NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS

Tikhonov Moscow Institute of Electronics and Mathematics

Resiliently Synchronizing SFN Networks: Combining Precise Time Signals from GPS and Longwave Radio Stations

by

Maxim Emelyanenko

Graduate Qualification Work – Bachelor Thesis

Bachelor of Science in Information Security (10.03.01)

Authored by: Maxim Emelyanenko

Department of Information Security

Group BIB201

Certified by: Yaroslav Mesheryakov

Associate Professor, Thesis Supervisor

Moscow 2024

Contents

Annotation

This thesis introduces a novel hybrid approach to enhance the resilience of time synchronization within Single Frequency Networks (SFN), crucial for digital broadcasting systems. The traditional reliance on Global Navigation Satellite Systems (GNSS) exposes SFNs to vulnerabilities including geophysical phenomena like ionospheric variations due to solar activity fluctuations, and deliberate disruptions such as spoofing and jamming attacks.

To address these vulnerabilities, this work integrates terrestrial longwave radio transmitters into the time transfer framework as a supplementary precise time signal source. By leveraging the stable and reliable nature of longwave signals, a dual-check mechanism is implemented, enhancing the SFN's reliability significantly. This integrated system provides a robust fallback, ensuring continuous operation even when GNSS services are compromised.

Experimental validations confirm the effectiveness of this approach, demonstrating a reduction in time transmission errors and an enhanced resilience against spoofing and jamming within SFNs.

The findings lay a foundation for future research and development in robust hybrid time synchronization technologies, offering a scalable and effective solution to safeguard critical infrastructure against single-source dependency risks.

Аннотация

В работе представлен новый метод синхронизации времени в одночастотных сетях (SFN) для повышения устойчивости систем цифрового вещания. Описанный метод устраняет критические уязвимости, присущие традиционно используемым для синхронизации времени глобальным навигационным спутниковым системам (GNSS). К числу факторов, к которым уязвимы сети SFN, работающие на основе сигналов GNSS, относятся как геофизические явления, например, ионосферные изменения в результате перепадов солнечной активности, так и преднамеренные воздействия.

Решение описанной проблемы осущствлено путем интеграции в службу точного времени наземных длинноволновых радиопередатчиков в качестве дополнительного источника сигналов точного времени. В работе предложена более устойчивая, по сравнению с имеющейся, структура передачи параметров точного времени. Для этого использована двухуровневая проверка, значительно повышающая надежность работы SFN. Такая интеграция наземных длинноволновых радиопередатчиков в службу точного времени

использует стабильный и надежный характер длинноволновых сигналов для создания отказоустойчивого механизма, обеспечивающего непрерывную работу в условиях потенциальных отказов работы GNSS.

Экспериментальная проверка подтвердила эффективность этого метода: в результате его применения снизилось количество ошибок передачи времени и усилилась защита сети SFN от спуфинга и атак глушения.

Результаты создают основу для будущих исследований и разработок в области надежной гибридной системы синхронизации точного времени. Предложено масштабируемое и эффективное решение для защиты критически важных инфраструктур против внешних факторов.

Keywords: Time Transfer, Time Cross-checking, GNSS, SFN, Spoofing

1 Introduction

The rapid evolution of single frequency networks (SFNs) necessitates heightened precision in timing synchronization to accommodate the increasing data throughput demands. This is particularly critical with the advent of advanced broadcasting standards such as Digital Video Broadcasting – Terrestrial (DVB-T) and its enhancement in DVB-T2. While the focus is on DVB-T2 due to its extensive use in Russia [1], the findings and methodologies proposed by this paper are applicable to other digital broadcasting standards such as ATSC, ISDB-T, and DTMB, which are used in different regions around the world. These standards generally utilize an orthogonal frequency-division multiplexing (OFDM) modulation, which imposes stringent synchronization requirements to prevent inter-symbol interference and maximize bandwidth efficiency [2–5].

Historically, Global Navigation Satellite Systems (GNSS) such as GPS and GLONASS have been instrumental in achieving high precision time synchronization across the extensive and geographically dispersed infrastructure of SFNs. However, GNSS systems have numerous vulnerabilities, including susceptibility to ephemeris errors, ionospheric delays, solar activities, and security threats like jamming and spoofing. These vulnerabilities are detailed in Section 1.2.3, highlighting the critical need for alternative or complementary systems to enhance reliability and integrity in SFN synchronization.

An alternative approach, inspired by the eLoran system, is explored in this study. eLoran is a terrestrial-based navigation system developed to provide a reliable backup for GNSS, ensuring resilience against a single point of failure in critical synchronization applications [6]. Further discussions on eLoran's applicability and implementation are presented in Section 1.2.5.

Building upon the eLoran principles, this thesis examines the feasibility of utilizing terrestrial radio stations, specifically those managed by the Russian State Service for Time, Frequency, and Earth Rotation Parameters, alongside stations in Germany, the United Kingdom, and Japan. These stations are renowned for their picosecond-level precision in time signal broadcasting and represent a promising alternative for enhancing synchronization accuracy and resilience in SFNs [7].

1.1 Objective

This thesis aims to develop and validate resilient time transfer methodologies that enhance the resilience of synchronization in Single Frequency Networks (SFN), crucial for reliable digital

broadcasting. Given the inherent vulnerabilities of Global Navigation Satellite Systems (GNSS) such as GPS and GLONASS — ranging from ionospheric variations to intentional disruptions like spoofing and jamming — this work proposes the integration of terrestrial longwave radio signals as a supplementary precision time source. This hybrid synchronization approach seeks to mitigate the risks associated with GNSS dependencies by ensuring a continuous operation through alternative time signals, thereby fostering enhanced resilience and reliability of SFN infrastructures worldwide. The ultimate goal is to demonstrate that the integration of these terrestrial signals can significantly reduce synchronization errors and safeguard against the potential failure of GNSS, thus providing a more stable and secure broadcasting environment.

This paper is organized in the following way: Section 1.2 provides a theoretical background on the working principles and vulnerabilities of GNSS and the synchronization requirements of SFNs, Section 1.3 details the methods employed in the experimental evaluation, Section 2 discusses the integration and testing of time transfer techniques, and Section 3 explores resiliency improvements to the time transfer techniques against GNSS jamming and spoofing scenarios. Finally, Section 4 concludes with a summary of the findings and implications for future research.

1.2 Theoretical Background

1.2.1 Time Transfer Approaches

Time synchronization in Single Frequency Networks (SFNs) can be primarily categorized into the one-way and two-way synchronization techniques, each having its unique application and accuracy profile suitable for different levels of system demands.

One-Way Time Transfer: In the one-way time transfer, a time signal is sent from a source to a receiver without requiring any information to be sent back to the source. This method is simpler and often used where high precision is not paramount. However, its accuracy can be affected by asymmetries in signal path delays, which are not typically compensated for in the one-way method. This approach is often employed in systems where the absolute precision of synchronization is less critical but where the simplicity and cost-effectiveness of the setup are prioritized.

Two-Way Time Transfer: Two-way time transfer involves sending a time signal from a source to a receiver, which then sends a return signal back to the source. The time of flight for the signals is measured and used to adjust and correct any delay, thus providing a higher level of accuracy compared to the one-way transfer. This method compensates for path delay

variations and is crucial in applications requiring high precision, such as in telecommunications and high-frequency trading systems [8].

Stratum Levels in Network Synchronization: In network time synchronization, stratum levels define the hierarchy of sources providing the timing. A stratum 1 time server is directly connected to a primary time source, such as a GPS or a radio clock. Lower stratum numbers represent closer proximity to the primary time standard, and thus generally higher accuracy. Devices downstream, such as stratum 2 servers, synchronize to stratum 1 servers, adding a degree of latency and potential error as the stratum number increases. This hierarchical setup ensures that even if lower stratum devices have less precision, the system as a whole can maintain a robust and scalable time synchronization infrastructure.

1.2.2 Working Principle of GNSS

Global Navigation Satellite Systems (GNSS) operate on the principle of trilateration, utilizing signals from multiple satellites to pinpoint a receiver's location. Each satellite transmits messages that include the satellite's position and the time the message was sent. Receivers calculate the distance to each satellite by measuring the time delay of the received signals, which, when combined with the positions of at least four satellites, allow for accurate determination of the receiver's three-dimensional position and time.

GNSS signals are encoded using a technique known as spread spectrum. Each satellite broadcasts a unique pseudorandom noise (PRN) code, which helps to distinguish between signals from different satellites and assists in precise timing measurements necessary for trilateration.

Real-Time Kinematic (RTK): RTK is a technique used to enhance the precision of position data derived from satellite-based positioning systems. It uses measurements of the phase of the satellite's signal carrier wave in addition to the information content of the signal and relies on a single reference station to provide real-time corrections, achieving precision down to centimeters [9, 10].

Differential Carrier Phase Time Transfer: This method utilizes the carrier phase of GNSS signals for high-precision time transfer. By comparing the phase difference of the carrier signals received from a satellite at two or more locations (typically a reference station and a remote station), the relative timing errors can be minimized significantly. This technique is especially useful in applications requiring extremely high accuracy and stability in time synchronization [11].

1.2.3 GNSS Vulnerabilities

Global Navigation Satellite Systems (GNSS) face a spectrum of vulnerabilities that impact the accuracy and reliability of synchronization in Single Frequency Networks (SFN). Technical errors such as ephemeris inaccuracies and time system errors can lead to significant deviations in positioning and timing signals [12]. Additionally, natural phenomena like ionospheric and tropospheric delays, compounded by solar activities, cause signal distortion and interference [13].

Beyond natural and technical issues, intentional interferences such as jamming and spoofing present severe threats. Jamming disrupts GNSS reception, while spoofing misleads receivers with counterfeit signals. Notably, incidents in South Korea due to GPS jamming by North Korea exemplify the growing risk and sophistication of these attacks [14]. The intrinsic low power of GNSS signals heightens their susceptibility to such disruptions [15].

The dependency of SFNs on accurate time synchronization makes them particularly prone to the adverse effects of GNSS vulnerabilities. These issues underline the urgent need for developing and integrating alternative synchronization methods to ensure the security and reliability of critical infrastructures like digital broadcasting services. The exploration of robust countermeasures is crucial to mitigate the impact of GNSS vulnerabilities on SFN operations, as emphasized by recent studies on the consequences of GNSS disruptions [16].

1.2.4 SFN Synchronization

Single Frequency Networks (SFNs) necessitate precise synchronization of broadcast signals across multiple transmitters to ensure seamless delivery and prevent signal interference, crucial for maintaining high network efficiency and signal quality. Traditionally, SFNs have relied on GNSS systems like GPS and GLONASS for synchronization. Although effective, this method introduces vulnerability due to its single point of failure nature, highlighting the need for more resilient synchronization techniques.

An alternative, the Integrated Time Transfer via the Nimbra transport platform, embeds time signals within the data stream transmitted across physical networks. Each node in the network utilizes these embedded signals to synchronize its operations. While this method enhances resilience against GNSS vulnerabilities such as jamming and spoofing, it depends heavily on the network infrastructure's integrity, which can be compromised by high latency or inconsistent network designs [17, 18].

The shift towards such alternative synchronization methods introduces significant technical

and infrastructural challenges. Establishing a terrestrial time signal network, for instance, demands considerable investment and meticulous planning to achieve the necessary coverage, precision, and reliability. Moreover, integrating multiple synchronization sources adds complexity to synchronization algorithms and system designs, necessitating ongoing research and development to ensure effective implementation across diverse network configurations.

1.2.5 eLoran

eLoran, an enhancement of the original Long Range Navigation (Loran) system, provides a terrestrial-based navigational and timing solution that can complement or replace GNSS systems like GPS in certain scenarios. Utilizing a network of high-powered, low-frequency radio transmitters, eLoran offers broad coverage and can penetrate environments where GNSS signals are weak or blocked. Its resilience against interference and spoofing attacks enhances its suitability for critical infrastructure roles [6].

The low frequency of eLoran signals ensures reliable reception under adverse conditions, positioning eLoran as a dependable option for secure time and frequency dissemination. Designed to address the vulnerabilities of satellite-based systems, particularly under conditions of natural or human-made disruptions, eLoran supports continued service availability even when GNSS signals fail.

Global efforts to expand eLoran infrastructure indicate a strategic commitment to strengthen communication, navigation, and timing services against emerging threats to GNSS reliability. The incorporation of eLoran into technological frameworks aims to enhance operational continuity and security, striving for a GNSS-independent, robust global positioning network.

1.2.6 RBU Time Signal Radio Station

The RBU longwave time signal and standard-frequency radio station is operated by the Russian Television and Radio Broadcasting Network (RTRN) and overseen by the All-Russian Scientific Research Institute for Physical-Engineering and Radiotechnical Metrology (VNIIFTRI). This facility offers a stable and accurate time signal source, boasting a relative frequency error that does not exceed $2 \cdot 10^{-12}$ Hz [19].

The RBU signal employs a DXXXW modulation scheme to disseminate time and frequency unit measurements. The transmission consists of a sinusoidal carrier wave at a frequency of $66666.\overline{6}$ Hz, which is interrupted every 100 ms for 5 ms. Following this brief interruption, the carrier undergoes narrowband phase modulation for 80 ms using subcarrier frequencies

of 100 Hz or 312.5 Hz, each with a modulation index of 0.698. The 312.5 Hz subcarrier represents binary ones and marks the start of each second and minute in the transmitted time scale, while the 100 Hz subcarrier marks binary zeros and fills remaining intervals free of data transmission. A timing layout fragment of the DXXXW signal is depicted in Figure 1.

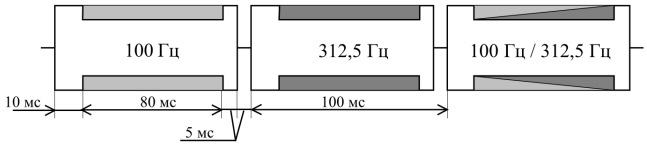


Figure 1: DXXXW signal fragment.

The RBU time code transmits detailed time-related information, crucial for synchronization and timekeeping across vast networks. The data structure includes:

1. **Time Code Elements:** year, month, day of the week, day of the month, hour, and minute, encoded in a binary-coded decimal (BCD) format, allowing receivers to decode the current date and time precisely.
2. **DUT1 and dUT1 Corrections** that provide adjustments to UT1 relative to UTC, catering for Earth's rotational irregularities.
3. **Second and Data bits** which include unary encoding to indicate adjustments such as DUT1 values, enhancing the accuracy of timekeeping.

Second markers are distinguished by a preceding 0.1-second interval modulated at 312.5 Hz, while minute markers are identified by two consecutive 0.1-second intervals prior to the second marker, each modulated at 312.5 Hz [19]. These modulation patterns allow accurate determination of the second and minute boundaries in Coordinated Universal Time (UTC).

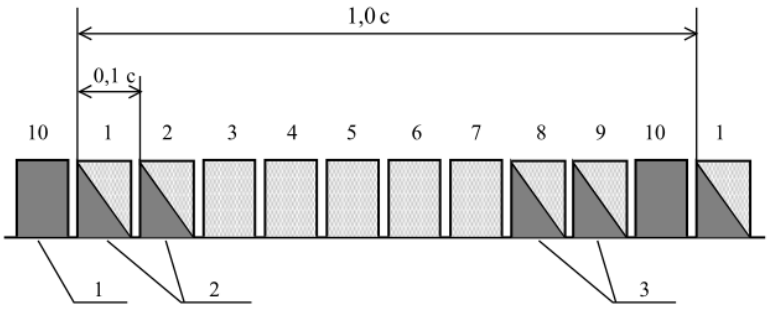


Figure 2: RBU signal payload structure:

1 – Start-of-the-Second marker; 2 – Information payload;

3 – Start-of-the-Minute marker.

The RBU signal comprises two critical features aiding in accurate UTC time phase determination:

1. The carrier wave undergoes brief interruptions for 5 ms every 100 ms, denoted with empty space between bits of RBU signal payload on Figure 2. These interruptions provide precise phase markers with ambiguity of 10.
2. Additionally, a 312.5 Hz carrier modulation marks the beginning of each second, essential for resolving the phase ambiguity associated with the pulse per second (PPS) signal. These modulations are denoted with dark grey color rectangles indexed with number 10 on Figure 2, while the light grey color illustrates 100 Hz modulations used to encode binary data which is irrelevant to the proposed approach.

Notably, selection of the RBU radio station serves exclusively as a practical example within this context. Other time signal stations, such as DCF-77 (Germany), WWWB (USA), and MSF (UK), are also operational globally. The optimal longwave time signal source for a specific application would be determined through a comprehensive evaluation that considers factors such as signal strength, propagation characteristics within the target deployment region, and accessibility. In essence, any longwave time signal station located within suitable geographic proximity to the receiver could be interchangeably employed within the proposed framework.

1.2.7 Longwave Time Signal Range Limitations

Investigations into the range limitations of longwave time signals are essential for understanding their effective use in various applications. Longwave (LW) signals, classified to encapsulate frequencies between 30 kHz and 300 kHz, are particularly suited for robust, long-distance transmission due to their ability to follow the Earth's curvature via ground wave propagation. This characteristic is instrumental in achieving extensive coverage, potentially spanning national boundaries.

Conversely, despite these capabilities, the range of longwave signals can still be constrained by several factors. Notably, the ionospheric conditions, which play a crucial role in radio wave propagation, introduce variability in signal reach. Solar activities, such as flares and sunspots, can significantly affect ionospheric density and thus influence the propagation paths and stability of longwave signals.

Furthermore, the topography and the electrical conductivity of the terrain over which the signal travels can also impact its effective range. Areas with higher conductivity, such as bodies

of water or wetlands, can enhance ground wave propagation, while rocky or mountainous regions may present challenges due to signal absorption and scattering [20, 21].

Empirical studies, such as those referenced in ITU-R recommendations, suggest that longwave time signals can reliably cover distances up to 2000 km under optimal conditions. However, these theoretical ranges often require validation through dedicated experimental research to confirm practical usability across such expanses, especially in diverse geographic settings like Russia's vast territories [7].

In conclusion, while longwave time signals have inherent advantages for wide-area coverage due to their propagation characteristics, their effective range is not absolute and is influenced by a complex interplay of atmospheric, topographical, and solar conditions. As such, continual monitoring and research are necessary to optimize their application in global time and frequency dissemination.

1.2.8 Software Defined Radio

Software Defined Radio (SDR) represents a transformative approach to radio design, enabling radios to be extensively configured and controlled via software. This technology allows for the replacement of traditional hardware components, such as mixers and filters, with software processing. This shift not only reduces physical constraints but also enhances the flexibility and adaptability of radio systems to new frequencies and functionalities through software updates.

A fundamental aspect of SDR technology is its reliance on in-phase (I) and quadrature (Q) components to digitally represent radio frequency signals. These components are essential for capturing all the information of a signal, including its amplitude and phase. The I and Q components represent the signal on orthogonal axes, with the I component aligned with the cosine wave and the Q component aligned with the sine wave. This method offers several significant advantages:

- **Complete Signal Representation:** I/Q data allows for the complete representation of a radio signal's phase and amplitude, which is crucial for accurately demodulating complex algorithms used in modern communication systems.
- **Improved Signal Integrity:** By processing signals in both the in-phase and quadrature dimensions, SDRs can more effectively distinguish between overlapping signals and reduce the likelihood of interference, leading to clearer and more reliable signal reception.
- **Flexibility in Signal Manipulation:** I/Q processing facilitates various signal manip-

ulations such as shifting, modulation, and demodulation directly in the digital domain, offering extensive control over signal characteristics with high precision.

One of the critical applications of SDR technology is in GNSS spoofing, an area of growing concern due to the potential security risks it poses. SDRs can generate and transmit radio signals that mimic GNSS signals, making them invaluable tools for both perpetrating and defending against spoofing attacks. This capability allows security professionals and researchers to simulate various spoofing scenarios to develop and test countermeasures that improve the resilience of GNSS receivers. Effective anti-spoofing techniques, leveraging the nuanced control offered by SDR, are vital for ensuring the security and reliability of GNSS-dependent systems [22].

1.3 Methods

Literature Review and Theoretical Framework

The research commenced with a literature review focusing on the current advancements and identified gaps in time synchronization technologies. This review concentrated on the DVB-T and DVB-T2 standards, Precision Time Protocol (PTP), and GNSS systems. The theoretical insights gained were pivotal in guiding the experimental approach and highlighting the need for robust synchronization solutions in Single Frequency Networks (SFNs).

Figure 3: Active ferrite rod antenna with an integrated LNA.

Antenna Design and Signal Reception

Critical to the experiment was the design and optimization of antennas for the robust reception of longwave time signals. The setup included an active ferrite rod antenna, seen on Figure 3. It is integrated with a low-noise amplifier (LNA) to enhance the reception quality of the longwave signals, essential for maintaining synchronization across SFN networks.

Experimental Setup

The core of the experimental setup involved a hardware platform centered around the STM32 microcontroller, known for its robust performance in real-time applications. This platform was integral as a primary timing source to synchronize SFN base stations, depicted in Figure 4.

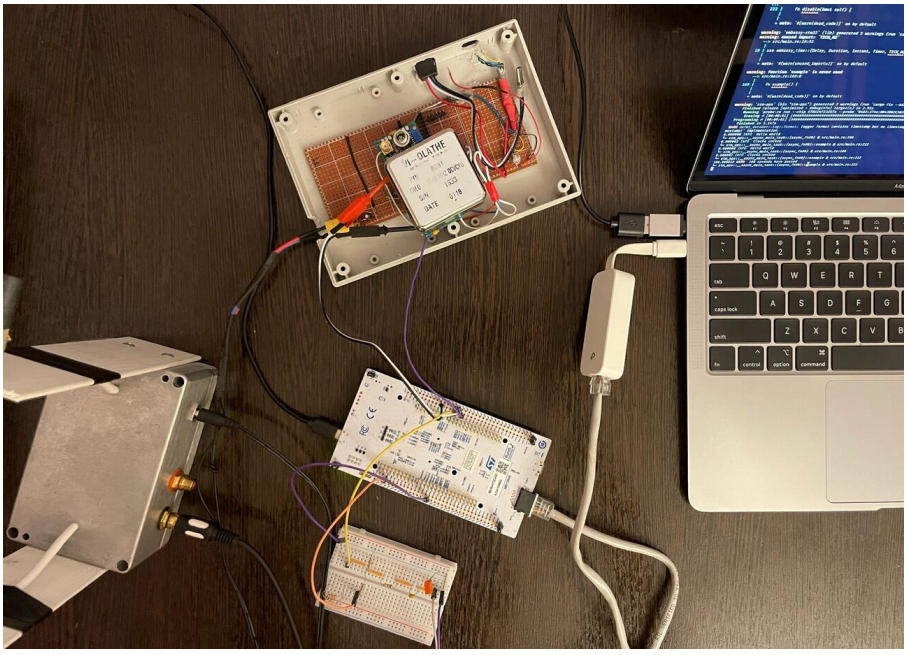


Figure 4: Experimental setup showing key components: the STM32-NUCLEO development board at the center, the OCVCXO crystal at the top, the ferrite rod antenna on the left, and a workstation equipped with DSP algorithms on the right.

Digital Signal Processing and Algorithm Development

Concurrently, novel Digital Signal Processing (DSP) algorithms were developed to facilitate the extraction and interpretation of time signals from the noise-laden data received by the antennas. These algorithms were crucial for ensuring accurate time synchronization, particularly in challenging signal environments. Furthermore, algorithms were developed to detect discrepancies in GNSS and terrestrial-based time signals, enabling an internal switching mechanism that maintains continuous synchronization of the SFN base stations.

System Integration and Testing

The system's accuracy and reliability were further bolstered by integrating oven-compensated voltage-controlled oscillators (OCVCXO). Known for their stable signal output over time, these devices were instrumental in maintaining synchronization precision. The entire setup's

effectiveness was rigorously validated through a series of empirical tests conducted under varied natural conditions and at different distances from the radio time signal source.

Statistical Analysis of Phase Errors

A key part of ensuring the reliability of time synchronization was the statistical analysis of phase measurement errors. This involved verifying the normality of the distribution of phase errors, which is critical for applying certain statistical techniques. For precise measurements, the study employed standard statistical formulas to determine the required sample size based on the desired Standard Error of the Mean (SEM) to ensure the accuracy of conclusions drawn from the data analysis.

GNSS Spoofing Resilience Evaluation

To assess the time transfer framework's resilience against GNSS spoofing, an SDR transmitter was used. The Great Scott Gadgets HackRF One, shown in Figure 5, was configured to simulate GNSS spoofing attacks to evaluate the system's defensive mechanisms.

This methodological approach underscores our dedication to advancing SFN synchronization technologies by addressing each component of the synchronization process, from signal reception and processing to hardware innovation and algorithmic development.

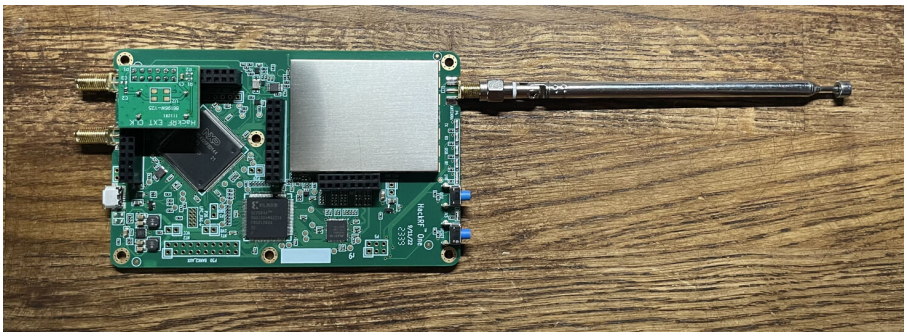


Figure 5: Great Scott Gadgets HackRF One with an RF shield.

2 Time Transfer

To design and evaluate resilient time transfer algorithms, it is imperative to develop a robust and generalized time and frequency system. This system must synchronize its time and frequency to Coordinated Universal Time (UTC) using multiple GNSS constellations while incorporating a sufficiently stable clock source to compensate for potential transient GNSS outages.

2.1 Experimental Setup for PPS Synchronization

In our experimental setup, the u-blox MAX-M10S GNSS receiver module serves as the primary source of GNSS-synchronized Pulse-Per-Second (PPS) signals. However, its internal Temperature-Compensated Crystal Oscillator (TCXO) lacks the long-term stability required to maintain accurate timekeeping in GNSS-deprived conditions [23, 24].

To address the stability issue during periods of GNSS signal degradation (e.g., urban canyons, tunnels, jamming, or spoofing), this paper proposes a designed generalized setup featuring an intermediate PPS source with enhanced long-term stability. Specifically, we incorporate a 10 MHz Oven-Controlled Voltage-Controlled Crystal Oscillator (OCVCXO) with a frequency tolerance of 0.1 ppm.

The intermediate stratum-1 PPS source is provided by an STM32 NUCLEO-H723ZG development kit equipped with an Arm® Cortex®-M7 CPU [25]. This development kit is synchronized to the stable PPS signal generated by the u-blox MAX-M10S module to correct the reference 10 MHz signal from the OCVCXO. The accurate and reliable PPS signal locally-derived by STM32 from the OCVCXO enables uninterrupted operation of DVB-T2 SFN transmitters, thereby mitigating service degradation during GNSS signal outages.

2.1.1 GNSS Module Configuration

The u-blox MAX-M10S GNSS receiver module is configured via a UART interface using a proprietary UBX datagram protocol. The protocol features commands such as `UBX-CFG-VALGET` and `UBX-CFG-VALSET` for reading and modifying configuration parameters respectively [26].

To maximize accuracy at the expense of convergence speed, the receiver is configured by modifying the following keys.

1. `CFG-NAVSPG-DYNMODEL = 0b0010`. To maximize accuracy, the receiver is configured

in static positioning mode, given its stationary nature in this application.

2. `CFG-SIGNAL-GLO_ENA = 0b1` and `CFG-SIGNAL-BDS_ENA = 0b1`.
   For improved precision and faster convergence, we enable GLONASS and BeiDou constellations, which are disabled by default.
3. `CFG-RATE-MEAS = 0b1111101000`. A slower position update rate enhances stability and accuracy. We configure the measurement rate of 0.1 Hz as it is not crucial for the scenario of time transfer.
4. `CFG-RATE-TIMEREF = 0b000`. Different GNSS constellations utilize distinct time systems. This configuration allows us to align PPS measurements to Coordinated Universal Time (UTC) [24]. This way multiple GNSS receivers from different vendors in an SFN will consistently output universally synchronized PPS.

To monitor the GNSS receiver's operational status, the `UBX-MON-COMMS` key can be configured to output a stream of NMEA, RTCM, SPARTN, and UBX datagrams. These datagrams contain comprehensive information about location, speed, time, signal spectrum, tracked satellites, satellites in view, carrier-to-noise density ratios (C/N_0), Position Dilution of Precision (PDOP), and other metrics [26]. The data can be visualized using the PyGPSClient GUI application, as shown in Figure 6 [27].

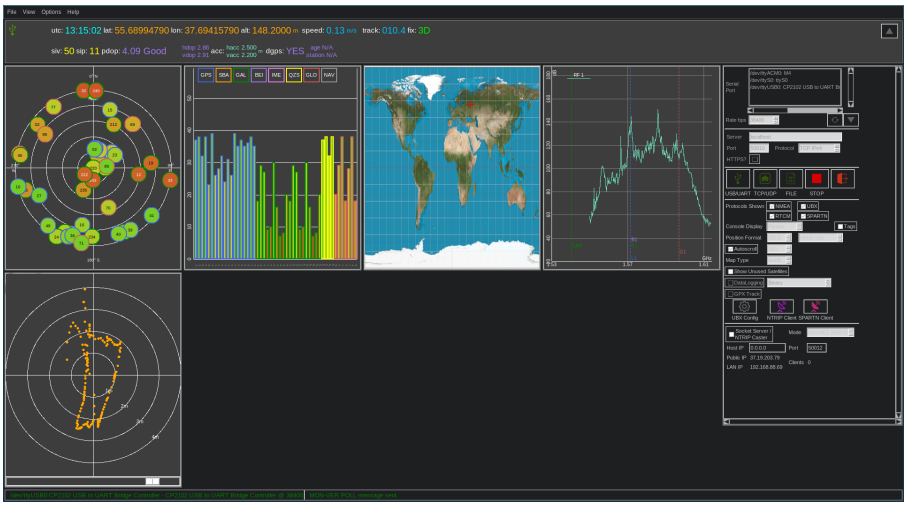


Figure 6: PyGPSClient displaying the operational status of the u-blox MAX-M10S GNSS receiver.

After obtaining a position fix, the output PPS signal from the GNSS receiver can be verified using a logic analyzer. The signal should exhibit a period of 1 Hz with a duty cycle of 10%. Figure 7 shows the PPS signal as visualized in DSView, a digital oscilloscope software [28]. An error of 1.7 microseconds can be observed in period calculation, as caused by sample rate.

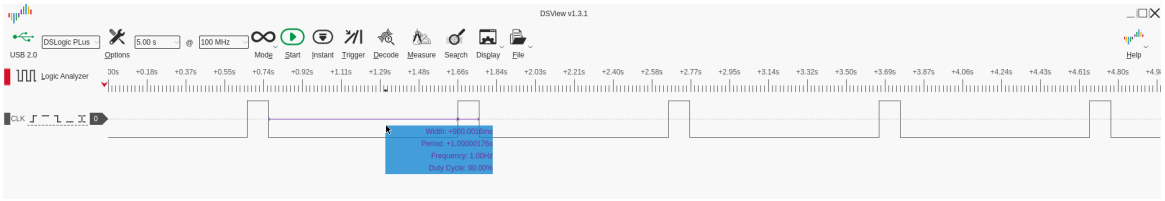


Figure 7: Representation of the PPS signal in DSView digital oscilloscope software.

2.1.2 PPS Registration with STM32

To achieve highly consistent latency between the PPS pulse generation on the GNSS receiver module and its registration on the STM32 microcontroller, careful hardware design is essential. One effective approach is to route the PPS signal to a General Purpose Input/Output (GPIO) pin configured as an input with a pull-down resistor. This pin can then be assigned to the External Interrupt/Event (EXTI) controller [29]. By setting the EXTI controller to trigger on the rising edge of the PPS signal, a hardware interrupt is dispatched with a configurable priority, allowing precise PPS pulse registration on the STM32 [30].

2.2 OCVCXO Frequency Correction

Crystal oscillator accuracy and precision are susceptible to variations in current, temperature fluctuations, vibration-induced noise, physical aging, ambient pressure, humidity, and electromagnetic fields [31]. To compensate for these factors, Voltage-Controlled Crystal Oscillators (VCXOs) employ a Phase-Locked Loop (PLL) circuits with frequency multipiers. By varying the voltage supplied to the PLL pin, the oscillator's frequency can be adjusted. The STM32H7 MCU provides multiple 12-bit Digital-to-Analog Converters (DACs) that allow precise voltage control via DAC peripheral hardware registers [29].

To accurately measure the error of the OCVCXO, the number of clock cycles it generates between two consecutive PPS pulses must be determined. The STM32H7's Reset and Clock Controller (RCC) provides a feature that facilitates this measurement. The RCC manages several internal clock sources, including a high-speed internal 64 MHz oscillator (HSI), low-power internal 4 MHz oscillator (CSI), accurate 48 MHz oscillator and a low-speed internal 32 kHz RC oscillator (LSI). The RCC also supports high-speed external oscillators (HSE) ranging from 4 to 48 MHz. The clock security system automatically switches to the HSI if a loss of the external clock is detected [29].

This paper proposes wiring the OCVCXO as the HSE of the STM32 and connecting the

GNSS receiver's PPS signal to a GPIO input pin. An interrupt should be configured with the highest priority, as described in Section 2.1.2, to register the PPS signal. An interrupt handler can then be programmed to follow this algorithm:

1. **Capture Cycle Count**:
   Read the value of the CPU Cycle Counter (CYCCNT) from the data watchpoint and trace unit (DWT) cycle count register [32].
2. **Compute the difference** between the current and previously captured CYCCNT values to determine the number of CPU cycles elapsed since the last interrupt.
3. **Adjust PLL Voltage**:
   - If the difference is greater than 10 million cycles, reduce the PLL voltage via the DAC, as the clock runs too fast.
   - If the difference is less than 10 million cycles, increase the PLL voltage to speed up the clock, as the clock runs too slow.

The accumulated difference in cycles can be used instead to account for the PLL convergence delay. A reference algorithm that accumulates the error over 100 PPS pulses is provided in Appendix A.

2.3 PPS Phase Synchronization

Once the frequency stability of the Oven-Controlled Voltage-Controlled Crystal Oscillator (OCVCXO) has been calibrated and corrected, the next imperative is to develop a strategy for generation of a synchronized Pulse-Per-Second (PPS) signal. This synchronization ensures temporal coherence between the local time signal source and Coordinated Universal Time (UTC).

A straightforward hardware solution involves utilizing frequency division circuitry via a sequence of binary counters, such as the Texas Instruments CD4040 series [33]. By setting a division factor of 10 million and providing an input from the OCVCXO clock, the carry-out signal from the final counter generates a stable 1 Hz output. To align this signal phase with the PPS output from the GNSS receiver, the GNSS PPS signal is routed to the reset pins of the binary counters. Consequently, the locally generated PPS signal maintains phase alignment with the GNSS PPS signal, thereby aligning with UTC through a cost-effective hardware solution.

However, this approach exhibits notable limitations, primarily the lack of configurability in phase offsets to account for time signal propagation errors. In contrast, the STM32 H7

MCU series offers integrated high-resolution timers that, when configured in pulse-width modulation (PWM) mode, can function as versatile frequency dividers. By dynamically adjusting the PWM duty cycle and precisely timing resets via software, it is possible to generate an accurate PPS signal with user-configurable phase offsets, thus providing more precise temporal synchronization [34].

To evaluate the generated PPS signal's temporal precision with respect to the reference GNSS signal, a methodology for capturing the relative phase offset between the two signals is necessary. While monitoring a generated a PPS signal with picosecond-level precision typically requires sophisticated instrumentation, monitoring the phase difference without signal generation is achievable with readily available microcontroller peripherals.

A detailed interrupt-driven approach to capture and monitor the phase offset between the OCVCXO-derived PPS and GNSS-derived PPS signals is outlined in Appendix A on line 38. The algorithm involves accumulating phase error measurements across multiple PPS pulses, thereby mitigating transient inaccuracies and enabling long-term, high-stability synchronization.

A detailed interrupt-driven approach to capture and monitor the phase offset between the OCVCXO-derived PPS and GNSS-derived PPS signals is outlined in Appendix A. The algorithm involves accumulating phase error measurements across multiple PPS pulses, thereby mitigating transient inaccuracies and enabling long-term, high-stability synchronization.

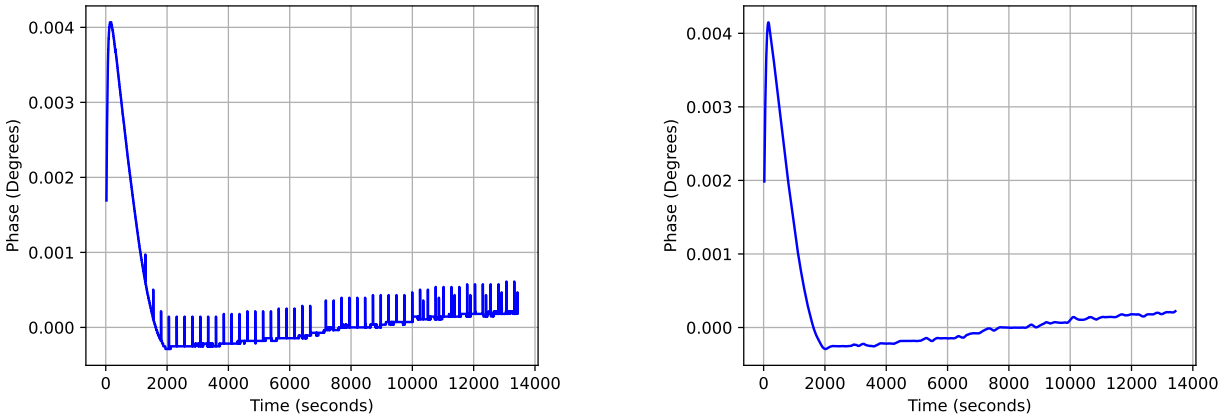
2.4 Results Overview

A comprehensive time transfer experiment was conducted over a span of 13,000 seconds (approximately 3.6 hours), during which multiple metrics pertaining to the Oven-Controlled Voltage-Controlled Crystal Oscillator (OCVCXO) were recorded. The experiment began with a cold-start of the GNSS receiver and the OCVCXO at room temperature, providing a full view of the crystal's warm-up phase and subsequent stabilization.

For the evaluation, key metrics such as the phase and frequency of the oscillator were analyzed. The phase was calculated using the formula:

$$\phi(T_{\text{interrupt}}) = f(T_{\text{interrupt}}) \mod 10000000$$

where $f(T)$ is the value of the CYCCNT cycle count register at time T.



(a) Unfiltered

(b) Smoothed with Savitzky-Golay

Figure 8: Relative OCVCXO observed phase value over time.

The warm-up phase of the OCVCXO is observed during the initial 2,000 seconds, as depicted in Figure 8a. During this interval, significant frequency drift is noticeable due to thermal effects as the crystal's oven stabilizes. The cumulative frequency error and phase offset stabilize gradually as the crystal reaches its operating temperature.

After the warm-up period, periodic noise in the form of 10 Hz jumps is visible in the raw phase data (Figure 8a) and in the histogram of instantaneous frequency offset values (Figure 9). These jumps are attributed to debounce effects in the GNSS PPS signal and software interrupt latencies, representing measurement artifacts rather than inherent issues with the OCVCXO. To mitigate this noise, a Savitzky-Golay filter was applied, producing a smoothed frequency stability profile (Figure 8b) and cumulative relative frequency error (Figure 10).

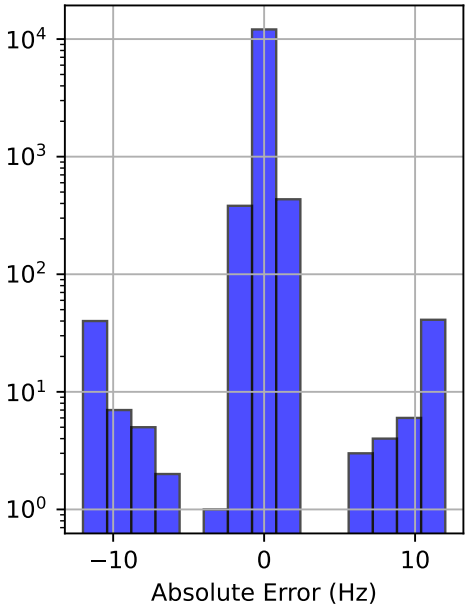


Figure 9: OCVCXO frequency offsets distributions with two peaks at ±10 Hz, caused by PPS signal debounce.

The Savitzky-Golay filter is a digital smoothing filter that applies a polynomial fitting over a sliding window to preserve important features like peak heights and widths while reducing noise. The polynomial coefficients are computed using the least-squares method. For a given window size N and polynomial

order k, the smoothed value y'_i of data point y_i is calculated as:

$$y'_i = \sum_{j=-m}^{m} c_j y_{i+j}$$

where: $m = \frac{N-1}{2}$, and c_j are the filter coefficients derived from polynomial fitting [35].

In this experiment, a window size of 301 samples and a polynomial order of 3 were chosen to balance the trade-off between noise reduction and curve shape distortion.

The smoothed results highlight the OCVCXO's frequency stability over the entire duration of the experiment, with a frequency change of 10 Hz over 11,000 seconds post-warmup. The relative frequency error is 0.09 parts per billion (ppb), calculated as:

$$\frac{10 \text{ Hz}}{11000 \cdot 10\text{MHz}} = 9 \times 10^{-11} \approx 0.09 \text{ ppb}$$

Simultaneously, phase stability achieved a drift of 6×10^{-4} degrees, equivalent to approximately 2 microseconds, over the same period.

The linear downward trend in cumulative relative frequency error suggests that the OCVCXO was oscillating 0.0009 Hz below the target 10 MHz frequency, with the limited 12-bit resolution of the Digital-to-Analog Converter (DAC) unable to fully compensate. Despite this, the observed frequency stability remains within operational margins for SFN applications [4, 36–39].

These results do not account for the drift of the GNSS receiver's internal Temperature-Compensated Crystal Oscillator (TCXO), which could impact the accuracy of the measurements. Comprehensive validation of this experimental setup requires precision instrumentation, such as geodetic antennas with GNSS phase-locking capability [40, 41].

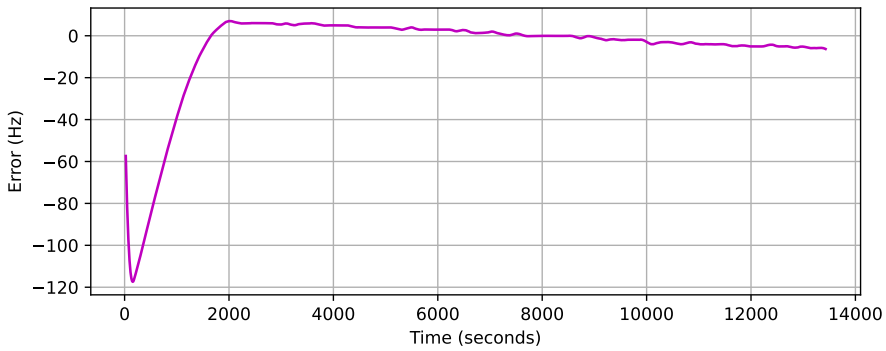


Figure 10: Cumulative relative OCVCXO observed frequency error smoothed with Savitzky-Golay filter.

2.5 Conslusions

In summary, this foundational time transfer experiment demonstrated the efficacy of OCVCXO frequency correction and PPS phase synchronization using STM32 microcontrollers. The subsequent section addresses resilient time transfer in the context of GNSS spoofing, introducing innovative algorithms and methodologies to maintain synchronization accuracy despite intentional interference.

With a foundational time transfer setup in place, the next section focuses on enhancing time transfer resilience, particularly against GNSS spoofing. Here, this paper delves into innovative algorithms and methodologies designed to maintain accurate and reliable synchronization despite the presence of intentional signal interference.

3 Resilient Time Transfer

The resilience of time transfer is paramount for reliable GNSS-based services due to the increasing vulnerability of these services to spoofing and interference attacks. The proposed framework for resilient time transfer relies on incorporating diverse timing sources and validating the consistency of GNSS-derived timing through cross-checking methodologies with alternative references, following the principles outlined by Zhang et al. in "Protecting GNSS-based Services using Time Offset Validation" [42, 43].

3.1 Concept overview

To improve resilience against spoofing attacks, an effective time cross-checking method must be developed using alternative timing sources, specifically focusing on absolute and relative time validation. This section explores and refines methodologies to enhance GNSS-based time transfer resilience using external time sources, with specific emphasis on detecting intentional time shifts.

The validation framework proposed by Zhang et al. involves two primary strategies:

1. **Absolute Time Validation:** By comparing the absolute difference between GNSS-provided time and an alternative source:

$$f(t) = f(|T_{\text{ext}}(t) - T_{\text{GNSS}}(t)|)$$

   where T_{ext} represents the time from an external source and T_{GNSS} denotes the GNSS-derived UTC time. The result is then compared to the accuracy threshold of the external source, ε_{ext}.

2. **Relative Time Validation:** This method examines the difference in elapsed time between consecutive GNSS-provided intervals:

$$f(t) = f(|\Delta T_{\text{ext}}(t) - \Delta T_{\text{GNSS}}(t)|)$$

   where ΔT_{ext} and ΔT_{GNSS} represent the elapsed intervals of external and GNSS time, respectively.

This paper advocates for the incorporation of an alternative time source derived from a longwave time signal and standard-frequency radio station into a resilient time transfer

framework. The specific station under consideration is the RBU facility operating near Moscow, Russia.

Longwave radio transmissions offer several advantages as a time reference. They demonstrate superior penetration capabilities through dense urban environments compared to GNSS signals [44]. Additionally, the physics of radio wave propagation at these frequencies makes them demonstrably less susceptible to spoofing attacks [45].

To achieve resilient time transfer with RBU radio time signal, the following steps are undertaken:

1. **Antenna Setup:** An antenna designed and tuned to the $66.\overline{6}$ kHz frequency of the RBU signal should be employed to maximize reception quality.
2. **Signal Processing Algorithm:** Digital signal processing algorithms are proposed to extract relative timing information from the RBU signal to supplement GNSS-based timing:
   (a) **Preprocessing:** Filter and demodulate the received longwave signal to reduce noise and interference.
   (b) **Clock Recovery and Decoding:** Extract the timing markers and calculate the time signal phase.
   (c) **Time Offset Calculation:** Compare the RBU time phase with the UTC time phase to compute the time offset.
3. **Time Cross-Checking and Validation:** The time offset between the GNSS and RBU signals is monitored over a rolling window to detect discrepancies. A threshold is set based on the accuracy of the RBU signal to flag spoofing attempts and disable time transfer with the compromised GNSS receiver.

For illustrative purposes, Figure 11 depicts a time and frequency transfer system. The system comprises a stratum-1 time and frequency signals source, realized by an STM32 development board driven by a High-Speed External (HSE) clock in the form of an Oven-Controlled Voltage-Controlled Crystal Oscillator (OCVCXO). Additionally, it includes a u-blox MAX-M10S GNSS receiver module, an RBU antenna, and a personal computer configured for digital signal processing of the digitized signal received from the RBU-tuned antenna.

To facilitate system performance evaluation under GNSS spoofing conditions, a Software Defined Radio (SDR) HackRF One transmitter is employed to simulate a GPS constellation signal in Section 3.3.1.

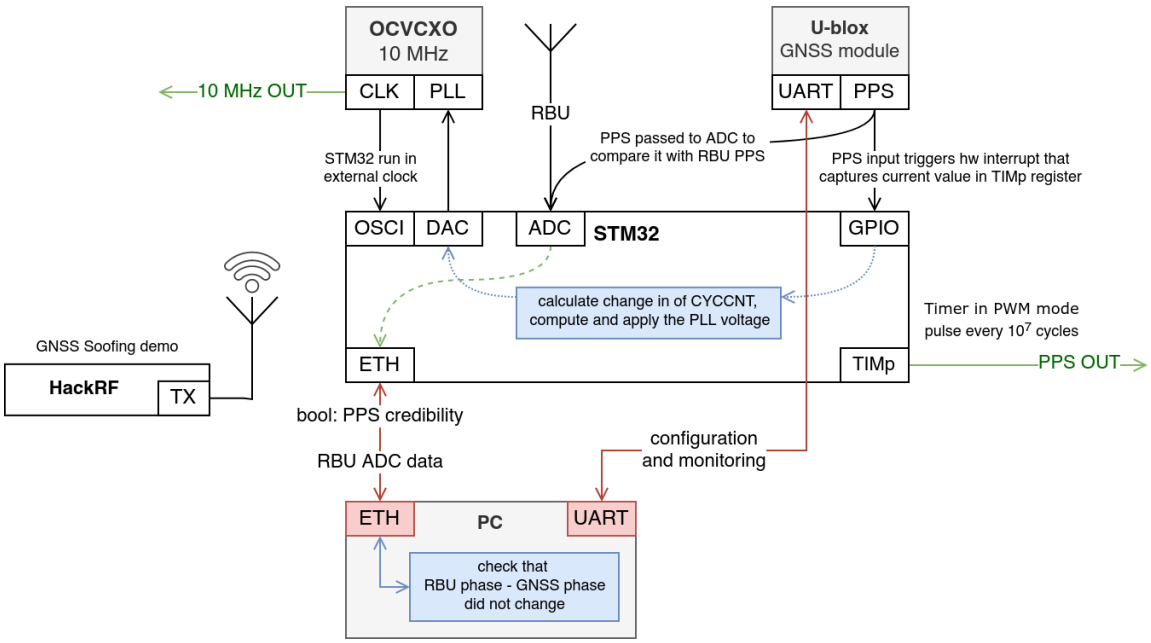


Figure 11: Resilient time and frequency transfer system schematic outlining proposed hardware and software architecture.

3.2 RBU Signal Processing

3.2.1 Antenna Construction

Historically, longwave time signals have been received using ferrite rod antennas, a design often employed in wristwatches and other portable devices [46]. These antennas are constructed with a ferrite bar wrapped in a conductive wire.

For effective reception of the RBU longwave time signal at $66.\overline{6}$ kHz, an antenna specifically designed and tuned to this frequency is required. The ferrite rod antenna offers high selectivity and signal sensitivity due to the ferrite core's magnetic properties, which concentrate the magnetic flux and enhance signal reception.

A ferrite rod antenna suitable for receiving the RBU signal typically has the following specifications [47]:

- **Ferrite Rod Dimensions:** The rod should be 100-150 mm in length and 10-12 mm in diameter. A ferrite material with high permeability (μ) is recommended, such as N48 or N87, to enhance signal sensitivity.
- **Wire Gauge:** Enamel-coated copper wire (magnet wire) of 0.1-0.2 mm in diameter.
- **Number of Loops:** 100-200 turns around the ferrite rod to achieve resonance at $66.\overline{6}$ kHz.

The inductance (L) of the coil is calculated using:

$$L = \frac{\mu_0 \mu_r A N^2}{l}$$

where: μ_0 is the permeability of free space ($4\pi \times 10^{-7}$ H/m), μ_r is the relative permeability of the ferrite material, A is the cross-sectional area of the ferrite rod, N is the number of turns, l is the length of the coil.

The resonant frequency (f) of the antenna is determined using the following formula:

$$f = \frac{1}{2\pi\sqrt{LC}}$$

where L is the inductance of the coil, and C is the capacitance of the tuning capacitor. To tune the antenna to the RBU signal frequency, a parallel capacitor C is selected to resonate at $66.\overline{6}$ kHz.

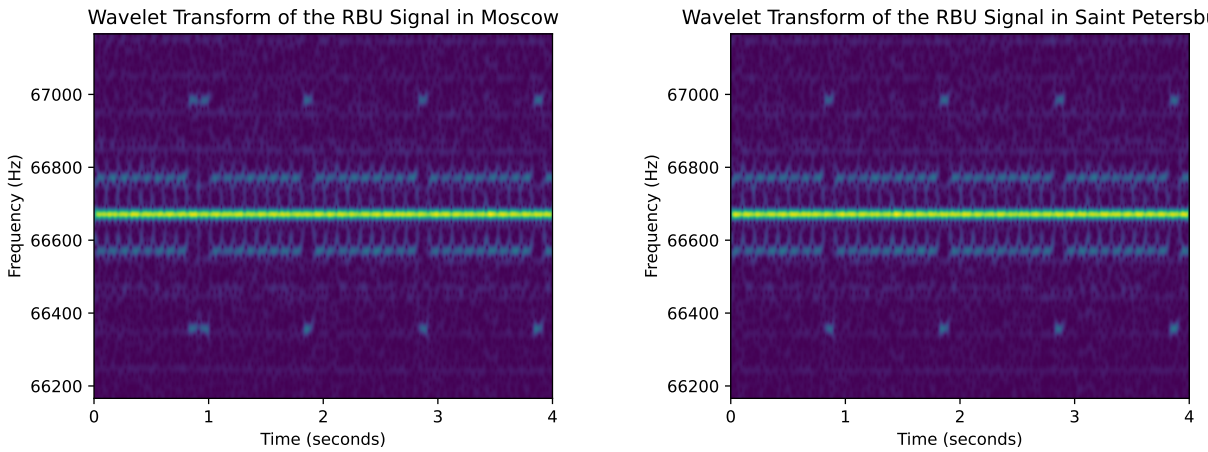
3.2.2 Experimental Validation of Antenna Performance

To quantitatively assess the efficacy of the proposed longwave time signal reception strategy, field measurements were conducted at two geographically distinct locations: Moscow, Russia (in close proximity to the RBU transmitter), and Saint Petersburg, Russia (approximately 600 kilometers apart). The rationale behind this experiment design was to evaluate the signal degradation over a varying propagation distance.

Wavelet analysis, a time-frequency signal processing technique, was employed to characterize the spectral properties of the received RBU signal at both locations. Wavelet transforms decompose a signal into its constituent time-frequency components, enabling the visualization of both signal frequency content and its variation over time. This analysis method is particularly well-suited for non-stationary signals, such as the RBU transmission which incorporates timing markers superimposed on a carrier wave.

The wavelet power spectra for the RBU signal received in Moscow and Saint Petersburg are presented in Figures 12a and 12b, respectively. As anticipated, the signal exhibits a dominant spectral component at the expected RBU carrier frequency of $66.\overline{6}$ kHz.

Encouragingly, the wavelet power spectra for both recordings reveal minimal spectral noise or interference across the frequency band of interest, signifying successful reception and demodulation of the RBU signal. Furthermore, the presence of a clear and well-defined time-domain marker within the Moscow recording (Figure 12a, near the one-second time mark) underscores the viability of extracting timing information from the received signal using the proposed digital signal processing algorithms.



(a) Moscow

(b) Saint Petersburg

Figure 12: Wavelet transform of the RBU signal recorded at different distances from the transmitter.

The successful reception and characterization of the RBU signal at both test locations, even across a substantial propagation distance, validates the potential of incorporating longwave time signal references into a resilient GNSS time transfer framework. Following efforts will concentrate on the development and implementation of robust digital signal processing algorithms to extract timing information from the RBU signal for time transfer applications.

3.2.3 ADC Circular Buffer DMA Configuration

The STM32 H7 microcontroller series incorporates multiple 16-bit Analog-to-Digital Converters (ADCs) that can be extended to 16-bit resolution via sigma-delta modulation. To facilitate continuous signal acquisition with minimal computational overhead, the integrated Direct Memory Access (DMA) peripheral is configured in Circular mode. This allows seamless data transfer from the ADC to memory while minimizing CPU involvement.

DMA enables efficient data transfer by offloading memory transactions from the CPU, thus allowing uninterrupted signal acquisition. In Circular mode, the DMA controller automatically wraps around to the initial memory address once the end of the buffer is reached, forming a circular buffer that continuously stores new ADC data. The key configuration parameters include:

1. **Source and Destination Addresses:** Defines the ADC data register as the source and a pre-allocated memory buffer as the destination.
2. **Transfer Size:** Specifies the number of data elements to be transferred per DMA transaction.

3. **Increment Modes:** Configures the source as non-incremental and the destination as incremental to support circular buffering.
4. **Circular Mode Activation:** Enables automatic wrapping to the beginning of the buffer.

The Circular mode ensures continuous signal acquisition without data loss while minimizing the load on the CPU, which only needs to configure the DMA transfer.

The STM32 H7 ADCs offer multiple conversion modes to accommodate various signal acquisition needs:

1. **Single:** Converts a single input channel, operating in single-shot or continuous mode.
2. **Scan:** Converts a predefined set of input channels in single-shot or continuous mode.
3. **Discontinuous:** Converts a subset of predefined input channels at each trigger signal.

For this resilient time transfer application, this paper proposes adoption of Single Mode with continuous conversion to monitor a single RBU signal channel. This approach ensures consistent signal acquisition with minimal overhead, providing the data required for robust time validation.

The ADC-to-DMA data transfer mechanism proposed by this paper operates as follows:

1. **ADC Conversion:** The ADC continuously converts the analog signal to a digital representation and stores the result in its data register.
2. **DMA Transfer into memory:** The DMA controller autonomously transfers the ADC result to a circular buffer in memory.
3. **Transfer to PC:** The buffer is transferred from memory to a personal computer for Digtal Signal Processing (DSP).

The ADC Circular Buffer DMA configuration enables continuous signal acquisition, facilitating seamless integration into the proposed resilient time transfer framework. By carefully configuring the ADC conversion mode and DMA transfer parameters, a reliable data pipeline is established for the consistent decoding and validation of time signals.

3.2.4 ADC Data Transfer

Efficient and reliable transfer of the digitized radio signal from the embedded microcontroller to a personal computer (PC) is crucial for advanced digital signal processing (DSP) tasks. These tasks, often demanding high computational resources, are suboptimal for execution on

embedded devices due to their limited processing capabilities. This section discusses various communication interfaces for transmitting digitized signals effectively, thereby enabling the implementation of sophisticated DSP algorithms essential for accurate time signal decoding and validation.

The choice of an appropriate communication interface is governed by several pivotal factors:

- **Data Rate Requirements:** Predicated on the ADC's sampling rate and resolution, a minimum data transfer rate is necessary to handle the output effectively. Given a typical sampling rate between 1 and 2 Msps with 16-bit resolution, the interface must support data rates of at least 32 Mbps.
- **Distance and Network Complexity:** The distance between the embedded device and the PC determines the selection of the communication interface. Shorter distances may benefit from simpler, direct communication methods such as SPI, whereas longer distances necessitate more robust network protocols like Ethernet.
- **Real-Time Constraints and Determinism:** Applications requiring stringent real-time performance and deterministic data delivery may prioritize interfaces that provide guaranteed timing and delivery mechanisms.
- **Hardware Availability and Development Effort:** The choice of interface is also influenced by the existing hardware capabilities of the embedded system and the PC, and the complexity involved in developing necessary drivers and software.

Several interfaces are assessed based on their suitability for the outlined criteria:

- **Universal Asynchronous Receiver/Transmitter (UART):** Although widely used due to its simplicity, UART's limited throughput, typically only in the kbps range, is inadequate for our needs.
- **Serial Peripheral Interface (SPI):** SPI can potentially provide high data rates suitable for short-distance, high-speed transfers. Nevertheless, its applicability is diminished by practical constraints on distance and the absence of sophisticated error handling mechanisms in the STM32 H7 microcontroller.
- **Inter-Integrated Circuit (I2C):** Suitable for low-speed, multi-device networks, the I2C's lower data rate limits its utility for high-throughput applications.
- **Controller Area Network (CAN):** Though robust and reliable, CAN's primary use in industrial control rather than high-speed data transfer renders it suboptimal for this application.

- **Universal Serial Bus (USB):** USB supports high data rates and broad compatibility with PCs. However, it requires complex driver development, potentially complicating deployment across various operating systems.
- **Ethernet:** With inherent support for high-speed network communication and minimal additional hardware requirements due to the integrated Ethernet controller in the STM32 H7 series, Ethernet is ideally suited for this application. It supports data rates sufficient for our needs and incorporates DMA offloading for efficient data transfers.

Given its high data throughput capabilities, integral support in our chosen hardware platform, and the provision of advanced network features, Ethernet is selected as the communication interface for data transfer. Ethernet's attributes on the H7 include:

- 100 Mbit/s data rate capabilities, meeting application's requirements.
- Built-in support in the STM32 H7 series, eliminating the need for hardware integration.
- DMA capability that facilitates efficient data offloading, conserving CPU resources for other critical tasks.
- Support for advanced network features such as hardware-accelerated checksumming, flow control, and hardware Precision Time Protocol (PTP) for accurate time synchronization, which are essential for precision and data integrity in time-sensitive applications [48].

The employment of Ethernet not only aligns with the technical demands of the project but also enhances the system's scalability and flexibility for potential future upgrades or modifications.

The final stage in our data acquisition and processing pipeline involves decoding the digitized RBU signal. This phase is critical as it transforms raw data into actionable timing information. Decoding strategies involve several sophisticated algorithms, including phase-locked loops (PLLs), cross-correlation analysis, and Savitzky-Golay filtering to extract and refine the timing signals. This section delineates the algorithmic approach and evaluates its efficacy through simulation and empirical testing.

3.2.5 Signal Decoding

The accurate decoding of time signals involves the extraction of phase information from the carrier wave interruptions and modulations at the 312.5 Hz frequency, as previously mentioned in Section 1.2.6. This process is essential to accurately compute relative phase differences and thus derive precise time measurements.

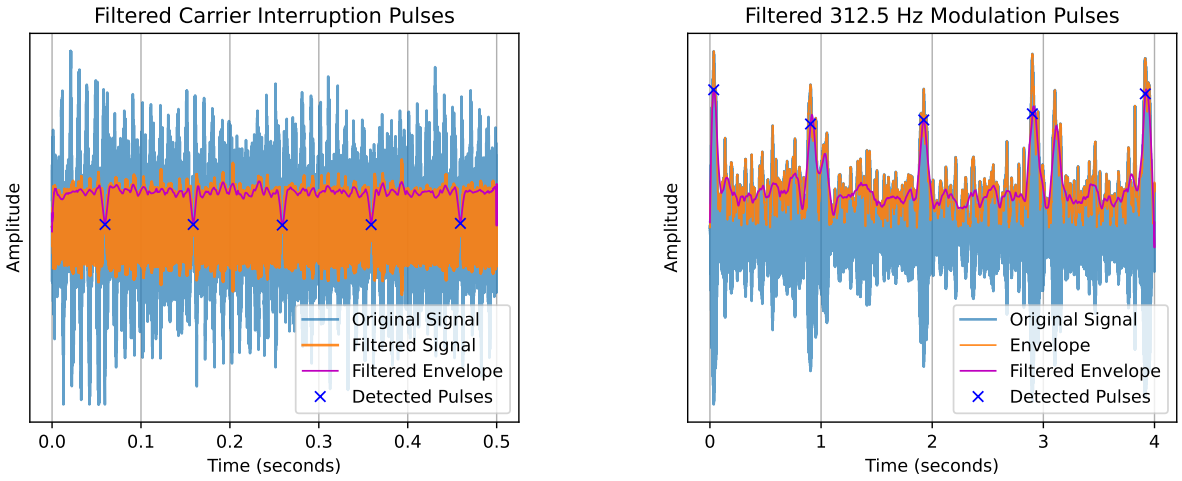
In this context, the 312.5 Hz subcarrier modulation is crucial, as it correlates directly to the 1 Hz signal that marks the beginning of UTC seconds. Figure 1 illustrates a representative segment of the DXXXW signal.

To isolate these characteristics, an initial filtering stage employs a bandpass filter to capture the 10 Hz carrier frequency. Following this, an algorithm detects periodic reductions in amplitude to ascertain the phase of the 10 Hz signal. This filtering strategy effectively highlights the signal interruptions critical for phase calculations.

With the phase of the 10 Hz signal established, resolving the ambiguity of the 1 Hz PPS signal's phase involves searching for peaks corresponding to the 312.5 Hz modulation preceding each second mark. These detections are facilitated by either employing a wavelet analysis, as demonstrated in Section 3.2.2, or by utilizing a more focused filter that isolates only the target frequency.

For digital signal processing, the SciPy library is utilized extensively due to its comprehensive support for DSP algorithms [49]. The steps involved are as follows and are executed in separate threads [50]:

1. **10 Hz Interruption Detection:** A Finite Impulse Response (FIR) bandpass filter with 1001 taps is implemented to isolate the 10 Hz carrier interruptions. The choice of taps and filter characteristics are based on the signal's attributes and desired resolution. The filtered signal's envelope is then calculated using the Hilbert transform, followed by further smoothing with a Savitzky-Golay filter to reliably identify signal interruptions.
2. **312.5 Hz Modulation Detection:** Similar steps are repeated. Instead, due to the longer duration of these modulations (90 ms), a FIR filter with 18001 taps is used, increasing bandwidth necessary for detecting these events.



(a) Isolated 10 Hz interruptions

(b) Isolated 312.5 Hz modulations

Figure 13: Carrier signals filtered at target frequencies after DSP.

The results from the DSP stages are further visualized using a Matplotlib library [51] in Figure 13a for the 10 Hz carrier interruptions and Figure 13b for the 312.5 Hz modulations. Each figure displays the signal after the DSP and peak detection steps have been applied, showcasing the target frequencies' signal characteristics.

Following the isolation and detection of the 10 Hz carrier interruptions and 312.5 Hz modulations through digital signal processing (DSP) techniques, a crucial step involves analyzing the peak timing information to compute the phases of the target signals. This phase information is instrumental in deriving precise time measurements from the RBU signal.

To facilitate the consistent representation and analysis of recurring signals, this paper advocates the use of modular arithmetic for phase calculations. The phase of a recurring signal is determined by calculating the modulo of the peak occurrence times relative to the signal's period. For instance, a peak of the 312.5 Hz modulated signal occurring at 2542 milliseconds since the start of the recording would be considered to have a phase of 542 milliseconds, calculated as 2542 $\bmod 1000$, because the period of this signal is 1 second. Similarly, for 10 Hz interruptions, which have a period of 100 milliseconds, the modulus of 100 milliseconds is used.

Modular representations simplify algorithmic accumulation and averaging of recorded signals to achieve target phase detection accuracy.

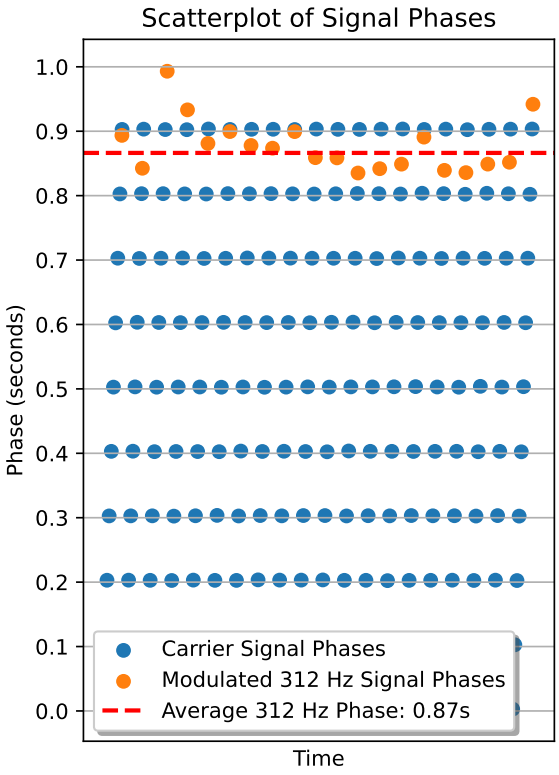


Figure 14: Scatter plot displaying the phase values of detected signals peaks. The x-axis represents the time of peak occurrence since the recording began, and the y-axis shows the phase of each peak modulo 1 second, highlighting the periodic nature of the peak occurrences.

Figure 14 depicts the phases of all detected peaks across two analyzed signals, corresponding to 1-second intervals. The consistent and near-linear distribution of the carrier interruption peak phases demonstrates the effectiveness of the DSP algorithms in accurately identifying these markers. Notably, there are minimal deviations from the expected linear trend, suggesting successful peak detection with negligible errors.

However, the peak phases of the 312.5 Hz modulation exhibit a slightly higher degree of variability compared to the carrier interruption peaks due to it's wider modulation time frame. While the majority of these peaks cluster around a central value, there are a few outliers evident in the figure.

To mitigate the impact of these outliers and achieve a more robust phase estimation for the 312.5 Hz modulation, averaging and outlier removal techniques are employed. By calculating the average phase value across a defined window of consecutive peaks, outliers can be removed and phase recomputed with higher degree of precision.

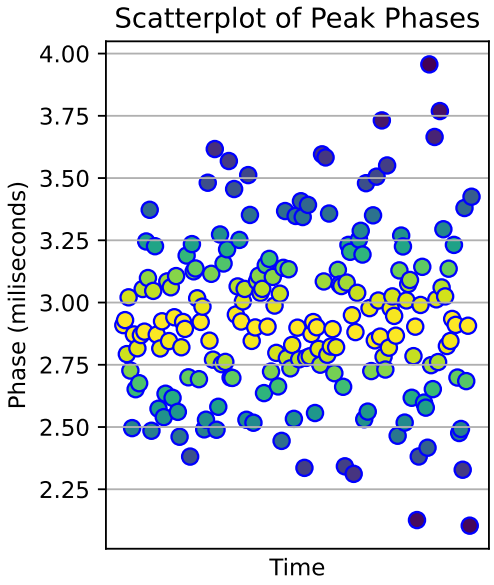


Figure 15: Phase values of carrier interruption peaks computed relative to it's period of 100 milliseconds with color representing phase-value density.

Figure 15 presents a scatter plot depicting the phases of individual carrier interruption peaks over a 30-second signal duration (encompassing 300 detected peaks). This dataset allows for a comprehensive evaluation and estimation of the phase error distribution.

Achieving a high degree of accuracy in time measurements is contingent upon the precision with which phase values are extracted and analyzed. This paper addresses the methodologies employed to evaluate and enhance the precision of phase measurements derived from the RBU carrier interruptions.

To ensure the reliability of the time transfer technique, it is imperative to ascertain that the phase measurements errors adhere to a predictable statistical distribution. Initial analysis involves plotting the distribution of carrier interruption phases to ascertain their conformity to a normal distribution, which is a prerequisite for applying certain statistical techniques.

The histogram in Figure 16 illustrates that the phase values closely follow a normal distribution. This finding permits the application of parametric statistical methods for further analysis. Subsequently, the standard deviation (denoted as σ) of the phase measurements is calculated, revealing a value of approximately 0.3 milliseconds. This measurement indicates the spread of the phase values around the mean, serving as a metric of variability and precision.

In the context of refining measurement precision, it is essential to ascertain the adequate number of samples needed to achieve a specified level of accuracy. The desired precision, defined in terms of the standard error of the mean (SEM), is set at 20 microseconds. This value is conservatively chosen based on the consideration that light travels approximately 6

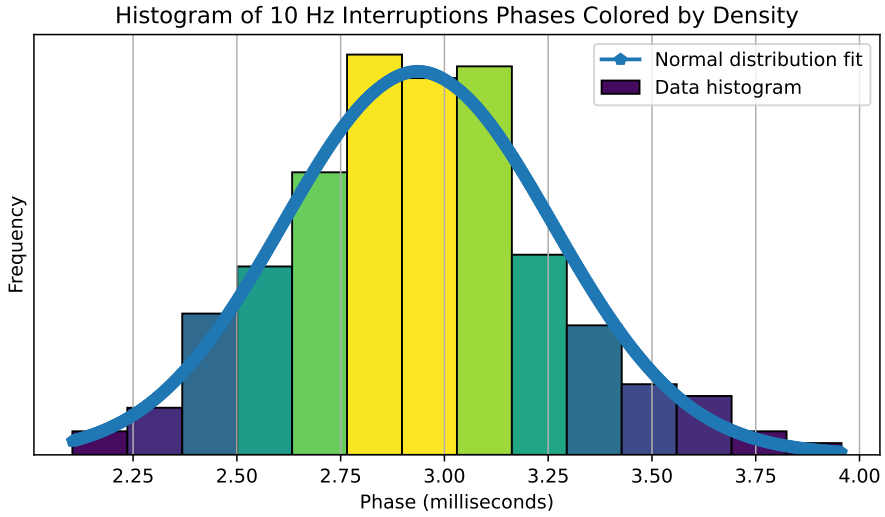


Figure 16: Distribution of carrier interruption phases, highlighting the Gaussian nature of the data. The x-axis represents the time of interruption occurrence since the recording began, and the y-axis shows the phase of each interruption modulo 100 milliseconds.

kilometers in this duration, relating to less five percent of the operational range of typical DVB-T2 systems, which extends up to 130 kilometers [52]. Furthermore, the DVB-T2 provider has the option to reduce the target SEM by increasing the sample size, which in turn affects the convergence speeds.

Using the standard deviation obtained from the histogram analysis, the required sample size n is calculated using the SEM equation:

$$\text{SEM} = \frac{\sigma}{\sqrt{n}}; \qquad n = \left(\frac{\sigma}{\text{SEM}}\right)^2$$

Given $\sigma = 0.3$ milliseconds and $\text{SEM} = 0.02$ milliseconds, the computation yields a sample size of approximately 289 measurements. This calculation dictates the minimum number of phase measurements needed to achieve the desired level of precision under normal distribution assumptions.

3.3 GNSS Jamming and Spoofing

To anticipate and counteract potential threats in GNSS services, it is imperative to simulate the actions of an attacker. This simulation uses readily available tools to demonstrate potential vulnerabilities. A popular tool in these simulations is the *HackRF One* software-defined radio (SDR) from Great Scott Gadgets.

The `osqzss/gps-sdr-sim` project is a prime example of an open-source initiative that will be utilized in this study to conduct experimental spoofing simulations. This software is highly cited and extensively employed within the community for GNSS signal simulation, demonstrating its robustness and reliability in such applications [53, 54].

3.3.1 HackRF SDR Configuration

The maximum range for GNSS spoofing achieved with a HackRF One sans external amplifiers is confined to roughly 5 meters, a finding documented by Songala et al. [55]. This constraint is crucial for conducting experiments within a safe and controlled environment. To further mitigate risks, precautions were taken to comply with local regulation, execute the experiments in secluded locations and restrict transmission durations, ensuring minimal possibility of causing unintended disruptions.

Ephemeris data, which includes detailed information about the positions of satellites in orbit, is crucial for generating accurate GNSS signals. This data, typically provided in RINEXv3 format, is essential for ensuring that the simulated signals reflect current satellite positions [56].

For the experiments, an antenna with a size of 10 cm was used. This dimension corresponds to half the wavelength of the carrier frequency, calculated by the equation:

$$\text{Wavelength} = \frac{c}{f} = \frac{3 \times 10^8 \text{ m/s}}{1575 \times 10^6 \text{ Hz}} \approx 19 \text{ centimeters},$$

where c is the speed of light and f is the frequency.

```
./gps-sdr-sim -e ./ephemeris.rinex3 \  # Ephemeris data file
    -l "55.753,37.648,120" \ # Coordinates and height
    -b 8 \                   # 8-bit data format for HackRF DAC
    -o - |                   # Pipe output to standard output
    hackrf_transfer -t - \   # Read data from standard input
        -f 1575420000 \      # gnss carrier frequency of 1575.42MHz
        -s 2600000 \         # 2.6 Msps, maximum supported HackRF sample rate
```

Listing 1: Command to simulate and transmit GNSS signal using HackRF.

After setting up the HackRF with GPS-SDR-SIM and obtaining the latest ephemeris data, signal simulation and transmission can commence as shown in Listing 1.

Following successful signal transmission, the GNSS receiver typically converges on the spoofed coordinates and time. This result is depicted in Figure 17, showing a PyGPSClient interface reporting the manipulated positioning.

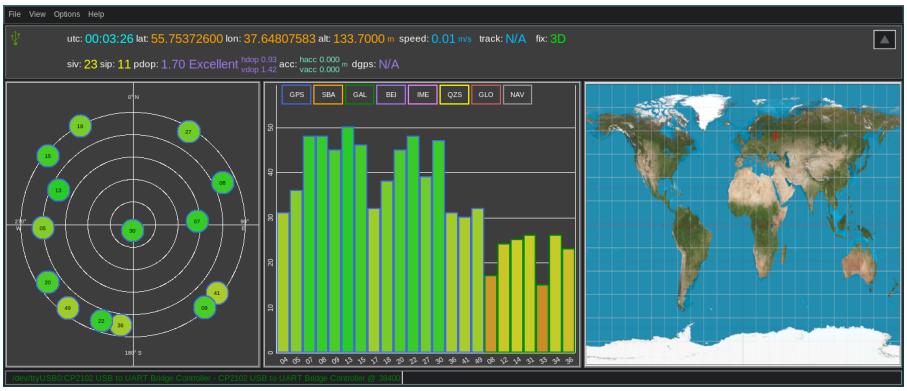


Figure 17: PyGPSClient reporting spoofed position and time.

The proximity of the transmitter to the GNSS receiver results in exceptionally high Carrier-to-Noise ratio (C/N_0) values, indicative of robust signal reception. These values, depicted by green bins in the middle subplot of Figure 17, suggest an unnaturally strong signal relative to conventional GNSS reception levels. Elevated C/N_0 values can serve as indicators of spoofing, offering a method for detection based on signal strength anomalies [57, 58].

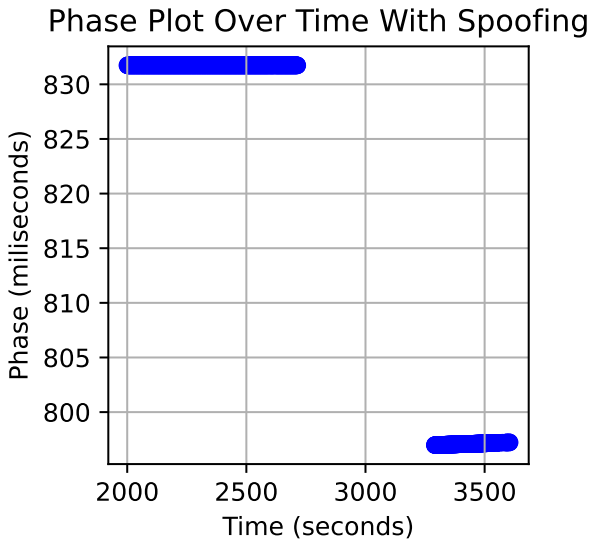


Figure 18: Phase plot showing the effect of GNSS spoofing.

Additionally, plotting the phase of an incoming PPS signal as illustrated by Figure 18, reveals a conspicuous change following the activation of the HackRF transmission. The interruption in continuity between the 2750 and 3250 second marks is attributable to the GNSS receiver's reconvergence on the new, spoofed signal.

These observations not only underscore the efficacy and availability of the spoofing technique but also highlight how discrepancies in C/N_0 and phase continuity can be exploited to detect and mitigate spoofing attacks.

3.4 Spoofing Detection Algorithm

The work presented in this thesis proposes a comprehensive approach to enhancing the resilience of GNSS time transfer by integrating longwave time signals and standard-frequency radio stations as supplementary sources. This hybrid method addresses vulnerabilities in GNSS, such as spoofing and signal loss, by providing an independent verification mechanism through terrestrial radio signals. The conclusion of this research not only encapsulates the findings but also lays the groundwork for future innovations in the field of time synchronization with alternative time sources.

The proposed method systematically leverages the stable and reliable nature of longwave radio signals to augment GNSS time synchronization. The methodology involves several key steps:

1. **Phase Establishment:** Initially, the correct relative phase of the radio time signal is compared to UTC time, established and stored. This phase acts as a benchmark for subsequent measurements.
2. **Synchronization Delay:** While the GNSS receiver's pulse-per-second (PPS) provides primary time data, its synchronization is not immediately applied. Instead, synchronization values are computed and temporarily held.
3. **Parallel Processing:** Concurrently, the radio time signal undergoes digital signal processing at a server to calculate time offsets accurately.
4. **Validation and Application:** Time offsets are retransmitted to a time transfer device. Synchronization adjustments are only applied if the newly computed phase from the radio time station does not deviate beyond a predefined threshold from the stored phase.
5. **Continuous Operation:** This process is iterative, ensuring ongoing verification and adjustment of time synchronization, thereby safeguarding against errors and potential spoofing attacks.

4 Conclusions

The significance of this research lies in its potential to provide another decentralized time source and increase the reliability of time transfer systems in critical infrastructure. By integrating a fail-safe mechanism that cross-verifies GNSS data with longwave radio signals, the system maintains high accuracy and robustness against disruptions. This method significantly mitigates the risks associated with GNSS dependency, offering a more secure and resilient framework for time-sensitive applications.

The experimental validations confirm the effectiveness of this approach, with significant improvements in the reduction of time transmission errors and enhanced resilience against disruptions. These findings lay a solid foundation for future research and the development of advanced hybrid time synchronization systems.

Looking forward, this thesis opens several avenues for further research:

- **Enhanced DSP Techniques:** Advanced digital signal processing algorithms can be explored to further reduce the time offset calculation latency and improve the precision of phase measurements.
- **Machine Learning Models:** Implementing machine learning techniques to predict and correct discrepancies in time signals before they affect the system could offer a proactive approach to time synchronization.
- **Broader Signal Integration:** Expanding the range of signal sources, including other radio frequencies and emerging satellite systems, could enhance the redundancy and reliability of the time transfer framework.

Finally, this thesis not only advances the academic field of time synchronization but also provides a practical blueprint for deploying resilient time transfer systems. The methodologies and insights derived from this study contribute to the body of knowledge necessary for designing future-proof synchronization systems that can operate reliably under adversarial conditions.

References

[1] Министерство цифрового развития, связи и массовых коммуникаций РФ. *О федеральной целевой программе "Развитие телерадиовещания в Российской Федерации на 2009-2018 годы"* . Последняя редакция введена в действие с 27 декабря 2018 года постановлением Правительства РФ от 14 декабря 2018 г. №1559. 2009. URL: http://docs.cntd.ru/document/902161938 (visited on 03/22/2024).

[2] European Telecommunications Standards Institute. "Digital Video Broadcasting (DVB), Framing Structure, Channel coding, and Modulation for digital terrestrial television broadcasting system (DVB-T2)." In: ETSI EN 302755v1.1.1. Sept. 2009.

[3] Min Liang, Wei Chen, and Bi-qi Long. "OFDM Timing Synchronization Schemes in SFN Channels." In: *2007 2nd IEEE Conference on Industrial Electronics and Applications*. 2007, pp. 1535–1538. DOI: 10.1109/ICIEA.2007.4318664.

[4] Md Sarwar Morshed. "Synchronization Performance in DVB-T2 system." MA thesis. 2009.

[5] Eduardo Garro et al. "Layered Division Multiplexing With Distributed Multiple-Input Single-Output Schemes." In: *IEEE Transactions on Broadcasting* PP (Apr. 2018), pp. 1–10. DOI: 10.1109/TBC.2018.2823643.

[6] Gregory W. Johnson et al. "An Evaluation of eLoran as a Backup to GPS." In: *2007 IEEE Conference on Technologies for Homeland Security*. 2007, pp. 95–100. DOI: 10.1109/THS.2007.370027.

[7] ITU Radiocommunication Assembly. *Standard Frequencies and Time Signals*. Tech. rep. TF.768-2. Question ITU-R 106/7. International Telecommunication Union, 1997. URL: http://www.itu.int.

[8] *Recommendation ITU-R TF.1153-4: The operational use of two-way satellite time and frequency transfer employing pseudorandom noise codes*. TF Series: Time signals and frequency standards emissions. International Telecommunication Union (ITU). Aug. 2015.

[9] Xia Xue, Honglei Qin, and Hui Lu. "High-precision time synchronization of kinematic navigation system using GNSS RTK differential carrier phase time transfer." In: *Measurement* 176 (2021), p. 109132. ISSN: 0263-2241. DOI: https://doi.org/10.1016/

j.measurement.2021.109132. URL: https://www.sciencedirect.com/science/article/pii/S0263224121001597.

[10] John Fischer and Paul E. Myers. "Methods for Subnanosecond Time Synchronizing using RTK Receivers and Devices Thereof." Patent Application Publication US 2015/0268352 A1. Related U.S. Provisional Application No. 61/815,116, filed on Apr. 23, 2013. Sept. 24, 2015.

[11] Cillian O'Driscoll. "GNSS Solutions: Carrier phase and its measurement for GNSS." In: *Inside GNSS* 5.4 (July 2010), 18–22.

[12] Andrew Dempster. "How vulnerable is GPS?" In: *Position* 20 (2005), pp. 64–67.

[13] Transportation Infrastructure. "Vulnerability assessment of the transportation infrastructure relying on the global positioning system." In: *Center, John A. Volpe Nat. Transp. Syst., Tech. Rep* (2001).

[14] J. Seo and M. Kim. "eLoran in Korea – Current status and future plans." In: *Proceedings of the European Navigation Conference*. Vienna, 2013, pp. 23–27.

[15] Ali Khalajmehrabadi et al. "Real-Time Rejection and Mitigation of Time Synchronization Attacks on the Global Positioning System." In: *IEEE Transactions on Industrial Electronics* 65.8 (2018), pp. 6425–6435. DOI: 10.1109/TIE.2017.2787581.

[16] Juraj Machaj et al. "Impact of GPS interference on time synchronization of DVB-T transmitters." In: vol. 2021. Hindawi Limited, 2021, pp. 1–11.

[17] Bengt Hellstrom. "GPS-free synchronization of Digital Terrestrial TV and Mobile TV distribution networks." In: NID2890 A2. Sweden, Aug. 2007.

[18] Nicholas Ciarleglio, Thomas Edwards, and Robert Welch. "Large scale PTP: How big can it get?" In: *SMPTE 2016 Annual Technical Conference and Exhibition*. 2016, pp. 1–13. DOI: 10.5594/M001705.

[19] FGUP VNIIFTRI. *Bulletin V 16 / 2018: Standard Time and Frequency Signals*. Federal Agency for Technical Regulation and Metrology. Principal Metrological Center of the State Service of Time and Frequency of the Russian Federation. Moscow, Russia, 2018.

[20] Alexandru Rusu. "possible applications of eloran system for positioning and timing synchronization in underground mining." In: June 2017. DOI: 10.5593/sgem2017/22/S09.057.

[21] Harald Dalichau. "Evaluation of different frequency bands regarding their qualification for Inhouse Powerline Communication." In: *5th International Symposium on Powerline Communications, Lund University, Malmö, Sweden, 4th-6th April*. 2001.

[22] Bhaskara Satyanarayana Margana et al. "A Simple SDR based Method to Spoof Low-End GPS aided Drones for Securing Locations." In: *2021 IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-of-Things (RAAICON)*. 2021, pp. 32–36. DOI: 10.1109/RAAICON54709.2021.9929965.

[23] u-blox AG. *MAX-M10S: Standard precision GNSS module*. UBX-20035208. Data sheet. Version R05. Mar. 2024.

[24] u-blox AG. *MAX-M10S: Standard precision GNSS module*. UBX-20053088. Integration manual. Version R04. June 2023.

[25] STMicroelectronics. *STM32H7 Nucleo-144 boards (MB1364)*. UM2407. User manual. Version Rev 4. Oct. 2023.

[26] u-blox AG. *u-blox M10 SPG 5.10: Standard precision GNSS firmware*. UBX-21035062. Interface description. Version R03. June 2023.

[27] SEMU Consulting. *Python Graphical GPS Client Application supporting NMEA, UBX, RTCM3 , NTRIP and SPARTN Protocols*. GitHub repository. Accessed: May 12, 2024. 2022. URL: https://github.com/semuconsulting/pygpsclient.

[28] DreamSourceLab. *DSView: An open source multi-function instrument for everyone*. https://github.com/DreamSourceLab/DSView. Accessed: May 12, 2024. 2024.

[29] STMicroelectronics. *STM32H723xE/G Datasheet*. Version Rev 4. DS13313, Production Data. Nov. 2023.

[30] STMicroelectronics. *Programming Manual: STM32F7 Series and STM32H7 Series Cortex-M7 processor programming manual*. Version Rev 5. PM0253. June 2019.

[31] Hui Zhou et al. "Frequency accuracy & stability dependencies of crystal oscillators." In: *Carleton University, Systems and Computer Engineering, Technical Report SCE-08-12* (2008).

[32] STMicroelectronics. *Reference Manual: STM32H723/733 Value Line Advanced Arm-based 32-bit MCUs*. Version Rev 3. RM0468. Dec. 2021.

[33] Texas Instruments Inc. *CD4020B, CD4024B, CD4040B Types*. SCHS030D. Data sheet. Version Rev D. Revised December 2003. Acquired from Harris Semiconductor. Dec. 2003.

[34] Xiangjie Kong et al. "Design of PPS Self-Calibration High-Precision Frequency Meter Based on STM32 Single Chip Microcomputer." In: *2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT)*. 2023, pp. 302–306. DOI: 10.1109/EIECT60552.2023.10442747.

[35] Ronald W. Schafer. "What Is a Savitzky-Golay Filter? [Lecture Notes]." In: *IEEE Signal Processing Magazine* 28.4 (2011), pp. 111–117. DOI: 10.1109/MSP.2011.941097.

[36] Caiwei Li et al. "Planning Large Single Frequency Networks for DVB-T2." In: *Broadcasting, IEEE Transactions on* PP (Sept. 2015), pp. 1–1. DOI: 10.1109/TBC.2015.2419179.

[37] European Broadcasting Union. *Frequency and Network Planning Aspects of DVB-T2*. EBU Technical Report TECH 3348. Geneva: European Broadcasting Union, May 2011.

[38] International Telecommunication Union. *Frequency and network planning aspects of DVB-T2*. ITU-R Report BT.2254-4. Geneva: International Telecommunication Union, Oct. 2020.

[39] Iñaki Eizmendi et al. "Empirical DVB-T2 Thresholds for Fixed Reception." In: *Broadcasting, IEEE Transactions on* 59 (June 2013), pp. 1–11. DOI: 10.1109/TBC.2013.2241358.

[40] P. Baeriswyl et al. "Time transfer with geodetic GPS receivers using code and phase observations." In: *Tenth European Frequency and Time Forum EFTF 96 (IEE Conf. Publ. 418)*. 1996, pp. 430–435. DOI: 10.1049/cp:19960090.

[41] Jerome Delporte et al. "GPS Carrier-Phase Time Transfer Using Single-Difference Integer Ambiguity Resolution." In: *International Journal of Navigation and Observation* 2008 (Jan. 2008). DOI: 10.1155/2008/273785.

[42] Kewei Zhang, Marco Spanghero, and Panagiotis Papadimitratos. "Protecting GNSS-based Services using Time Offset Validation." In: *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2020, pp. 575–583. DOI: 10.1109/PLANS46316.2020.9110224.

[43] u-blox AG. *Positioning Implementation: GNSS, aiding, hybrid positioning and Cell-Locate®*. UBXDOC-686885345-1826. Application Note. Version R02. C1-Public. Apr. 2024.

[44] ITU. "Ground-wave propagation curves for frequencies between 10 kHz and 30 MHz." In: *Recommendation International Telecommunication Union* 02 (2007), pp. 368–369.

[45] Christoph Classen. "Jamming the RIAS. Technical Measures against Western broadcasting in East Germany (GDR) 1945–1989." In: *Airy Curtains in the European Ether*. Nomos Verlagsgesellschaft mbH & Co. KG. 2013, pp. 321–346.

[46] Michael Lombardi. "Time Measurement." In: Feb. 2014, 21 p.

[47] Jingcheng Li et al. "A method for estimating the low frequency coupling characteristics of a ferrite-cored rod antenna to a long conductor." In: *Progress In Electromagnetics Research M* 75 (Jan. 2018), pp. 193–203. DOI: 10.2528/PIERM18081507.

[48] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE Std 1588-2008. Institute of Electrical and Electronics Engineers, 2008.

[49] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[50] O. Tange. "GNU Parallel - The Command-Line Power Tool." In: *;login: The USENIX Magazine* 36.1 (Feb. 2011), pp. 42–47. URL: http://www.gnu.org/s/parallel.

[51] J. D. Hunter. "Matplotlib: A 2D graphics environment." In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[52] LS telcom AG. *Planning DVB-T2: Advance and Challenge*. White Paper WHP01_3. LS telcom AG, Oct. 2010.

[53] Takuji Ebinuma. *GPS-SDR-SIM: Software-Defined GPS Signal Simulator*. GitHub repository. Accessed: May 12, 2024. 2023. URL: https://github.com/osqzss/gps-sdr-sim.

[54] Weike Feng et al. "Software-Defined Radio Implemented GPS Spoofing and Its Computationally Efficient Detection and Suppression." In: *IEEE Aerospace and Electronic Systems Magazine* 36.3 (2021), pp. 36–52. DOI: 10.1109/MAES.2020.3040491.

[55] Komal Kumar Songala et al. “Simplistic Spoofing of GPS Enabled Smartphone.” In: *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*. 2020, pp. 460–463. DOI: 10.1109/WIECON-ECE52138.2020.9397980.

[56] Werner Gurtner and Lou Estey. *RINEX: The Receiver Independent Exchange Format, Version 3.00*. Astronomical Institute, University of Bern and UNAVCO, Boulder, CO. Nov. 2007.

[57] Jinyuan Liu et al. “Robust Spoofing Detection for GNSS Array Instrumentation Based on C/N0 Difference Measurements.” In: *IEEE Transactions on Instrumentation and Measurement* 72 (2023), pp. 1–11. DOI: 10.1109/TIM.2023.3328684.

[58] Min Deng et al. “GNSS Spoofing Detection Based on Abnormal Receiver Noise and Carrier-to-Noise Ratio Metric.” In: *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*. 2022, pp. 1306–1311. DOI: 10.1109/ICCT56141.2022.10072895.

A Interrupt Handler Algorithm

```rust
#[embassy_executor::task]
async fn pps_sync_task(
    mut pps_exti_input: ExtiInput<'static>,
    mut pll_vcxo: DacChannel<'static, DAC1, 1, NoDma>,
) -> ! {
    let mut pll_dac_val: u16 = 2369;
    pll_vcxo.set(Value::Bit12Right(pll_dac_val));

    let mut count = 1;
    let mut sum = 0u64;
    const AVERAGING_COUNT: u64 = 100;
    let mut before = 0;

    loop {
        pps_exti_input.wait_for_rising_edge().await;
        let now = Instant::now().as_ticks();

        let diff = now - before;

        sum += diff;
        if count >= AVERAGING_COUNT {
            let avg = sum as f64 / AVERAGING_COUNT as f64;
            info!("Average diff: {}", avg);

            if sum > 10_000_000 * AVERAGING_COUNT {
                pll_dac_val -= 1;
            } else if sum < 10_000_000 * AVERAGING_COUNT {
                pll_dac_val += 1;
            }

            info!("Set DAC to: {}", pll_dac_val);
            pll_vcxo.set(Value::Bit12Right(pll_dac_val));

            sum = 0;
            count = 1;
        } else {
            count += 1;
            info!("PHASE: {}", now % 10_000_000);
        }

        before = now;
    }
}
```