

References

1. Hogan M., Piccarreta B. Interagency Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT). Gaithersburg: National Institute of Standards and Technology, 2018. – P. 185. – DOI: 10.6028/NIST.IR.8200.
2. Fagan M., Megas K. N., Scarfone K., Smith M. IoT Device Cybersecurity Capability Core Baseline. Gaithersburg: National Institute of Standards and Technology, 2020. – P. 23. – DOI: 10.6028/NIST.IR.8259A.
3. Ravidas S., Lekidis A., Paci F., Zannone N. Access control in Internet-of-Things: A survey // Journal of Network and Computer Applications, 2019. – Vol. 144. – Pp. 79–101. – DOI: 10.1016/j.jnca.2019.06.017.
4. Ragothaman K., Wang Y., Rimal B., Lawrence M. Access Control for IoT: A Survey of Existing Research, Dynamic Policies and Future Directions // Sensors, 2023. – Vol. 23. – No 4. – Art. 1805. – DOI: 10.3390/s23041805.
5. Ahsan M. S., Pathan A.-S. K. A Comprehensive Survey on the Requirements, Applications, and Future Challenges for Access Control Models in IoT: The State of the Art // IoT, 2025. – Vol. 6. – Art. 9. – DOI: 10.3390/iot6010009.
6. Khan M. A., Khan M. A., Jan S. U., Ahmad J., Jamal S. S., Shah A. A., Pitropakis N., Buchanan W. J. A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT // Sensors, 2021. – Vol. 21. – Art. 7016. – DOI: 10.3390/s21217016.
7. Empl P., Böhm F., Pernul G. Process-Aware Intrusion Detection in MQTT Networks // Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy (CODASPY '24). Porto, Portugal, 2024. – Pp. 91–102. – DOI: 10.1145/3626232.3653271.
8. Kayes A. S. M., Rahayu W., Dillon T., Chang E. A Survey of Context-Aware Access Control Mechanisms for Cloud and Fog Networks: Taxonomy and Open Research Issues // Sensors, 2020. – Vol. 20. – No 9. – Art. 2464. – DOI: 10.3390/s20092464.
9. Kalaria R., Kayes A. S. M., Rahayu W., Pardede E., Shahraki A. S. Adaptive context-aware access control for IoT environments leveraging fog computing // International Journal of Information Security, 2024. – Vol. 23. – Pp. 3089–3107. – DOI: 10.1007/s10207-024-00866-4.
10. Butun I., Sari A., Österberg P. Hardware Security of Fog End-Devices for the Internet of Things // Sensors, 2020. – Vol. 20. – No. 20. – Art. 5729. – DOI: 10.3390/s20205729.
11. Hindy H., Bayne E., Bures M., Atkinson R., Tachtatzis C., Bellekens X. Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset) // arXiv, 2020. arXiv:2006.15340.
12. Vaccari I., Chiola G., Aiello M., Mongelli M., Cambiaso E. MQTTset, a New Dataset for Machine Learning Techniques on MQTT // Sensors, 2020. – Vol. 20. – No 22. – Art. 6578. – DOI: 10.3390/s20226578.
13. Mudgerikar A., Sharma P., Bertino E. Edge-Based Intrusion Detection for IoT devices // ACM Transactions on Management Information Systems, 2020. – Vol. 11. – No. 4. – Art. 18. – DOI: 10.1145/3382159.
14. Asulba B., Souto P. F., Almeida L. Bringing IoT Intrusion Detection to the Edge // Proceedings of the 8th International Conference on Future Networks & Distributed Systems (ICFNDS '24). Marakech, Morocco, 2024. New York: ACM, 2024. – P. 10. – DOI: 10.1145/3726122.3726166.
15. Kotenko I. V., Saenko I. B., Lipatnikov V. A., Andreev I. A. Metod dinamicheskogo obnaruzheniya kiberugroz v raspredelennyh sistemah Interneta veshchej na osnove generativnyh modelej // Prikladnaya informatika, 2026. – Vol. 21. – No. 2. – Pp. 102–118. – DOI: 10.37791/2687-0649-2026-21-2-102-118.
16. Lipatnikov V. A., Shevchenko A. A., Kosolapov V. S., Sokol D. S. Metod obespecheniya informacionnoj bezopasnosti seti VoIP-telefonii s prognozom strategii vtorzenij narushitelya // Informacionno-upravlyayushchie sistemy, 2022. – No 1. – Pp. 54–67. – DOI: 10.31799/1684-8853-2022-1-54-67.
17. Lipatnikov V. A., Shevchenko A. A., Melehov K. V., Tkachev D. F. Metodika povysheniya zashchishchennosti seti peredachi dannyh ob'ektov kriticheskoy informacionnoj infrastruktury pri mnogoehtapnyh atakah // Informacionno-upravlyayushchie sistemy, 2024. – № 1. – Pp. 44–55. – DOI: 10.31799/1684-8853-2024-1-44-55.



ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ПОИСКА И ИДЕНТИФИКАЦИИ СИГНАЛОВ ДЛЯ SDR НА ОСНОВЕ ПЛАТФОРМЫ GNU RADIO

Баймаков Д. А.¹, Ткачев Д. Ф.², Воробьев П. В.³

DOI:10.21681/3034-4050-2026-3-40-48

Ключевые слова: программно-определяемое радио, GNU Radio, USRP B210, обнаружение БПЛА, спектральный анализ, Python-блоки, идентификация сигналов, база сигнатур.

Аннотация

Цель работы: создание законченного решения, позволяющего в реальном времени сканировать заданные диапазоны частот, автоматически фиксировать превышения порогового уровня мощности, определять параметры обнаруженных сигналов и идентифицировать их на основе предварительно сформированной базы сигнатур.

Результаты исследования: создан программно-аппаратный стенд для мониторинга электромагнитной обстановки в диапазонах работы каналов управления беспилотных летательных аппаратов (БПЛА) с использованием технологии программно-определяемого радио (SDR). Стенд реализован на базе приёмника USRP B200/B210 и открытой среды GNU Radio, развёрнутой в виртуальной машине Oracle VirtualBox. Подробно описаны этапы построения потокового графа обработки сигналов, разработки встроенных Python-блоков для порогового детектирования, измерения центральной частоты и ширины спектра, а также создания базы сигнатур с функцией автоматической идентификации. Приведены ключевые фрагменты программного кода и результаты экспериментальной проверки системы. Обсуждаются факторы, влияющие на точность измерений, и перспективы дальнейшего развития комплекса.

Практическая ценность заключается в создании целостной архитектуры программного комплекса, объединяющей в реальном времени спектральный анализ, измерение параметров сигнала и его идентификацию на основе базы сигнатур с использованием только открытых программных средств и доступного SDR-оборудования.

Введение

Беспилотные летательные аппараты (БПЛА) широко используют для управления и передачи телеметрии нелегализуемые диапазоны частот 2,4 ГГц, 5,8 ГГц, а также 433 МГц и 900 МГц [1]. Обнаружение и идентификация таких сигналов в реальной электромагнитной обстановке является актуальной задачей как для целей безопасности объектов, так и для противодействия несанкционированному применению дронов [2]. Технология программно-определяемого радио (SDR) предоставляет гибкий инструмент для построения недорогих и перестраиваемых систем радиомониторинга [3]. В отличие от традиционных аппаратных анализаторов спектра, SDR-приёмник в связке с открытым программным обеспечением позволяет не только визуализировать спектр, но и реализовывать произвольные алгоритмы цифровой обработки сигналов, адаптированные под конкретную задачу [4].

Целью настоящей работы является разработка законченного программного решения, позволяющего в реальном времени сканировать заданные диапазоны частот, автоматически фиксировать превышения порогового уровня мощности, определять центральную частоту и ширину полосы обнаруженного излучения, а также сохранять спектрограммы в базу данных с возможностью последующей идентификации вновь появляющихся сигналов. В статье последовательно описан процесс создания такого комплекса на базе приёмопередатчика USRP B200/B210 и среды GNU Radio, включая установку драйверов, построение потокового графа, разработку специализированных Python-блоков и реализацию подсистемы хранения и сравнения сигнатур.

Аппаратная и программная основа стенда

В качестве SDR-приёмника использовалось устройство Ettus Research USRP B200 (в дальнейшем применим и B210), подключаемое

1 **Баймаков Данил Алексеевич**, оператор научной роты Военной академии связи им. Маршала Советского Союза С. М. Буденного. Санкт-Петербург, Россия. E-mail: baymakovdaniil@gmail.com

2 **Ткачев Дмитрий Федорович**, начальник научно-исследовательского отдела научно-исследовательского центра Военной академии связи им. Маршала Советского Союза С. М. Буденного. Санкт-Петербург, Россия. E-mail: dimas.portnoy@inbox.ru

3 **Воробьев Павел Владимирович**, научный сотрудник научно-исследовательского отдела научно-исследовательского центра Военной академии связи им. Маршала Советского Союза С. М. Буденного. Санкт-Петербург, Россия. E-mail: pvorobyev98@gmail.com

к персональному компьютеру через интерфейс USB 3.0. Приёмник обеспечивает непрерывный частотный диапазон от 70 МГц до 6 ГГц и мгновенную полосу пропускания до 56 МГц, что с запасом перекрывает каналы управления большинства коммерческих БПЛА⁴. Для калибровки и тестовых излучений применялся передатчик HackRF One, способный генерировать немодулированную несущую или FM-тон в диапазоне 1 МГц – 6 ГГц.



Рис. 1. Аппаратная часть стенда

Обработка сигналов выполняется в среде GNU Radio, выбранной благодаря открытой архитектуре, богатой библиотеке стандартных блоков и возможности расширения функционала с помощью модулей на Python⁵. Установка производилась в два этапа:

1. На хост-системе (Windows) устанавливался менеджер пакетов Conda и дистрибутив Radioconda, включающий все необходимые зависимости Python.
2. Основная работа велась в виртуальной машине Oracle VirtualBox с образом instant-gnuradio на базе Linux, что обеспечило корректное взаимодействие с драйверами UHD (USRP Hardware Driver).

После подключения USRP B200 и активации компонента Cypress Westbridge в настройках виртуальной машины устройство определялось в GNU Radio как `uhd.usrp_source`, готовое к использованию в потоковых графах.

Программная часть комплекса построена по двухуровневой архитектуре. Ядром является граф обработки сигналов, сгенерированный в GNU Radio Companion в виде Python-класса `Base_detection`. Этот класс обеспечивает ряд действий:

1. Настройка и запуск приёмника USRP с заданными центральной частотой, частотой дискретизации и коэффициентом усиления.
2. Преобразование потока комплексных отсчётов в спектр мощности с помощью БПФ.
3. Визуализация спектра и «водопада» в реальном времени.
4. Вызов встроенных Python-блоков для анализа спектра.

Поверх этого низкоуровневого слоя работает управляющее приложение `MyEnhancedApp`, которое наследует класс `Base_detection` и расширяет его функциональность:

1. Обрабатывает сообщения от блоков-детекторов.
2. Ведёт базу данных известных сигналов в формате JSON.
3. Реализует логику записи фрагментов IQ-данных при обнаружении нового сигнала.
4. Предоставляет графический интерфейс для отображения результатов идентификации и управления записью.

Такая архитектура позволила отделить базовую функциональность SDR от высокоуровневой логики приложения, что значительно упростило разработку и отладку.

Настройка параметров сканирования: ширина полосы и частотное разрешение

Одной из ключевых особенностей разработанного комплекса является возможность гибкой настройки параметров сканирования в зависимости от решаемой задачи. В текущей конфигурации ширина анализируемой полосы составляет 2,4 МГц, что достаточно для обнаружения большинства узкополосных сигналов, однако для полноценного захвата широкополосных сигнатур БПЛА (например, каналов управления с FHSS, занимающих до 20 МГц) этого может оказаться недостаточно. В то же время прямое увеличение полосы обзора ведёт к росту вычислительной нагрузки и снижению частотного разрешения, что негативно сказывается на точности измерения параметров сигнала.

Основными переменными, определяющими ширину полосы сканирования и разрешение спектрограммы, являются `samp_rate` (частота дискретизации, Гц) и `fft_length` (размер блока БПФ). Ширина отображаемого спектра равна `samp_rate`, а частотное разрешение (расстояние между соседними бинами) вычисляется как отношение `samp_rate / fft_length`. Например, при `samp_rate = 2,4e6` и `fft_length = 2048` разрешение составляет около 1,17 кГц, а общая полоса – 2,4 МГц. Если требуется расширить полосу

⁴ Ettus Research. USRP B200/B210 Product Overview. URL: <https://www.ettus.com>

⁵ GNU Radio Documentation. URL: <https://wiki.gnuradio.org>

до 10 МГц, необходимо увеличить `samp_rate` до 10e6, однако при сохранении `fft_length = 2048` разрешение упадёт до 4,88 кГц, что может затруднить идентификацию узкополосных компонент сигнала. Для компенсации этого эффекта следует пропорционально увеличить `fft_length` (например, до 8192), что восстановит прежнее разрешение, но потребует больше вычислительных ресурсов.

Дополнительно в коде присутствует переменная `bandwidth`, задающая полосу пропускания аналогового фильтра USRP (по умолчанию 10e6). При изменении `samp_rate` рекомендуется синхронизировать `bandwidth` с новой частотой дискретизации, чтобы избежать алиасинга. Все перечисленные переменные доступны для редактирования в секции `#Variables` файла `Base_detection.py` и могут быть изменены как вручную, так и через графический интерфейс GNU Radio Companion.

Таким образом, выбор параметров сканирования сводится к поиску компромисса между шириной обзора, точностью измерений и вычислительной нагрузкой на компьютер. Практика показала, что для большинства задач обнаружения БПЛА оптимальным является соотношение `samp_rate = 10e6`, `fft_length = 8192`, обеспечивающее полосу 10 МГц при разрешении около 1,22 кГц, что приемлемо для идентификации типовых сигнатур.

Теоретические аспекты обнаружения сигналов БПЛА

Прежде чем перейти к описанию программных блоков, кратко рассмотрим факторы, влияющие на эффективность обнаружения сигналов БПЛА.

Большинство современных дронов используют для управления и передачи видео диапазоны 2,4 ГГц (2,400–2,4835 ГГц) и 5,8 ГГц (5,725–5,850 ГГц). Некоторые модели применяют 433 МГц (например, Chitwan QW350-02) или 868/915 МГц (Autel EVO II) [5]. Каналы управления часто используют широкополосные сигналы с псевдослучайной перестройкой рабочей частоты (FHSS) или расширением спектра, что требует от системы мониторинга достаточной мгновенной полосы пропускания и высокой чувствительности.

Мощность теплового шума на входе приёмника определяется выражением:

$$P_{\text{ш}} = k \cdot T \cdot B \cdot K_{\text{ш}}, \quad (1)$$

где k – постоянная Больцмана; T – шумовая температура, B – ширина полосы, $K_{\text{ш}}$ – коэффициент шума [6].

Таким образом, при расширении полосы в k раз мощность шума возрастает в k раз, что при фиксированной мощности сигнала приводит к снижению отношения сигнал/шум (SNR) на $10 \log_{10}(k)$ дБ. С другой стороны, для корректного воспроизведения широкополосных сигналов необходимо, чтобы полоса пропускания приёмника была не меньше ширины спектра сигнала. В разработанном комплексе частота дискретизации выбрана равной 2,4 МГц, что обеспечивает компромисс между чувствительностью и достаточной полосой обзора для обнаружения большинства каналов управления БПЛА.

Отношение сигнал/шум и точность измерений. Как показано в работах [7, 8], SNR напрямую влияет на точность измерения навигационных параметров и, как следствие, на точность позиционирования. В контексте данной задачи SNR определяет надёжность детектирования сигнала и точность определения его центральной частоты и ширины полосы. Для повышения SNR в программном комплексе применяется накопление спектров (интегрирование) и сглаживание скользящим средним.

Потоковый граф GNU Radio и пользовательские блоки

Потоковый граф, реализованный в классе `Base_detection`, включает следующие ключевые блоки:

1. `uhd.usrp_source` — источник IQ-отсчётов от USRP.
2. `stream_to_vector` и `keep_one_in_n` — преобразование потока в векторы и децимация для снижения вычислительной нагрузки.
3. `fft_vcc` — вычисление спектра мощности с использованием окна Блэкмана-Харриса для уменьшения растекания спектра.
4. `complex_to_mag_squared` и `moving_average` — вычисление квадрата модуля и сглаживание для получения усреднённого спектра.
5. `nlog10_ff` — перевод в логарифмический масштаб (дБ).
6. `qtgui_vector_sink_f` и `qtgui_waterfall_sink_c` — визуализация спектра и водопада.

Для реализации специфических функций детектирования и анализа сигналов были разработаны три встроенных Python-блока.

Блок `bandwidth_detector` является центральным элементом анализа. На вход он получает вектор спектральных отсчётов в дБ, а на выходе формирует сообщения с параметрами обнаруженного сигнала. Основные методы блока:

1. `estimate_noise_floor` — оценивает уровень шума как медианное значение спектра, что обеспечивает устойчивость к выбросам.

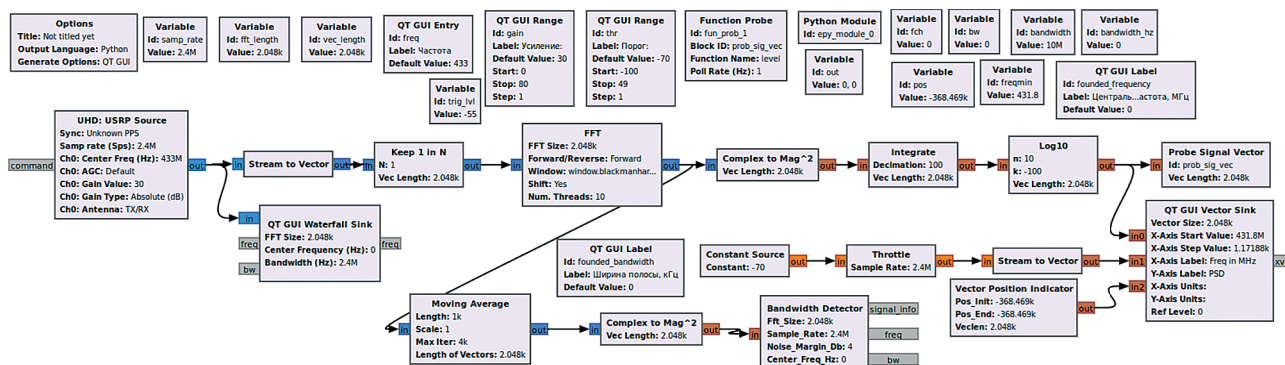


Рис. 2. Поточковый граф программы

2. `detect_signal_center` – находит индекс бина с максимальной амплитудой и сравнивает его с заданным порогом `threshold_db`.
3. `find_signal_bandwidth` – от найденного центра ищет границы влево и вправо до тех пор, пока значение не опустится ниже уровня `noise_floor + noise_margin_db`. Разность граничных частот, умноженная на частотное разрешение, даёт ширину полосы.

Блок публикует сообщения через порты `bw` и `freq`, передавая словарь с полями `bandwidth_hz`, `center_freq_hz`, `snr_db` и др. Ниже приведён фрагмент метода обнаружения центра сигнала:

```
python
def detect_signal_center(self,
spectrum_db):
    peak_bin = np.argmax(spectrum_db)
    if spectrum_db[peak_bin] > self.
threshold_db:
        return peak_bin
    return None
```

В модуле `epy_module_0` реализована вспомогательная функция `detect`, используемая для инициализации переменных в `Base_detection`. Она выполняет поиск каналов в спектре, мощность которых превышает порог, и возвращает частоту первого обнаруженного канала и его длину в бинах. Для уточнения центра используется средневзвешенное значение индексов с весами, равными мощности:

```
python
def avg_freq_pwr(data):
    total_power = np.sum(data)
    if total_power == 0: return 0
    indices = np.arange(len(data))
    return np.sum(indices * data) /
total_power
```

Блок `epy_block_0` генерирует вектор-маску, в которой заданный диапазон индексов (от `pos_init` до `pos_end`) заполняется значением 1000, а остальные элементы – -1000. Маска используется для визуального выделения на спектре

полосы обнаруженного сигнала (зелёная линия на графике).

Base detection				
Name	Size	Modified	Star	
Base_detection.py	18.9 kB	Yesterday	☆	
BD_Launcher.py	14.1 kB	Yesterday	★	
BD_Launcher.py.bak	13.2 kB	Mon	☆	
epy_block_0.py	2.3 kB	Yesterday	☆	
epy_block_1.py	3.9 kB	Yesterday	☆	
epy_module_0.py	3.3 kB	Yesterday	☆	
__pycache__	4 items	Yesterday	☆	
Signal_recording.grc	39.8 kB	Mon	☆	

Рис. 3. Составляющие элементы программы

Алгоритм обнаружения и измерения параметров сигнала

Полный цикл обработки одного кадра спектра выглядит следующим образом:

1. Поток IQ-отсчётов с выхода USRP буферизуется в вектор длиной `fft_length` (2048).
2. Выполняется БПФ с окном Блэкмана – Харриса, вычисляется квадрат модуля, производится сглаживание и логарифмирование.
3. Полученный спектр в дБ подаётся на вход блока `bandwidth_detector` и одновременно отображается на графике.
4. Блок `bandwidth_detector` оценивает шум, ищет пик выше порога и определяет границы сигнала.

Если сигнал обнаружен, блок публикует сообщение с параметрами.

Приложение `MyEnhancedApp` получает сообщение, обновляет информационную панель и, если сигнал не найден в базе, активирует кнопку записи.

Экспериментальная проверка с тестовым FM-тоном шириной 200 кГц на частоте 434,8 МГц показала, что центральная частота определяется с погрешностью не более ± 2 кГц, а ширина

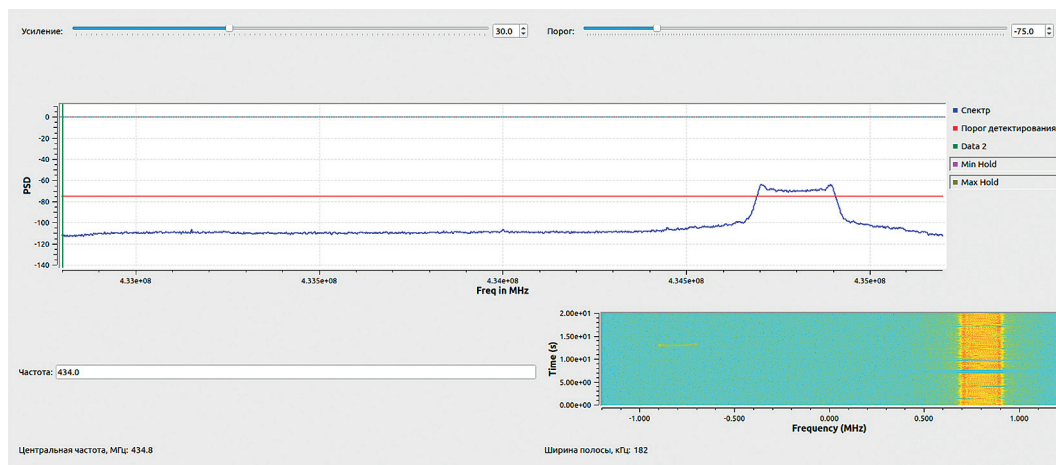


Рис. 4. Проверка работоспособности программы на определение параметров сигнала

полосы – с точностью около 5 %. Погрешность обусловлена конечной крутизной спадов спектра реального сигнала и дискретностью БПФ.

База данных сигналов и алгоритм идентификации

Для решения задачи идентификации была разработана подсистема хранения и сопоставления сигнатур, реализованная в классе MyEnhancedApp.

Каждая запись в базе описывается классом KnownSignal (dataclass) и содержит поля: id, name, center_freq_hz, bandwidth_hz, а также допуски freq_tolerance_hz (по умолчанию 5 кГц) и bw_tolerance_hz (30 кГц).

Класс SignalDatabase управляет списком сигналов в памяти и синхронизирует его с JSON-файлом на диске. При запуске приложения база загружается, а при добавлении нового сигнала – автоматически сохраняется.

При поступлении параметров обнаруженного сигнала вызывается метод find_matching_signal, который последовательно сравнивает частоту и ширину полосы с каждой записью. Если разности не превышают допусков, сигнал считается известным. Ниже представлен фрагмент кода метода сопоставления:

```
python
def find_matching_signal(self,
center_freq_hz, bandwidth_hz):
    for signal in self.signals:
        if abs(center_freq_hz - signal.
center_freq_hz) <= signal.freq_
tolerance_hz \
        and abs(bandwidth_hz - signal.
bandwidth_hz) <= signal.bw_ tolerance _
hz:
        return signal
    return None
```

Такой подход устойчив к небольшим погрешностям измерений и естественным флуктуациям параметров сигнала.

Механизм записи и пополнения базы

При обнаружении сигнала, отсутствующего в базе, оператору предоставляется возможность записать 100-миллисекундный фрагмент IQ-данных и сохранить его в базу. Для этого в MyEnhancedApp реализованы:

1. file_sink – блок GNU Radio, подключаемый параллельно к выходу USRP и управляемый программно.
2. record_timer – таймер Qt, ограничивающий длительность записи.
3. Диалоговое окно с полем ввода имени сигнала.

После завершения записи временный файл temp_signal.dat переименовывается в signal_<id>_<name>.dat, а параметры сигнала добавляются в JSON-базу. Такой подход позволяет накапливать библиотеку сигнатур, которую в дальнейшем можно использовать для автоматической идентификации типов БПЛА.

```
1 [
2 {
3   "id": 6,
4   "name": "Test1",
5   "center_freq_hz": 433450000.0,
6   "bandwidth_hz": 5000.0,
7   "freq_tolerance_hz": 30000,
8   "bw_tolerance_hz": 100000
9 },
10 {
11   "id": 7,
12   "name": "Test2",
13   "center_freq_hz": 432850000.0,
14   "bandwidth_hz": 5000.0,
15   "freq_tolerance_hz": 30000,
16   "bw_tolerance_hz": 100000
17 },
18 {
19   "id": 8,
20   "name": "Test3",
21   "center_freq_hz": 433740500.0,
22   "bandwidth_hz": 53000.0,
23   "freq_tolerance_hz": 30000,
24   "bw_tolerance_hz": 100000
25 }
26 ]
```

Рис. 5. Содержимое JSON-файла базы сигналов

Экспериментальные результаты

Работоспособность комплекса проверялась в лабораторных условиях с использованием генератора HaskRF и реальных сигналов БПЛА.

Основные достигнутые характеристики:

1. Диапазон сканирования: 70 МГц – 6 ГГц (ограничен возможностями USRP).
2. Мгновенная полоса анализа: 2,4 МГц (может быть увеличена ценой снижения чувствительности).

3. Пороговая чувствительность: около -100 дБм при полосе 200 кГц (с накоплением 100 спектров).

4. Точность определения центральной частоты: ± 2 кГц.

5. Погрешность измерения ширины полосы: не более 5 % для сигналов с крутыми фронтами.

Как показано в [8, 10], увеличение полосы пропускания ведёт к снижению SNR и, как следствие, к росту вероятности ошибок. В нашем

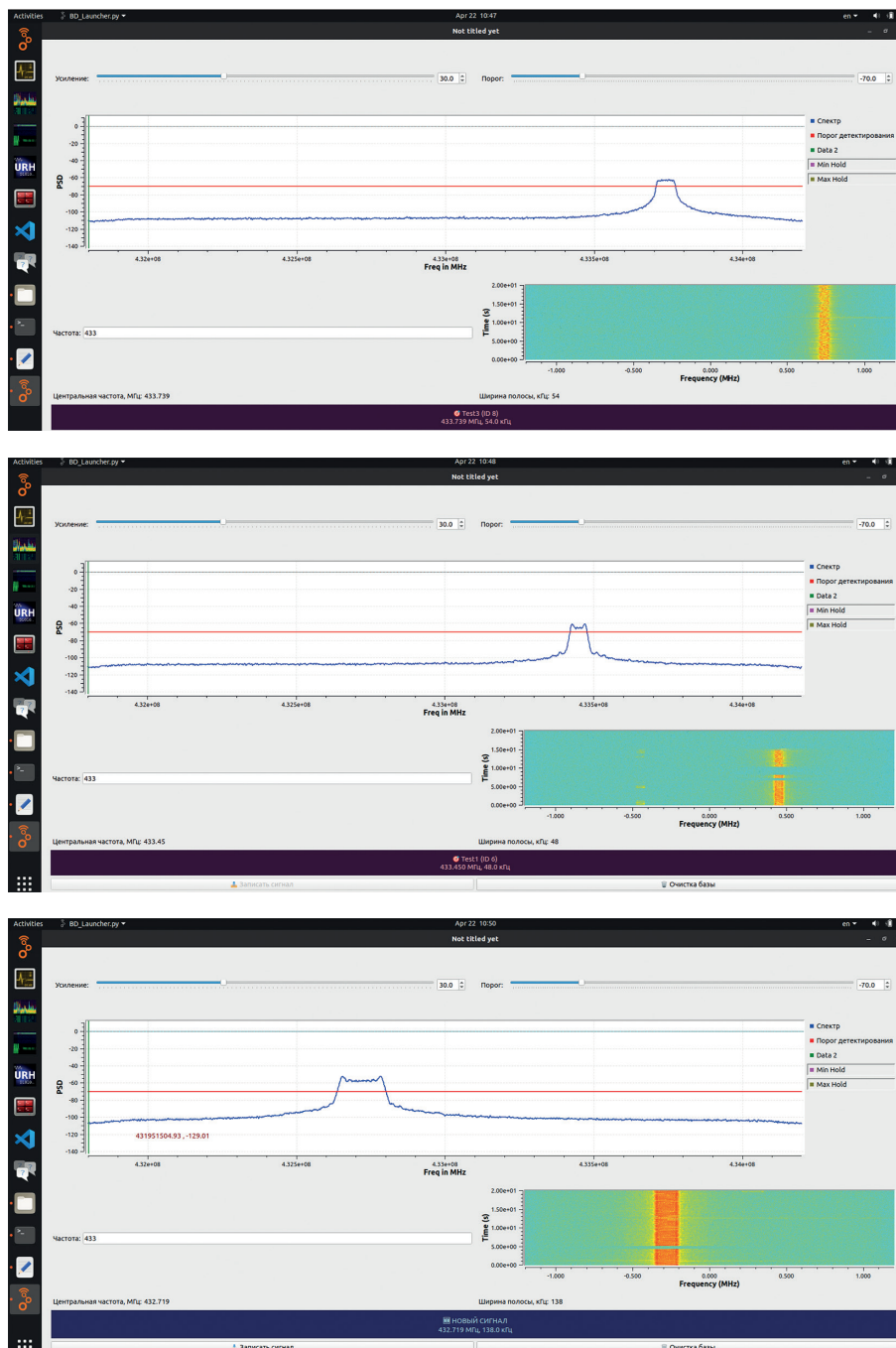


Рис. 6. Проверка работоспособности идентификации сигналов

комплексе выбор частоты дискретизации 2,4 МГц и размера БПФ 2048 даёт частотное разрешение около 1,17 кГц, что достаточно для различения каналов управления большинства дронов. При необходимости повышения точности можно увеличить длину БПФ или уменьшить полосу обзора, сузив фильтр.

В ходе эксперимента были записаны сигналы нескольких типов. При повторном появлении сигнала программа корректно идентифицировала его как известный, отобразив на информационной панели имя и ожидаемые параметры. Для неизвестного сигнала, сгенерированного HackRF, было предложено сохранить его в базу с произвольным именем.

В отличие от традиционных спектроанализаторов, предлагаемый комплекс не только визуализирует спектр, но и автоматически измеряет параметры сигнала и ведёт базу сигнатур. По сравнению с коммерческими системами радиомониторинга его стоимость на порядок ниже, а открытая архитектура позволяет адаптировать алгоритмы под конкретные задачи.

В результате эксперимента (рис. 6) была проведена проверка алгоритма идентификации сигналов. При излучении первых двух сигналов из базы (рис. 5) они были определены программой как известные, а также выведены их исходные параметры. При излучении ранее неизвестного сигнала программа даёт возможность его сохранить, предоставляя кнопку «Записать сигнал» активной.

Направления дальнейшего развития

Созданный программно-аппаратный комплекс представляет собой законченное решение для радиомониторинга, но его потенциал может быть расширен в следующих направлениях:

Автоматизация записи: реализация полностью автоматического режима, при котором любой новый сигнал записывается без участия оператора.

Углублённый анализ: переход от сравнения по двум параметрам к корреляционному анализу спектров или использованию классификаторов на основе машинного обучения для повышения точности идентификации.

Интеграция с системами противодействия: дополнение комплекса блоком подавления (jammer) и алгоритмом автоматического сопровождения цели.

Распределённая сеть мониторинга: синхронизация нескольких SDR-приёмников по сигналам GPS PPS для реализации TDoA-пеленгации [9, 10].

Заключение

В результате выполненной работы с нуля разработан и экспериментально апробирован программно-аппаратный комплекс на основе открытой среды GNU Radio и SDR-приёмника USRP B200/B210, предназначенный для решения задач поиска, измерения параметров и идентификации сигналов БПЛА. Последовательно решены задачи развёртывания виртуализированного программного окружения, построения потокового графа обработки сигналов в GNU Radio Companion, создания специализированных встроенных блоков на языке Python для порогового детектирования, автоматического определения центральной частоты и ширины спектра, а также реализации подсистемы хранения и сопоставления сигнатур в формате JSON с пользовательским интерфейсом управления записью.

Экспериментальные испытания подтвердили работоспособность и устойчивость предложенных алгоритмов. Комплекс обеспечивает мониторинг в диапазоне 70 МГц – 6 ГГц с мгновенной полосой анализа 2,4 МГц (расширяемой до 10 МГц при сохранении приемлемого частотного разрешения порядка 1,2 кГц), точность определения центральной частоты ± 2 кГц и погрешность измерения ширины полосы не более 5 %. Достигнутая пороговая чувствительность составляет около -100 дБм в полосе 200 кГц, что достаточно для обнаружения каналов управления большинства коммерческих БПЛА. Функция идентификации на основе базы сигнатур с допусками по частоте и ширине полосы показала корректное распознавание ранее записанных сигналов в условиях лабораторных экспериментов.

Научная значимость работы заключается в демонстрации целостной архитектуры программного SDR-решения, объединяющего в реальном времени спектральный анализ, извлечение параметров сигналов и их идентификацию исключительно на базе открытого программного обеспечения и доступного аппаратного обеспечения. Практическая ценность состоит в создании недорогого и гибкого инструмента радиомониторинга, который может быть использован как в исследовательских целях, так и в качестве основы для построения специализированных систем обнаружения БПЛА.

К ограничениям текущей реализации следует отнести ручной режим инициирования записи новых сигналов, фиксированные допуски при идентификации, а также зависимость вычислительной нагрузки от выбранной частоты дискретизации и длины БПФ. Дальнейшее развитие комплекса предполагает внедрение

автоматического режима пополнения базы сигнатур, применение методов машинного обучения для повышения точности идентификации за счёт анализа формы спектра, интеграцию с модулями подавления (jammer) и синхронизацию нескольких приёмников по сигналам GPS PPS для реализации разностно-дальномерной

пеленгации (TDoA). Полученные результаты подтверждают перспективность использования технологий SDR и открытых программных платформ для создания эффективных и экономичных систем радиомониторинга в интересах противодействия несанкционированному применению беспилотных летательных аппаратов.

Литература

1. Лысов А. В. Основные характеристики БПЛА, используемые на поле боя. – СПб: Лаборатория ППШ, 2024. – 357 с.
2. Боев Н. М. Анализ радиолиний связи с беспилотными летательными аппаратами // Вестник СибГАУ, 2012. – № 2. – С. 86–91.
3. Костин М. С., Ярлыков А. Д. Архитектурно-конфигурируемые SDR-технологии радиомониторинга и телеметрии. – М.: Инфра-Инженерия, 2021. – 147 с.
4. Галкин В. А. Основы программно-конфигурируемого радио. – М.: Горячая линия – Телеком, 2013. – 372 с.
5. Робенков Д. Н., Демченко А. А. Построение системы связи с беспилотным летательным аппаратом // СПб-ГУПС, 2023. – С. 278–280.
6. Шалаев А. А. Оценка фактических значений отношения сигнал/шум в радионавигационных системах на основе экспериментальных данных // Crede Experto, 2025. – № 4. – С. 60–75.
7. Арефьев Р. О., Скрипник О. Н., Арефьева Н. Г. Опыт использования программно-определяемых GNSS приёмников // Crede Experto, 2022. – № 1. – С. 88–99.
8. Семенко А. И., Бокла Н. И., Домрачева Е. А. Определение помехозащищённости телекоммуникационных систем с многопозиционным фазоманипулированным сигналом // Доклады АН ВШ РФ, 2017. – № 2 (35). – С. 97–106.
9. Гилев И. В., Канавин С. В. Программа автоматизации определения местоположения мобильного источника радиоизлучения с использованием приемника программно-определяемого радио // Вестник Воронежского института МВД России, 2025. – № 3. – С. 150–161.
10. Арзуманян А. Г., Севоян О. Ж., Гомцян О. А. Синхронная система обработки и передачи данных с привязкой к шкале единого мирового времени // Информационные технологии и телекоммуникации, 2016. – Т. 4. – № 2. – С. 12–19.

SOFTWARE PACKAGE FOR SEARCHING AND IDENTIFYING SIGNALS FOR SDR BASED ON THE GNU RADIO PLATFORM

Baymakov D. A.⁶, Tkachev D. F.⁷, Vorobyov P. V.⁸

Keywords: software-defined radio, GNU Radio, USRP B210, UAV detection, spectral analysis, Python blocks, signal identification, signature database.

Abstract

The goal of the work is to create a complete solution that allows scanning specified frequency ranges in real time, automatically recording exceedances of the threshold power level, determining the parameters of the detected signals and identifying them based on a pre-formed database of signatures.

Results of the study: a hardware and software stand has been created for monitoring the electromagnetic environment in the operating ranges of unmanned aerial vehicles (UAV) control channels using software-defined radio (SDR) technology. The test bench is implemented on the basis of the USRP B200/B210 receiver and the GNU Radio open environment deployed in the Oracle VirtualBox virtual machine for threshold detection, measurement of the center frequency and spectrum width, as well as the creation of a signature database with an automatic identification function. The key fragments of the program code and the results of the experimental verification of the system are presented. Factors affecting the accuracy of measurements and prospects for further development of the complex are discussed.

The practical value lies in the creation of an integral architecture of the software package that combines in real time spectral analysis, measurement of signal parameters and its identification based on a database of signatures using only open software and available SDR equipment.

6 **Danil A. Baymakov**, operator of the scientific company of the Military Academy of Communications named after Marshal of the Soviet Union S. M. Budyonny. St. Petersburg, Russia. E-mail: baymakovdanil@gmail.com

7 **Dmitry F. Tkachev**, Head of the Research Department of the Research Center of the Military Academy of Communications named after Marshal of the Soviet Union S. M. Budyonny. St. Petersburg, Russia. E-mail: dimas.portnoy@inbox.ru

8 **Pavel V. Vorobyov**, Researcher of the Research Department of the Research Center of the Military Academy of Communications named after Marshal of the Soviet Union S.M. Budyonny. St. Petersburg, Russia. E-mail: pvorobyev98@gmail.com