Development of a Passive RF Spectrum Intelligence Platform for Drone Signal Detection and Analysis using HackRF One and Python

Maj Gagandeep Jaswal
Military College of Telecommunication Engineering (MCTE), Mhow

---***---

Abstract - *The increasing use of unmanned aerial systems (UAS) in military, commercial and civilian applications has significantly increased the importance of passive radio frequency (RF) monitoring for spectrum awareness and authorised signal analysis. Conventional drone detection techniques rely primarily on radar, electro-optical sensors or active RF interrogation, which may not always be suitable for laboratory research, training environments or passive spectrum observation. Software Defined Radio (SDR) offers a flexible and cost-effective approach for analysing RF activity without transmitting energy into the electromagnetic spectrum. This paper presents the design of a Python-based Passive RF Spectrum Intelligence Platform using HackRF One for real-time spectrum monitoring, RF signal visualisation and candidate signal analysis. The proposed software continuously scans user-defined frequency ranges, performs Fast Fourier Transform (FFT)-based spectrum analysis, generates waterfall visualisation and records candidate RF events for subsequent analysis. The platform incorporates a graphical user interface developed using PyQt6 together with SQLite-based event management to organise signal metadata and recorded IQ data. The modular software architecture enables future integration of advanced signal classification algorithms while maintaining separation between passive RF observation and protocol-specific decoding. The proposed platform is intended for authorised spectrum monitoring, education and software defined radio research.*

Keywords: Software Defined Radio, HackRF One, Passive RF Monitoring, Spectrum Intelligence, FFT, Waterfall Display, IQ Recording, Python, Electromagnetic Spectrum Operations.

1. INTRODUCTION

The electromagnetic spectrum has become one of the most valuable operational resources supporting communication, navigation, surveillance, intelligence gathering and command-and-control functions. Modern wireless systems occupy increasingly congested portions of the radio spectrum, making efficient spectrum monitoring essential for both research and operational applications. Among these wireless platforms, unmanned aerial systems (UAS) represent one of the fastest-growing technologies and employ a variety of RF communication techniques for control, telemetry and payload transmission.

Software Defined Radio (SDR) has transformed RF research by replacing dedicated hardware processing with programmable software implementations. Instead of requiring separate hardware receivers for different communication standards, SDR platforms digitise the received spectrum and implement signal processing algorithms in software. This flexibility enables rapid development of customised RF analysis tools and supports experimentation across multiple communication bands.

HackRF One is a widely used SDR platform capable of receiving signals over a broad frequency range extending from approximately 1 MHz to 6 GHz. Combined with Python and modern scientific computing libraries, HackRF One provides an accessible platform for implementing spectrum analysis, RF signal visualisation and data acquisition systems. Python libraries such as NumPy, SciPy, PyQt6 and PyQtGraph enable development of sophisticated graphical interfaces together with real-time digital signal processing algorithms.

The objective of this work is to design a passive RF spectrum intelligence platform capable of continuously monitoring selected frequency bands, visualising spectrum occupancy, identifying candidate RF transmissions and maintaining a searchable database of observed events. Rather than attempting universal decoding of proprietary or encrypted communication protocols, the proposed platform concentrates on passive spectrum observation, RF signal characterisation and recording of IQ samples for authorised offline analysis. This design philosophy provides a flexible foundation for future enhancement while maintaining compatibility with laboratory research and educational applications.

2. Literature Review

Drone detection technologies have traditionally relied upon radar, acoustic sensors, electro-optical imaging and RF monitoring systems. Radar systems provide long-range detection capabilities but often encounter challenges when observing small unmanned aerial vehicles possessing limited radar cross sections. Electro-optical and infrared sensors offer high target identification accuracy under favourable environmental conditions but their effectiveness may degrade during adverse weather, poor illumination or line-of-sight obstruction.

Passive RF monitoring has emerged as an important complementary technology because it does not require transmission of electromagnetic energy. Instead, passive systems analyse naturally occurring RF emissions generated

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 08 | Aug 2026 www.irjet.net p-ISSN: 2395-0072

by authorised communication devices operating within the monitored spectrum. Such systems are particularly useful for laboratory experimentation, electromagnetic spectrum analysis and communication research where spectrum occupancy rather than target engagement represents the primary objective.

Recent developments in Software Defined Radio have significantly reduced the cost of implementing sophisticated RF monitoring systems. SDR devices such as HackRF One, RTL-SDR and USRP platforms provide researchers with programmable access to radio frequency signals over wide bandwidths. Combined with open-source software frameworks including GNU Radio and Python-based scientific libraries, these platforms enable implementation of spectrum analysers, protocol analysers and custom signal processing algorithms.

Although several commercial spectrum monitoring solutions exist, many require specialised hardware and proprietary software environments that restrict flexibility and increase deployment cost. An open, Python-based architecture provides greater extensibility, allowing researchers to integrate customised signal detection algorithms, graphical interfaces, database management systems and future machine learning modules within a single software environment.

3. Research Motivation

The increasing complexity of modern RF environments has created a requirement for flexible spectrum monitoring platforms capable of supporting research, education and authorised laboratory experimentation. Existing commercial solutions often provide limited opportunities for algorithm development and software customisation. Similarly, many SDR demonstration applications concentrate on signal reception without integrating spectrum visualisation, event management and structured recording of RF observations. The motivation behind this work is to develop an integrated software platform that combines RF spectrum monitoring, graphical visualisation and database management within a single Python application. Such a platform enables researchers to investigate spectrum occupancy, analyse candidate RF signals and organise recorded observations for future processing. The modular design also permits future incorporation of intelligent signal classification algorithms while maintaining compatibility with standard SDR hardware.

4. Proposed System Architecture

The proposed Passive RF Spectrum Intelligence Platform has been designed as a modular software architecture integrating Software Defined Radio (SDR), Digital Signal Processing (DSP), database management and graphical visualisation within a single Python application. The system enables passive observation of radio frequency activity by continuously monitoring user-defined frequency ranges and analysing received RF energy without transmitting electromagnetic signals.

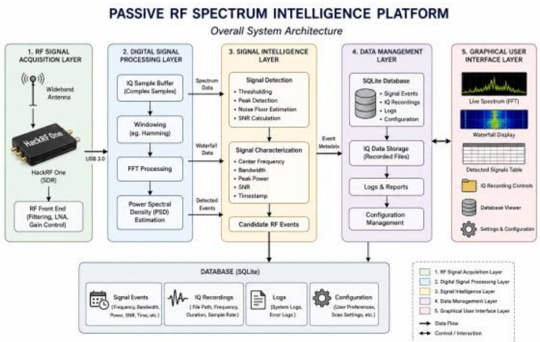
The overall architecture consists of five major functional layers: RF Signal Acquisition Layer

4.1. Digital Signal Processing Layer
4.2. Signal Intelligence Layer
4.3. Data Management Layer
4.4. Graphical User Interface Layer

The RF Signal Acquisition Layer interfaces with HackRF One to receive digitised In-phase (I) and Quadrature (Q) samples over the selected frequency range. The digitised samples are processed by the Digital Signal Processing Layer, where FFT-based spectrum analysis, power estimation and waterfall generation are performed.

The Signal Intelligence Layer analyses the processed spectrum to identify candidate RF transmissions based on measurable parameters including signal bandwidth, peak power, occupied bandwidth and spectral characteristics. Candidate events are recorded in a structured SQLite database together with frequency, timestamp and signal metadata.

The graphical interface provides operators with a real-time spectrum analyser, waterfall display, detected signal table and IQ recording controls, allowing authorised users to monitor RF activity and review previously recorded observations.



[1] Fig. 1 Overall System Architecture

Figure Caption: Overall architecture of the proposed Passive RF Spectrum Intelligence Platform.

5. Hardware Design

The hardware architecture has been intentionally designed to minimise complexity while providing sufficient flexibility for RF experimentation and software development. The primary hardware component is HackRF One, which

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 08 | Aug 2026 www.irjet.net p-ISSN: 2395-0072

functions as a broadband Software Defined Radio receiver capable of digitising RF signals over a wide frequency range. The host computer executes the developed Python application and performs all digital signal processing functions. Communication between HackRF One and the host computer is established through a USB interface, enabling high-speed transfer of digitised IQ samples.

The hardware configuration consists of:

5.1. HackRF One SDR
5.2. Personal Computer/Laptop
5.3. Wideband Antenna
5.4. USB 3.0 Interface
5.5. External Power (if required)

The SDR receives RF signals through the connected antenna and digitises them into complex IQ samples. These samples are forwarded to the host computer, where all spectrum processing and visualisation operations are performed entirely in software.

The use of software-based signal processing significantly improves flexibility compared to dedicated spectrum analysers since processing algorithms can be modified without requiring hardware redesign.

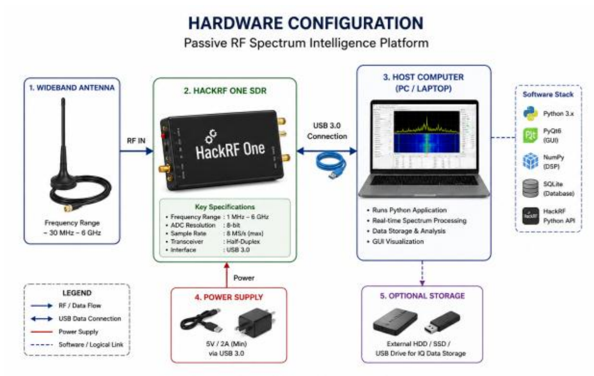


Fig. 2 Hardware Configuration of the Proposed Platform

[2] Fig. 2 Hardware Configuration

Figure Caption: Hardware configuration of the proposed RF Spectrum Intelligence Platform.

6. Software Architecture

The software has been developed using Python because of its extensive scientific computing ecosystem, cross-platform compatibility and rapid development capability. The software follows a modular architecture to simplify maintenance, future expansion and integration of additional SDR devices.

The primary software modules include:

6.1 HackRF Interface Modules

This module establishes communication with HackRF One and manages SDR configuration parameters such as centre frequency, sample rate and receiver gain. The module provides an abstraction layer that isolates hardware-specific operations from higher-level processing modules.

6.2 Spectrum Processing Module

This module performs FFT computation on incoming IQ samples to generate the real-time spectrum display. Power Spectral Density (PSD) estimation is performed to calculate signal strength across the monitored frequency range.

The processed spectrum is simultaneously utilised for:

6.2.1 Live spectrum display
6.2.2 Waterfall generation
6.2.3 Peak detection
6.2.4 Signal bandwidth estimation

6.3 Signal Detection Module

The signal detection module continuously evaluates the processed spectrum and identifies candidate RF transmissions exceeding configurable detection thresholds. Detected signals are characterised using parameters such as:

6.3.1 Centre Frequency
6.3.2 Signal Bandwidth
6.3.3 Peak Power
6.3.4 Signal-to-Noise Ratio (SNR)
6.3.5 Detection Timestamp

These parameters are stored within the database for future analysis.

6.4 Database Module

SQLite has been selected as the backend database because it requires no dedicated database server while providing reliable storage for recorded observations.

The database stores:

6.4.1. Signal Events
6.4.2. IQ Recording Metadata
6.4.3. Application Logs
6.4.4. User Configuration

This structured storage enables rapid retrieval of previously recorded observations for offline analysis.

6.5 User Interface Module

The graphical interface has been developed using PyQt6 to provide an interactive dashboard supporting real-time monitoring.

The interface consists of:

6.5.1. Live FFT Spectrum
6.5.2. Waterfall Display
6.5.3. Frequency Control Panel
6.5.4. Detected Signal Table
6.5.5. IQ Recording Controls
6.5.6. Database Viewer

The modular user interface permits future addition of advanced analysis tools without affecting existing software functionality.

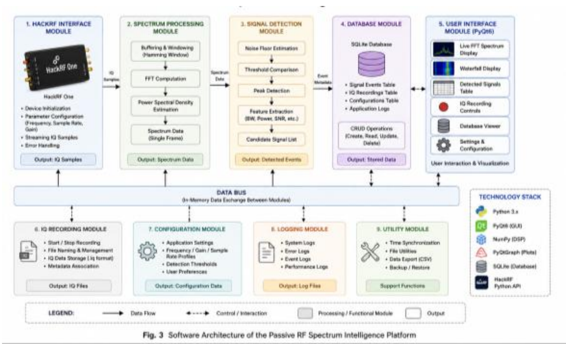


Fig. 3 Software Architecture

Figure Caption: Functional software modules developed in Python.

7. Signal Processing Methodology

The proposed platform performs digital signal processing using Fast Fourier Transform (FFT) to convert the received time-domain IQ samples into their corresponding frequency-domain representation.

The received IQ samples are first buffered before applying a suitable window function to reduce spectral leakage. The windowed samples are then transformed using FFT to obtain the frequency spectrum.

The frequency spectrum is analysed to determine:

7.1. Peak Signal Strength
7.2. Occupied Bandwidth
7.3. Average Noise Floor
7.4. Spectral Occupancy

Signals exceeding the configured detection threshold are marked as candidate RF transmissions and stored in the event database.

The waterfall display is generated by continuously stacking successive FFT outputs, enabling operators to observe temporal changes in spectrum occupancy.

Future versions of the platform may incorporate additional DSP algorithms for signal classification and modulation recognition while preserving the existing modular architecture.

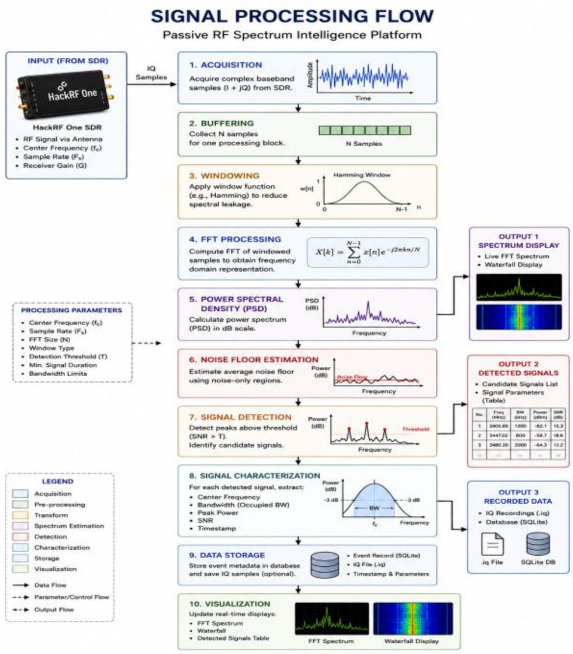


Fig. 4 Signal Processing Flow

Figure Caption: FFT-based signal processing workflow employed in the proposed platform.

8. REFERENCES

[1] S. Haykin, Communication Systems, 5th ed. New York, NY, USA: Wiley, 2009.

[2] J. G. Proakis and M. Salehi, Digital Communications, 5th ed. New York, NY, USA: McGraw-Hill, 2008.

[3] S. M. Kay, Fundamentals of Statistical Signal Processing, Upper Saddle River, NJ, USA: Prentice Hall, 1998.

[4] B. Sklar, Digital Communications: Fundamentals and Applications, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2001.

[5] R. G. Lyons, Understanding Digital Signal Processing, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2011.

[6] M. Ossmann, "HackRF One Documentation," Great Scott Gadgets, 2025. Available: HackRF Documentation

[7] M. Ossmann, "HackRF One Hardware Specifications," Great Scott Gadgets, 2025. Available: HackRF One Hardware Guide

[8] GNU Radio Project, "GNU Radio Documentation," Available: GNU Radio Documentation

[9] M. Ossmann, "Third-Party Software Compatible with HackRF," Great Scott Gadgets, 2025. Available: HackRF Software Support

[10] R. Akeela and B. Dezfouli, "Software-defined Radios: Architecture, State-of-the-art, and Challenges," IEEE Communications Surveys & Tutorials, 2018.

[11] F. Maggi and A. Guglielmini, "RFQuack: A Universal Hardware-Software Toolkit for Wireless Protocol Security Analysis and Research," 2021.

[12] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming and A. Burg, "An Open-Source LoRa Physical Layer Prototype on GNU Radio," 2020.

[13] R. Zitouni, H. Bouaroua and B. Senouci, "Hardware-Software Codesign for Software Defined Radio: IEEE 802.11p Receiver Case Study," 2020.

[14] Python Software Foundation, "Python 3 Documentation." Available: Python Documentation

[15] NumPy Developers, "NumPy Documentation." Available: NumPy Documentation

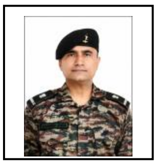
[16] SciPy Community, "SciPy Documentation." Available: SciPy Documentation

[17] PyQt Project, "PyQt6 Documentation." Available: PyQt Documentation

[18] SQLite Consortium, "SQLite Documentation." Available: SQLite Documentation

[19] ITU-R Recommendation SM.329-12, Unwanted Emissions in the Spurious Domain, International Telecommunication Union.

[20] IEEE Standard 802.11-2020, IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks.



Description " Maj Gagandeep Jaswal is an officer of the Corps of Signals, Indian Army, currently serving at the Military College of Telecommunication Engineering (MCTE), Mhow. His areas of interest include Software Defined Radio (SDR), Electronic Warfare, Electromagnetic Spectrum Operations (EMSO), wireless communications, and RF signal processing, with a focus on developing indigenous spectrum intelligence and passive RF monitoring solutions using Python and SDR technologies. "