DOI 10.24412/1829-0450-fm-2026-1-51-60
УДК 519.688

Поступила: 09.03.2026.
Сдана на рецензию: 10.03.2026г.
Подписана к печати: 27.03.2026г.

A SOFTWARE-DEFINED RADIO APPROACH TO LoRA-BASED COMMUNICATION FOR UNMANNED AERIAL VEHICLES

***H. Tumanyan*[1], *L. Kirakosyan*[2]**

[1]*Yerevan State University*
[2]*Russian-Armenian University*
hovhannes.tumanyan1@edu.ysu.am, kirakosyan.lilia@student.rau.am

ABSTRACT

As Unmanned Aerial Vehicles (UAVs) continue to expand their operational reach, the critical need for highly resilient, long-distance data links has driven significant interest in Low Power Wide Area Network (LPWAN) technologies. LoRa modules are commonly used for UAV telemetry; however, they typically operate on a fixed frequency channel, which makes the communication link vulnerable to jamming, whereas SDR-based systems can mitigate this issue by changing the operating frequency. SDRs offer incredible flexibility, allowing us to dynamically change frequencies to avoid active RF jamming and create custom, wideband channels that standard chips cannot support. Using BladeRF 2.0 micro xA4 SDRs and custom GNU Radio processing, we built a continuous, Frequency Division Duplex (FDD) LoRa link. By artificially expanding the bandwidth to 3 MHz and placing the uplink and downlink on entirely separate frequency bands, the system effortlessly handles the data flow required by Ground Control Stations. This design preserves the ruggedness of LoRa's Chirp Spread Spectrum while eliminating traditional bottlenecks, easily supporting MAVLink v2 encryption overhead, and entirely removing turnaround delays. Real-world flight tests with a Raspberry Pi 5 companion computer confirmed the system's stability, maintaining a flawless, low-latency connection at 400 meters using only the SDR's unamplified native power

Keywords: Software-Defined Radio (SDR), Unmanned Aerial Vehicles (UAV), LoRa, Telemetry, BladeRF.

I. Introduction

The rapid proliferation and expanding operational scope of autonomous UAVs necessitate highly reliable, low-latency, and long-range communication architectures. For small-to-medium autonomous platforms operating on frameworks like ArduPilot [1] and PX4 [2], the MAVLink protocol [3] has emerged as the ubiquitous standard for bidirectional telemetry, parameter synchronization, and command-and-control (C2) data streams [1]. As UAV applications transition from localized line-of-sight flights to complex, autonomous Beyond Visual Line of Sight

(BVLOS) operations such as infrastructure inspection, search and rescue, and large-scale agricultural surveying the demands placed on aerial communication networks have scaled exponentially.

To achieve BVLOS operations, Long Range (LoRa) [4] modulation is frequently evaluated by the aerospace and robotics communities. LoRa relies on a proprietary physical layer utilising Chirp Spread Spectrum (CSS) [5]. By sweeping a continuous frequency chirp across a defined bandwidth, CSS spreads the narrowband signal over a wider spectrum, yielding a processing gain that permits signal demodulation near or below the ambient noise floor [5]. Consequently, LoRa exhibits exceptionally high resilience to multipath fading, and narrowband interference – all of which are critical factors in dynamic aerial environments. Furthermore, LoRa's low power requirements make it highly suitable for constrained aerial platforms where power budgets are strictly allocated.

This study presents an SDR-based architecture designed to circumvent these hardware limitations. By migrating the LoRa physical layer into software via GNU Radio [6] and deploying it on BladeRF 2.0 micro xA4 platforms [7], the system attains dynamic frequency agility to evade active RF jamming, and the operational bandwidth is artificially extended to an unprecedented 3 MHz. Furthermore, a continuous FDD link is established by utilizing highly asymmetric frequency bands. This approach provides a high-capacity, zero-turnaround-latency alternative to conventional LPWAN setups, successfully bridging the gap between high-throughput local telemetry and robust long-range CSS links. The system is empirically validated through active flight testing utilizing a Raspberry Pi 5 [8] companion computer to handle the dense baseband processing requirements.

II. Related Work and Telemetry Paradigms

To fully contextualise the necessity of an SDR-driven LoRa link, it is crucial to evaluate the existing concepts of UAV telemetry and the academic efforts to reverse-engineer LPWAN protocols. The industry currently relies on a fragmented ecosystem of RF solutions, none of which perfectly address the dual requirements of high bandwidth and long-range resilience.

A. Existing UAV Telemetry Architectures

Current UAV telemetry systems exhibit specific performance trade-offs when applied to the bidirectional demands of the MAVLink protocol:

- **Narrowband FSK Radios:** Operating primarily in the 433 MHz or 915 MHz ISM bands [9], traditional SiK-based [10] radios utilise basic Frequency Shift Keying (FSK) [12] and Frequency Hopping Spread Spectrum (FHSS) [9]. While theoretical air data rates can reach 250 kbps, practical long-range implementations often restrict this to 64 kbps or lower to maintain link integrity and sensitivity. Crucially, the TDM switching latency

heavily chokes the link during bidirectional communications, adding tens of milliseconds of latency to every request-response transaction [1], [10].

- **Broadband Line-of-Sight (Wi-Fi):** Standard 2.4 GHz or 5 GHz IEEE 802.11 telemetry [17] provides exceptional throughput (exceeding 10 Mbps) but experiences rapid free-space path loss and extreme susceptibility to urban RF interference. Lacking the signal processing gain of CSS, these systems are strictly limited to short-range, line-of-sight (LOS) operations, typically failing beyond a few hundred meters in cluttered environments.
- **Control-Centric LPWANs (ExpressLRS/Crossfire)** [12]**:** Modern open-source control links [12] utilise Semtech LoRa chips to achieve incredibly low latency and extended range. However, their physical layer scheduling prioritizes unidirectional RC control packets. The bi-directional telemetry bandwidth is intentionally constrained (often limited to < 2 kbps) to ensure primary C2 command delivery, rendering them unsuitable for full MAVLink parameter synchronization, map polygon transfers, or cryptographic key exchanges.
- **Cellular LTE/5G Telemetry:** While cellular networks offer theoretical infinite range [13], they are subject to variable latency, cellular tower handoff drops, and altitude-based signal degradation (as cellular antennas are down-tilted toward the ground). Furthermore, they require reliance on third-party infrastructure, making them unsuitable for remote, off-grid robotic operations.

B. SDR Implementations of LoRa

The foundation of software-defined LoRa communication was significantly advanced by the reverse engineering and implementation efforts of the academic community. Tapparel et al. [14] and the EPFL Telecommunications Circuits Laboratory [15] successfully developed gr-lora_sdr [16], a fully functional open-source GNU Radio implementation of the LoRa transceiver. Their extensive signal analysis provided the essential building blocks for generating, interleaving, and decoding LoRa modulated signals purely in software, allowing researchers to bypass the rigid constraints of commercial Semtech silicon [15]. While SDR platforms have been extensively utilised in UAV research for spectrum monitoring, anti-jamming applications, and custom waveform design, the application of SDRs to artificially expand the LoRa physical layer for high-throughput, full-duplex robotic telemetry remains largely unexplored in contemporary literature. Previous SDR implementations of LoRa have predominantly focused on IoT gateway emulation or security auditing at standard bandwidths (125 kHz). This paper proposes a UAV telemetry system that integrates the LoRa protocol with SDR, building upon existing implementations to enable flexible and robust telemetry communication.

III. Theoretical Background: MAVLink and CSS

To mathematically justify the necessity of an expanded 3 MHz bandwidth, the payload structure of the target protocol and the physical layer equations governing CSS must be analyzed.

A. MAVLink Protocol Structure and Demands

MAVLink is a highly efficient, lightweight serialisation protocol optimized for constrained radio links. The introduction of MAVLink v2 significantly increased packet overhead to support modern features. A standard MAVLink v2 frame consists of a 10-byte header (comprising a Magic Marker 0xFD, Payload Length, Incompatibility Flags, Compatibility Flags, Sequence number, System ID, Component ID, and a 24-bit Message ID), followed by up to 255 bytes of payload, a 2-byte CRC-16-CCITT, and an optional 13-byte cryptographic signature [3]. In standard flight scenarios, flight controllers stream critical state data (e.g., ATTITUDE, GLOBAL_POSITION_INT, VFR_HUD) at frequencies between 4 Hz and 50 Hz. This establishes a baseline continuous bandwidth requirement of approximately 3.5 to 4.5 kbytes/s (28 to 36 kbps).

While standard telemetry links support this baseline, the protocol's point-to-point mechanisms for mission uploads and parameter synchronization create massive burst-throughput requirements. During a standard GCS connection sequence, the request for the complete parameter tree (often containing over 1,200 configuration parameters) can instantaneously saturate narrowband links, causing buffer overflows and packet loss in half-duplex radios [1].

B. LoRa CSS Mathematical Foundations

LoRa utilizes frequency chirps to represent symbols [5]. The instantaneous frequency of a LoRa chirp $f(t)$ varies linearly over time T_{sym}:

$$f(t) = f_0 + \frac{BW}{T_{sym}} t$$

Where f_0 is the starting frequency, $\boldsymbol{BW}$ is the bandwidth, and T_{sym} is the symbol duration. The relationship between the symbol rate (R_s), Bandwidth ($\boldsymbol{BW}$), and Spreading Factor($\boldsymbol{SF}$) is defined mathematically as:

$$R_s = \frac{BW}{2^{SF}}$$

The resulting physical bit rate (R_b), incorporating the Forward Error Correction Coding Rate ($\boldsymbol{CR}$), is calculated as:

$$R_b = SF \cdot \frac{BW}{2^{SF}} \cdot CR$$

In commercial hardware, the maximum bandwidth is 500 kHz. At ***SF***=7 and ***CR***=4/5, the maximum theoretical bit rate of commercial LoRa is constrained to approximately 21.9 kbps [18]. This rate is mathematically insufficient to process uncompressed MAVLink streams alongside concurrent parameter downloads, firmly establishing the necessity for an SDR-driven approach for multi-megahertz bandwidth expansion.

IV. System Architecture and Hardware architecture

To address the latency and bandwidth limitations outlined above, we propose a comprehensive SDR-based hardware and software architecture designed for aerial integration. To mitigate the latency of conventional TDD, our system implements a Dual-Frequency FDD architecture using BladeRF SDRs. The GCS transmits on an uplink of 383.75 MHz and receives on a 637.9 MHz downlink, while the UAV operates inversely. The large frequency separation between the uplink and downlink allows the BladeRF's internal RF front-end filters to strongly suppress leakage from the local transmitter, enabling full-duplex operation with minimal mutual interference. The physical backbone relies on the BladeRF's Analog Devices AD9361 transceiver, which easily supports the arbitrary baseband sampling rates up to 61.44 MSPS required for this architecture [7].

On the UAV side, the BladeRF is paired with a Raspberry Pi 5 single-board computer [8], which serves as the MAVLink companion computer. Processing a 6 Msps, 3 MHz wide Chirp Spread Spectrum signal in GNU Radio requires substantial real-time CPU cycles for FFT[18] peak detection, phase tracking, and deinterleaving. A 6 Msps complex64 stream equates to a continuous 48 MB/s data transfer over the USB bus [18].

V. GNU Radio Software Implementation

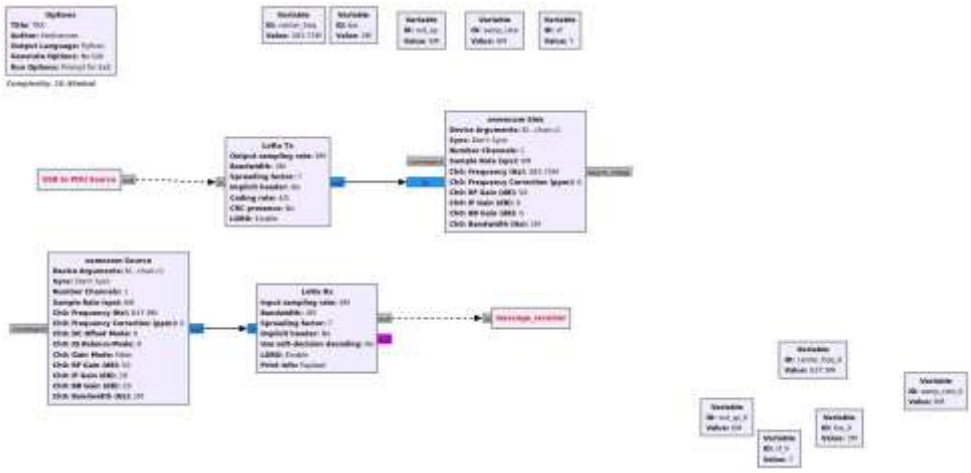
The software bridging the SDR hardware and the MAVLink protocol utilizes a highly optimized GNU Radio Companion (GRC) flowgraph built upon the modified gr-lora_sdr module.

A. Specific PHY Parameterisation

To attain the necessary throughput for MAVLink while maximising the processing gain of CSS, the variables within the GNU Radio flowgraph are defined meticulously:

- **Sample Rate:** 6 Msps. This baseband sampling rate interfaces the GNU Radio software with the Osmocom hardware blocks [20], safely satisfying the Nyquist-Shannon sampling [21] theorem for the 3 MHz signal width by maintaining an oversampling ratio of 2.
- **Bandwidth:** 3 MHz. By expanding the software LoRa bandwidth to 3 MHz, the chirp slope steepens drastically, proportionally increasing the theoretical maximum bit rate to accommodate GCS parameter downloads.

- **Spreading Factor:** SF7. As the lowest standard spreading factor, SF7 minimizes the time- on-air per symbol, prioritizing extremely low-latency delivery for joystick operations over extended range limits.
- **Coding Rate:** 4/5. Appending one parity bit per four data bits supplies adequate forward error correction (FEC) utilising Hamming (8,4) codes [11], suitable for aerial multipath environments without excessively bloating the packet overhead.
- **Low Data Rate Optimization (LDRO):** Enabled. Although conventionally utilized only for high-SF configurations where symbol times exceed 16.38 ms, enabling LDRO assists the SDR receiver algorithms in maintaining symbol timing stability. It effectively halves the number of bits encoded per symbol, increasing resilience against fractional frequency offsets
- and clock drifts between the independent, un-synchronized oscillators of the two BladeRF boards.
- **Implicit Header & CRC:** The LoRa physical header is configured to explicitly declare payload length and coding rate. The physical layer CRC is explicitly disabled, as the upper MAVLink protocol natively implements highly robust CRC-16-CCITT cryptographic hashes per packet, making PHY-layer CRCs redundant.



B. Transmit Chain Processing

The transmit chain begins with a custom GNU Radio Source block that parses the asynchronous serial or UDP stream originating from QGroundControl or the Flight Controller. The block buffers incoming bytes until a complete MAVLink packet is formed, constructing discrete GNU Radio Protocol Data Units (PDUs) [6] to prevent fragmentation across the air interface.

The PDUs are passed to the LoRa TX Block, which executes a rigorous sequence of physical layer transformations [5]:

1. **Data Whitening:** An XOR cipher utilizing a predefined linear-feedback shift register (LFSR) sequence randomizes the data to ensure uniform spectral energy distribution and prevent DC biasing in the RF amplifier chain [15].
2. **Hamming Encoding and Interleaving:** The 4/5 FEC is applied, and the resulting bits are diagonally interleaved across a defined matrix size to protect against burst errors caused by momentary RF fading or impulsive noise [15].
3. **Gray Indexing:** The interleaved bits are mapped to integer values representing the starting frequency offset of each individual chirp [15].
4. **Chirp Synthesis:** The baseband complex I/Q samples are mathematically generated, synthesizing a signal that sweeps from -1.5 MHz to 1.5 MHz relative to the center frequency at exactly 6 Msps.

C. Receive Chain Processing

The incoming 6 Msps baseband stream from the Osmocom Source is fed into the computationally intensive LoRa RX block. The process involves:

1. **Preamble Synchronization:** The receiver detects a defined sequence of identical up-chirps.
2. **De-Chirping:** The synchronised signal is multiplied by a locally generated down-chirp. This mathematical operation transforms the frequency-swept LoRa symbols into discrete, stationary frequency tones.
3. **FFT Peak Detection:** FFT [18] is applied to the de-chirped signal. The frequency bin containing the maximum spectral energy peak directly correlates to the Gray-indexed integer of the transmitted symbol [18].
4. **De-Interleaving and Decoding:** The integer is converted back to bits, de-interleaved across the matrix, de-whitened via the LFSR, and parsed back into a MAVLink PDU. A custom sink block finally routes the raw byte stream to the local serial/UDP socket connected to the GCS.

VI. Experimental Flight Validation

To explicitly validate the theoretical capabilities and stability of the proposed SDR-based LoRa architecture, comprehensive real-world flight tests were conducted in an open-field environment.

A. Experimental Setup and Hardware Integration

The UAV test platform was equipped with a standard open-source flight controller Pixhawk 6c [19] utilising the MAVLink v2 protocol [3]. The flight controller's telemetry UART port was bridged via a USB-to-Serial adapter to the Raspberry Pi 5 companion computer. The Raspberry Pi 5 executed the GNU Radio gr-lora_sdr flowgraph, generating the data stream and pushing it to the onboard BladeRF 2.0 micro SDR. The Ground Station node consisted of a laptop running

QGroundControl[2]. The laptop was tethered to a secondary BladeRF 2.0 micro via USB 3.0. Both SDR nodes utilised basic, unamplified omnidirectional antennas tuned appropriately for the 383.75 MHz and 637.9 MHz frequencies. The antennas were mounted vertically to ensure matching polarisation during level flight.

B. Flight Test Performance and Observations

During dynamic, multi-axis flight operations, including rapid altitude changes, banking maneuvers, and high-speed passes, the system successfully initialised and maintained a continuous, bidirectional FDD MAVLink connection between the UAV and QGroundControl up to a radial distance of approximately **400 meters**. Across the evaluated operational conditions, the implemented FDD link demonstrated stable communication performance. The GCS received continuous telemetry updates at a sustained rate without packet interruption. Furthermore, command-and-control messages and full parameter tree refresh operations were handled concurrently with minimal end-to-end latency.

C. Analysis of the 400-Meter Unamplified Range

The effective range of 400 meters observed during this unamplified test aligns precisely with the expected empirical performance profile of the utilised hardware. When establishing a wideband 3 MHz channel, the thermal noise floor ($10\log_{10}(BW)$) is intrinsically elevated by 7.78 dB compared to a narrowband 125 kHz signal. This flight experiment empirically confirms that artificially widening the LoRa bandwidth via SDR successfully processes the high-throughput MAVLink protocol without crashing, or dropping state packets. The Raspberry Pi 5 and the BladeRF successfully managed the immense data rates required for 3 MHz FDD CSS modulation in an active, vibrating airborne environment without triggering thermal throttling or USB bus exhaustion.

VII. Conclusion

This paper presents a UAV telemetry system implemented with SDR, using a customised *gr- lora_sdr* on BladeRF 2.0 micro xA4 hardware to expand LoRa's physical layer to 3 MHz, enabling a high-throughput, low-latency communication link suitable for continuous MAVLink data exchange. The 3 MHz bandwidth accommodates the cryptographic overhead of MAVLink v2 message signing, while the 254 MHz separation between uplink and downlink frequencies prevents local receiver desensitization. Empirical flight tests validated the system, sustaining a stable bidirectional MAVLink connection with QGroundControl at 400 meters using only the native 6.3 mW transmit power of the SDR interfaced with an airborne Raspberry Pi 5, combining the robustness and Doppler tolerance of CSS with the continuous throughput required for modern secure and autonomous UAV operations.

REFERENCES

1. ArduPilot Development Team, MAVLink Telemetry Bandwidth Requirements and High Latency Modes, ArduPilot Documentation, 2022.
2. Dronecode Project, QGroundControl User Guide: Telemetry and Parameter Synchronization. [Online]. Available: docs.qgroundcontrol.com.
3. Dronecode Project, Message Signing (Authentication) – MAVLink Guide, MAVLink Documentation, 2023.
4. *Seller O.and Sornin N.* Low Power Long Range Transceiver (LoRa), Semtech Application Note AN1200.22.
5. *Vangelista L.* Frequency Shift Chirp Modulation: The LoRa Modulation // IEEE Signal Processing Letters, vol. 24, no. 12, PP. 1818–1821, Dec. 2017.
6. GNU Radio Foundation, "GNU Radio Manual and C++ API Reference: Message Passing and PDUs, 2022.
7. Nuand LLC, BladeRF 2.0 Micro Architecture and Specifications, Nuand Documentation.
8. Raspberry Pi Ltd, "Raspberry Pi 5 Product Brief and BCM2712 SoC Architecture, 2023.
9. Federal Communications Commission (FCC), Part 15 Radio Frequency Devices, 47 CFR §15.247.
10. ArduPilot Development Team, SiK Telemetry Radio Specifications and TDM Latency, ArduPilot Wiki.
11. Proakis J. and Salehi M. Digital Communications, 5th ed. McGraw-Hill, 2008.
12. ExpressLRS Open Source Project, ExpressLRS Documentation: Telemetry Bandwidth Limits and Scheduling. [Online].
13. 3GPP (3rd Generation Partnership Project), Release 15/16: UAV Communications and Aerial UE Support.
14. *Tapparel J., Afisiadis O., Mayoraz P., Balzotti A. and Burg A.* gr-lora: A GNU Radio Implementation of the LoRa PHY Layer, IEEE Transactions on Communications, 2020.
15. Telecommunications Circuits Laboratory (TCL), Reverse Engineering of the LoRa PHY Layer, EPFL, Tech. Rep., Feb. 2020. Available: https://www.epbgdfl.ch/labs/tcl/wp- content/uploads/2020/02/Reverse_Eng_Report.pdf
16. *Tapparel J.* gr-lora_sdr repository, GitHub [Online]. Available: https://github.com/tapparelj/gr-lora_sdr.
17. The Things Network (TTN), LoRa Airtime Calculator [Online]. Available: https://www.thethingsnetwork.org/airtime-calculator.
18. Oppenheim A. and Schafer R. Discrete-Time Signal Processing, 3rd ed. Pearson, 2010.
19. Holybro, Pixhawk 6C Flight Controller Technical Specifications and Architecture, 2022.
20. Osmocom (Open Source Mobile Communications), gr-osmosdr: GNU Radio block for interfacing with SDR hardware. [Online].
21. *Shannon C.* A Mathematical Theory of Communication, originally published in Bell System Technical Journal, Vol. 27. PP. 379–423 and 623–656, July and October 1948.

МЕТОД ОРГАНИЗАЦИИ СВЯЗИ LoRA В БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТАХ НА ОСНОВЕ ПРОГРАММНО-ОПРЕДЕЛЯЕМОГО РАДИО

О. Туманян[1], Л. Киракосян[2]
[1] Ереванский государственный университет
[2]Российско-Армянский университет
hovhannes.tumanyan1@edu.ysu.am, kirakosyan.lilia@student.rau.am

АННОТАЦИЯ

В данной статье речь идет о том, что, поскольку радиус действия беспилотных летательных аппаратов (БПЛА) продолжает расширяться, острая необходимость в высоконадежных каналах передачи данных на большие расстояния вызывает значительный интерес в области энергоэффективных сетей дальнего радиуса действия (LPWAN). Модули LoRa часто используются для телеметрии БПЛА; однако обычно они работают на фиксированном частотном канале, что делает канал связи уязвимым к радиоэлектронному подавлению (глушению). В то же время системы на базе программно-определяемого радио (SDR) могут решить эту проблему за счет изменения рабочей частоты. Технология SDR предлагает гибкость, позволяя динамически менять частоты для избежания активных радиопомех и создавать настраиваемые широкополосные каналы, которые стандартные чипы поддерживать не могут. Используя SDR BladeRF 2.0 micro xA4 и специализированную обработку сигналов в GNU Radio, мы создали непрерывный канал связи LoRa с частотным дуплексным разделением (FDD). За счет искусственного расширения полосы пропускания до 3 МГц и размещения восходящего (uplink) и нисходящего (downlink) каналов в совершенно разных частотных диапазонах, система без труда справляется с потоком данных, необходимым для наземных станций упавления. Данная архитектура сохраняет надежность метода расширения спектра на основе линейной частотной модуляции (CSS), используемого в LoRa, при этом устраняет традиционные узкие места системы. Она легко справляется с дополнительной вычислительной нагрузкой, связанной с шифрованием MAVLink v2, и полностью исключает задержки, возникающие при переключении направления передачи.

Ключевые слова: Программно-определяемое радио (SDR), беспилотные летательные аппараты (БПЛА), LoRa, телеметрия, BladeRF.