UDC 621.39.83 **doi: 10.26906/SUNZ.2025.4.176-179**

Mykola Bikchentayev, Bohdan Boriak

National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine

DESIGN AND IMPLEMENTATION OF A SOFTWARE-DEFINED SPECTRUM ANALYZER BASED ON PLUTO-SDR

Abstract. This article studies how a Software-Defined Radio (SDR) can work as a real-time spectrum analyzer and compares two spectrum analyzer program designs for PlutoSDR. **The aim of the article**: to demonstrate how an SDR can be used as a spectrum analyzer that makes fast sweeps and clearly displays data over wide frequency spans. We compare two approaches to the program design: (1) a step-by-step sweep and (2) a fast sweep with fewer tuner steps and RBW/VBW used during signal processing. **The results obtained**: the second program has shorter sweep times and more stable power level estimates across the band. It also makes burst signals easier to see and supports continuous monitoring like an entry-level Real-Time Spectrum Analyzer (RTSA). **Conclusions**: an SDR configured with proper RBW/VBW and efficient rendering can provide good-enough real-time spectrum analysis for educational and engineering tasks.

Keywords: software-defined radio, SDR, spectrum analyzer, signal processing.

Introduction

Wireless signals are now an essential part of daily life: phones, Wi-Fi, Bluetooth, IoT devices, sensors in factories, RF trackers all share the same air [1]. Since many devices communicate at once, they can interfere with each other and cause slow data transfer or dropouts. Some bands are licensed and tightly managed [2]. Others (like 2.4 and 5.6 GHz) are unlicensed and represent free, public portion of the radio spectrum that anyone can use, so devices must use low power, short bursts, and polite spectrum access (i.e., a technique called Listen Before Talk, LBT) [3]. Designers try to keep energy inside the right channel (limit “adjacent-channel” leakage) and follow basic rules on power and emissions, while still providing good speed and battery life.

Real-world RF environment changes quickly—often in milliseconds. Signals can hop in frequency, start and stop in short bursts, or collide with strong nearby transmitters. Classic sweep tools (i.e., swept-tuned spectrum analyzers that tune across the band and measure one narrow slice at a time) often miss these brief events, so modern practice observes the spectrum as it changes over time. Even simple checks, such as confirming that a device keeps its energy inside the assigned channel, need an instrument that can see fast behavior, not only slow averages. This need leads directly to a different kind of tool [1].

A Real-Time Spectrum Analyzer (RTSA/RSA) is a test instrument that continuously watches the radio spectrum. It can catch very short events, trigger when a chosen condition occurs (for example, a burst above a limit or a mask violation), record the signal around that moment, and let us study it by frequency and by time. In practice, an RSA helps engineers find and explain interference, out-of-band emissions, spurious tones, and timing issues. Because it captures and stores the exact data segment, teams can repeat the analysis, consult the requirements, and issue fixes—useful both for specialists and enthusiasts. For this purpose different analyzer architectures exist [1].

RSA Architecture Types

Swept Spectrum Analyzer (SA). An SA, also called superheterodyne spectrum analyzer, downconverts the input and sweeps a narrow filter (resolution bandwidth, RBW) across the chosen frequency span, measuring one small slice at a time (Fig. 1, a). This approach gives strong dynamic range and accurate results for steady, continuous signals. However, because it looks at only one slice at any instant, rapid changes in a signal may be missed if they happen between slices [1].

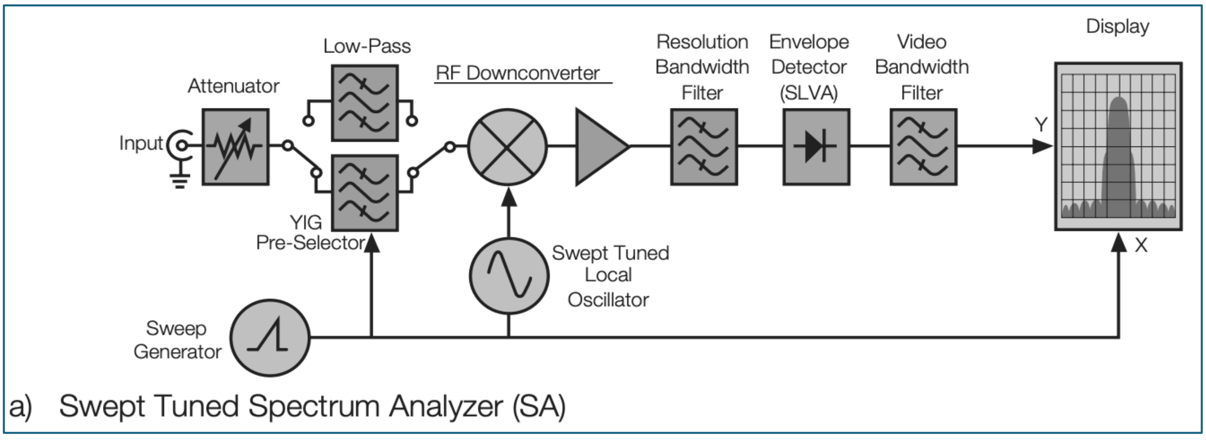

Fig. 1. Block diagram of SA analyser

Vector Signal Analyzer (VSA). A VSA downconverts a signal to IF/baseband, digitizes it, and uses digital downconversion, filtering, and an FFT to produce spectrum/time views and modulation metrics (e.g., EVM, channel power).

Because it analyzes stored IQ blocks, it is excellent for known, repeatable waveforms (Fig. 2, b). Its limits are that it is blind between captures (so rare or very short events can be missed).

Thus, a VSA is powerful tool, but it is not designed for continuous, gap-free watching of the spectrum [1].

Real-Time Spectrum Analyzer (RTSA/RSA). "Real-time" originated in early digital system simulations and means the simulation processes events as fast as the real system does (Fig. 2, c). An RTSA continuously watches a chosen slice of the radio band. The signal is sampled fast enough to satisfy Nyquist criteria to catch very short events like brief bursts, hops, or unwanted tones [1].

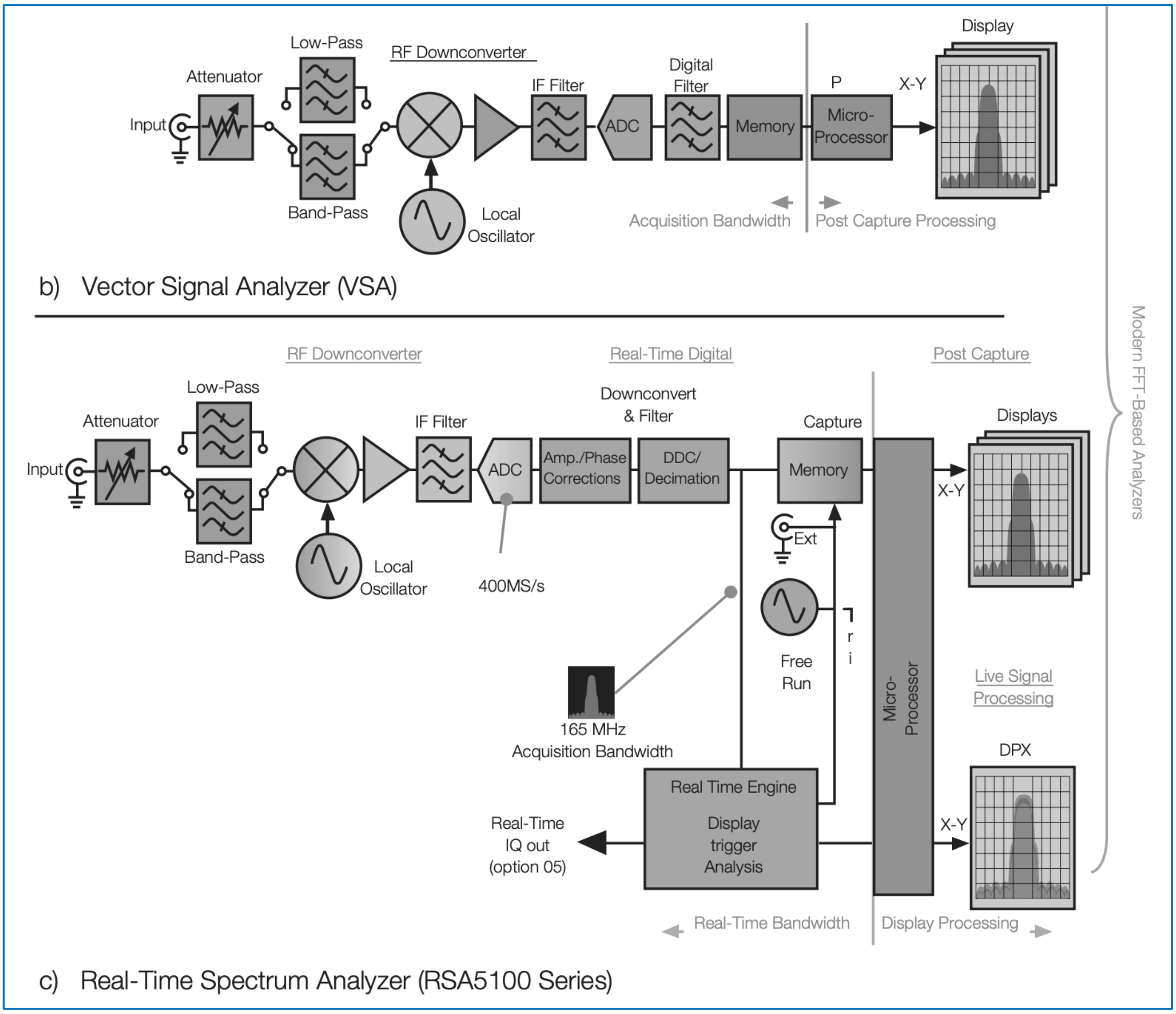

Fig 2. Block diagram of VSA and RTSA analyzers

SDR as RSA

Software-Defined Radio (SDR) can be programmed to act as an RSA. Due to the nature of SDR, this approach offers flexible, affordable solution that can be adapted to a given project. The viewing range and detection rules can be set in software. We can also record data for replay and sharing, and as well as plug it into tools or tests—all without buying a new RSA device each time a change is needed.

Compared with small swept analyzers (e.g. TinySA), an SDR-based setup is better at catching short, changing signals and avoids the blind spots that appear when reading the signal one slice at a time as all these issues can be addressed in the programming. Although professional benchtop RSAs deliver the highest calibrated accuracy and widest coverage, they are usually much more expensive; an SDR provides a "good enough" real-time visibility at a fraction of the cost. The main trade-off is setup effort: a reliable radio front end and a capable computer are needed, plus a bit of tuning, to make sure everything runs smoothly. We will use PlutoSDR as a basis for our RSA. The program that will monitor the spectrum will be based off the existing ADALM Pluto SDR Spectrum Analyzer publicly available on GitHub under the MIT license [4] (Fig. 3). This program uses Python as a programming language and pyadi-iio library to communicate with PlutoSDR. Below you can see a block diagram of this program.

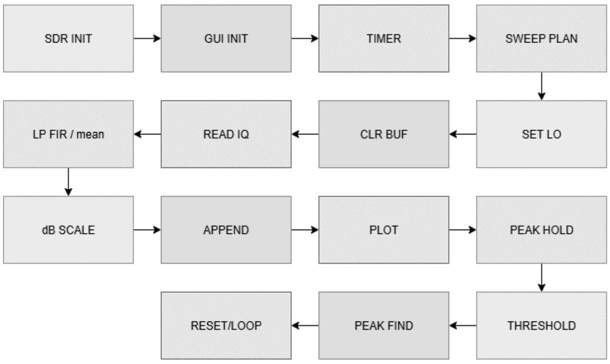

Fig 3. Block diagram of ADALM-Pluto-Spectrum-Analyzer program

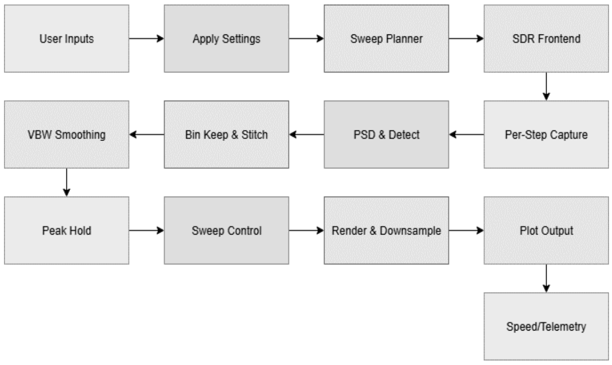

Fig 4. Block diagram of a "fast spectrum analyzer" program

The program begins with SDR INIT and GUI INIT: it connects to PlutoSDR, sets sample rate, gain (AGC or manual), and buffer sizes, then builds the plot, controls, markers, and status labels. A TIMER (fires every 10–30 Hz) breaks work into small chunks so the UI stays smooth. Each timer tick follows the SWEEP PLAN: SET LO tunes to the next center frequency, CLR BUF drops stale samples, and READ IQ captures a fresh block of IQ data. LP FIR / mean applies a filter to reduce jitter. dB SCALE converts to log units, APPEND stores frequency–level pairs, and PLOT draws these pairs across the span.

PEAK HOLD keeps the highest level for each frequency. THRESHOLD compares the current (or peak-hold) amplitude to a user-set threshold. If exceeded, trigger a visual cue (and optionally a log/notification). PEAK FIND runs at sweep end to list local maxima. RESET/LOOP wraps to the start of the frequency plan and continues the program. This ensures continuous, real-time monitoring: the loop repeats with the same parameters unless changed by the user, refreshing traces and peak-hold indefinitely.

Although this program offers helpful overlays, markers, threshold alerts, live point-by-point updates, and a UI with lots of inputs, it has limits on wide spans. It changes the tuner at every step, clears buffers, and draws each new point on the screen, so sweeping is slow, and very short bursts can be missed between steps. Its detector uses simple smoothing rather than standard RBW/VBW, so levels and narrow peaks may be less reliable. The edges between chunks of processed data can add small errors (stitching artifacts), and heavy screen updates can make the app feel busy and laggy when there are many points. Thus, a sweep from 100 MHz to 5.8 GHz may take up to 2 minutes.

For this reason, we need a program that runs the radio at a higher sample rate to reduce LO hops, sets the FFT size from a chosen RBW for clear and predictable resolution, and applies VBW after detection on linear power to smooth the trace in a proper, comparable way. It should draw once per sweep with fewer points, so the UI stays smooth. Block diagram of the new program can be seen in Fig. 4.

The sweep starts with User Inputs and Apply Settings: a user chooses start/stop, RBW (detail), VBW (smoothing), gain, and view mode (clean, with less points displayed on the chart, or granular).

The program validates these, computesFFT size from RBW, clears old peaks/buffers, and initializes timers. Next, Sweep Planner splits the span into overlapping steps for processing and builds one global frequency axis. In SDR Frontend, PlutoSDR is configured (sample rate, RF bandwidth, gain) and will tune to each step central frequency and stream IQ data.

Per step, Per-Step Capture tunes to the center and reads a fixed IQ block. PSD & Detect converts it with an FFT to power-versus-frequency and scales to dB. Bin Keep & Stitch takes only the middle “good” bins and places them into the global trace, so all steps join into one spectrum. After detection, VBW Smoothing applies a gentle time average at each frequency to reduce flicker. Peak Hold stores the highest seen level per bin to catch short bursts. Sweep Control advances steps, and at the end records sweep time for the speed readout.

For display, Render & Downsample keeps the UI fast by lightly averaging (clean view) and drawing fewer points while preserving shape. Plot Output shows live dB versus frequency, with an optional dashed peak trace. Speed/Telemetry reports last sweep time and sweeps per second. In contrast, with the first program, this one is built for speed and stable results on wide spans: it has a faster sample rate, needs fewer tuner steps, sets FFT size from RBW (for proper frequency detail), and applies VBW after detection to smooth the trace in a standard way. It renders once per sweep and limits plotted points, so the UI stays smooth, and the levels are more trustworthy across the whole band. This makes this program work faster over big frequency ranges, gives us faster sweeps, cleaner displays, and more consistent measurements. In this case a sweep from 100 MHz to 5.8 GHz takes around 400 ms.

Conclusions

An SDR working as a real-time spectrum analyzer can watch one band continuously, trigger on quick events, and save IQ data for later study. These features support faster interference hunting, clearer compliance checks, and an open toolchain that we can improve over time. Two spectrum analyzer program designs for PlutoSDR were compared. The first design is simple and shows each point on the screen, but it becomes slow for wide spans and may miss narrow or short signals. The second design reduces tuner steps, sets FFT size from the chosen RBW, applies VBW after detection, and draws once per sweep. As a result, it gives faster scans, a cleaner display, and more stable levels across the band. The performance of the program is limited only by the chosen RBW/VBW, memory needs for long captures, and SDR computational resources, nevertheless, the proposed method is practical for study and engineering work. Future work may include adding a better level calibration, custom triggers, and report export to bring the system closer to professional instruments while keeping the benefits of open, software-based tool.

The screenshot of a program can be seen on Fig. 5.

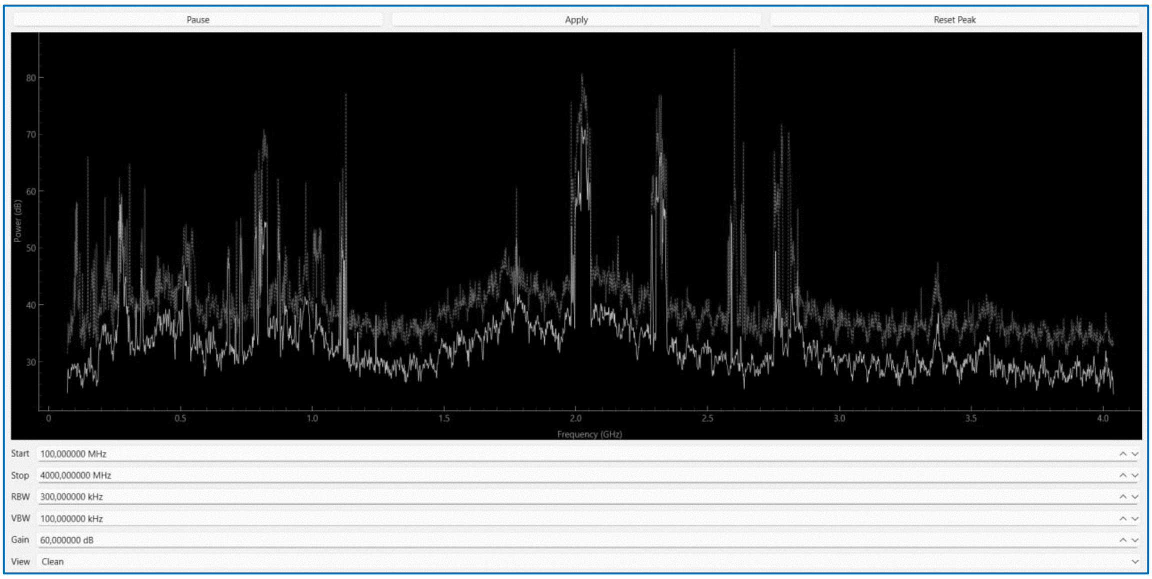

Fig 5. "Fast spectrum analyzer" program UI

References

1. Tektronix. Spectrum Analyzer How-To Guide: What They Are, What They Measure, & How to Use Them. Tektronix. URL: https://www.tek.com/en/documents/primer/what-spectrum-analyzer-and-why-do-you-need-one
2. Schirn A. IEEE 1932.1-2024: Licensed and Unlicensed Wireless Networks. The ANSI Blog. 20.05.2025 URL: https://blog.ansi.org/ansi/ieee-1932-1-2024-licensed-and-unlicensed-spectrum/
3. Wirepas. How wireless mesh and WLAN can live happily together. Wirepas Blog. 11.06.2019 URL: https://wirepas.com/blog/how-wireless-mesh-and-wlan-can-live-happily-together/
4. Fromconcepttocircuit. ADALM-Pluto-Spectrum-Analyzer: Real-time spectrum analyzer for ADALM Pluto SDR with GUI, peak hold, and markers. Репозиторій GitHub. URL: https://github.com/fromconcepttocircuit/ADALM-Pluto-Spectrum-Analyzer

Received (Надійшла) 20.08.2025
Accepted for publication (Прийнята до друку) 29.10.2025

Відомості про авторів / About the Authors

Бікчентаєв Микола Олексійович – аспірант, Національний університет «Полтавська політехніка ім. Юрія Кондратюка», Полтава, Україна;
Mykola Bikchentayev – PhD student, National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine;
e-mail: niko.bikcheb@gmail.com; ORCID Author ID: https://orcid.org/0009-0008-1350-516X.

Боряк Богдан Радиславович – кандидат технічних наук, доцент кафедри автоматики, електроніки та телекомунікацій, Національний університет «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна;
Bohdan Boriak – PhD, Associate Professor at the Department of Automation, Electronics and Telecommunications, National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine;
e-mail: boriakbr@nupp.edu.ua, ORCID Author ID: http://orcid.org/0000-0002-8114-7930;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=58983614200.

Дослідження використання Pluto-SDR як аналізатора спектру

М. О. Бікчентаєв, Б. Р. Боряк

Анотація. У статті досліджується використання програмно-керованого радіо (SDR) у ролі аналізатора спектру та порівнюються два підходи до розробки програми аналізатора для PlutoSDR. **Мета статті** — показати, як SDR можна використати у ролі аналізатора спектру, що здатен швидко здійснювати обробку широкого радіочастотного діапазону та наочно відображати отримані дані. Ми порівнюємо два підходи до проектування: (1) покроковий аналіз та (2) швидкий аналіз який здійснює менша переналаштувань частоти та застосує RBW/VBW під час обробки сигналу. **Результати**: другий варіант має вищу швидкість роботи та більш стабільні оцінки рівня потужності по заданому діапазону. Він також краще фіксує короткочасні сигнали і при цьому безперервно здійснює обробку сигналу у вибраному діапазоні, подібно до RTSA початкового рівня, на кшталт TinySA. **Висновок**: SDR із коректно налаштованими RBW/VBW та оптимізованим процесом відображення результатів може забезпечити достатньо якісний аналіз спектру в реальному часі.

Ключові слова: програмно-кероване радіо, SDR, аналізатор спектру, обробка сигналів.