

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ДИАГНОСТИКИ RFID-СЧИТЫВАТЕЛЯ

Аннотация: В статье рассматривается актуальная задача разработки автоматизированной системы диагностики RFID-считывателей, что обусловлено растущей потребностью в эффективном контроле качества данного оборудования. Целью исследования является создание программного обеспечения для автоматизированной проверки параметров RFID-считывателей с использованием среды Node-RED и базы данных MySQL. В ходе работы реализованы алгоритмы последовательной проверки базовых параметров антенны, контроля рабочей частоты в диапазоне 866,9–866,93 МГц, автоматической коррекции частоты и измерения уровней мощности излучения при 10, 15 и 20 dBm. Разработан пользовательский интерфейс для визуализации процесса диагностики и хранения результатов в базе данных. Полученные результаты демонстрируют успешную реализацию комплексной системы диагностики, включающей автоматическую обработку данных, формирование отчётов и удобный интерфейс оператора. Разработанное решение позволяет существенно повысить эффективность процесса диагностики RFID-считывателей за счёт минимизации человеческого фактора и автоматизации рутинных операций.

Ключевые слова: RFID-считыватель, Node-RED, диагностика, база данных, программное обеспечение, интерфейс.

Samsonov N.A.
Student
NIU MIET
Russia, Zelenograd

DEVELOPMENT OF SOFTWARE FOR RFID-READER DIAGNOSTICS

Abstract: The article discusses the current task of developing an automated system for diagnosing RFID readers, which is due to the growing need for effective quality control of this equipment. The purpose of the study is to create software for automated testing of RFID reader parameters using the Node-RED environment and the MySQL database. The study implemented algorithms for sequential testing of basic antenna parameters, monitoring the operating frequency in the range of 866.9-866.93 MHz, automatic frequency correction, and measuring radiation power levels at 10, 15, and 20 dBm. A user interface has been developed to visualize the diagnostic process and store the results in a database. The results demonstrate the successful implementation of a comprehensive diagnostic system that includes automatic data processing, report generation, and a user-friendly interface. The developed solution significantly improves the efficiency of the diagnostic process.

Keywords: RFID reader, Node-RED, diagnostics, database, software, interface.

Важнейшим этапом становится выбор программной среды, где происходит практическая реализация алгоритмов автоматизированной проверки RFID-считывателя. На данном шаге важно перенести ранее описанную последовательность действий в программную форму, и оптимизировать выстроить такую структуру, в которой каждая операция диагностики будет выполняться последовательно, с минимальным участием оператора. В этом и состоит основное преимущество выбранной среды Node-RED: она позволяет представить логику работы системы в

виде наглядной цепочки взаимосвязанных потоков, где отдельные блоки отвечают за запуск процедуры, передачу команд, обработку результатов и формирование итогового вывода [1].

Практическая реализация алгоритмов в Node-RED строится вокруг потоковой логики. Внутри среды каждый этап диагностической процедуры оформляется как отдельный функциональный узел или группа узлов, а передача информации между этапами осуществляется в виде сообщений. За счёт этого появляется возможность явно задать маршрут прохождения данных: от момента запуска проверки до получения окончательного результата.

Проверка начинается с измерения мощности антенны `imp open` и `imp load`. Именно поэтому в Node-RED выстраивается отдельный поток, связанный не с общим запуском считывателя, а с фиксацией двух исходных состояний измерительной схемы. Сначала определяется значение `imp open`, которое отражает состояние пассивного излучения антенны. Затем фиксируется `imp load`, когда антенна уже подключена и система работает в другой конфигурации.

Этот этап является одним из самых важных во всей процедуре, хотя внешне он выглядит как подготовительный. Ошибка здесь не всегда заметна сразу, но потом она влияет на трактовку всех последующих результатов.

В программе этот участок реализован как два отдельных последовательных процесса. Сначала выполняется первое измерение. Только после его завершения система переходит ко второму. Поток операций для диагностики параметров `imp open` и `imp load` приведен на рисунке 1.

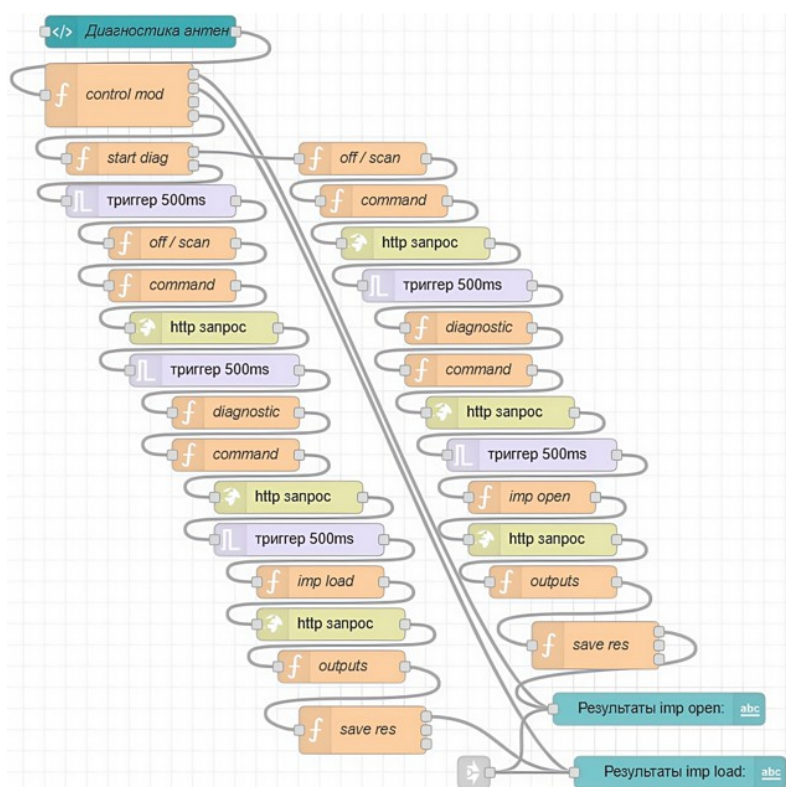


Рисунок 1. Поток операций для диагностики imp open и imp load

После фиксации базовых состояний система переходит к более содержательной части проверки, к контролю рабочей частоты считывателя. Этот этап является ключевым, потому что именно здесь становится понятно, можно ли вообще переходить к дальнейшим измерениям параметров рабочих мощностей. Если частота изначально выходит за допустимые пределы, то последующая диагностика мощности теряет часть своей ценности. По этой причине в программе заложен не простой контроль значения, а логика обязательного сопоставления текущей частоты с рабочим диапазоном 866,9-866,93 МГц. Если измеренное значение попадает в этот интервал, поток Node-RED не задерживается и сразу переводит систему к следующему этапу. Но если частота оказывается выше или ниже допустимого уровня, алгоритм не продолжает проверку формально, а сначала вычисляет величину отклонения и только затем запускает процедуру автоматической коррекции частоты.

Именно такая схема делает диагностику действительно корректной, а не формальной, поскольку программа не просто фиксирует проблему, а пытается привести устройство в корректное рабочее состояние. Лишь после этого начинается проверка уровней мощности излучения. Она выполняется последовательно при 10 dBm, 15 dBm и 20 dBm. Поток операций, разработанный для автоматической коррекции, приведен на рисунке 2.

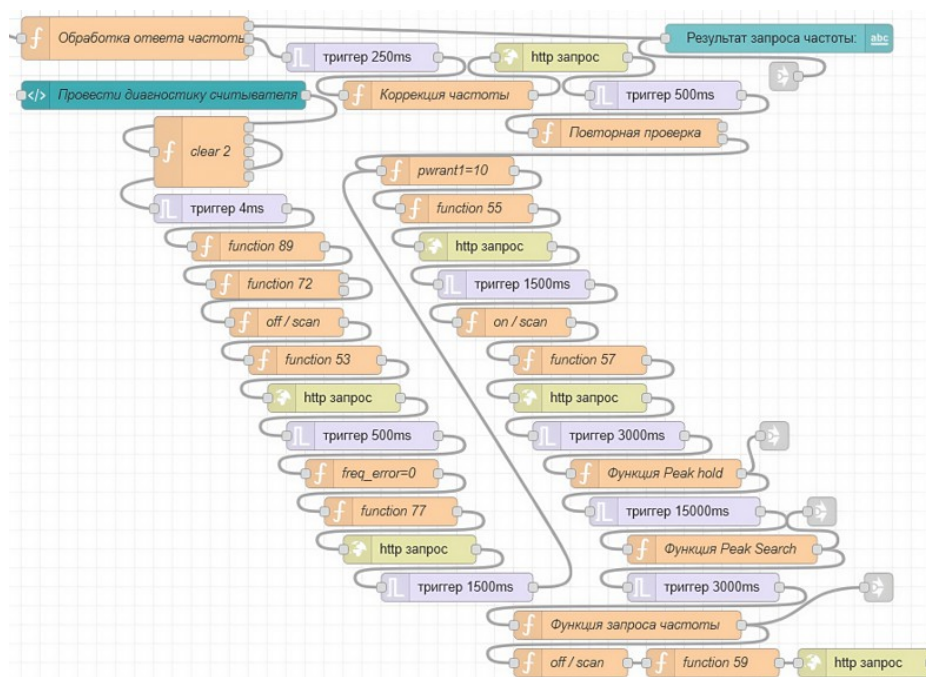


Рисунок 2. Поток операций для проверки и автоматической коррекции частоты

После того как частота приведена к требуемому диапазону, система переходит к основной части проверки, к измерению излучаемой мощности RFID-считывателя. Именно на этом этапе становится видно, насколько устройство сохраняет устойчивость в рабочих режимах и можно ли считать его параметры пригодными для дальнейшей эксплуатации. Поток операций для автоматической диагностики мощностей приведен на рисунке 3.

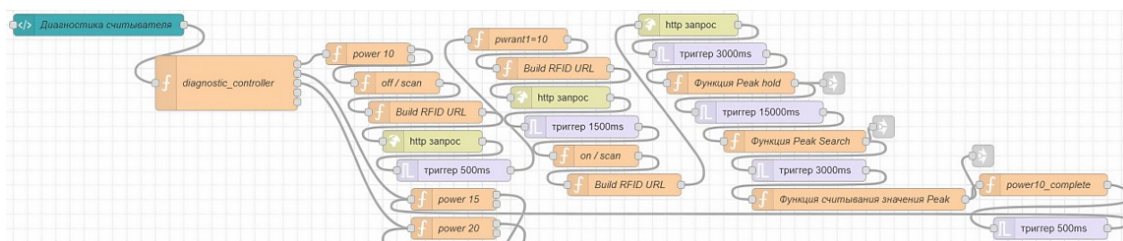


Рисунок 3. Поток операций для автоматической диагностики мощностей 10dBm, 15 dBm, 20 dBm в среде разработки Node-RED

Особое значение имеет программная обработка полученных данных. На уровне алгоритмов диагностики этого недостаточно - система должна уметь привести поступающую информацию к единому виду, перед отправкой отчета в базу данных. Это делает процедуру более прозрачной и облегчает контроль. Подобная архитектура особенно полезна, поскольку позволяет сразу готовить массив диагностических значений для последующей записи в базу данных и не оставлять часть обработки на ручные действия оператора [1].

С практической стороны Node-RED даёт ещё одно существенное преимущество - возможность быстрой доработки реализованных алгоритмов [2].

В результате программная реализация алгоритмов в Node-RED позволяет перевести процедуру проверки RFID-считывателя в форму целостного автоматизированного процесса. Каждый этап – от запуска системы и настройки параметров до получения результатов и их оценки – становится частью единого потока, контролируемого программной логикой. Это сокращает количество ручных операций, повышает воспроизводимость диагностики, упрощает сопровождение программного обеспечения и создаёт основу для дальнейшей интеграции с базой данных и интерфейсом оператора. Именно поэтому реализация алгоритмов в выбранной среде разработки выступает центральным шагом в создании пульта автоматизированной проверки RFID-считывателя и

непосредственно подводит к следующему подпункту, связанному с подключением базы данных [2].

Подключение базы данных является важным этапом реализации программно-аппаратного комплекса, поскольку именно на этом уровне результаты автоматизированной проверки RFID-считывателя переходят из состояния временно полученных измерений в форму структурированной информации, пригодной для хранения, последующего анализа и формирования отчётности. Это необходимо для фиксации результата конкретной проверки и накопления диагностической истории, сопоставления измерений между собой и повышения общей прозрачности процесса контроля.

В рамках поставленной задачи в качестве системы хранения данных используется MySQL. Использование реляционной базы данных позволяет организовать хранение результатов в упорядоченном виде, где каждая запись содержит набор параметров, связанных с конкретной процедурой проверки RFID-считывателя. Например, если устройство после его производства при первой проверке показало неудовлетворительные результаты, его отправили на доработку и затем провели диагностику повторно, чтобы убедиться, что оно исправно.

С программной точки зрения подключение MySQL в составе разрабатываемого пульта должно обеспечивать устойчивый обмен данными между логикой, реализованной в Node-RED, и таблицами базы данных. Node-RED выполняет роль промежуточного координирующего слоя: с одной стороны, он принимает результаты измерений и обработки, а с другой - преобразует их в форму, пригодную для занесения в MySQL.

Для диагностической системы важно не просто сохранить итоговую отметку о том, исправен считыватель или нет. Важно понимать, как именно вёл себя RFID-считыватель во время испытания. В MySQL хранится весь набор ключевых параметров, который формируется в

процессе проверки. Дополнительно в запись целесообразно включать дату и время проверки, серийный номер устройства.

Подключение базы данных заметно меняет и саму организацию работы оператора. Без MySQL результаты либо остаются в оперативной памяти программы, либо отображаются один раз на экране и потом требуют ручного переноса в журнал, таблицу или отчёт. Это неудобно. И, что важнее, это снова возвращает человеческий фактор. При использовании MySQL диагностические данные сразу попадают в постоянное хранилище. За счёт этого сокращается количество ручных действий, снижается риск ошибок при переписывании значений и появляется единый формат хранения результатов

Последовательность взаимодействия между Node-RED и базой данных является важной связкой. После завершения измерений система должна не просто передать набор отдельных чисел, а сформировать логически завершённую запись, относящуюся к конкретному сеансу диагностики. Для этого в потоке Node-RED выполняется объединение всех значений в структурированный массив, после чего создаётся запрос на запись в MySQL.

Подключение базы данных MySQL является одной из ключевых составляющих разработки пульта автоматизированной проверки RFID-считывателя. Интеграция Node-RED с MySQL позволяет связать этапы диагностики, хранения данных и формирования отчётности в единый информационный контур, что повышает удобство эксплуатации, снижает влияние человеческого фактора и делает работу пульта более завершённой с практической точки зрения.

Проектирование интерфейса пульта автоматизированной проверки RFID-считывателя является важным этапом разработки программно-аппаратного комплекса, поскольку именно через интерфейс обеспечивается практическое взаимодействие оператора с системой.

С учётом выбранной среды разработки целесообразно реализовать программный интерфейс на базе Node-RED, так как данная среда позволяет выстроить внутреннюю логику автоматизированной проверки, и организовать удобное визуальное представление её результатов [3].

Основная задача программного интерфейса заключается в том, чтобы представить оператору все ключевые этапы проверки в последовательной и понятной форме.

Экран текущей диагностики в таком комплексе нужен для отображения данных и постоянного контроля над ходом проверки. Центральная панель интерфейса должна показывать финальный результат и всю логику движения проверки. На рисунке 4 представлен интерфейс пульта.

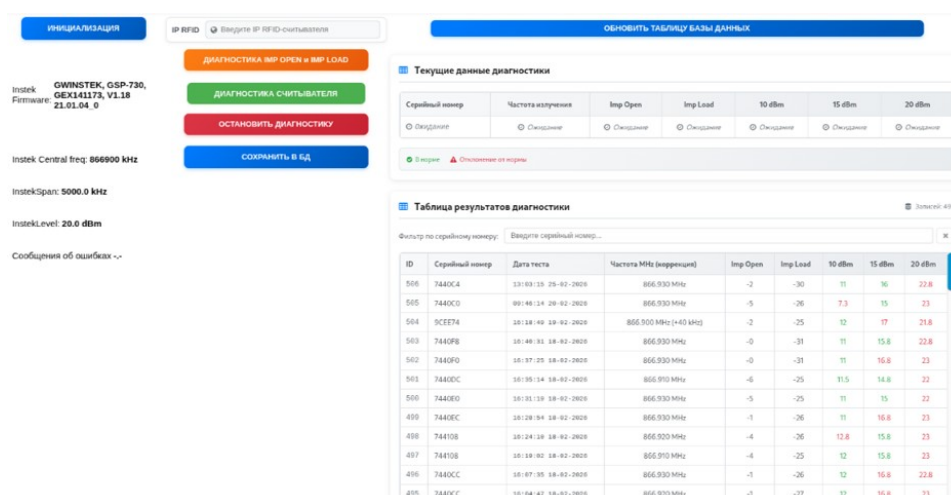


Рисунок 4. Интерфейс пульта проверки целиком

Первым этапом на экран последовательно выводятся результаты измерения базовых состояний антенны, imp open и imp load рисунок 5.

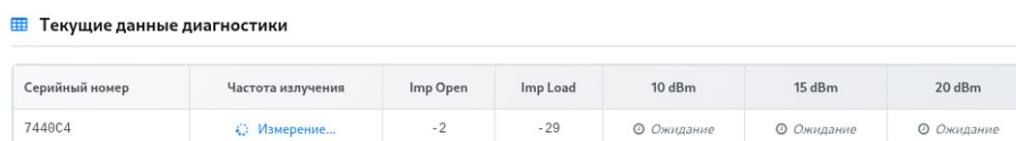


Рисунок 5. Результаты измерений imp open и imp load в интерфейсе

Далее происходит измерение рабочей частоты, для этого в интерфейсе реализована анимация и надпись, что происходит измерение параметра. Удовлетворяющая частота отображается в таблице рисунок 6.

Текущие данные диагностики

Серийный номер	Частота излучения	Imp Open	Imp Load	10 dBm	15 dBm	20 dBm
7440C4	866.930 MHz	-2	-28	Измерение...	Ожидание	Ожидание

Рисунок 6. Отображение частоты если она удовлетворяет нужному промежутку

Если частота считывателя выходит за пределы диапазона 866,9-866,93 МГц, то в рабочем окне необходимо сразу показывать фактическое значение частоты, и запускать коррекцию. Данный принцип отображения реализован в интерфейсе пульта и представлен на рисунке 7.

Текущие данные диагностики

Серийный номер	Частота излучения	Imp Open	Imp Load	10 dBm	15 dBm	20 dBm
744040	866.870 MHz Коррекция...	-3	-22	Ожидание	Ожидание	Ожидание

Рисунок 7. Отображение этапа коррекции частоты

После автоматической коррекции на рисунке 8 представлен статус частоты:

Текущие данные диагностики

Серийный номер	Частота излучения	Imp Open	Imp Load	10 dBm	15 dBm	20 dBm
744040	866.900 MHz	-3	-22	Измерение...	Ожидание	Ожидание

Рисунок 8. После автоматической коррекции поле отображает частоту, которая регламентирована

Диагностика считывателя выстроена как последовательная проверка при уровнях мощности 10 dBm, 15 dBm и 20 dBm. По этой причине в интерфейсе необходимо выделить отдельную зону, где результаты по каждому режиму будут показаны отдельно. Оператору важно быстро

понять, как считыватель ведёт себя на каждом уровне мощности и есть ли отклонение в каком-то конкретном режиме. Именно поэтому наиболее удачным решением выглядит табличное представление данных. Реализация данного блока интерфейса представлена на рисунке 9.

Текущие данные диагностики

Серийный номер	Частота излучения	Imp Open	Imp Load	10 dBm	15 dBm	20 dBm
7440C4	866.930 MHz	-2	-29	11	16	22.8

Рисунок 9. Результаты всей диагностики

Важным элементом интерфейса является блок взаимодействия с базой данных и отчётностью. После завершения диагностики оператор должен видеть, что данные успешно сохранены в MySQL. В прикладном плане это особенно удобно, поскольку объединяет в одном рабочем окне сразу три функции: проведение диагностики, фиксацию результатов рисунок 10.

ОБНОВИТЬ ТАБЛИЦУ БАЗЫ ДАННЫХ

Текущие данные диагностики

Серийный номер	Частота излучения	Imp Open	Imp Load	10 dBm	15 dBm	20 dBm
7440C4	866.930 MHz	-2	-29	11	16	22.8

В норме Отклонение от нормы

Таблица результатов диагностики Записей: 500

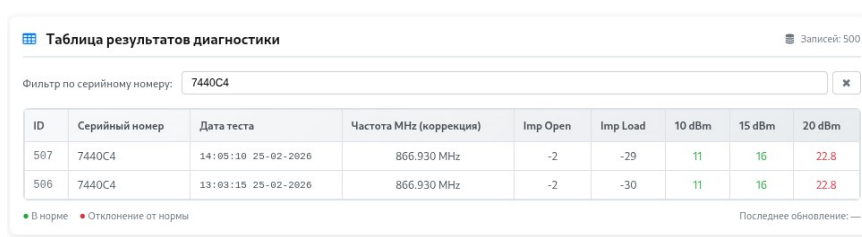
Фильтр по серийному номеру:

ID	Серийный номер	Дата теста	Частота MHz (коррекция)	Imp Open	Imp Load	10 dBm	15 dBm	20 dBm
507	7440C4	14:05:10 25-02-2026	866.930 MHz	-2	-29	11	16	22.8
506	7440C4	13:03:15 25-02-2026	866.930 MHz	-2	-30	11	16	22.8
505	7440C0	09:46:14 20-02-2026	866.930 MHz	-5	-26	7.3	15	23
504	9CEE74	16:18:49 19-02-2026	866.900 MHz (+40 kHz)	-2	-25	12	17	21.8
503	7440F8	16:40:31 18-02-2026	866.930 MHz	-0	-31	11	15.8	22.8
502	7440F0	16:37:25 18-02-2026	866.930 MHz	-0	-31	11	16.8	23
501	7440DC	16:35:14 18-02-2026	866.910 MHz	-6	-25	11.5	14.8	22
500	7440E0	16:31:19 18-02-2026	866.930 MHz	-5	-25	11	15	22
499	7440EC	16:28:54 18-02-2026	866.930 MHz	-1	-26	11	16.8	23
498	744108	16:24:10 18-02-2026	866.920 MHz	-4	-26	12.8	15.8	23
497	744108	16:19:02 18-02-2026	866.910 MHz	-4	-25	12	15.8	23
496	7440CC	16:07:35 18-02-2026	866.930 MHz	-1	-26	12	16.8	22.8

Рисунок 10. Таблица результатов диагностики отображает все диагностики, которые хранятся в БД

С эргономической точки зрения интерфейс должен быть спроектирован таким образом, чтобы оператор выполнял минимальное число действий. Для этого важно придерживаться последовательной структуры экранных блоков, избегать перегруженности текстом и не допускать дублирования информации в разных частях окна.

В интерфейсе реализован поиск по серийному номеру диагностируемого устройства. Введя фильтр по серийному номеру, отобразятся конкретные диагностики исключительно данного устройства. Это требуется для удобства сравнения результатов, например если устройство при первой диагностике не удовлетворяло требованиям, затем была выполнена доработка и повторная проверка рисунок 11.



The screenshot shows a web interface titled "Таблица результатов диагностики" (Diagnostic Results Table). It features a search filter "Фильтр по серийному номеру: 7440C4" and a table with two rows of data. The table columns are: ID, Серийный номер (Serial number), Дата теста (Test date), Частота MHz (коррекция) (Frequency MHz (correction)), Imp Open, Imp Load, 10 dBm, 15 dBm, and 20 dBm. The first row (ID 507) shows a test at 14:05:10 on 25-02-2026 with a frequency of 866.930 MHz. The second row (ID 506) shows a test at 13:03:15 on 25-02-2026 with the same frequency. The values for 10 dBm, 15 dBm, and 20 dBm are highlighted in green, indicating they are within normal ranges.

ID	Серийный номер	Дата теста	Частота MHz (коррекция)	Imp Open	Imp Load	10 dBm	15 dBm	20 dBm
507	7440C4	14:05:10 25-02-2026	866.930 MHz	-2	-29	11	16	22.8
506	7440C4	13:03:15 25-02-2026	866.930 MHz	-2	-30	11	16	22.8

Рисунок 11. Пример использования фильтра поиска диагностики по конкретному серийному номеру

Проектирование интерфейса пульта автоматизированной проверки RFID-считывателя должно быть ориентировано на обеспечение наглядного и удобного взаимодействия оператора с системой. Программный интерфейс на базе Node-RED позволяет объединить управление диагностической процедурой, отображение промежуточных результатов, контроль частоты, вывод данных по мощностям 10 dBm, 15 dBm и 20 dBm, а также сохранение информации в MySQL и формирование отчёта в едином рабочем пространстве. Аппаратная часть при этом должна выполнять вспомогательную роль, обеспечивая надёжность эксплуатации и удобство подключения оборудования [1].

Данная разработка была применена и испытана, на серии испытаний на рисунке 10 мы видим 507 диагностик. На рисунках выше приведены результаты испытаний приборов с помощью разработанного программного обеспечения.

Использованные источники:

1. Лямов, Ю. О. Опыт использования среды потокового программирования Node-RED в курсе «Интернет вещей» [Электронный ресурс] / Ю. О. Лямов, Л. Г. Галимова // Путь к инновационному развитию России : сборник научных трудов. — 2023. — С. 283–286. — URL: <https://elibrary.ru/item.asp?id=60780342> (дата обращения: 31.07.2026).
2. Шалыгин, М. Г. Автоматизация измерений, контроля и испытаний [Электронный ресурс] : учебное пособие для вузов / М. Г. Шалыгин, Я. А. Вавилин. — 4-е изд., стер. — Санкт-Петербург : Лань, 2025. — 172 с. — URL: <https://lanbook.com/catalog/inzhenerno-tekhnicheskie-nauki/avtomatizatsiya-izmereniy-kontrolya-i-ispytaniy/> (дата обращения: 31.07.2026).
3. Заяц, А. М. Проектирование и разработка web-приложений. Введение в frontend и backend разработку на JavaScript и Node.js [Электронный ресурс] : учебное пособие для СПО / А. М. Заяц, Н. П. Васильев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 120 с. — URL: <https://dokumen.pub/web-frontend-backend-javascript-nodejs-3.html> (дата обращения: 31.07.2026).