International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

SkySense: LoRa-Enabled Autonomous Weather Monitoring Drone

Prof. Sharda Mungale, Vineet Wankhede, Sanwedana Lokhande, Sejal Vaidya, Aditya Khandalkar
Department of Industrial IOT, Priyadarshini College of Engineering, Nagpur, Maharashtra, India

Abstract: Traditional meteorological data collection relies heavily on fixed monitoring posts, creating significant blind spots across agricultural fields, elevated terrains, and industrial sectors where physical access is restricted. To bridge this gap, this project develops SkySense, an unmanned aerial system featuring an autonomous quadcopter equipped with environmental sensors that streams atmospheric telemetry back to a base station via long-range radio transmissions. Flight stabilisation and route navigation are governed by a customised control board driven by an STM32F411 chip. Concurrently, a secondary STM32F401 processor interfaces with a DHT22 ambient sensor and an ENS160 air purity module, multiplexing sensor telemetry into compact packets dispatched over a point-to-point LoRa channel. A ground-side ESP32 unit captures these frames, presents telemetry and link metrics locally on a TFT display, and routes records to Google Cloud over Wi-Fi. This documentation details the underlying hardware selection, system topology, data formatting, and theoretical calculations covering link margin, radio airtime, and hover endurance, alongside a structured verification protocol for future field trials.

Keywords: UAV, Weather Monitoring, LoRa, Air Quality, STM32, Internet of Things

1. INTRODUCTION

Ambient metrics such as relative humidity, atmospheric temperature, and air purity serve as critical variables for precision farming, public health safeguards, industrial compliance, and emergency management. However, existing observation networks consist primarily of stationary ground installations. While precise locally, these stations involve high deployment costs and remain sparsely distributed, rendering them ineffective for capturing microclimates within valleys, orchards, or isolated industrial corridors. Deploying manual spot-checks in such remote or rugged terrain is labour-intensive and hazardous.

Deploying multirotor unmanned aerial vehicles offers a viable solution to these spatial limitations. Quadcopters can navigate to targeted coordinates, hold position to sample ambient air at precise elevations, and execute repeatable flight paths. Although prior atmospheric UAV studies highlight these operational advantages, they also emphasise inherent constraints regarding payload limits, flight endurance, and aerodynamic turbulence generated by rotor downwash [1], [2]. Furthermore, communication infrastructure presents a distinct challenge; cellular networks and local Wi-Fi cannot be depended upon in remote deployment zones.

To overcome connectivity barriers, SkySense utilises LoRa technology, a spread-spectrum modulation scheme engineered for long-range, low-power telemetry packets of minimal size [3], [4]. Environmental telemetry frames require only tens of bytes and updates at multi-second intervals, fitting the operational profile of LoRa links. Distinguishing between physical-layer LoRa modulation and the network-layer LoRaWAN architecture, which requires centralised gateways and server infrastructure [5], SkySense implements a direct, point-to-point radio link between the aircraft and a base receiver, keeping the cloud-upload pipeline entirely independent.

1.1 Problem Statement and Objectives

The problem addressed here is how to obtain georeferenced temperature, humidity and air-quality readings from places that fixed stations do not cover, without relying on cellular or Wi-Fi coverage at the drone. The objectives are to:

- define an architecture that joins a custom STM32-based quadcopter, a weather and air-quality payload and a LoRa telemetry link;
- specify the components, packet format and operating procedure of a student-scale prototype;

- estimate airtime, free-space link margin and hover endurance analytically; and
- define a measurement protocol that will allow the design to be evaluated honestly once it is built and flown.

1.2 Scope and Status of Work

This paper follows a precise functional classification framework:

- **Defined:** Hardware configurations and interface schematics that are locked into project documentation.
- **Proposed:** System features incorporated into the architectural blueprint but not yet physically implemented or tested.
- **Implemented:** Capabilities that have been assembled and verified for functional execution.
- **Validated:** Features subjected to rigorous experimental protocols with documented performance logs.

Table 1 summarises the current implementation milestones of the project. At this stage, all operational features remain categorised under the Proposed tier pending empirical flight testing. Multi-drone deployment and cloud automation are reserved for future development phases.

Table 1. Status register of SkySense features (to be updated by the authors)

Feature	Status	Evidence required before status changes
Hardware selection and interface plan	Defined	Project component documentation
Stabilised flight with RC control	Proposed	Flight-controller logs of a test flight
GPS-assisted modes (hold, return home, waypoint)	Proposed	Logged flight showing each mode working
Weather and air-quality payload	Proposed	Bench comparison against a reference instrument
Packet formation and LoRa transmission	Proposed	Serial logs of frames sent and received
ESP32 receiver, TFT display and cloud upload	Proposed	Screenshots and cloud records from a live run
Range, packet delivery rate, endurance	Proposed	Field-test tables following the protocol in Section 6

Table 1 states what is claimed; it will be revised if and when tests are completed.

2. LITERATURE REVIEW

2.1 Drone-Based Environmental Monitoring

Environmental data collection using unmanned aerial vehicles has seen significant development over recent years. Early foundational approaches include Ya'acob et al.'s design of a UAV-based monitoring framework [6] and Sajid et al.'s deployment of multi-sensor wireless nodes mounted directly on quadcopters [7]. Alternative architectures studied by Potter et al. leveraged drones as mobile collection hubs within broader sensor networks [8], while Jumaah et al. concentrated heavily on airborne particulate (PM2.5) tracking [9]. Although these studies validate aerial environmental sensing, they primarily focus on particulate capture or ground-node data harvesting rather than streaming real-time, integrated meteorological parameters over long-range radio links.

2.2 Meteorological Measurement from Small UAVs

Mounting weather instrumentation onto a multirotor involves complex aerodynamic and thermal considerations rather than simple hardware attachment. As discussed by Jacob et al. [2] and shown in the campaign comparison by Barbieri et al. [10], variables such as rotor downwash, battery and component thermal radiation, and slow sensor response times can introduce significant measurement biases if placement is neglected. Furthermore, wind parameters on multirotors

are frequently calculated directly from the vehicle's inertial motion rather than using physical anemometers [11], [12]. This justifies SkySense's approach of relying on dedicated temperature, humidity, and gas indicators while avoiding direct wind speed claims.

2.3 LoRa and UAV Communication

The LoRa literature keeps coming back to one trade-off. A higher spreading factor gives a longer reach and better receiver sensitivity, but the data rate drops and every packet stays on the air for longer [3], [13]. Field measurements show that the range people actually get depends heavily on where the antennas are placed and on the terrain around them [4], [14]. A comparison of low-power wide-area technologies lists LoRa among the choices that suit short, infrequent messages [15], and Adelantado et al. point out where LoRaWAN runs into duty-cycle and capacity limits [5]. Two studies bring drones into LoRa systems. Delafontaine et al. used a drone inside an existing LoRa network to help locate nodes, and tested the idea in the field [16]. Behjati et al. proposed LoRa for an Internet of Drones, in which a drone carrying a LoRa gateway collects readings from sensors on farms, and their work combines real measurements with simulation [17]. In both studies the drone acts as a mobile gateway for sensors on the ground. Neither treats the drone itself as the source of the weather data.

Table 2. Comparison of selected related approaches (summarised from titles and abstracts)

Ref.	Main focus	Role of the UAV / link	Difference from SkySense
[6]	UAV environment monitoring	UAV as sensing platform	LoRa-based telemetry is not the central theme
[7]	Multi-sensor node on a quadcopter	Airborne sensing node	Focus on node design; long-range link not the subject
[8]	Drone-enabled wireless sensor network	UAV collects data from ground sensors	Drone as collector, not as weather source
[9]	UAV-based PM2.5 monitoring	Airborne air-quality sensing	Particulates rather than weather plus gas indicators
[16]	Drone-aided localisation in LoRa networks	Drone as mobile gateway for node localisation	Localisation task, not weather sensing
[17]	LoRa-enabled Internet of Drones for livestock	Drone with LoRa gateway for ground sensors	Drone as gateway; data originate on the ground

Table 2 compares the roles of the UAV and the radio link; the entries describe each paper's stated focus and should be checked against the full texts before submission.

2.4 Research Gap

A narrow but defensible gap can be identified. UAV environmental monitoring and LoRa-UAV integration have each been studied, but a drone that carries its own weather and air-quality payload, streams georeferenced observations over a point-to-point LoRa link to a ground station, and forwards them to a cloud dashboard is less clearly documented in the sources reviewed here. SkySense does not claim novelty in any single component. Its contribution is a specified integration on low-cost STM32 and ESP32 hardware and a transparent evaluation plan. A systematic literature search would be needed before any stronger statement of novelty is made.

3. PROPOSED SYSTEM ARCHITECTURE

3.1 Overview

SkySense is divided into an airborne segment and a ground segment, as shown in Fig. 1. The airborne segment has two microcontrollers with separate duties. An STM32F411CEU6 Black Pill is the flight controller: it reads the inertial, compass, barometer, GPS and radio-control inputs, runs attitude estimation and PID loops, drives four ESCs, and manages arming and failsafe. A separate STM32F401CCU6 Black Pill is the weather and telemetry node: it reads the DHT22 and ENS160, receives flight data from the flight controller, builds the packet and passes it to the LoRa

module. The ground segment is an ESP32 with a LoRa receiver and TFT display, which also uploads data to Google Cloud over Wi-Fi. Flight control and telemetry are kept apart so that a fault in the payload code cannot affect flight safety.

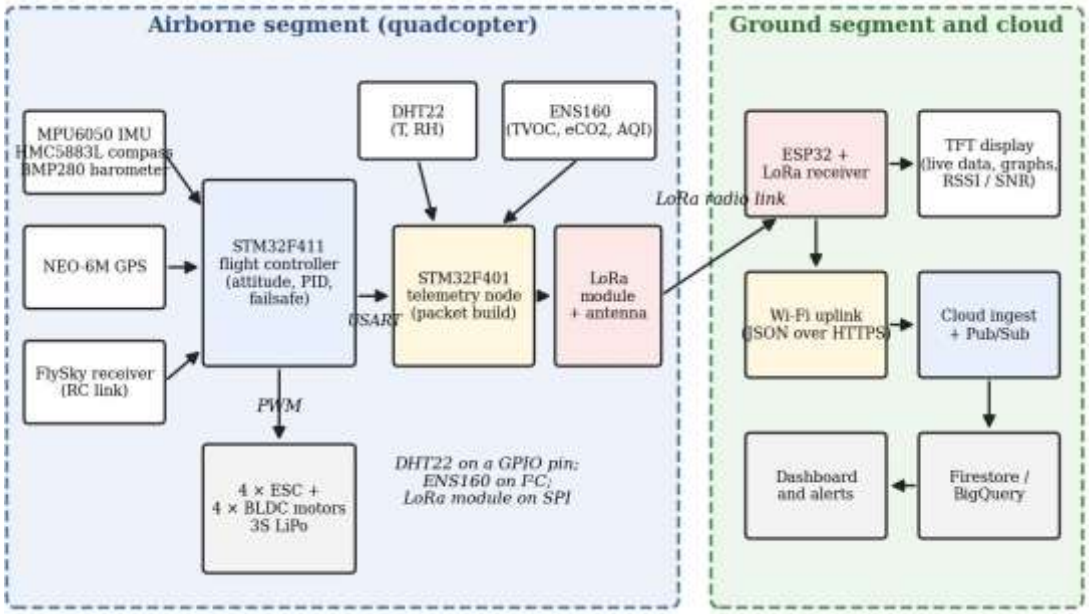


Fig. 1 Proposed block architecture of SkySense

Fig. 1 is an original block diagram prepared for this paper. Arrows show data or control flow; the pilot's FlySky transmitter reaches the flight controller through the receiver and is independent of the LoRa telemetry link.

3.2 Data Flow

The DHT22 supplies temperature and relative humidity through a single-wire digital interface, and the ENS160 supplies TVOC, equivalent CO2 (eCO2) and an air-quality index (AQI) over IZC. The telemetry node also receives GPS position, altitude, speed, battery voltage and flight status from the flight controller over a USART link. It combines these into one packet and sends it to the LoRa module over SPI. The ESP32 ground station receives the packet, verifies its checksum, records the received signal strength indicator (RSSI) and signal-to-noise ratio (SNR), shows the values on the TFT display and, when a Wi-Fi connection is available, uploads a JSON message to the cloud. Missing packet numbers reveal lost packets, which is how the packet delivery rate in Section 6 is determined.

3.3 Proposed Packet Format

The telemetry packet contains a packet counter, timestamp, GPS position, altitude, speed, battery voltage, temperature, humidity, TVOC, eCO2, AQI and a CRC-8 or CRC-16 checksum. To keep airtime short, a fixed-length binary frame of 32 bytes is proposed (Table 3). The field sizes and resolutions are design proposals and may change after the sensor ranges and the LoRa module are confirmed.

Table 3. Proposed 32-byte application payload

Field	Bytes	Description / proposed encoding
Packet ID	2	Incrementing counter; used to detect loss
Timestamp	4	UTC seconds, or milliseconds since boot
Latitude, longitude	4 + 4	Signed integers, degrees × 10^7
Altitude	2	0.1 m steps, signed
Speed	2	0.1 m/s steps
Battery voltage	2	Millivolts

Temperature	2	Signed, 0.1 °C steps
Relative humidity	2	0.1 %RH steps
TVOC	2	ppb
eCO2	2	ppm
AQI	1	ENS160 air-quality index
Status flags	1	GPS fix, sensor fault, low battery
Checksum	2	CRC-16 (a CRC-8 is the alternative named in the project notes); the LoRa PHY CRC is separate

Table 3 gives the frame assumed in the airtime estimates of Section 5.

The ground station reformats each decoded packet as a JSON message with the same fields plus RSSI, SNR and a UTC timestamp. Both LoRa modules must use identical frequency, bandwidth, spreading factor, coding rate and sync word.

4. HARDWARE AND SOFTWARE REQUIREMENTS

This section lists the components of the SkySense prototype as they appear in the project's component documentation, along with the manufacturers' data sheets [18]-[24]. The documentation does not name the LoRa module or the operating frequency, so those two items are left as placeholders. Table 4 gives the full list.

Table 4. Components of the SkySense prototype

Component	Qty	Main use	Connected to
STM32F411CEU6 Black Pill	1	Primary flight controller	IMU, compass, barometer, GPS, RC receiver, ESCs, telemetry node
MPU6050	1	Accelerometer and gyroscope	STM32F411 (IZC)
HMC5883L	1	Magnetic compass / heading	STM32F411 (IZC)
BMP280	1	Barometric pressure and relative altitude	STM32F411 (IZC or SPI)
NEO-6M GPS	1	Position, speed, course and time	STM32F411 (USART)
FlySky FS-i6S transmitter and 10-channel receiver	1 + 1	Pilot control commands	STM32F411
1000 kV BLDC motor	4	Thrust	ESC outputs
30 A ESC	4	Motor speed control	STM32F411 PWM, battery power
3S 2200 mAh 35C LiPo	1	Main power source	ESCs, power distribution, regulators
Component	**Qty**	**Main use**	**Connected to**
DHT22	1	Ambient temperature and humidity	STM32F401 (single-wire GPIO)
ENS160	1	TVOC, eCO2 and AQI	STM32F401 (IZC)
STM32F401CCU6 Black Pill	1	Weather and telemetry node	DHT22, ENS160, LoRa module, STM32F411
LoRa transmitter module	1	Sends telemetry to ground [model and frequency to be confirmed]	STM32F401 (SPI)
LoRa-enabled ESP32	1	Ground LoRa receiver and Wi-Fi gateway	TFT display, Wi-Fi, Google Cloud

TFT display	1	Telemetry, graphs and signal data	ESP32
Google Cloud services	–	Storage, visualisation and alerts	ESP32 via Wi-Fi

Table 4 is taken from the project component documentation; part numbers are as stated there.

4.1 Flight Controller: STM32F411CEU6 Black Pill

The STM32F411CEU6 is a 32-bit Arm Cortex-M4 chip with a floating-point unit. It runs at up to 100 MHz, has 512 KB of flash and 128 KB of SRAM, and uses 3.3 V logic [18]. Its timers give separate PWM outputs for the four ESCs, and the USART, IZC and SPI ports are enough to connect the GPS, receiver, sensors and the link to the telemetry node. In the project plan the MPU6050 and HMC5883L share the IZC bus, the BMP280 sits on IZC or SPI, the NEO-6M is on a USART, and the FlySky receiver is read through a serial (SBUS or i-BUS) input or by PWM capture. Because the board is a 3.3 V device, a 5 V GPS output should not go straight to a GPIO pin; the module must be confirmed 3.3 V tolerant, or a level shifter is needed.

4.2 Navigation and Inertial Sensors

MPU6050. The MPU6050 is a 6-axis IMU. It has a 3-axis accelerometer (±2 to ±16 g) and a 3-axis gyroscope (±250 to ±2000 degrees/s), both read at 16-bit resolution over IZC, usually at address 0x68 [19]. It should be fixed firmly near the centre of gravity, away from the ESCs, power wires and motors. The gyroscope is calibrated with the drone sitting still, and a low-pass or complementary filter removes much of the vibration noise.

HMC5883L. The HMC5883L is a 3-axis digital compass (±1.3 to ±8.1 gauss, IZC address 0x1E). It gives the heading with respect to magnetic north and helps correct the slow yaw drift of the gyroscope [20]. A small mast above the frame keeps it clear of high-current wires and steel parts, and it needs a calibration outdoors to remove hard-iron and soft-iron offsets.

BMP280. The BMP280 is a digital barometer for roughly 300 to 1100 hPa. Its data sheet quotes a relative accuracy of about ±0.12 hPa and an absolute accuracy of about ±1 hPa, with IZC or SPI output [21]. The flight controller turns pressure changes into relative altitude. Open-cell foam over the sensor reduces noise from wind and propeller pressure, the sensor should stay away from ESC and battery heat, and the ground-level pressure at take-off is used as the reference.

NEO-6M GPS. The u-blox NEO-6M outputs latitude, longitude, altitude, ground speed, course, satellite count and UTC time as NMEA sentences over UART. Typical horizontal accuracy is about 2.5 m CEP with a good view of the sky, at a typical 1 Hz update rate [22]. Its antenna goes on a raised mast, laid flat and away from the LoRa antenna, power wires and ESCs. GPS-dependent modes should be enabled, and the home position stored, only once a stable 3D fix is available.

4.3 Radio Control and Propulsion

The FlySky FS-i6S transmitter and its 10-channel receiver give the pilot manual control. A typical channel plan assigns roll, pitch, throttle and yaw to CH1 to CH4, arm/disarm to CH5, flight mode (angle, horizon, altitude hold, GPS hold) to CH6, return-to-home or GPS enable to CH7, and buzzer/LED control, telemetry request and a reserved function to CH8 to CH10. The flight controller must monitor signal validity and apply a failsafe (cut throttle or controlled landing or return home) if the link is lost, and must not arm unless the RC link, IMU calibration and safety conditions are valid.

Four 1000 kV brushless motors, each driven by a 30 A ESC, provide thrust. A 1000 kV motor on a 3S battery (11.1 V nominal) has an approximate no-load speed of 1000 × 11.1 = 11,100 rpm; the loaded speed is lower. Two motors turn clockwise and two counter-clockwise, and propeller direction must match. A separate 5 V UBEC is preferred over the ESC's built-in BEC for powering the STM32 boards.

The 3S 2200 mAh 35C LiPo battery (11.1 V nominal, 12.6 V fully charged) supplies the ESCs directly, with regulated 5 V or 3.3 V supplies for the electronics. Its 35C rating corresponds to a maximum continuous discharge capability of 35 × 2.2 A = 77 A. The battery should not be discharged below about
3.5 V per cell under load, should be stored at about 3.8 V per cell, and should be charged only with a balance charger

in a LiPo-safe bag, with a low-voltage alarm or voltage telemetry in use.

4.4 Weather and Air-Quality Payload

DHT22 (AM2302). The DHT22 is a digital sensor for temperature (−40 to +80 °C, typically ±0.5 °C, 0.1 °C resolution) and relative humidity (0 to 100 %RH, typically ±2 %RH, 0.1 %RH resolution), using a single-wire protocol [23]. It can be read at most once every two seconds, and the data line needs a 4.7 kfi to 10 kfi pull-up resistor. The sensor has to sit in open, ventilated air, away from motor wash and from ESC and battery heat, and it must not be sealed in an enclosure. Compared with IZC environmental sensors it is slow, so it is fine for periodic weather readings but not for fast control feedback.

ENS160. The ENS160 is a digital metal-oxide multi-gas sensor. It reports TVOC (0 to 65,000 ppb), eCO2 (400 to 65,000 ppm) and an AQI over IZC or SPI, at address 0x52 or 0x53 depending on the ADDR pin [24]. The operating range is about −40 to +85 °C and 5 to 95 %RH non-condensing, and the readings typically need under three minutes of warm-up to settle. If the library supports it, DHT22 temperature and humidity can be passed to the ENS160 for compensation. The sensor should be shielded from rain, dust, propeller wash, battery fumes, adhesives and ESC heat. Its output is best read as a relative air-quality indicator and not a laboratory-grade concentration, and the eCO2 value is calculated from the sensor's response, not measured directly.

4.5 Telemetry Node, LoRa Link and Ground Station

The STM32F401CCU6 (Arm Cortex-M4, 84 MHz, 256 KB flash, 96 KB SRAM, 3.3 V logic) reads the DHT22 and ENS160, receives flight data from the STM32F411 and sends the packet to the LoRa module. It should not perform flight-critical control; stabilisation, motor control and failsafe remain on the STM32F411.

The link between the drone and the ground station is a point-to-point sub-GHz LoRa radio link. The authors still have to confirm the module model and the carrier frequency, and the frequency must be one that is allowed for unlicensed use in the country of operation. In India, 865–867 MHz is the band generally associated with licence-exempt LoRa-class devices [25], but the current rules should be checked before any transmission. Range depends on antenna design, transmit power, spreading factor, terrain, vegetation and interference, so no range figure is assumed here.

The ESP32 ground station receives the packets, decodes them, shows a dashboard on the TFT display, computes link quality and plots trends. The display shows flight status (arming, mode, GPS fix, satellites), position and distance from home, altitude and speed, battery voltage and a low-battery warning, temperature and humidity, TVOC, eCO2 and AQI, and RSSI, SNR, packets received and packet-loss percentage, together with graphs of altitude, battery voltage, temperature, humidity, TVOC and RSSI. If no packet arrives for several seconds, the display shows "NO SIGNAL" and not stale values.

4.6 Cloud Integration

The ESP32 connects to a router or hotspot over Wi-Fi and sends the telemetry as JSON over HTTPS to an authenticated ingest endpoint (Cloud Functions or API Gateway). From there, Google Cloud Pub/Sub buffers the messages and Firestore or BigQuery keeps the live and historical records. Looker Studio or a custom web dashboard shows maps, graphs and alerts, and Cloud Functions can send email, SMS or webhook alerts for low battery, lost signal, poor air quality or a geofence breach [26]. Google Cloud IoT Core has been retired, so it is not used. The upload needs internet at the ground station; the local TFT display and any local log work without it.

4.7 Software

The software consists of embedded firmware on the STM32F411 (sensor fusion, PID loops, RC decoding, GPS parsing, failsafe), embedded firmware on the STM32F401 (sensor reading, packet building, LoRa transmission), ESP32 firmware (LoRa reception, display, JSON creation, HTTPS upload) and the cloud services listed above. The exact libraries and firmware versions are to be added by the authors.

5. METHODOLOGY AND WORKING PRINCIPLE

5.1 Operating Procedure

The operating sequence is shown in Fig. 2. Before flight, the operator powers the electronics and allows the ENS160 to warm up, calibrates the IMU and compass, waits for a stable GPS 3D fix and stores the home position. Each sensor should first be tested individually, the motor direction and propeller type should be checked on the bench without propellers, and the RC failsafe should be set. The pilot arms the vehicle and flies it manually or in an assisted mode. Autonomous behaviour is limited to the GPS-assisted modes of the flight controller (position hold, return home and waypoint flight), which depend on the compass, barometer and GPS and are marked as proposed in Table 1. A pilot must stay ready to take over, because autonomous flight is different from remote-controlled flight and does not remove the need for supervision.

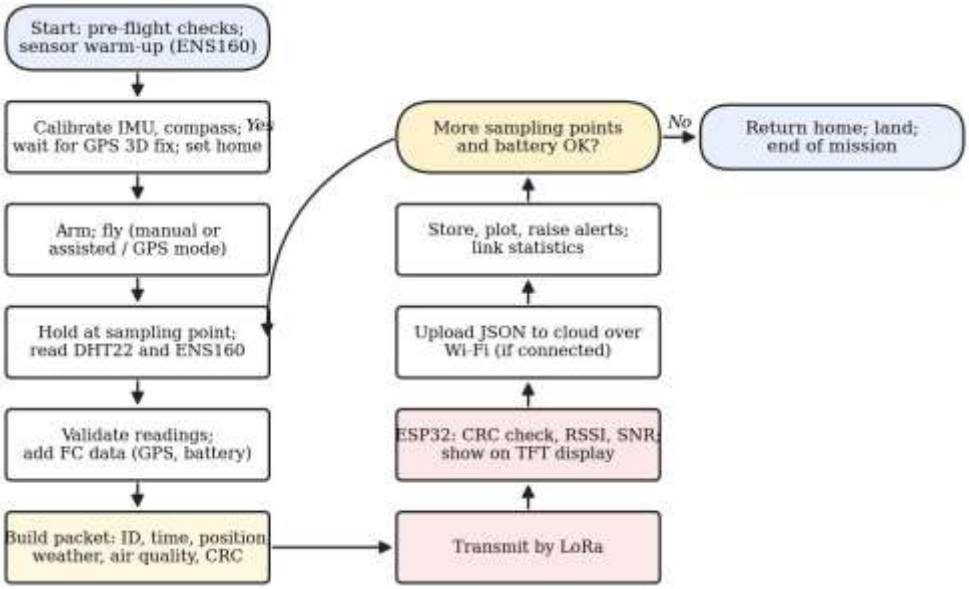


Fig. 2 Proposed operating workflow of SkySense

Fig. 2 is an original flowchart. The loop continues while sampling points remain and the battery is above the threshold.

At each sampling point the vehicle holds or slows briefly so that the air near the sensors can settle after the vehicle has passed. This reduces the influence of the vehicle's own wake and of sensor response time [2]. The telemetry node reads the DHT22 no more than once every two seconds and the ENS160 at its own update rate. It checks that the values are plausible, adds the latest flight data and a checksum, and transmits the packet. The ground station then checks, displays, logs and uploads it.

5.2 Telemetry Node Algorithm

```
initialise(DHT22, ENS160, LoRa, USART_to_FC)
wait until ENS160 warm-up complete
id <- 0
loop every T_s seconds (T_s >= 2 s):
    gps, alt, speed, batt, status <- read_from_FC()
    temp, rh <- read_DHT22()
    tvoc, eco2, aqi <- read_ENS160()   (compensated with temp, rh)
    if any value out of valid range: set flag SENSOR_FAULT
    pkt <- encode(id, time, gps, alt, speed, batt, temp, rh, tvoc, eco2, aqi,
flags)
    pkt <- pkt + checksum(pkt)
    LoRa.send(pkt)
    id <- id + 1
```

Algorithm 1. Telemetry node loop (pseudocode)

5.3 Analytical Models

The LoRa radio parameters follow standard expressions [3], [27]. For spreading factor SF, bandwidth BW and coding-rate index CR (1 for 4/5), the symbol duration and nominal bit rate are:

$$T_{sym} = 2^{SF} / BW \quad (1)$$

$$R_b = SF \cdot (BW / 2^{SF}) \cdot 4/(4 + CR) \quad (2)$$

The time on air of a frame with payload length PL bytes is the preamble duration plus the payload duration [27]:

$$T_{air} = (n_{pre} + 4.25) \cdot T_{sym} + n_{payload} \cdot T_{sym} \quad (3)$$

where n_pre is the number of preamble symbols and n_payload is the number of payload symbols given in the manufacturer's application note. For a 32-byte payload (Table 3), 125 kHz bandwidth, CR 4/5, 8 preamble symbols, explicit header and radio CRC enabled, Eq. (3) gives the values in Table 5. These are calculated values and not measurements.

Table 5. Calculated airtime and bit rate for a 32-byte payload (125 kHz, CR 4/5)

SF	T_sym (ms)	Bit rate (bps)	Time on air (ms)
7	1.02	5469	71.9
9	4.10	1758	246.8
10	8.19	977	452.6
12	32.77	293	1810.4

Table 5 shows that the same frame takes about 25 times longer to send at SF12 than at SF7. The DHT22 limit of one reading per two seconds also sets a lower bound on the sampling period.

For the radio path, the free-space path loss with distance d in km and frequency f in MHz is

$$FSPL\ (dB) = 20 \cdot \log10(d) + 20 \cdot \log10(f) + 32.44 \quad (4)$$

At 865 MHz this gives about 91.2 dB at 1 km, 105.2 dB at 5 km and 111.2 dB at 10 km. The link margin is then

$$M = P_t + G_t + G_r - L - S_{rx} \quad (5)$$

where P_t is the transmit power, G_t and G_r are the antenna gains, L is the total path loss and S_rx is the receiver sensitivity. Free-space loss is an optimistic bound for an airborne transmitter with a clear line of sight. Real links lose additional signal to terrain, vegetation, antenna orientation and the airframe, and the radio horizon and local obstructions will limit range before free-space loss does. The achievable range must therefore be measured, as proposed in Section 6.

Two quantities are used in the evaluation. Packet delivery rate is PDR = N_rx / N_tx × 100 %, where N_tx is counted from packet IDs. Sensor error against a reference is summarised by bias and root-mean-square error, RMSE = sqrt((1/n) Σ (x_i − r_i)Z).

Flight time can be estimated to first order as

$$t = \eta \cdot C / I_{avg} \quad (6)$$

where C is the battery capacity, η the usable fraction and I_avg the mean current. The component documentation gives a power-budget example in which each motor and ESC draws an average of 8 A in hover, so that I_avg is about 32 A. With C = 2.2 Ah and η = 1, Eq. (6) gives about 4.1 minutes, and with η = 0.8 it gives about 3.3 minutes. The 8 A figure is an assumed example and not a measurement. Battery sag, control corrections, wind and the added payload will shorten flight time, so the real value must be measured with the payload fitted. This estimate shows that endurance, and therefore coverage per flight, will be short on this airframe and battery.

6. EVALUATION PLAN AND DISCUSSION

No experimental results are reported in this paper. The prototype has not been flown or range-tested at the time of writing, and no measured data are presented. This section defines how the design should be evaluated and what the analytical work in Section 5 does and does not show. Section 6.3 is reserved for the authors' data.

6.1 Evaluation Parameters

Table 6. Proposed performance parameters and measurement methods

Parameter	Measurement method	Reported as
Temperature and humidity accuracy	Co-locate with a reference instrument on the bench and outdoors in shade; repeat with rotors running on the ground	Bias, RMSE, number of samples
Air-quality indicators	Compare ENS160 response with a reference or a known source in a controlled test after warm-up	Relative response; no absolute accuracy claimed
Rotor-induced effects	Compare hover readings at different mounting positions against a ground reference	Difference vs. reference (°C, %RH)
GPS and altitude consistency	Compare logged position and barometric altitude with a known point	Error (m)
Communication range	Ground and flight tests at increasing distance with logged RSSI and SNR, for each SF	Maximum distance with PDR above a stated threshold
Packet delivery rate	Count received packets against packet IDs	PDR (%), by distance and SF
Latency	Compare onboard timestamp with reception time on a time-synchronised ground computer, and with cloud arrival time	Mean and maximum (s)
Data reliability	Fraction of packets with valid checksum and plausible values	Valid packet ratio
Endurance and energy	Log battery voltage and current in hover and mission flight with payload	Flight time (min), mean current (A)
Parameter	**Measurement method**	**Reported as**
Environmental conditions	Repeat tests at different times of day, wind levels and humidity	Metrics above, per condition

Table 6 lists the metrics to be reported; thresholds must be fixed before testing.

6.2 Test Stages

Testing should proceed from the least to the most risky. Stage 1 is a bench test of each sensor and of the telemetry chain, including the LoRa link to the ESP32, the display and the cloud upload. Stage 2 is a ground range test with the transmitter at a height similar to that of the drone. Stage 3 is a propeller-free and then tethered or low-altitude hover test to check electrical interference between the radio, flight controller and sensors and to verify IMU orientation and ESC response. Stage 4 is a flight in an open area far from people, vehicles, buildings and airports, with a pilot ready to take over. Each stage should be completed and documented before the next begins.

6.3 Results (Reserved)

[Insert measured data, tables and graphs here after testing. Each figure or table must state the date, location, weather conditions, the radio settings (module, frequency, SF, BW, CR, power, antenna), the payload mass and the reference instrument used.]

6.4 Discussion of Expected Behaviour

The airtime analysis shows a clear trade-off. At SF7 the 32-byte frame occupies the channel for about 72 ms, whereas at SF12 it occupies it for about 1.8 s. A lower spreading factor suits short links and frequent updates, while a higher one improves sensitivity at the cost of data rate and energy per frame. Because the DHT22 is read at most every two seconds and the values change slowly, an update every few seconds is likely to be enough, which allows a conservative spreading factor. These statements describe the expected behaviour of the modulation and not the performance of a built system.

Sensor behaviour is likely to be the larger source of uncertainty. Heating by the electronics, rotor downwash and slow humidity response can produce errors larger than the data-sheet accuracy. The ENS160's metal-oxide element also responds to several gases and to environmental conditions, so its outputs are relative indicators. For these reasons the planned experiments compare hover and ground readings with a reference before any claim of accuracy is made.

7. APPLICATIONS, LIMITATIONS AND CHALLENGES

7.1 Applications

Agriculture: sampling microclimate across fields and orchards to support irrigation and crop-protection decisions. **Environmental observation:** repeated surveys of valleys, lake shores and forest edges where fixed stations are not available. **Remote and industrial-area monitoring:** readings of temperature, humidity and relative air-quality changes from locations that are hard to reach or unsafe to enter. **Disaster preparedness:** quick checks of local conditions before and during floods, storms or heat events, as a complement to official forecasts and not a replacement for them.

7.2 Limitations and Challenges

Radio: LoRa has a small bandwidth, so only compact messages can be sent. Obstructions, vegetation, interference and antenna orientation can all shorten the range, and regulations limit frequency, power and sometimes duty cycle. **Sensors:** the DHT22 is slow and affected by self-heating and rotor effects, and the ENS160 gives relative indicators that need a warm-up period and are not laboratory-grade. Calibration drift adds to the error. **Payload and endurance:** in the documented example a 2200 mAh 3S battery gives only a few minutes of hover, and extra payload cuts this further, so coverage per flight is small. **Weather exposure:** rain, strong wind and low temperatures can damage the vehicle or weaken the batteries, and the sensors must be protected from rain and dust while still sampling open air. **Flight safety:** RC link loss, GPS loss, battery failure or a software fault all need a failsafe response, and LiPo batteries need careful handling. **Cloud dependence:** the cloud dashboard needs internet at the ground station, so the local display and logging should keep working without it. **Regulation:** drones in India fall under the Drone Rules, 2021, which cover classification, permitted airspace and operation [28]. These rules and the radio rules change over time and should be checked before every test campaign.

8. CONCLUSION

This paper has presented SkySense, a design for a LoRa-enabled weather-monitoring quadcopter built on an STM32F411 flight controller and an STM32F401 weather and telemetry node. The drone sends temperature, humidity, air-quality indicators, position, altitude, speed and battery voltage to an ESP32 ground station, which displays the data and forwards it to Google Cloud. The components, architecture, packet format, operating procedure and analytical estimates of airtime, path loss and hover endurance were described, and a plan for evaluating sensor accuracy, range, delivery rate, latency and endurance was defined. Related work has examined UAV environmental sensing and LoRa-UAV integration separately; SkySense proposes to combine them on low-cost hardware with a

transparent evaluation. The system is at the design stage and no performance has been demonstrated. Its limitations include the sensor exposure problem, short flight time and limited payload, the dependence of LoRa range on the environment, and regulatory restrictions.

9. FUTURE WORK

The next step is to build the prototype, run the tests in Section 6 and replace the reserved results section with real measurements. After that, the sensors can be calibrated in a controlled chamber, the mounting mast improved and different mounting positions compared. If the DHT22 turns out to be a limit, a faster or more accurate humidity sensor can replace it, and a larger battery or a more efficient airframe would help endurance. The system could also grow to several drones, connect the ground station to a LoRaWAN network [5], use richer dashboards, be compared with nearby weather stations and forecasts, and apply data analysis to the records it collects.

10. REFERENCES

[1] T. F. Villa, F. Gonzalez, B. Miljievic, Z. D. Ristovski and L. Morawska, "An overview of small unmanned aerial vehicles for air quality measurements: Present applications and future prospectives," Sensors, vol. 16, no. 7, Art. 1072, 2016.

[2] J. D. Jacob, P. B. Chilson, A. L. Houston and S. W. Smith, "Considerations for atmospheric measurements with small unmanned aircraft systems," Atmosphere, vol. 9, no. 7, Art. 252, 2018.

[3] A. Augustin, J. Yi, T. Clausen and W. M. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," Sensors, vol. 16, no. 9, Art. 1466, 2016.

[4] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen and J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," Int. J. Distributed Sensor Networks, vol. 13, no. 3, 2017.

[5] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne, "Understanding the limits of LoRaWAN," IEEE Commun. Mag., vol. 55, no. 9, pp. 34–40, 2017.

[6] N. Ya'acob, M. Zolkapli, J. Johari, A. L. Yusof, S. S. Sarnin and A. Z. Asmadinar, "UAV environment monitoring system," in Proc. 2017 Int. Conf. Electrical, Electronics and System Engineering (ICEESE), IEEE, 2017, pp. 105–109.

[7] M. Sajid, Y. J. Yang, G. B. Kim and K. H. Choi, "Remote monitoring of environment using multi-sensor wireless node installed on quad-copter drone," in Proc. 2016 IEEE Int. Symp. Robotics and Intelligent Sensors (IRIS), 2016, pp. 213–216.

[8] B. Potter, G. Valentino, L. Yates, T. Benzing and A. Salman, "Environmental monitoring using a drone-enabled wireless sensor network," in Proc. 2019 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2019, pp. 1–6.

[9] H. J. Jumaah, B. Kalantar, A. A. Halin, S. Mansor, N. Ueda and S. J. Jumaah, "Development of UAV-based PM2.5 monitoring system," Drones, vol. 5, no. 3, Art. 60, 2021.

[10] L. Barbieri et al., "Intercomparison of small unmanned aircraft system (sUAS) measurements for atmospheric science during the LAPSE-RATE campaign," Sensors, vol. 19, no. 9, Art. 2179, 2019.

[11] P. P. Neumann and M. Bartholmai, "Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit," Sensors and Actuators A: Physical, vol. 235, pp. 300–310, 2015.

[12] K. Meier, R. Hann, J. Skaloud and A. Garreau, "Wind estimation with multirotor UAVs," Atmosphere, vol. 13, no. 4, Art. 551, 2022.

[13] M. Bor, J. Vidler and U. Roedig, "LoRa for the Internet of Things," in Proc. Int. Conf. Embedded Wireless Systems and Networks (EWSN), 2016, pp. 361–366.

[14] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hänninen and M. Pettissalo, "On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology," in Proc. 14th Int. Conf. ITS

Telecommunications (ITST), 2015.

[15] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," ICT Express, vol. 5, no. 1, pp. 1–7, 2019.

[16] V. Delafontaine, F. Schiano, G. Cocco, A. Rusu and D. Floreano, "Drone-aided localization in LoRa IoT networks," in Proc. 2020 IEEE Int. Conf. Robotics and Automation (ICRA), 2020, doi: 10.1109/ICRA40945.2020.9196869.

[17] M. Behjati, A. B. Mohd Noh, H. A. H. Alobaidy, M. A. Zulkifley, R. Nordin and N. F. Abdullah, "LoRa communications as an enabler for Internet of Drones towards large-scale livestock monitoring in rural farms," Sensors, vol. 21, no. 15, Art. 5044, 2021, doi: 10.3390/s21155044.

[18] STMicroelectronics, "STM32F411xC/xE and STM32F401xB/xC datasheets," data sheets (Arm Cortex-M4 microcontrollers).

[19] InvenSense, "MPU-6000 and MPU-6050 product specification," data sheet.

[20] Honeywell, "HMC5883L 3-axis digital compass IC," data sheet.

[21] Bosch Sensortec, "BMP280 digital pressure sensor," Data sheet BST-BMP280-DS001.

[22] u-blox, "NEO-6 series u-blox 6 GPS modules," data sheet.

[23] Aosong Electronics, "AM2302 (DHT22) digital temperature and humidity sensor," data sheet.

[24] ScioSense, "ENS160 digital metal-oxide multi-gas sensor," data sheet.

[25] LoRa Alliance, "RP002 LoRaWAN regional parameters" (IN865–867 channel plan), technical specification. [Verify the current version and the applicable Indian rules before use.]

[26] Google Cloud, "Pub/Sub, Cloud Functions, Firestore and BigQuery documentation," [Online]. Available: https://cloud.google.com/docs

[27] Semtech Corp., "SX1276/77/78/79 – 137 MHz to 1020 MHz low power long range transceiver," data sheet; and "AN1200.13: LoRa modem designer's guide."

[28] Ministry of Civil Aviation, Government of India, "Drone Rules, 2021," notified August 2021 (as amended).