware Guide, Wiring Reference & Code Explanation*

Arduino Nano 33 BLE Sense + GT-U7 GPS + RFM95 LoRa | Barometric Altitude | SF9 | 2 Hz

## 1. Hardware Overview

**The flight logger is a compact airborne telemetry unit. Version 2 changes the altitude source from GPS to the built-in LPS22HB barometric pressure sensor, which provides higher resolution, faster update rates, and immunity to GPS altitude noise. LoRa spreading factor has been changed from SF12 to SF9 to support a 2 Hz (500 ms) transmission rate.**

| Component | Role | Interface |
|---|---|---|
| Arduino Nano 33 BLE Sense | Main processor (nRF52840, 3.3V) | USB, I2C, SPI, Serial1 |
| GT-U7 GPS Receiver | Position (lat/lon), speed, satellites — NOT altitude | UART 9600 baud (Serial1) |
| RFM95 LoRa Radio | Wireless telemetry to base station at SF9 / 2 Hz | SPI (D10–D13) |
| LSM9DS1 IMU (built-in) | Accelerometer, gyroscope, magnetometer for orientation | I2C (internal, library-managed) |
| LPS22HB Barometer (built-in) | Pressure sensor for altitude — PRIMARY altitude source | I2C (internal, library-managed) |

**NOTE:** ***The Nano 33 BLE Sense operates at 3.3 V logic. All peripherals must be 3.3 V compatible. The RFM95 operates at 3.3 V. The GT-U7 TX pin outputs 3.3 V UART by default.***

## 2. Wiring Diagrams

### 2.1 RFM95 LoRa Module to Nano 33 BLE Sense

**The RFM95 uses SPI. The Nano 33 BLE Sense hardware SPI pins are D11 (MOSI), D12 (MISO), D13 (SCK). All lines are 3.3 V — no level shifting is required between the Nano 33 and the RFM95.**

| Signal | RFM95 Pin | Nano 33 BLE Sense Pin | Notes |
|---|---|---|---|
| NSS (Chip Select) | NSS | D10 | SPI chip select, active low |
| MOSI | MOSI | D11 | SPI data: Nano → RFM95 |

| | | | |
|---|---|---|---|
| MISO | MISO | D12 | SPI data: RFM95 → Nano |
| SCK | SCK | D13 | SPI clock |
| RESET | RST | D9 | Module reset (active low) |
| DIO0 | DIO0 | D2 | TX/RX done interrupt |
| Power | VCC | 3.3V | Use external 3.3V LDO for reliability (see note) |
| Ground | GND | GND | Common ground |

**NOTE:** ***Add a 100 nF ceramic decoupling capacitor as close as possible between the RFM95 VCC and GND pins. The RFM95 draws up to 120 mA during transmit. Power it from a dedicated AMS1117-3.3 regulator rather than the Nano 3.3V pin, which is limited to ~50 mA.***

## 2.2 GT-U7 GPS Module to Nano 33 BLE Sense

**GPS uses Serial1 (hardware UART, D0 RX / D1 TX) for reliable 9600 baud NMEA parsing. GPS altitude is NOT used; only latitude, longitude, speed, and satellite count are extracted.**

| Signal | GT-U7 Pin | Nano 33 BLE Sense Pin | Notes |
|---|---|---|---|
| GPS UART TX | TX | D0 (Serial1 RX) | GPS sends NMEA sentences to Nano |
| GPS UART RX | RX | D1 (Serial1 TX) | Optional; used only for config commands |
| Power | VCC | 3.3V | GT-U7 has internal 3.3V LDO; verify your module |
| Ground | GND | GND | Common ground |

## 2.3 Built-in Sensors (No Additional Wiring)

**Both the LSM9DS1 IMU and the LPS22HB barometric sensor are soldered directly onto the Nano 33 BLE Sense PCB. They are accessed through the I2C bus internally. No wires, connectors, or additional components are needed for these sensors.**

| Sensor | IC | Function |
|---|---|---|
| IMU | LSM9DS1 | Accelerometer (±4g), gyroscope (±2000°/s), magnetometer (±4 gauss) |

| | | |
|---|---|---|
| Barometer | LPS22HB | Pressure 260–1260 hPa, resolution 1/4096 hPa, ODR 1–75 Hz |

**NOTE:** ***Mount the Nano 33 BLE Sense away from direct sunlight, heat sources, and airflow across the PCB. These factors change the air temperature near the LPS22HB and introduce altitude measurement errors through adiabatic pressure changes.***

# 3. LPS22HB Barometric Pressure Sensor & Altitude

## 3.1 Why Barometer Instead of GPS for Altitude

**GPS altitude has several serious limitations for flight logging:**

- Update rate: Standard GPS modules (including the GT-U7) output position data at only 1 Hz. This is far too slow to capture rapid altitude changes during aggressive flight manoeuvres.
- Noise: GPS altitude typically has 3–10 m of noise even with a clear sky view. This noise cannot be filtered away without introducing unacceptable lag.
- Vertical dilution: GPS is inherently less accurate in the vertical axis (VDOP is typically 1.5–2× the horizontal DOP). A fix with 3 m horizontal accuracy may have 6–10 m vertical error.

The LPS22HB overcomes all of these issues. It updates at 25 Hz (configurable to 75 Hz), has a pressure resolution of 1/4096 hPa (corresponding to ~0.1 m altitude resolution at sea level), and its readings are smooth and consistent.

## 3.2 Hypsometric (Barometric) Altitude Formula

**Atmospheric pressure decreases with altitude in a well-characterised way described by the international standard atmosphere. Given a reference pressure P_ref measured at the launch site, the altitude above that point is:**

```
h_AGL = 44330 * ( 1 - (P / P_ref) ^ 0.1903 )
```

**Where:**

- h_AGL is the altitude above the launch site in metres (AGL = Above Ground Level)
- P is the current barometric pressure in hPa (hectopascals / millibars)
- P_ref is the base pressure measured at the launch site in hPa
- 0.1903 = 1/5.255 — the adiabatic lapse-rate exponent for dry air
- 44330 is the nominal scale height in metres for the standard atmosphere

Because we use P_ref from the launch site itself (not a fixed value like 1013.25 hPa), the formula gives altitude relative to the launch point regardless of the actual elevation or local weather. This self-referencing approach eliminates the need for QNH correction or knowledge of the true sea-level pressure.

The constant 0.1903 is derived from the gas law for a dry adiabatic lapse rate. For short flights (under ~3000 m AGL) and moderate temperature ranges, the error introduced by assuming dry air is less than 0.5%.

## 3.3 Base Pressure Calibration

**On power-on, the code accumulates 50 consecutive LPS22HB readings at 25 Hz (approximately 2 seconds) and averages them to establish the base pressure. The averaging eliminates startup transients in the sensor and short-term pressure fluctuations from wind gusts or handling.**

```
#define BARO_BASE_SAMPLES  50   // 50 samples at 25 Hz = ~2 seconds
```

**After calibration:**

- The 1-Euro altitude filter is reset so it initialises at 0 m AGL.
- The packet transmitter starts. All altitude values are relative to this base pressure.
- If the aircraft is moved significantly in altitude before power-on (e.g., carried up a hill), simply power off and on again at the launch point to recalibrate.

**NOTE:** *The LPS22HB is sensitive to rapid pressure changes caused by airflow across the module. If the PCB is in an open frame or exposed to propeller wash, shelter the sensor from direct air impingement using a small piece of open-cell foam over the sensor. This is common practice in barometric altitude sensing.*

### 3.4 Temperature Effects

**The barometric formula assumes a fixed temperature relationship with altitude. In reality, temperature at the launch site affects the air density. A 10°C error in assumed temperature introduces approximately 0.3% altitude error (about 3 m per 1000 m AGL). For most RC flight operations the error is negligible. The LPS22HB includes an internal temperature sensor; future firmware versions could use it to apply a temperature correction.**

## 4. LoRa Radio Configuration

### 4.1 Spreading Factor Selection for 2 Hz Rate

**The spreading factor (SF) controls how a LoRa symbol is spread over time in the frequency domain. Higher SF = more chips per symbol = longer time on air = better range, but lower data rate.**

**The Time-on-Air (ToA) for a given packet depends on SF, bandwidth, coding rate, and payload size. For this flight logger (approximately 48-byte CSV payload, 125 kHz BW, 4/5 CR):**

| SF | ToA (48 B) | Max Pkt Rate | vs SF7 Range | Suitability |
|---|---|---|---|---|
| SF7 | ~123 ms | 8.1 pkt/s | Baseline | Too short range for flight logging |
| SF8 | ~216 ms | 4.6 pkt/s | +2.5 dB | Short range, fast |
| SF9 (USED) | ~370 ms | 2.7 pkt/s | +5 dB | Supports 2 Hz with 130 ms headroom |
| SF10 | ~698 ms | 1.4 pkt/s | +7.5 dB | Cannot support 2 Hz reliably |
| SF11 | ~1233 ms | 0.8 pkt/s | +10 dB | Cannot support 2 Hz |
| SF12 | ~2302 ms | 0.43 pkt/s | +12.5 dB | Cannot support 2 Hz |

**SF9 is the optimal choice: it is the highest spreading factor that fits within the 500 ms transmit interval,**

**and still provides a 5 dB range improvement over SF8 and approximately twice the range of SF7. The 130 ms headroom between the end of transmission and the next scheduled packet provides adequate time for IMU and GPS processing.**

### 4.2 Complete LoRa Settings Table

| Parameter | Setting | Code |
|---|---|---|
| Frequency | 915 MHz (USA) | LoRa.begin(915E6) |
| Spreading Factor | SF9 | LoRa.setSpreadingFactor(9) |
| Signal Bandwidth | 125 kHz | LoRa.setSignalBandwidth(125E3) |
| Coding Rate | 4/5 | LoRa.setCodingRate4(5) |
| TX Power | 20 dBm | LoRa.setTxPower(20) |
| CRC | Enabled | LoRa.enableCrc() |
| TX Interval | 500 ms (2 Hz) | #define LORA_TX_INTERVAL_MS 500 |

**WARNING: EU 868 MHz DUTY CYCLE: The 1% duty-cycle rule in 868 MHz EU regions limits SF9 (370 ms ToA) to one packet per 37 seconds on a single channel. For EU operation, implement multi-channel frequency hopping or revert to SF12 at a longer interval. The 915 MHz USA band has no duty-cycle restriction.**

## 5. 1-Euro Filter for Barometric Altitude

### 5.1 Why Filter Barometric Altitude?

**Although the LPS22HB is far less noisy than GPS altitude, its readings still contain a small amount of noise from electrical interference, mechanical vibration transmitted to the PCB, and turbulent airflow around the aircraft. At 25 Hz, each raw reading may vary by approximately ±0.2–0.5 m from the true altitude. Without filtering, this noise would be transmitted directly in the LoRa packet.**

**A fixed-cutoff low-pass filter introduces a compromise: low cutoff = smooth static readings but lag during climbs and dives; high cutoff = responsive during climbs but noisy at rest. The 1-Euro filter eliminates this trade-off by adapting its cutoff based on signal velocity.**

### 5.2 How the 1-Euro Filter Works

**At each sample the filter:**

- Computes the first derivative (rate of change) of the altitude signal. This derivative is itself low-pass filtered to prevent a single noisy sample from incorrectly triggering a high-cutoff response.
- Computes an adaptive cutoff frequency: cutoff = minCutoff + beta * |filtered_derivative|. When the aircraft is stationary, the derivative is near zero and the cutoff stays at minCutoff (heavy smoothing). When the aircraft is climbing or diving rapidly, the derivative is large, the cutoff rises, and the filter becomes more transparent (minimal lag).
- Applies a standard first-order low-pass filter at the adaptive cutoff to produce the final smoothed altitude.

The update equations are:

```
dt         = time since last sample (seconds)
dx         = (rawAlt - prevAlt) * freq
smoothedDx = alpha(dCutoff) * dx + (1 - alpha(dCutoff)) * prevDx
cutoff     = minCutoff + beta * |smoothedDx|
alpha_c    = 1 / (1 + 1/(2*pi*cutoff*dt))
filtered   = alpha_c * rawAlt + (1 - alpha_c) * prevFiltered
```

### 5.3 Parameter Selection for LPS22HB at 25 Hz

| Parameter | Value | Units | Rationale |
|---|---|---|---|
| freq | 25 | Hz | Matches LPS22HB ODR — the filter operates at sensor sample rate |
| minCutoff | 0.5 | Hz | At rest: 0.5 Hz cutoff removes the ~0.3 m pressure noise to under 0.05 m RMS |
| beta | 0.2 | Hz/(m/s) | A 1 m/s climb rate raises the cutoff to ~0.7 Hz. A 5 m/s climb raises it to 1.5 Hz. Fast ascents are tracked with only 1–2 sample delay. |
| dCutoff | 1.0 | Hz | Standard derivative filter — prevents single-spike noise from opening the cutoff |

**To increase smoothing (e.g., for an aircraft with heavy vibration), lower beta to 0.05 and raise minCutoff to 1.0. To improve tracking of very fast altitude changes, raise beta to 0.5. Start with the defaults and tune based on logged data from a test flight.**

### 5.4 Relationship to Transmission Rate

**The 1-Euro filter runs at 25 Hz (every 40 ms) inside the barometer polling routine, providing a continuously updated, smoothed altitude estimate. The LoRa transmitter reads the latest filtered value every 500 ms. This means 12–13 filter updates occur between each transmitted packet, ensuring the transmitted altitude is always the most recently filtered value rather than a stale or unfiltered reading.**

## 6. Kalman Filtering for Orientation

### 6.1 Sensor Fusion Problem

**Three sensors contribute to orientation measurement. Each has complementary strengths and weaknesses:**

- Accelerometer: Noisy but provides absolute pitch and roll from gravity. Useless during linear acceleration (aircraft accelerating in a straight line displaces the gravity vector).
- Gyroscope: Precise short-term angular rate, but drifts over time as integration error accumulates (bias).
- Magnetometer: Provides absolute yaw referenced to magnetic north, but susceptible to local magnetic interference from motors, battery cables, and electronics.

The Kalman filter is the statistically optimal linear estimator for fusing these sensors.

## 6.2 Two-State Kalman Filter (Pitch and Roll)

**The state vector is [angle, gyro_bias]. The Kalman filter alternates between two steps every IMU sample:**

### Predict Step — trust the gyroscope

```
rate   = gyroRate - estimated_bias

angle += dt * rate

P[0][0] += dt*(dt*P[1][1] - P[0][1] - P[1][0] + Q_angle)

P[1][1] += Q_bias * dt
```

**The covariance matrix P tracks the uncertainty in both state variables. Q_angle and Q_bias are process noise: how much the true angle and bias can change per unit time.**

### Update Step — correct with accelerometer

```
S = P[0][0] + R_measure    // Innovation covariance

K = [P[0][0]/S, P[1][0]/S] // Kalman gain

y = accelAngle - angle      // Innovation (residual)

angle += K[0] * y

bias  += K[1] * y
```

**R_measure is the measurement noise of the accelerometer-derived angle. The Kalman gain K automatically balances the trust placed in the gyro prediction versus the accelerometer measurement based on their relative uncertainties.**

## 6.3 Magnetometer Yaw with Tilt Compensation

**Gravity has no yaw component, so yaw cannot be derived from the accelerometer. The magnetometer provides absolute yaw, but its reading is distorted if the aircraft is tilted. Tilt compensation projects the magnetometer vector onto the horizontal plane using the already-computed pitch and roll:**

```
Mx = mx*cos(pitch) + mz*sin(pitch)

My = mx*sin(roll)*sin(pitch) + my*cos(roll) - mz*sin(roll)*cos(pitch)

rawYaw = atan2(-My, Mx)  [normalised to 0-360 degrees]
```

**A scalar Kalman filter smooths the resulting yaw to remove magnetometer noise without introducing a bias estimator (yaw has no integrating sensor).**

# 7. Telemetry Packet Format

**The flight logger transmits a comma-separated ASCII string at 2 Hz (every 500 ms). The packet contains 8 fields. GPS altitude has been removed; altitude is now provided exclusively by the barometric sensor.**

| Field | Type | Unit | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| LAT | float | degrees | GPS latitude — 6 decimal places (~0.1 m resolution) |
| LON | float | degrees | GPS longitude — 6 decimal places |
| BARO_AGL | float | metres | 1-Euro filtered altitude above launch base from LPS22HB. Resolution ~0.1 m. |
| PITCH | float | degrees | Kalman-filtered pitch. Positive = nose up. |
| ROLL | float | degrees | Kalman-filtered roll. Positive = right wing down. |
| YAW | float | degrees | Kalman-filtered magnetic heading. 0 = North, 90 = East, 0-360°. |
| SPEED | float | km/h | GPS ground speed |
| SATS | integer | - | Number of GPS satellites used in fix |

**Example packet: 51.501234,-0.124567,12.84,2.4,-1.1,274.5,15.3,8**

**Approximate payload length: 48 bytes. SF9 Time-on-Air: ~370 ms.**

# 8. Code Walkthrough

## 8.1 Setup

- IMU.begin() initialises the LSM9DS1 over internal I2C. Prints the accelerometer sample rate for verification.
- BARO.begin() initialises the LPS22HB at its default 25 Hz ODR. The code immediately halts with an error if this fails, as altitude is the primary sensor.
- The base-pressure accumulator begins running immediately after BARO.begin(). The 1-Euro filter is reset and LoRa transmission starts only after the full 50-sample average is complete.
- LoRa.begin(915E6) initialises the RFM95. SF9, 125 kHz BW, 4/5 CR, 20 dBm, and CRC are then configured. A mismatch with the base station on any of these parameters will cause zero packets to be received.
- Serial1.begin(9600) opens the hardware UART for the GT-U7 GPS.

## 8.2 Main Loop Structure

**The loop() runs without any blocking delay calls. All operations are triggered by elapsed time or data availability checks:**

- GPS bytes from Serial1 are fed to gps.encode() one character at a time, every loop iteration.
- updateOrientation() checks IMU.accelerationAvailable() and runs the Kalman filter only when new data is ready. The dt is measured in microseconds for numerical accuracy at 100+ Hz.
- updateBaroAltitude() is called every 40 ms (BARO_POLL_INTERVAL_MS). It reads BARO.readPressure(), runs the base calibration accumulator or the 1-Euro filter, and updates baroAGL.
- sendLoRaPacket() is called every 500 ms once base pressure is set. It formats the 8-field CSV, calls LoRa.endPacket(false) (blocking mode), and mirrors the packet to USB serial.

### 8.3 Why endPacket(false) (Blocking)?

**The LoRa library supports both blocking (false) and non-blocking (true) transmit modes. Blocking mode is used here because the loop timing is simple and predictable: the Nano 33 pauses for ~370 ms during transmit, then resumes. Non-blocking mode would allow the CPU to continue running during transmit but would require interrupt or polling logic to detect transmission completion. Since the barometer and IMU can tolerate a 370 ms gap (data is buffered in the sensor FIFOs), blocking mode is the simpler and safer choice.**

## 9. Troubleshooting

### 9.1 Barometric Sensor

| Symptom | Cause & Fix |
|---|---|
| "LPS22HB barometer init failed" | Ensure board is set to "Arduino Nano 33 BLE Sense" in Arduino IDE. Library Arduino_LPS22HB must be installed. The sensor is not present on standard Nano 33 BLE (non-Sense variant). |
| Base pressure never sets (counter stuck) | BARO.readPressure() may return 0 on the first few calls after begin(). The code guards against this. If stuck past 10 seconds, check library version (>=1.0.0) and confirm correct board selection. |
| AGL reading drifts slowly upward at rest | Normal: atmospheric pressure changes with weather. A 1 hPa drop in pressure corresponds to ~+8 m indicated altitude. For long sessions, consider periodic recalibration. |
| AGL noisy / jumpy | Airflow over the PCB. Shield the LPS22HB from direct propeller wash with open-cell foam. Alternatively lower 1-Euro beta to 0.05 for heavier smoothing. |
| AGL reads altitude at boot instead of 0 | Base pressure not yet set when first packet is sent. The code will not transmit until basePressureSet = true. This is working as designed. |

### 9.2 LoRa (SF9)

| Symptom | Cause & Fix |
|---|---|
| "LoRa init failed" | Check SPI wiring D10-D13. Confirm 3.3V supply to RFM95. D10 must be NSS (not NSS of another SPI device sharing the bus). |
| Base station receives no packets | Both ends MUST use SF9, 125 kHz, 915 MHz, CRC enabled. Any mismatch means zero packets. Check base station sketch has been updated. |
| Packets received at <2 Hz | Confirm LORA_TX_INTERVAL_MS = 500. If the base station is printing debug that takes time, ensure it is not calling delay(). |

| | |
|---|---|
| Occasional missed packets at 2 Hz | SF9 ToA is 370 ms; 500 ms interval leaves only 130 ms. A slow IMU read or large GPS sentence can cause a packet to be delayed by one cycle. Normal for SF9 at this rate. |

## 9.3 IMU / Orientation

| Symptom | Cause & Fix |
|---|---|
| "LSM9DS1 IMU init failed" | Same board selection issue as barometer. Requires Nano 33 BLE Sense board package. |
| Pitch/roll drift at rest | Normal gyro integration behaviour. The Kalman bias estimator corrects drift within ~10 s of power-on. Increase Q_bias if residual drift persists. |
| Yaw erratic or 180° wrong | Magnetic interference. Move away from motors, battery wiring, or steel fasteners. Calibrate the magnetometer by rotating the aircraft in a figure-8 pattern. |

## 9.4 GPS

| Symptom | Cause & Fix |
|---|---|
| LAT/LON shows 0.0 in packets | GPS has no fix. Wait for the GT-U7 fix LED to blink at 1 Hz. Latitude/longitude default to 0 until gps.location.isValid() is true. |
| GPS fix takes >5 minutes | Cold start with no almanac. First boot in a new location can take up to 5 minutes. Ensure clear sky view. Subsequent boots are faster (warm start). |
| SATS shows 0 | No fix yet. Note: altitude is not affected — it comes from the barometer, not GPS. |

## 9.5 Library Installation Summary

| Library | Search Name | Author |
|---|---|---|
| Arduino LSM9DS1 (IMU) | Arduino_LSM9DS1 | Arduino |
| Arduino LPS22HB (Barometer) | Arduino_LPS22HB | Arduino |
| LoRa (Sandeepmistry) | LoRa | Sandeep Mistry |
| TinyGPS++ | TinyGPSPlus | Mikal Hart |

**NOTE:** ***Do NOT install RadioHead. Use only the Sandeepmistry "LoRa" library. RadioHead is incompatible with the nRF52840 processor used on the Nano 33 BLE Sense.***