УТВЕРЖДАЮ

Технический директор
ООО «ОЛЬВИЯ»

_____ Ю.И. Константиниди

«12» _____06_____ 2020 г.

СИСТЕМА ИЗМЕРЕНИЯ СКОРОСТИ ДВИЖЕНИЯ ТРАНСПОРТНЫХ СРЕДСТВ «ДУЭТ»

Руководство пользователя программного обеспечения
БКЮФ.201219.029РП

2020 г.

Оглавление

Список сокращений....3
1. Назначение и область применения....4
2. Состав и принцип работы....5
3. Установка и удаление....7
4. Запуск и остановка....18
5. Работа с веб-интерфейсом....23
6. Материалы....48
7. Служебные данные....69
8. Устранение неисправностей....81
ПРИЛОЖЕНИЕ А....115
ПРИЛОЖЕНИЕ Б....174
ПРИЛОЖЕНИЕ В....181

Список сокращений

БД	–	База данных;
ГНСС	–	Глобальная навигационная спутниковая система;
ГРЗ	–	Государственный регистрационный знак;
ИС	–	Измерительная система;
ЛВС	–	Локальная вычислительная сеть;
ЛКМ	–	Левая кнопка мыши;
ОС	–	Операционная система;
ПДД	–	Правила дорожного движения;
ПО	–	Программное обеспечение;
ТС	–	Транспортное средство.
API	–	Application programming interface;
CRC	–	Cyclic redundancy check (Циклический избыточный код);
HTTP	–	HyperText Transfer Protocol Secure;
IEEE	–	Institute of Electrical and Electronics Engineers;
IP	–	Internet Protocol;
FTP	–	File Transfer Protocol (Протокол передачи файлов);
JSON	–	JavaScript Object Notation;
LSB	–	Least significant bit;
RDP	–	Remote Desktop Protocol (Протокол удаленного рабочего стола);
RPC	–	Remote Procedure Call (Удалённый вызов процедур);
UTC	–	Coordinated Universal Time (Всемирное координированное время);
XML	–	Extensible Markup Language (Расширяемый язык разметки).

1. Назначение и область применения

Программное обеспечение «ДУЭТ» предназначено для работы в составе одноимённой системы для измерения скорости движения ТС на протяженном участке дороги в автоматическом режиме.

Актуальная версия ПО «ДУЭТ», на которую распространяется данное руководство: **1.3.10 и старше**.

2. Состав и принцип работы

2.1. Состав системы

Система состоит из двух регистраторов и программного обеспечения (далее — ПО). В качестве регистраторов используются средства измерения утверждённого типа, изготавливаемые ООО «ОЛЬВИЯ»:

- измерители скорости радиолокационные многоцелевые с фотофиксацией «СКАТ» (далее — «СКАТ»);
- комплексы измерительные с фотофиксацией «СКАТ-ПП» (далее — СКАТ-ПП);
- измерители оптико-радиолокационные многоцелевые «ОРАКУЛ» (далее — «ОРАКУЛ»);
- комплекс измерительный с фотофиксацией «КРЕЧЕТ-СМ» (далее — «КРЕЧЕТ-СМ»).

ПО устанавливается на один из регистраторов Системы или на отдельный сервер, при этом в Систему могут входить два регистратора. В случае, если ПО устанавливается на регистратор, то он называется ведущим, второй — ведомыми.

2.2. Состав программного обеспечения

ПО распространяется в двух вариантах для операционных систем (далее — ОС):

- Windows;
- Linux.

ПО состоит из двух основных элементов:

- Основной исполняемый модуль (включает в себя вычислитель, формирователь материалов и встроенный веб-сервер) — duet_core;
- Веб-интерфейс пользователя.

Дополнительно ПО включает службу наблюдения за работой основного модуля — duet_supervisor.

2.3. Обслуживающий персонал

ПО «ДУЭТ» предназначено для обслуживания персоналом в составе минимум двух человек:

- Системный администратор;
- Оператор службы поддержки.

Системный администратор производит первичную установку системы и решает проблемы на системном уровне в случае их возникновения, используя графический веб-интерфейс, а также удалённый доступ к операционной системе с установленным ПО (регистраторы или отделенный сервер). Оператор производит настройки пользовательских параметров системы через веб-интерфейс и осуществляет первичную техническую поддержку.

2.4. Принцип работы системы

Принцип работы системы основан на измерении скорости движения ТС косвенным методом. Система измеряет интервал времени, за который ТС было пройдено расстояние между регистраторами.

Регистраторы осуществляют непрерывную фотосъемку зоны контроля. Первый регистратор при обнаружении в зоне контроля ТС (по распознанному ГРЗ) фиксирует изображение ТС, время фиксации и дистанцию от ТС до места установки регистратора. ТС, двигаясь по дороге, попадает в зону контроля следующего регистратора, который так же фиксирует изображение ТС, время фиксации и дистанцию от ТС до места установки следующего регистратора. Регистраторы в системе могут быть направлены как в одну сторону, так и разнонаправленно.

ПО Системы выполняет постоянный контроль баз данных регистраторов для получения информации о ТС. На основе полученных данных Система выполняет расчёт средней скорости ТС. Система также выкачивает по FTP необходимые предварительные материалы с регистраторов. Результатом работы является фотоматериал с вписанной служебной информацией. Вид результирующего фотоматериала может быть задан с помощью описания на языке разметки XML.

Измерение географических координат места установки регистраторов, входящих в Систему, осуществляется на основе данных, полученных от ГНСС на самих регистраторах.

3. Установка и удаление

3.1. Требования

ПО «ДУЭТ» предназначено для функционирования на отдельном выделенном сервере или аппаратной платформе средств измерения утверждённого типа «СКАТ», «СКАТ-ПП», «ОРАКУЛ», «КРЕЧЕТ-СМ», а также на отдельном выделенном сервере.

Вычислительная платформа, на которую производится установка ПО, должна удовлетворять следующим аппаратным требованиям:

- архитектура:
  - amd64 (Windows),
  - ARM (Linux),
- процессор (CPU) с 4 и более ядрами, с частотой не ниже 1,5 ГГц;
- оперативная память объемом не менее 4 Гб;
- SSD диск объемом не менее 256 Гб, структура диска должна содержать 2 раздела:
  - системный (C, не менее 25 Гб),
  - раздел данных (D, не менее 150 Гб).

Требования по предустановленному ПО для разных операционных систем:

1) Windows.

- операционная система MS Windows 7 Embedded x64 или Windows 10 LTSC x64;
- распространяемые пакеты:
  - Visual C++ Redistributable Packages for Visual Studio 2008 x86/x64;
  - Visual C++ Redistributable Packages for Visual Studio 2010 x86/x64;
  - Visual C++ Redistributable Packages for Visual Studio 2013 x86/x64;
  - Visual C++ Redistributable Packages for Visual Studio 2017 x86/x64.
- СУБД MySQL 5.7.6 или MariaDB 10.1.3 и старше:
- набор утилит для обслуживания БД (клиент БД SQL);
- сервер удалённых рабочих столов (обслуживание).

2) Linux.

- операционная система семейства Linux (Ubuntu, Debian или др.) с ядром не ниже 3.18.0;
- СУБД MySQL 5.7.6 или MariaDB 10.1.3 и старше:
- набор утилит для обслуживания БД (клиент БД SQL);
- удалённый доступ (ssh).

Пакеты для установки/обновления ПО «ДУЭТ» представляют собой файлы архива основного ПО с основным модулем, веб-интерфейсом и базовыми настройками предназначенные для работы в разных ОС:

- Windows:
  - duet-xxx-win64.zip
- Linux:
  - duet-xxx-linux-src.tar.gz

В имени пакета содержится информация о версии ПО «ДУЭТ»-"xxx". Например: duet-1.0.0-win64.zip.

Пакет ОС под семейства Linux собирается из исходных файлов из пакета прямо на целевой ОС или предварительно предустанавливается в образ этой ОС.

Ниже приводится процедура подготовки, установки и обновления ПО.

3.2. Конфигурации

ПО «ДУЭТ» может быть настроено по-разному в зависимости от конфигурации функционирования:

- сервер;
- регистратор.

Для обоих вариантов конфигураций в ПО «ДУЭТ» один из двух регистраторов становится **Ведущим (master)**, а другой — **Ведомым (slave).** Конфигурация выбирается на старте ПО, аргументами запуска исполняемого файла (см. Раздел 4 "Запуск").

Конфигурация «**сервер**» предполагает установку на выделенный сервер, который далее будет работать с двумя регистраторами удалённо. В данной конфигурации ПО «ДУЭТ» полностью управляет БД, установленной на сервере. Доступ к данным обоих регистраторов производится удаленно по ЛВС: к БД регистраторов через mysql клиент, к файлам материалов через FTP клиент. Клиенты для получения данных встроены в ПО «ДУЭТ». Ведущий и ведомый регистратор выбираются при дальнейшей настройке в веб-интерфейсе.

Конфигурация «**регистратор**» предполагает установку непосредственно на регистратор. В данной конфигурации ПО «ДУЭТ», установленное на один из регистраторов, использует СУБД, уже предустановленную на этом регистраторе, в которой автоматически создаёт отдельную БД для своих результатов.

- С регистратором, на котором установлено ПО «ДУЭТ», взаимодействие производится локально, без использования ЛВС: к БД регистраторов через mysql клиент, к файлам материалов через прямое обращение к каталогу материалов в файловой системе. Регистратор с установленным ПО «ДУЭТ» автоматически считается **Ведущим**.
- Со вторым регистратором, взаимодействие производится удаленно по ЛВС: к БД регистратора через mysql клиент, к файлам материалов через FTP клиент. Клиенты для получения данных встроены в ПО «ДУЭТ».

3.3. Профили приборов

ПО «ДУЭТ» подготовлено для работы с несколькими профилями, разработанными под каждый тип регистратора. Профиль выбирается автоматически при выборе определенного типа прибора в веб-интерфейсе. Ниже приведены параметры регистраторов, на которые рассчитано ПО в зависимости от профиля (табл. 3.1-3.4). Каждый параметр может быть переопределён в общем файле настроек ПО «ДУЭТ» (config.json) в разделах для Ведущего (master) и Ведомого (slave) приборов, если это потребуется для специально настроенных регистраторов данного конкретного типа. Для переопределения параметра в файле настроек для master/slave группы настроек значение параметра не должно быть пустое.

Таблица 3.1 — Профиль настроек регистратора «СКАТ»

Название параметра	**Параметр в файле настроек**	
Наименование БД	main_db_name	main
Имя пользователя БД	main_db_user	prometheus
Пароль пользователя БД	main_db_pass	*<*вычисляется*>
Путь к папке материалов всех фиксаций в файловой системе регистратора	fixations_dir	D:/arch/all_fixations

Примечание — Пароль пользователя БД генерируется автоматически по алгоритму, заложенному в регистраторе "СКАТ-ПП", см. документацию на регистратор.

Таблица 3.2 — Профиль настроек регистратора «СКАТ-ПП»

Название параметра	**Параметр в файле настроек**	
Наименование БД	main_db_name	main
Имя пользователя БД	main_db_user	prometheus
Пароль пользователя БД	main_db_pass	*<*вычисляется*>
Путь к папке материалов всех фиксаций в файловой системе регистратора	fixations_dir	D:/arch/all_fixations

Примечание — Пароль пользователя БД генерируется автоматически по алгоритму, заложенному в регистраторе "СКАТ-ПП", см. документацию на регистратор.

Таблица 3.3 — Профиль настроек регистратора «КРЕЧЕТ-СМ»

Название параметра	**Параметр в файле настроек**	
Наименование БД	main_db_name	main
Имя пользователя БД	main_db_user	prometheus
Пароль пользователя БД	main_db_pass	*<*вычисляется*>

Название параметра	Параметр в файле настроек	
Путь к папке материалов всех фиксаций в файловой системе регистратора	fixations_dir	D:/arch/all_fixations

Примечание — Пароль пользователя БД генерируется автоматически по алгоритму, заложенному в регистраторе "СКАТ-ПП", см. документацию на регистратор.

Таблица 3.4 — Профиль настроек регистратора «ОРАКУЛ»

Название параметра	Параметр в файле настроек	
Наименование БД	main_db_name	mary.main
Имя пользователя БД	main_db_user	prometheus
Пароль пользователя БД	main_db_pass	*<*вычисляется*>
Путь к папке материалов всех фиксаций в файловой системе регистратора	fixations_dir	/mnt/targets/ftp/all_fixations

Примечание — Пароль пользователя БД генерируется автоматически по алгоритму, заложенному в регистраторе "СКАТ-ПП", см. документацию на регистратор.

Доступ по FTP к всем типам регистраторов выполняется одинаково с настройками по умолчанию, указанными в табл. 3.5. Настройки могут быть изменены через config.json. Службе поддержки регистраторов необходимо открыть FTP доступ к материалам указанному пользователю.

Корень папки материалов по FTP для всех типов регистраторов должен находится внутри каталога всех фиксаций:

/all_fixations/

Таблица 3.5 — Удаленный доступ к файлам материалов по FTP

Название параметра	Параметр в файле настроек	
duet	ftp_user	duet
Пароль пользователя FTP	ftp_pass	duet
Порт FTP	ftp_port	21

3.4. Состав файлов под Windows

Общая структура каталогов и файлов ПО «ДУЭТ» под ОС Windows по умолчанию,

C:\duet	-	Основное ПО
\config	-	Каталог файлов настроек
\docroot	-	Каталог графического веб-интерфейса
\logs	-	Каталог протоколов работы (логи)
\templates	-	Шаблоны для создания материалов нарушений
***.dll**	-	Библиотеки ПО
***.ttf**	-	Дополнительные шрифты для шаблонов в **/templates**
privolvia.pem	-	Ключи для цифровых подписей материалов
pubolvia.pem		(RSA-SHA1)
privolvia.sxp	-	Ключи для цифровых подписей материалов
pubolvia.sxp		(ГОСТ)
duet_core.exe	-	Исполняемый файл ядра ПО
duet_supervisor.exe	-	Программа запуска и мониторинга **duet_core**
duet_start.bat	-	Скрипт запуска ПО в конфигурации **Регистратор**
duet_start_standalone.bat	-	Скрипт запуска ПО в конфигурации **Сервер**

Файлы настроек в папке **\config** :

config.json	-	Основные настройки
web_server.ini	-	Настройки ВЕБ-сервера для ВЕБ-интерфейса
users.json	-	Реестр пользователей ПО (см.п.5.2)

3.5. Первая установка под Windows

В данном разделе приведена процедура установки ПО «ДУЭТ» на регистраторы типа «СКАТ», «СКАТ-ПП», «КРЕЧЕТ-СМ» под ОС Windows, где ранее не было установлено других версий данного ПО.

3.5.1. Подготовка

Необходимое оборудование:

- регистратор или сервер, на который будет производиться установка;
- источник питания и кабель питания для устройства;
- набор для подключения:

  Вариант 1:

  - сервисный кабель для подключения устройства к рабочему компьютеру.

Вариант 2:

- монитор с VGA входом;
- компьютерная мышь;
- клавиатура;
- USB флешка;
- сервисный кабель для подключения устройства к монитору и USB.

Вариант 3:

- удалённый сетевой доступ к устройству (локальная сеть, частная защищённая сеть и др.);
- клиент подключения к удалённому рабочему столу устройства (для установки по сети).

3.5.2. Установка приложения

1. Из архива **duet-xxx-win64.zip** скопируйте содержимое в папку **C:\duet**
2. Настройте параметры в файле настроек **C:\duet\config\config.json**.
   1) Настройте данные для работы с сервисной БД ПО «ДУЭТ», в базовом файле настроек указаны параметры по умолчанию. Сервисная БД будет создана автоматически с указанным наименованием.

      Внимание! Необходимо использовать пользователя с правами на создание/редактирование БД и управления правами пользователей БД.

      - "parameters_db_name" - duet_service
      - "parameters_db_pass" - <password>
      - "parameters_db_user" - root

   2) Настройте данные для работы с БД результатов ПО «ДУЭТ», в базовом файле настроек указаны параметры по умолчанию. БД результатов будет создана автоматически с указанным наименованием.

      Внимание! Необходимо использовать пользователя с правами на чтение/редактирование таблиц БД.

      - "result_db_name" - duet_main
      - "result_db_pass" - <password>
      - "result_db_user" - root

   3) Доступ снаружи из других систем мониторинга для чтения данных БД результатов «ДУЭТ» (duet_main). Доступ выполняется через пользователя **"prometheus"**. Настройки пароля:

      Конфигурации **Регистратор** — изменений не требуется.

      Пользователь с паролем создаётся непосредственно самим ПО регистратора и дополнительных изменений не требуется.

Конфигурации **Сервер** — требуется задать пароль, пароль по умолчанию задан в файле настроек.

- "prometheus_db_pass" - <password>

4) Настройте папку материалов фиксаций ПО «ДУЭТ».
   - "result_directory" - D:/arch/duet

5) (опционально) Настройте папку временных файлов ПО «ДУЭТ» (например, "D:/arch/duet_tmp"). При значении в виде пустой строки ("") временные файлы, в том числе материалы с регистраторов, будут сохраняться в ту же папку, что и материалы — result_directory (действие по умолчанию):
   - "result_directory_tmp" - ""

6) Настройте максимальный размер папки создаваемых материалов ПО «ДУЭТ».

Внимание! Размер должен быть более 15 Гб. Большой размер увеличивает время подготовки на старте приложения.

Размер указывается в байтах (соответствует 20 Гб).

- "storage_limit_fixations" - 107374182400

3. Создайте ярлык для автозагрузки.

Конфигурации **Регистратор:**

В папке **C:\duet** создайте ярлык на файл **duet_start.bat**, скопируйте ярлык в автозагрузку.

- Windows 7 — папка Start\All Programs\Startup
- Windows 10 — папка по команде из Run: shell:startup

Конфигурации **Сервер** — требуется задать пароль, пароль по умолчанию задан в файле настроек.

В папке **C:\duet** создайте ярлык на файл **duet_start_standalone.bat**, скопируйте ярлык в автозагрузку.

- Windows 7 — папка Start\All Programs\Startup
- Windows 10 — папка по команде из Run: shell:startup

3.6. Обновление под Windows

В данном разделе приведена процедура обновления ПО «ДУЭТ».

3.6.1. Подготовка

Необходимое оборудование:

- регистратор или сервер, на который будет производиться установка;
- источник питания и кабель питания для устройства;
- набор для подключения:

Вариант 1:

- сервисный кабель для подключения устройства к рабочему компьютеру.

Вариант 2:

- монитор с VGA входом;
- компьютерная мышь;
- клавиатура;
- USB флешка;
- сервисный кабель для подключения устройства к монитору и USB.

Вариант 3:

- удалённый сетевой доступ к устройству (локальная сеть, частная защищённая сеть и др.);
- клиент подключения к удалённому рабочему столу устройства (для установки по сети).

Перед началом обновления необходимо остановить текущее ПО «ДУЭТ».

1) Завершить работу ПО можно, закрыв главное окно программы **duet_core**.

2) При проблемах завершения работы программы, необходимо завершить работу двух процессов в указанном порядке:

- **duet_supervisor**
- **duet_core**

3.6.2. Обновление приложения

1. Загрузите архив обновления **duet-xxx-win64.zip** (где «xxx» — номер версии) на регистратор или сервер с ПО «ДУЭТ» одним из способов:
   - «Подключение к удалённому рабочему столу» (RDP или другой протокол);
   - FTP (например, с помощью FileZilla).

2. Распакуйте архив нового дистрибутива **duet-xxx-win64.zip** во временное место на диске.
3. Скопируйте файлы и каталоги из распакованного архива в место установки ПО «ДУЭТ» (**C:\duet**), **исключая указанные**:

   \config - Каталог файлов настроек

4. Создайте ярлык для автозагрузки.

   Конфигурации **Регистратор:**

   В папке **C:\duet** создайте ярлык на файл **duet_start.bat**, скопируйте ярлык в автозагрузку.

   - Windows 7 — папка Start\All Programs\Startup
   - Windows 10 — папка по команде из Run: shell:startup

   Конфигурации **Сервер** — требуется задать пароль, пароль по умолчанию задан в файле настроек.

   В папке **C:\duet** создайте ярлык на файл **duet_start_standalone.bat**, скопируйте ярлык в автозагрузку.

   - Windows 7 — папка Start\All Programs\Startup
   - Windows 10 — папка по команде из Run: shell:startup

5. Следуйте дополнительным инструкциям по обновлению, если они предоставлены для данной конкретной версии ПО.

3.7. Удаление

В случае необходимости ПО «ДУЭТ» может быть удалено. Полное удаление также необходимо производить при установке новых несовместимых версий, что дополнительно указывается с обновлением.

3.7.1. Подготовка

Необходимое оборудование:

- регистратор или сервер, на который будет производиться установка;
- источник питания и кабель питания для устройства;
- набор для подключения:

Вариант 1:

- сервисный кабель для подключения устройства к рабочему компьютеру.

Вариант 2:

- монитор с VGA входом;
- компьютерная мышь;
- клавиатура;
- USB флешка;
- сервисный кабель для подключения устройства к монитору и USB.

Вариант 3:

- удалённый сетевой доступ к устройству (локальная сеть, частная защищённая сеть и др.);
- клиент подключения к удалённому рабочему столу устройства (для установки по сети).

Перед началом удаления необходимо остановить текущее ПО «ДУЭТ».

1) Завершить работу ПО можно, закрыв главное окно программы **duet_core**.

2) При проблемах завершения работы программы необходимо завершить работу двух процессов в указанном порядке:

- **duet_supervisor**
- **duet_core**

3.7.2. Удаление приложения

1. Удалите папку XML шаблонов фотоматериалов **templates**, если изменялся путь после установки (см. настройку в **C:\duet\config\config.json** — resultfile_templates_dir)
2. Удалите папку основного ПО: **C:\duet**
3. Удалите из автозагрузки ярлык на файл
   - **duet_start.bat** (конфигурация **Регистратор**)
   - **duet_start_standalone.bat** (конфигурация **Сервер**)

   Расположение в ОС:
   - Windows 7 — папка Start\All Programs\Startup
   - Windows 10 — папка по команде из Run: **shell:startup**

Папка материалов и БД «ДУЭТ» не удаляются, но могут быть удалены вручную.

3.8. Протоколы работы

Поддерживается несколько протоколов работы с внешними системами. Протокол выбирается в файле настроек **config.json**, поле **result_protocol** , например:

```
{
...
   "result_protocol": "default",
...
}
```

В понятие протокол вкладывается совокупность:

- сетевой протокол обмена данными (например, HTTP, JSON-RPC, SOAP т.п.);
- структура и формат хранения файлов материалов на диске (пути, форматы XML, JSON, JPG и т.п.);
- структура и формат БД для данных материалов (таблицы, поля, типы данных).

ВНИМАНИЕ! Протокол должен быть выбран один раз **после установки ПО перед запуском**. Его изменение в дальнейшем нежелательно, так как может привести к некорректной работе ПО. При изменении требуется провести удаление всех материалов с диска и из БД.

Поддерживаемые протоколы:

- **default** — протокол, по умолчанию, работает всегда, используется также для внутренних нужд функционирования системы (протокол и данные для веб-приложения и т.п.). Включает в себя:
  - MySQL БД main (с таблицами targets, events, materials), см. структуру в п.7.1;
  - Файлы на диске в отдельном каталоге со специальной структурой путей (см. Раздел 6 «Материалы»);
  - веб-протоколы — HTTP, JSON-RPC по HTTP (см. приложение «Описание веб-протоколов»).
- **samara —** специальный протокол использующий формат XML:
  - MySQL БД region_ru_samara (с таблицей xml_duet), см. структуру в п.7.1;
  - Файлы на диске в отдельном каталоге со специальной структурой путей (см. Раздел 6 «Материалы»);
  - веб-протоколы — HTTP/XML (см. приложение «Описание WEB протоколов»).

В следующем разделе подробно рассматривается первый запуск ПО «ДУЭТ», настройка и функционирование.

4. Запуск и остановка

ПО «ДУЭТ» готово к работе после установки и настроено на автоматический запуск при включении регистратора или выделенного сервера, в зависимости от конфигурации. В случае необходимости можно остановить работу ПО «ДУЭТ» и запустить вручную. Ниже описан ручной способ запуска и останова под ОС Windows.

Для запуска ПО в разных конфигурациях предусмотрено несколько вспомогательных файлов:

duet_start.bat - Скрипт запуска ПО в конфигурации **Регистратор**

duet_start_standalone.bat - Скрипт запуска ПО в конфигурации **Сервер**

Указанные вспомогательные файлы в свою очередь производят запуск программы супервизора ПО, которая запускает главное ядро ПО и контролирует его работу и выполняет перезапуск в критических ситуациях.

duet_supervisor.exe - Программа запуска и мониторинга **duet_core**

duet_core.exe - Исполняемый файл ядра ПО

Программу мониторинга **duet_supervisor** рекомендуется использовать как основной способ запуска приложения «ДУЭТ» для контроля и автоперезапуска ядра **duet_core** в случае возникновения проблем. Данная программа супервизор используется для запуска во вспомогательных **bat** файлах. Контроль производится периодически каждые 3 сек. запросами в ядро. Если ответ не получен в течение 20 сек. последует автоматический перезапуск. В начале работы duet_supervisor выделяет больший интервал ожидания (5 мин.) на выход duet_core в стабильное состояние, что необходимо для длительной очистки БД и каталога материалов при переполнении, при этом сохраняется базовый контроль над исполняемым процессом, который будет перезапущен и ранее, в случае сбоя.

Программа мониторинга и главный исполняемый файл ядра являются консольными приложениями и могут быть запущены с параметрами.

-h | --help - Отображение помощи по аргументам запуска

-v | --version - Отображение версии

-l | --log_level - Задать уровень протоколирования работы ПО, по умолчанию: DEBUG у duet_core, INFO у duet_supervisor

-m | --deploy_mode - Запуск в конфигурации:
m — module — на регистраторе
s — standalone — на выделенном сервере

-s | --simulator_mode arg - Отладочный запуск в режиме имитации нарушений:

	m — master — запуск на ведущем регистраторе с имитацией данных ведущего регистратора с реальными данными ведомого; b — both — запуск на выделенном сервере с имитацией ведущего и ведомого регистраторов.
--target arg	- (duet_supervisor) Путь к контролируемому приложению, по умолчанию duet_core
-p | --web_server_port arg	- (duet_supervisor) Порт для подключения и контроля за duet_core , по умолчанию 11000

Примеры отображения:

- Помощь duet_core

```
c:\duet>duet_core.exe -h
Duet v1.3.6
Duet average speed control system
Usage:  duet_core [options]
Supported options:
  -h [ --help ]                produce help message
  -v [ --version ]             display version
  -l [ --log_level ] arg (=DEBUG) application log level, [TRACE, DEBUG, INFO,
                                  WARNING, ERROR, CRITICAL, OFF]
  -m [ --deploy_mode ] arg (=m)   deploy mode, 'm' — module, 's' — standalone
  -s [ --simulator_mode ] arg (=none) simulator mode, 'm' — master, 'b' — both

Example run: ./duet_core
             ./duet_core -m s
             ./duet_core -l DEBUG -m s
```

- Информация о версии duet_core

```
c:\duet>duet_core.exe -v
Duet (duet_core) v1.3.6
```

- Помощь duet_supervisor

```
c:\duet>duet_supervisor.exe -h
Duet Supervisor v1.3.6
Duet average speed control system
Usage:  duet_supervisor [options]
Supervisor options:
  -h [ --help ]                produce help message
  -v [ --version ]             display version
  -l [ --log_level ] arg (=INFO)      supervisor log level, [TRACE, DEBUG,
                                  INFO, WARNING, ERROR, CRITICAL, OFF]
  --target arg                 duet_core target, executable or script.
                                  Pass args to target after '--'
  -p [ --web_server_port ] arg (=11000) duet_core web server port

Example run: ./duet_supervisor --target ./duet_core
```

./duet_supervisor --target ./duet_core -- -m m
./duet_supervisor -l DEBUG --target ./duet_core -- -m s -l DEBUG

- Информация о версии duet_supervisor

c:\duet>duet_supervisor.exe -v
Duet Supervisor (duet_supervisor) v1.3.6

4.1. Запуск программного обеспечения

Ручной запуск выполняется после подключения по удаленному доступу к операционной системе устройства:

Запуск ПО «ДУЭТ» можно выполнить:

- с помощью ярлыка в Автозагрузке.

  Конфигурация **Регистратор:** "**duet_start.bat**";

  Конфигурация **Сервер:** "**duet_start_standalone.bat**".

  Расположение в ОС:

  - Windows 7 — папка Start\All Programs\Startup;

  - Windows 10 — папка по команде из Run: **shell:startup**

- запуском программы напрямую:

  Конфигурация **Регистратор: "c:\duet\duet_start.bat**";

  Конфигурация **Сервер: "c:\duet\duet_start_standalone.bat**".

В случае успешного запуска на экране должно появиться консольное окно с информацией о запуске (рис. 4.1). Дальнейшее управление осуществляется через графический веб-интерфейс (см. раздел 5).

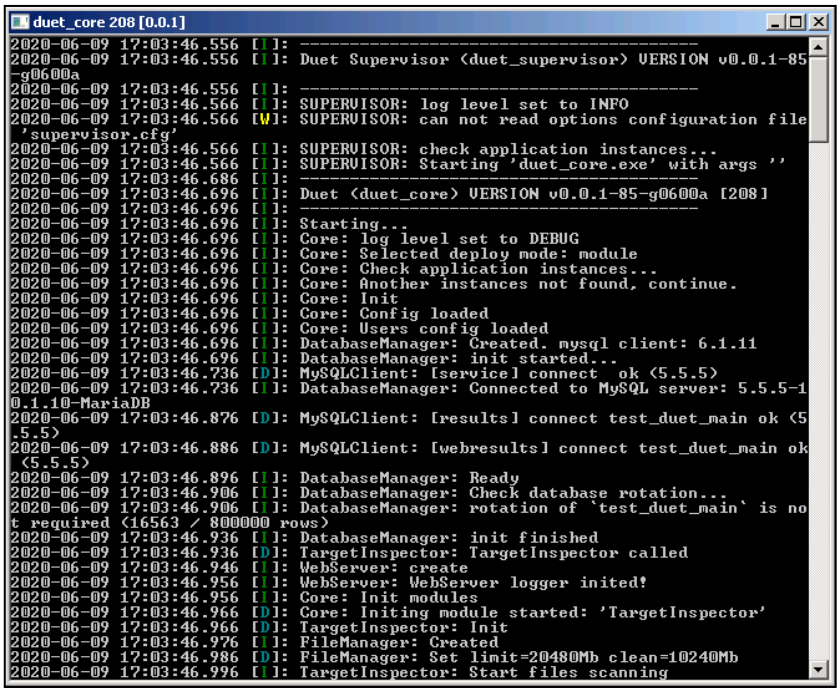


Рисунок 4.1 — Работающее ядро основного ПО

4.2. Остановка работы программного обеспечения

Ручная остановка работы ПО «ДУЭТ» может быть выполнена закрытием главного окна приложения **duet_core** (завершение работы приложения, рис. 4.2). При этом завершат работу процессы duet_core и duet_supervisor.

При проблемах завершения работы приложения необходимо завершить работу двух процессов в указанном порядке (рис. 4.3):

1) duet_supervisor
2) duet_core

ВНИМАНИЕ! *При завершении процесса* duet_core *без завершения* duet_supervisor *ПО будет продолжать перезапускаться!*

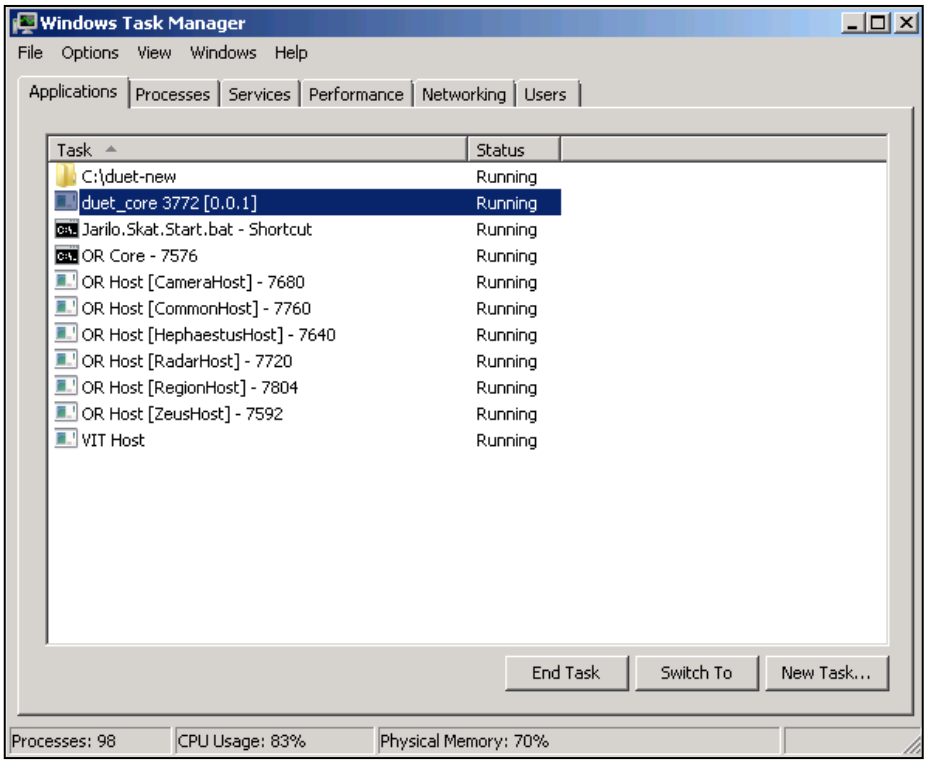


Рисунок 4.2 — Завершение работы приложения через Task Manager

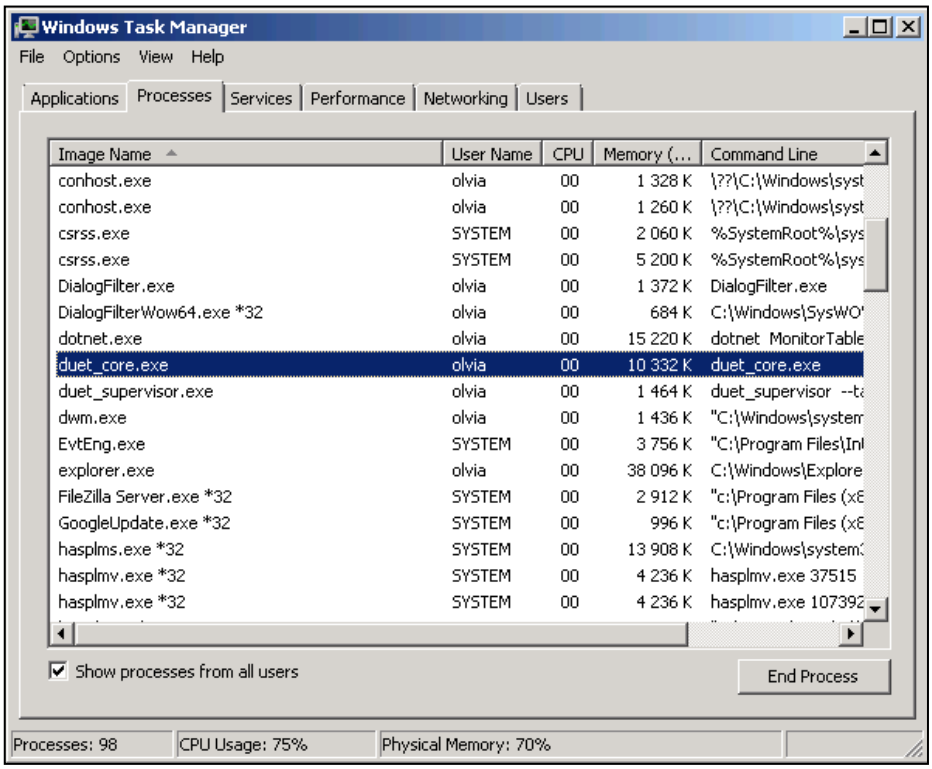


Рисунок 4.3 — Завершение работы процессов через Task Manager

5. Работа с веб-интерфейсом

Управление работой, настройка и мониторинг ПО «ДУЭТ» осуществляется через пользовательский веб-интерфейс.

5.1. Доступ к веб-интерфейсу

Для подключения к веб-интерфейсу необходимо перейти в браузере по адресу вида:

http://<IP-адрес устройства>:11000/

где IP-адрес представляет собой:

- в конфигурации Регистратор — IP-адрес регистратора с ПО «ДУЭТ»
- в конфигурации Сервер — IP-адрес выделенного сервера с ПО «ДУЭТ»

Поддерживаются современные браузеры: Google Chrome, Mozilla Firefox, Opera, Safari и другие. Версии не ранее 2019 г. выпуска.

Порт веб-интерфейса может быть изменен в конфигурации ПО. Этот порт одновременно используется и для других протоколов информационного взаимодействия: JSON-RPC по HTTP, HTTP/XML и т.п. Подробнее о протоколах указано в приложении «Описание веб-протоколов».

Конфигурация ВЕБ-сервера ПО находится в файле **web_server.ini.** Стандартное расположение файла при установке:

- Windows: ./config/web_server.ini (относительно папки ПО)
- Linux: /etc/duet/config/web_server.ini

Изменение конфигурации обычно не требуется, но может быть использовано для настройки адреса и порта веб-сервера, а также тонкой настройки задержек, кэширования и др. Изменение порта веб-сервера в **web_server.ini**:

```
[listener]
port=11000
```

При изменении порта в настройках web_server.ini требуется синхронно изменить порт в параметрах запуска программы **supervisor**, например, изменение на порт 9999 (по умолчанию 11000):

- Файл duet_start.bat:

```
@echo off
start duet_supervisor --target duet_core.exe -p 9999
```

- Файл duet_start_standalone.bat

```
@echo off
start duet_supervisor --target duet_core.exe -p 9999 -- -m s
```

ВНИМАНИЕ! Изменение настроек в файле конфигурации может привести к неработоспособности веб-интерфейса или ПО в целом!

5.2. Пользователи

ПО «ДУЭТ» поставляется с заданными профилями пользователей по умолчанию (табл. 5.1). Пользователей с одним видом прав в дальнейшем может быть несколько.

Таблица 5.1 — Пользователи

Имя	Права доступа	Пароль
Администратор	admin	password
Поверка	verify	password
Настройка	setup	password
Оператор	info	password
Журнал	journal	password

Дополнительно существует суперпользователь с максимальными правами (табл.5.2). Данный пользователь всегда только один.

Таблица 5.2 —. Системный пользователь

Имя	Права доступа	Пароль
root	root	pass

Реестр пользователей хранится в файле: **C:\duet\config\users.json**

Пароли пользователей хранятся в виде хэша MD5.

В зависимости от прав доступа пользователя меняются доступные функции и доступные пункты меню, например:

- «Администратор» имеет доступ ко всем возможным пунктам меню и разделам;
- «Журнал» имеет доступ к Журналам.

Примерная структура файла реестра пользователей **users.json** в JSON формате приведена в Листинге 5.1.

Листинг 5.1 — Пример структуры файла users.json

```
{
   "version":"1",
   "users": [
      {
         "name": "root",
         "fullname": "",
         "position": "",
         "password_hash": "bad65492ede82b516379ec0289a7335d",
         "rights": "root"
      },
      {
         "name": "Администратор",
         "fullname": "",
         "position": "",
         "password_hash": "bad65492ede82b516379ec0289a7335d",
         "rights": "admin"
      },
      {
         "name": "Поверка",
         "fullname": "",
         "position": "",
         "password_hash": "bad65492ede82b516379ec0289a7335d",
         "rights": "verify"
      },
      {
         "name": "Настройка",
         "fullname": "",
         "position": "",
         "password_hash": "bad65492ede82b516379ec0289a7335d",
         "rights": "setup"
      },
      {
         "name": "Оператор",
         "fullname": "",
         "position": "",
         "password_hash": "bad65492ede82b516379ec0289a7335d",
         "rights": "info"
      },
      {
         "name": "Журнал",
         "fullname": "",
         "position": "",
         "password_hash": "bad65492ede82b516379ec0289a7335d",
         "rights": "journal"
      }
   ]
}
```

Поля данных пользователя:

- name — имя пользователя, например «Администратор», обязательное;
- fullname — полное имя или подробности, например «Иван Иванович»;
- position — *не используется*;
- password_hash — MD5 хэш пароля пользователя, обязательное;

- rights — права пользователя, обязательное, варианты:

  - root — максимальный доступ, должен быть единственный;

  - admin — максимальный доступ, может быть несколько администраторов;

  - verify — доступ функциям поверки;

  - setup — доступ к настройке системы;

  - info — доступ к информации о системе;

  - journal — доступ к журналам.

ВНИМАНИЕ! В начале работы требуется **обязательно сменить пароли** всех пользователей в файле реестра пользователей!

5.2.1. Генерация новых паролей

Хэши MD5 паролей можно вычислить средствами в Windows 7/10:

1. Вызвать командную строку Windows (cmd).
2. Записать в файл требуемый пароль, например "mypass123", вручную или командой.

   <nul set /p="mypass123" > tempfile.txt

3. Вызвать получение хэша от файла.

3.1. (Windows 7) С помощью утилиты **certutil.**

Вызвать получение хэша от файла.

certutil -hashfile tempfile.txt MD5

Получим:

MD5 хеш файла tempfile.txt:

ba d6 54 92 ed e8 2b 51 63 79 ec 02 89 a7 33 5d

CertUtil: -hashfile — команда успешно выполнена.

MD5 хэш (без пробелов строчными символами):

bad65492ede82b516379ec0289a7335d

3.2. (Windows 10) С помощью powershell команды **Get-FileHash.**

Запустить powershell.

Вызвать получение хэша от файла.

Get-FileHash -Algorithm MD5 -Path .\tempfile.txt | Format-List

Получим:

Algorithm : MD5

Hash : BAD65492EDE82B516379EC0289A7335D

Path : C:\Users\olvia\tempfile.txt

MD5 хэш (без пробелов строчными символами):

bad65492ede82b516379ec0289a7335d

3.3. (Windows/Linux) С помощью утилиты **openssl.**

Вызвать получение хэша от файла.

openssl dgst -md5 tempfile.txt

Получим:

WARNING: can't open config file: /etc/ssl/openssl.cnf

MD5(tempfile.txt)= bad65492ede82b516379ec0289a7335d

MD5 хэш (без пробелов строчными символами):

bad65492ede82b516379ec0289a7335d

4. Полученный хэш необходимо вписать в файл **users.json** в поле **password_hash** для требуемого пользователя.

5.3. Вход

При успешном запуске ПО при вводе указанного в п. 5.1 адреса в браузере открывается стартовое окно идентификации пользователя с серийными номерами регистраторов ПО «ДУЭТ» (рис. 5.1). Необходимо ввести предоставленные данные в соответствующие поля и нажать кнопку «Вход».

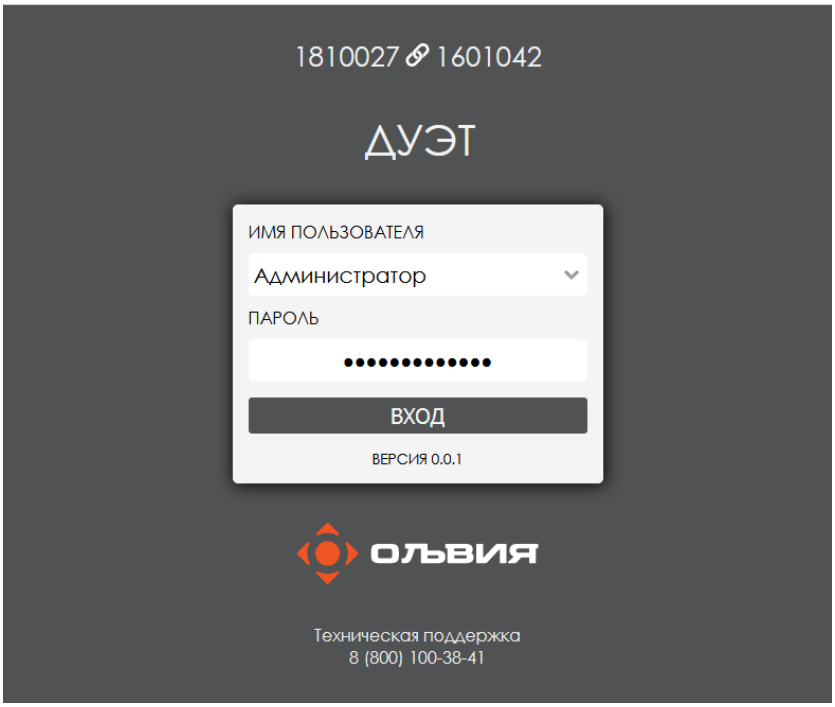


Рисунок 5.1 — Окно Входа

Если пользователь ввел некорректные данные, появляется сообщение «Неверное имя пользователя или пароль» (рис. 5.2).

Рисунок 5.2 — Окно Входа с сообщением о неверных данных

ВНИМАНИЕ! *Если окно идентификации не открылось в течение минуты, вероятно, запуск ПО не выполнен или выполнен с ошибками.*

После успешного входа осуществляется переход на главную страницу — «Контроль».

В начале работы необходимо произвести настройки на страницах разделов:

- «Администрирование»;
- «Настройка»;
- «Поверка».

5.4. Режимы работы

В ходе выполнения ПО «ДУЭТ» может находится в двух режимах:

- Настройка
- Контроль

Режим «Настройка» включен сразу после установки ПО, а также в него переходит ПО при любых изменениях настроек в веб-интерфейсе. В этом режиме не производится фиксаций нарушений ТС.

Режим «Контроль» переводит ПО в основной рабочий режим, в котором осуществляется подключение к регистраторам и начинает производиться сбор данных о проездах ТС с формированием материалов нарушений.

Переключение режима в Контроль производится на странице «Контроль».

5.5. Основные элементы интерфейса

При успешной авторизации отображаются страницы веб-интерфейса, состоящие из следующих элементов:

- Заголовок
- Рабочая область
- Меню

5.5.1. Заголовок

Заголовок страницы (рис. 5.3) включает:

- Кнопку выпадающего меню;
- Название системы;
- Название текущего раздела меню;
- Серийные номера регистраторов с индикатором связи между ними:
  - красный — связи нет или Контроль еще не производился;
  - белый — связь есть.
- Иконка режима работы, мигающая — выполнение перехода в режим (рис. 5.4);
- Иконка связи веб-интерфейса с устройством, на котором установлено ПО «ДУЭТ» (рис. 5.5).

Рис. 5.3. Вид заголовка страницы

а) Настройка

в) Контроль

Рисунок 5.4 — Отображение режимов работы в заголовке

Рисунок 5.5 — Вид иконки состояния связи

5.5.2. Меню

Меню включает все доступные для пользователя пункты с переходом к соответствующим разделам (рис. 5.6).

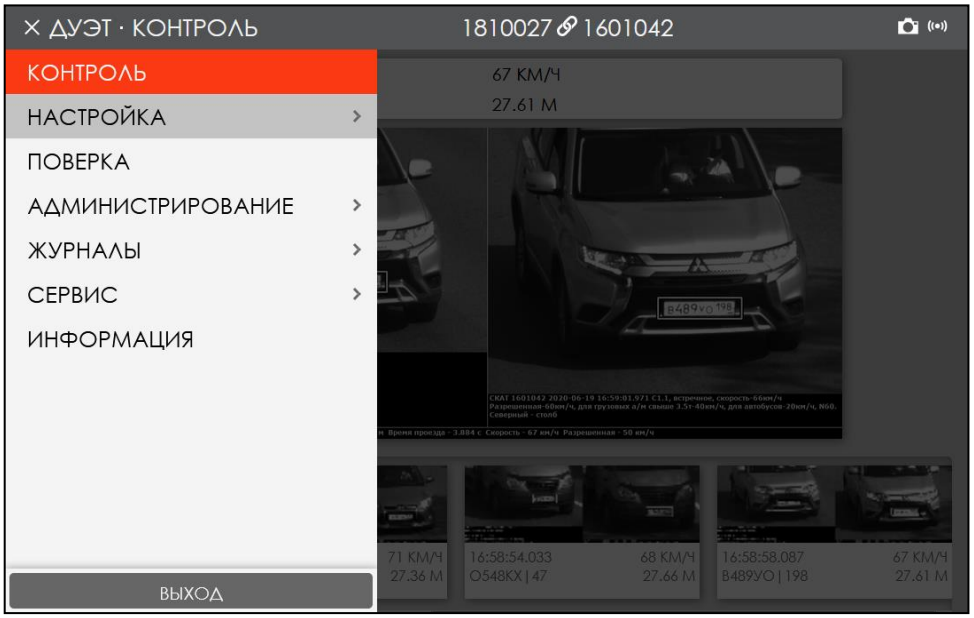


Рисунок 5.6 — Меню

5.6. Администрирование

Раздел «Администрирование» позволяет настраивать подключение к регистраторам и выполнять настройку формируемых материалов. Состоит из двух подразделов (рис. 5.7):

- параметры подключения;
- настройка материалов.

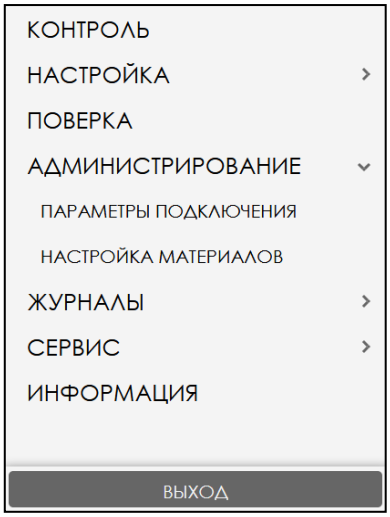


Рисунок 5.7 — Меню «Администрирование»

5.6.1. Параметры подключения

На странице настроек подключения указываются серийные номера подключаемых регистраторов и параметры доступа к ним. Набор настроек зависит от типа конфигурации ПО «ДУЭТ». Ведущий регистратор выбирается из списка по индексу: А / Б. Также можно указать или сформировать из приборов серийный номер самой системы.

В общих настройках указываются:

- Конфигурация — "НА СЕРВЕРЕ" / "НА РЕГИСТРАТОРЕ" — текущая конфигурация системы, изменение не допускается;
- Серийный номер системы — обычно в виде пары серийных номеров используемых регистраторов (например, 1908003/2002666), по кнопке можно сформировать из серийных номеров приборов и/или вписать вручную;
- Ведущий регистратор — "А" / "Б" — выбор регистратора, который играет роль ведущего из групп, показанных ниже;
- Использовать формат ГРЗ — ДА / НЕТ — использовать ли указание формата ГРЗ от регистраторов вручную, поля выбора формата появляются у каждой группы настроек регистраторов при использовании. Формат ГРЗ задается, исходя из особенностей распознавателя на конкретном регистраторе, можно указать:
  - "RUS" / "LAT" — набор символов;
  - "СТРОЧНЫЕ" / "ПРОПИСНЫЕ" — регистр символов;
  - "С РАЗДЕЛИТЕЛЕМ" / "БЕЗ РАЗДЕЛИТЕЛЯ" — наличие в номере символа разделителя в виде вертикальной черты "|".

Ожидаемые Форматы ГРЗ от регистраторов по умолчанию:

- «СКАТ» — RUS + СТРОЧНЫЕ + С РАЗДЕЛИТЕЛЕМ;
- «СКАТ-ПП» — RUS + СТРОЧНЫЕ + С РАЗДЕЛИТЕЛЕМ;
- «ОРАКУЛ» — LAT + СТРОЧНЫЕ + БЕЗ РАЗДЕЛИТЕЛЯ;
- «КРЕЧЕТ» — RUS + СТРОЧНЫЕ + С РАЗДЕЛИТЕЛЕМ.

Можно воспользоваться кнопкой рядом с переключателем "Использовать формат ГРЗ" для попытки автоопределения форматов (необходимо, чтобы регистраторы были доступны в это время, может потребоваться несколько попыток).

1) Конфигурация **Регистратор** (рис. 5.8).

Ведущим регистратором должен быть выбран тот, на котором установлено ПО «ДУЭТ». Для локального доступа к БД ведущего регистратора используется автоматически адрес 127.0.0.1 и прямой доступ в файловую систему.

- Ведущий регистратор:
  - Серийный номер;
  - Тип регистратора;
  - ГРЗ.

- Ведомый регистратор:
  - Серийный номер;
  - Тип регистратора;
  - IP-регистратора;
  - Имя пользователя FTP;
  - Пароль пользователя FTP;
  - ГРЗ.

2) Конфигурация **Сервер** (рис. 5.9).

Ведущий регистратор выбирается пользователем, может быть выбран любой. Для доступа к двум регистраторам, к БД и файлам материалов, используется доступ по ЛВС через IP-адрес.

- Ведущий регистратор
  - Серийный номер;
  - Тип регистратора;
  - IP-регистратора;
  - Имя пользователя FTP.
  - Пароль пользователя FTP;
  - ГРЗ.
- Ведомый регистратор
  - Серийный номер;
  - Тип регистратора;
  - IP-регистратора;
  - Имя пользователя FTP;
  - Пароль пользователя FTP.

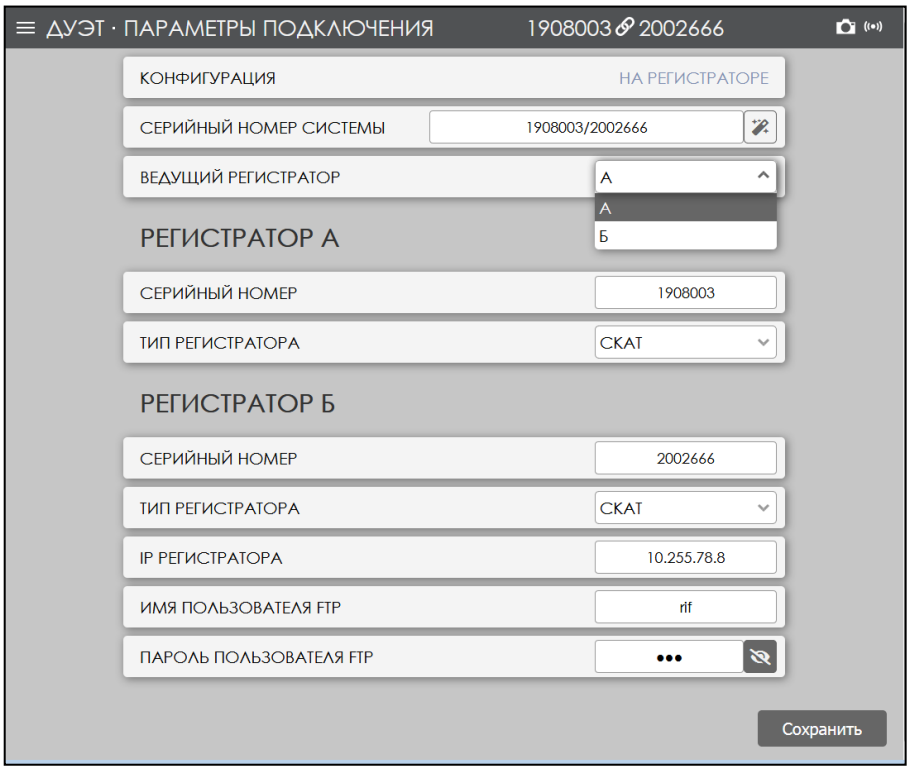


Рисунок 5.8 — Настройки подключения в конфигурации Регистратор

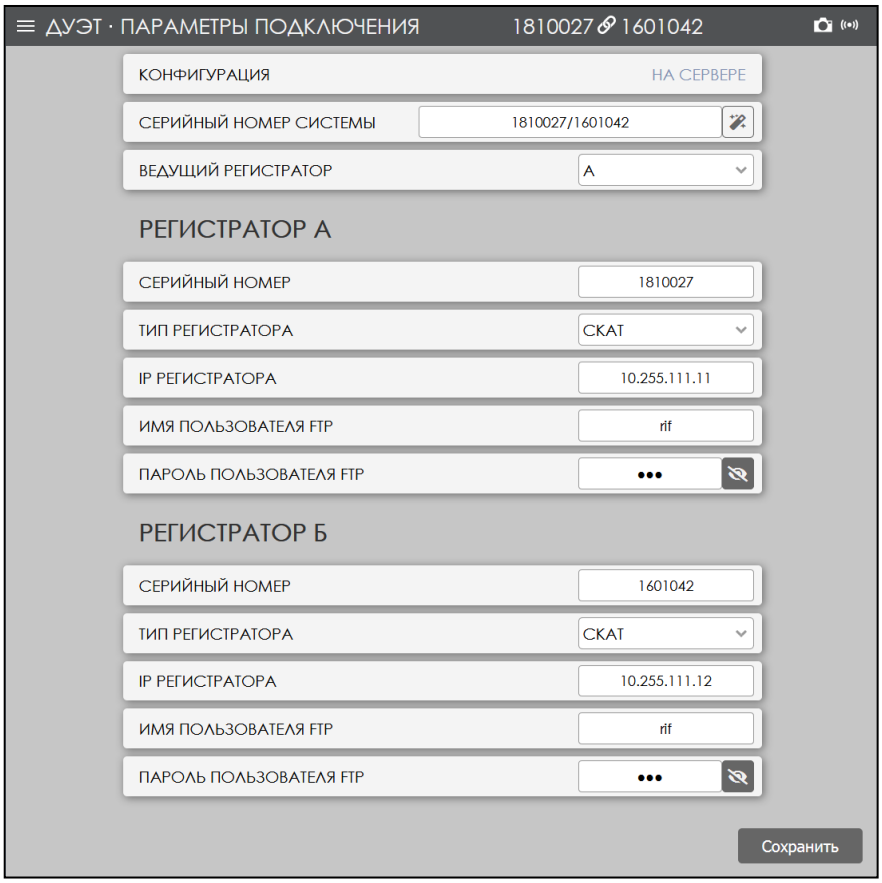


Рисунок 5.9 — Настройки подключения в конфигурации Сервер

Для двух регистраторов необходимо обязательно выбрать тип регистратора:

- «СКАТ»;
- «СКАТ-ПП»;
- «ОРАКУЛ»;
- «КРЕЧЕТ-СМ».

Для сохранения параметров нажмите кнопку «Сохранить». При выходе со страницы без сохранения параметров, всплывает сообщение «Имеются несохраненные изменения». При успешном сохранении параметров появляется сообщение «Параметры сохранены».

5.6.2. Настройка материалов

Раздел позволяет настроить параметры формирования материалов на события нарушений (рис. 5.10).

Задаются следующие параметры:

- Вид коллажа — выбор шаблона для фотоматериала, может быть выбран один из доступных файлов шаблонов XML из папки с шаблонами. Используется для формирования основного вида изображения в материале нарушения. «Папка с коллажами» показывает текущее место, где производится поиск файла коллажа формата XML;
- Формат основного материала — формат файла создаваемого материала нарушения, содержащего внедренные данные. На данный момент доступны варианты форматов:
  - protobuf;
  - jpg;
  - protobuf + jpg (два формата фотоматериала одновременно).
- Формат текстового файла — формат файла создаваемого текстового материала нарушения. На данный момент доступен один вариант — xml формат;
- Цифровая подпись — вид цифровой подписи, которой подписываются материалы нарушения. На данный момент доступен варианты:
  - RSA подписи;
  - ГОСТ подписи.

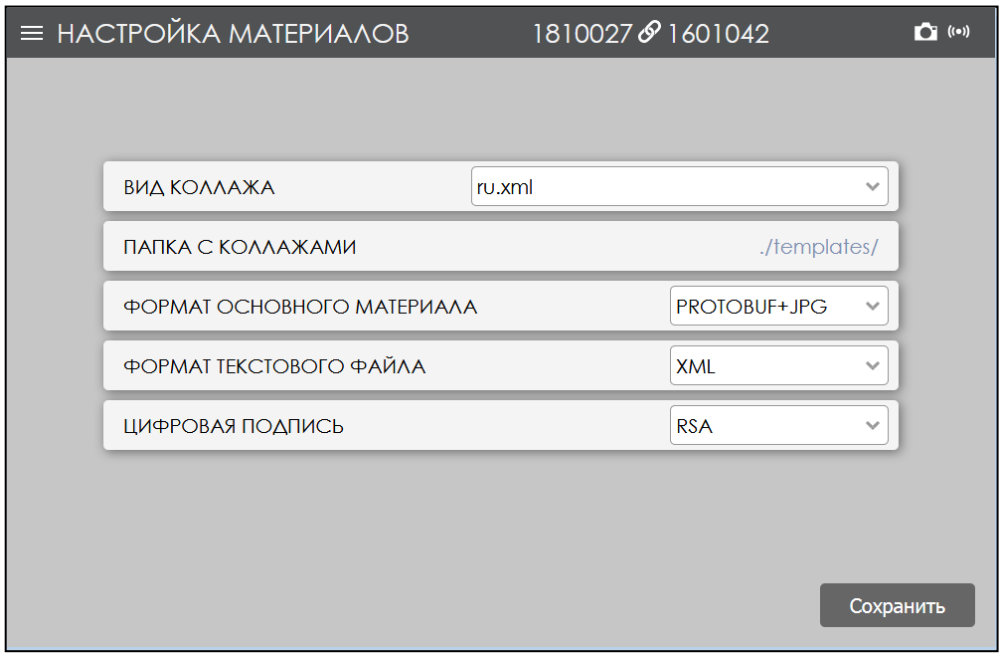


Рис. 5.10. Интерфейс настройки материалов

Для сохранения параметров нажмите кнопку «Сохранить». При выходе со страницы без сохранения параметров, всплывает сообщение «Имеются несохраненные изменения». При успешном сохранении параметров появляется сообщение «Параметры сохранены».

5.7. Настройка

Раздел «Настройка» позволяет выполнять настройки места расположения регистраторов и параметров контроля. Состоит из трех подразделов (рис. 5.11):

- место установки;
- параметры контроля;
- расположение регистраторов.

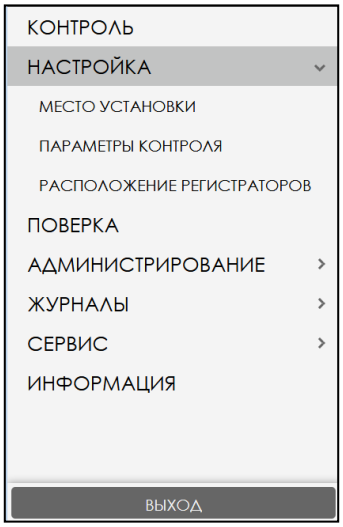


Рис. 5.11. Подразделы меню Настройки

Для сохранения параметров нажмите кнопку «Сохранить». При выходе со страницы без сохранения параметров, всплывает сообщение «Имеются несохраненные изменения». При успешном сохранении параметров появляется сообщение «Параметры сохранены».

5.7.1. Место установки

В данном разделе можно настроить данные о месторасположении системы. Указываются параметры по отношению к текущему **Ведущему** регистратору системы. Настройки будут использоваться при оформлении материалов фиксации событий (рис. 5.12):

- Расположение;
- Код ОКАТО;
- Направление движения — контролируемое направление движения с Ведущего регистратора:
  - встречное;
  - попутное;
  - любое;
- Направление встречного движения — текстовое описание встречного направления по отношению к Ведущему регистратору, например: «на ул. Ключевского» и т.п.;
- Направление попутного движения — текстовое описание попутного направления по отношению к Ведущему регистратору, например: «к ул. Ботаническая» и т.п.;
- Населенный пункт — находится ли система в населенном пункте или нет;
- Ограничение скорости (легковые ТС);
- Ограничение скорости (грузовые ТС).

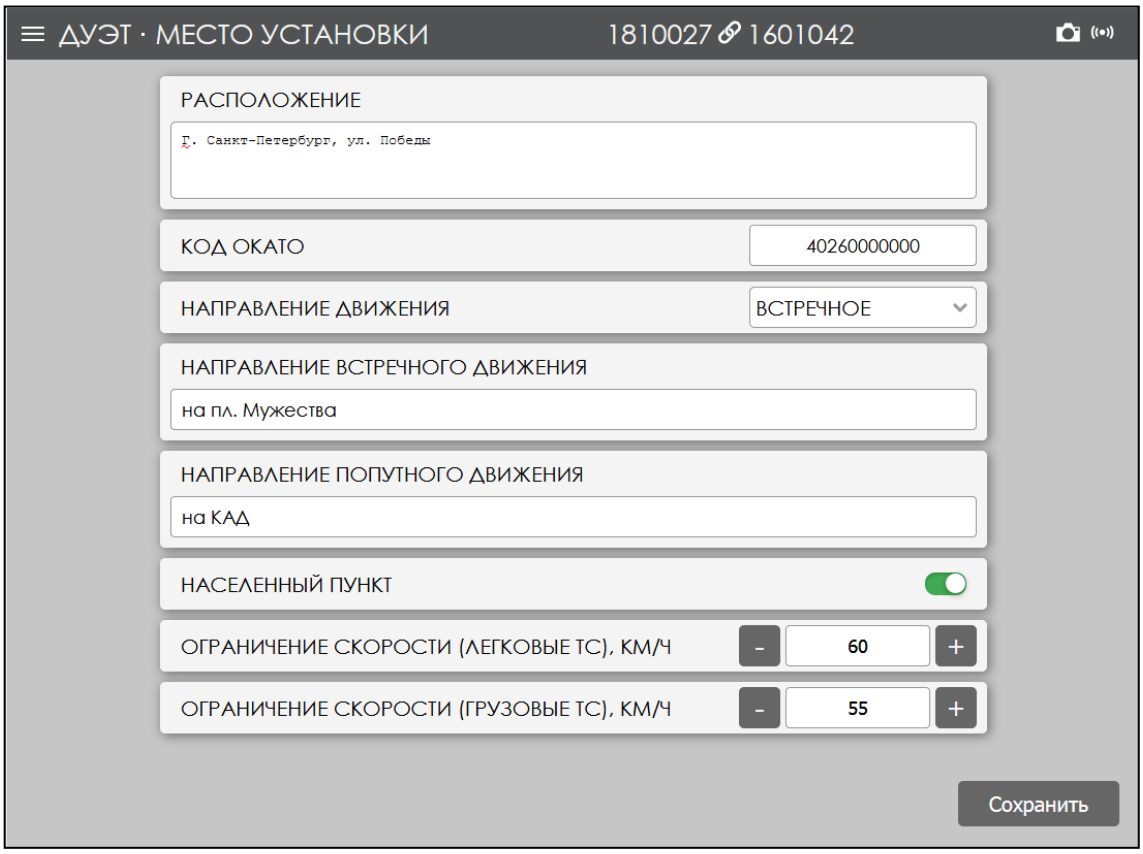


Рисунок 5.12 — Настройки места установки

Для сохранения параметров нажмите кнопку «Сохранить». При выходе со страницы без сохранения параметров, всплывает сообщение «Имеются несохраненные изменения». При успешном сохранении параметров появляется сообщение «Параметры сохранены».

5.7.2. Параметры контроля

На данной странице задаются параметры контроля зоны проезда (рис. 5.13):

- порог фиксации — порог фиксации скорости, выше которого производится фиксация нарушения скоростного режима и создаются материалы нарушения;
- приоритет ФМ мгновенной скорости — режим работы, при котором проверяются нарушения мгновенной скорости С1.1 с регистраторов. Если нарушение С1.1 есть, то нарушение средней скорости С1.2 не выносится. При этом запись о событии заносится в БД как С1.1, но файлы материалов на диске не формируются. В режиме работы по протоколу samara события С1.1 не заносятся в БД xml.

Для сохранения параметров нажмите кнопку «Сохранить». При выходе со страницы без сохранения параметров, всплывает сообщение «Имеются несохраненные изменения». При успешном сохранении параметров появляется сообщение «Параметры сохранены».

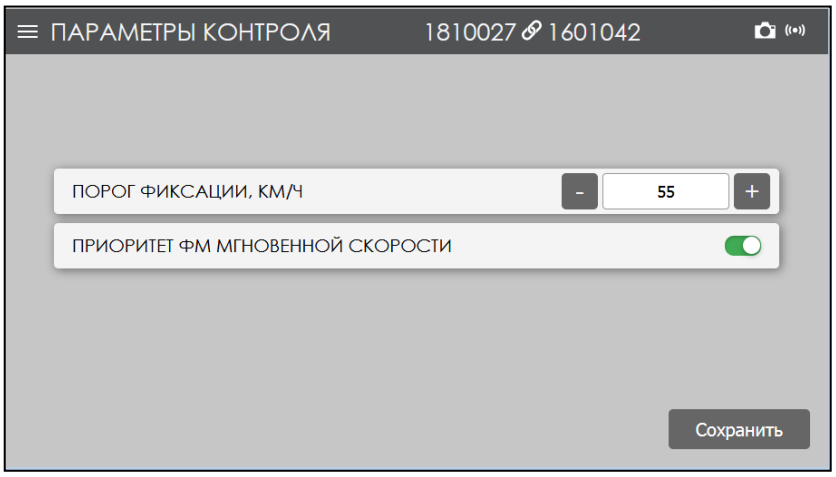


Рисунок 5.13 — Раздел «Параметры контроля»

5.7.3. Расположение регистраторов

В данном разделе задаются параметры взаимного расположения регистраторов с помощью кнопок смены положения под схематичными изображениями двух регистраторов:

- в одну сторону (рис. 5.14);
- разнонаправленно, друг на друга;
- разнонаправленно, друг от друга (рис. 5.15).

На данной странице также отображается текущая конфигурация ПО «ДУЭТ» и расстояние между регистраторами, заданное во время поверки. Ведущий регистратор на изображении отмечается **жирным** текстом серийного номера.

Также на данной странице задаются параметры:

- смещение по UTC — часовой пояс регистраторов, должен быть одинаков у обоих;
- параметры расположения Ведущего/Ведомого регистраторов:
  - расположение — текстовое описание месторасположения;
  - широта — координаты в градусах, с указанием полушария (опционально);
  - долгота — координаты в градусах, с указанием полушария (опционально).

Параметры расположения можно вписать вручную или попытаться прочитать с регистраторов с помощью кнопок поиска. Поиск данных доступен при корректно настроенных параметрах связи и наличии материалов на регистраторах. Максимальное время ожидания параметров — 10 сек.

ВНИМАНИЕ! *Параметры расположения регистраторов сохраняются один раз при настройке и используются только для дальнейших запросов состояния системы (getstatus в протоколе samara), в данные каждого нарушения автоматически идут непосредственно параметры расположения из очередного материала от регистраторов.*

Для сохранения параметров нажмите кнопку «Сохранить». При выходе со страницы без сохранения параметров, всплывает сообщение «Имеются несохраненные изменения». При успешном сохранении параметров появляется сообщение «Параметры сохранены».

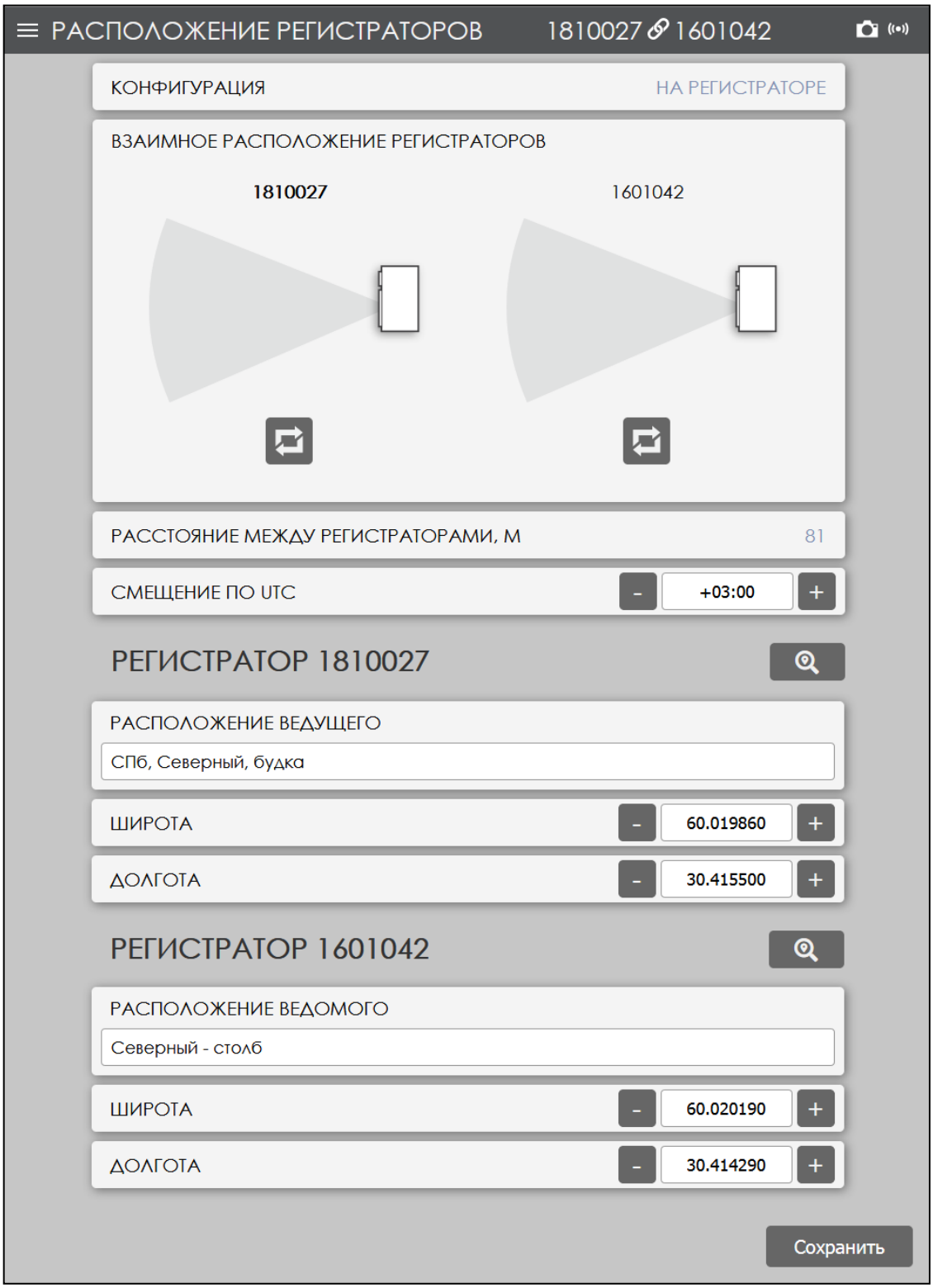


Рисунок 5.14 — Расположение регистраторов, направленных в одну сторону

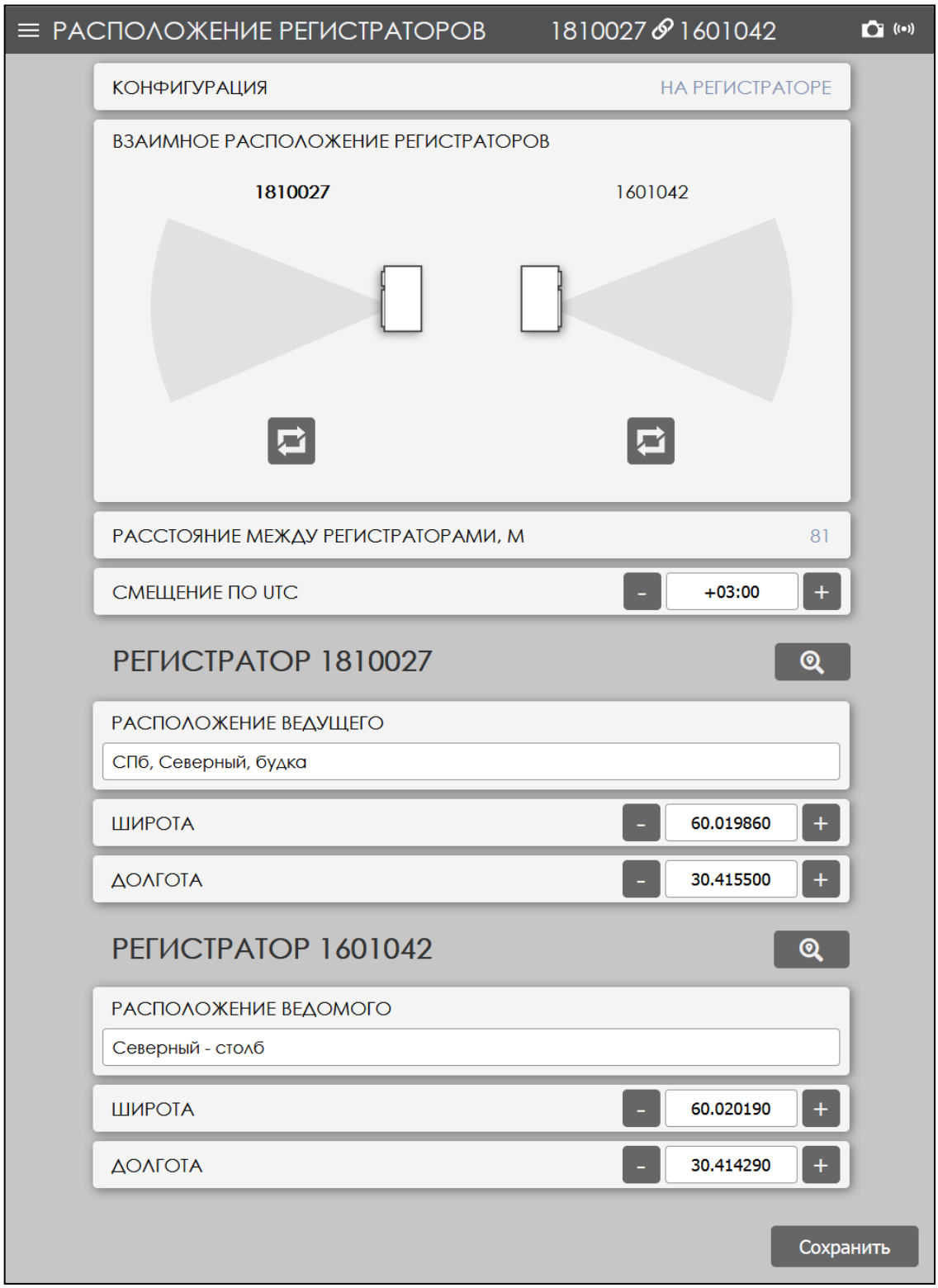


Рисунок 5.15 — Расположение регистраторов, направленных в разные стороны

5.8. Контроль

При успешной авторизации пользователя первой открывается главная страница Контроля (рис. 5.16). На странице отображается последняя фиксация в виде крупного изображения материала нарушения и история предыдущих фиксаций нарушений небольшими элементами.

Включение/Выключение «Контроля» выполняется с помощью кнопки в нижней части страницы. При включении начинают фиксироваться нарушения.

В области элемента каждого зафиксированного нарушения приводится информация о событии:

- время фиксации нарушения;
- вычисленная средняя скорость ТС;
- ГРЗ нарушителя;
- дистанция от нарушителя до Ведущего регистратора в метрах.

Рисунок 5.16 — Вид страницы «Контроль»

5.9. Поверка

Перед вводом системы в эксплуатацию должна быть проведена первичная поверка на месте её размещения. Поверка производится в соответствии с документом «Системы измерительные с фотофиксацией «ДУЭТ». Методика поверки» БКЮФ 201219.024МП.

В разделе веб-интерфейса «Поверка» вносится измерение расстояния между регистраторами, а также другие данные (рис. 5.17):

- номер свидетельства о поверке;

- дата проведения поверки;
- поверка действительная до (срок окончания);
- наименование поверяющей организации.

Для сохранения параметров нажмите кнопку «Сохранить». При выходе со страницы без сохранения параметров, всплывает сообщение «Имеются несохраненные изменения». При успешном сохранении параметров появляется сообщение «Параметры сохранены».

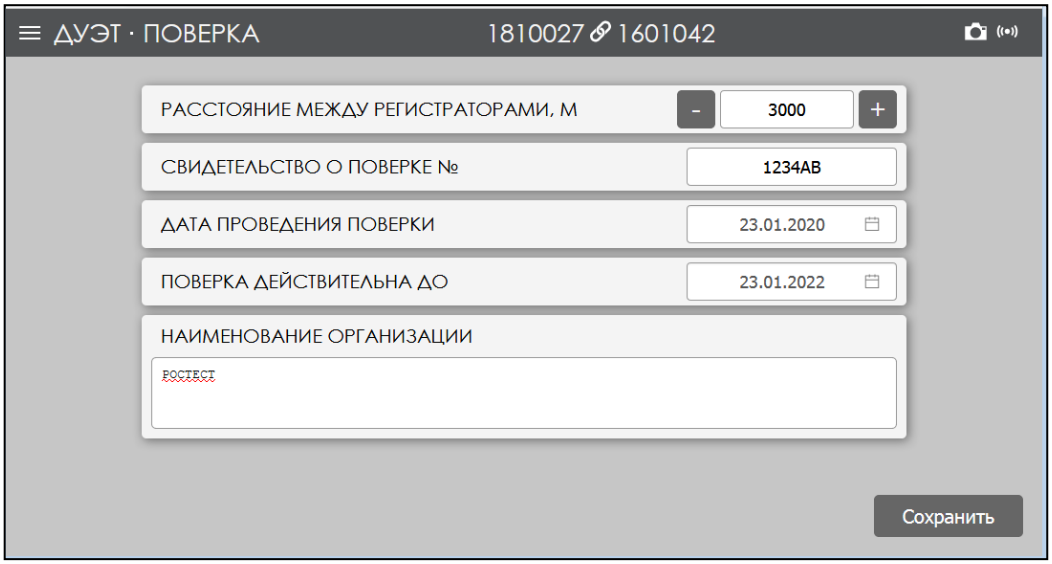


Рис. 5.17. Раздел параметров проведенной Поверки

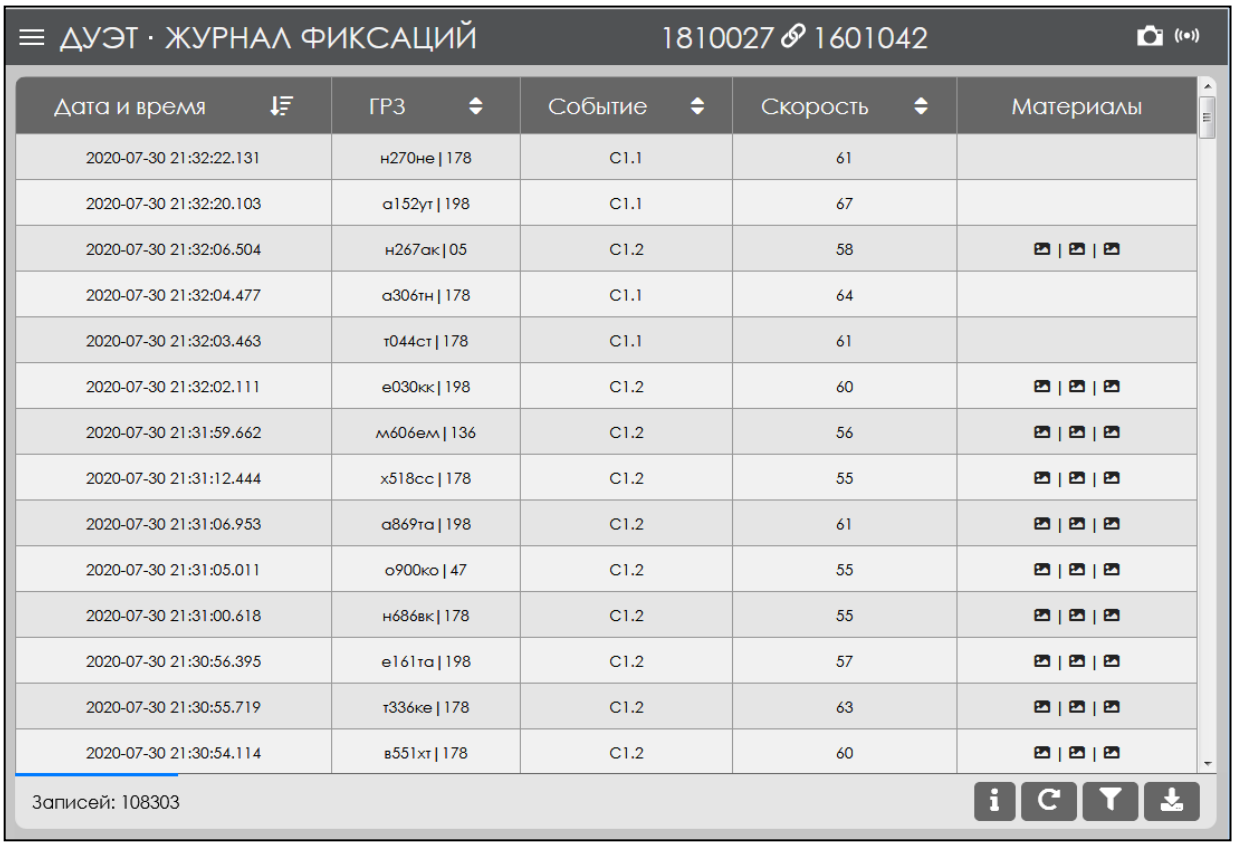
5.10. Журналы

5.10.1. Журнал фиксаций

Журнал фиксаций представляет собой таблицу, в которой отображается основная информация о зафиксированных нарушениях (рис. 5.18):

- дата и время;
- ГРЗ;
- событие (код правонарушения);
- скорость ТС;
- материалы.

Возможна сортировка по параметрам (столбцы).



ДУЭТ · ЖУРНАЛ ФИКСАЦИЙ 1810027 1601042

Дата и время	ГРЗ	Событие	Скорость	Материалы
2020-07-30 21:32:22.131	н270не | 178	С1.1	61	
2020-07-30 21:32:20.103	а152ут | 198	С1.1	67	
2020-07-30 21:32:06.504	н267ак | 05	С1.2	58	
2020-07-30 21:32:04.477	а306тн | 178	С1.1	64	
2020-07-30 21:32:03.463	т044ст | 178	С1.1	61	
2020-07-30 21:32:02.111	е030кк | 198	С1.2	60	
2020-07-30 21:31:59.662	м606ем | 136	С1.2	56	
2020-07-30 21:31:12.444	х518сс | 178	С1.2	55	
2020-07-30 21:31:06.953	а869та | 198	С1.2	61	
2020-07-30 21:31:05.011	о900ко | 47	С1.2	55	
2020-07-30 21:31:00.618	н686вк | 178	С1.2	55	
2020-07-30 21:30:56.395	е161та | 198	С1.2	57	
2020-07-30 21:30:55.719	т336ке | 178	С1.2	63	
2020-07-30 21:30:54.114	в551хт | 178	С1.2	60	

Записей: 108303

Рис. 5.18. Раздел «Журнал фиксаций»

Для просмотра материалов фиксации необходимо нажать на соответствующую иконку изображения. Внешний вид материала нарушения приведен на рис. 5.19. Также предусмотрена возможность скачивания изображений из материалов нажатием кнопки «Скачать».

Для нарушений С1.1 получение файлов материалов недоступно, так как они не создаются в режиме «приоритет ФМ мгновенной скорости».

Рисунок 5.19 — Материал фиксации нарушения

В нижней части страницы приведено общее количество записей, временной диапазон и расположены кнопки:

- информации об архиве;
- обновления данных на странице;
- настройки фильтра;
- сохранения/экспорта данных.

При нажатии на кнопку настройки фильтров открывается окно «Настройка фильтра», в котором выбираются параметры фильтрации данных в таблице (рис. 5.20). С помощью фильтра можно задать поиск:

- в пределах временного интервала;
- по знаку ГРЗ в виде "е396ов|178" или "е396*|178";
- по типу события/нарушения;
- по зафиксированной скорости ТС (от и до).

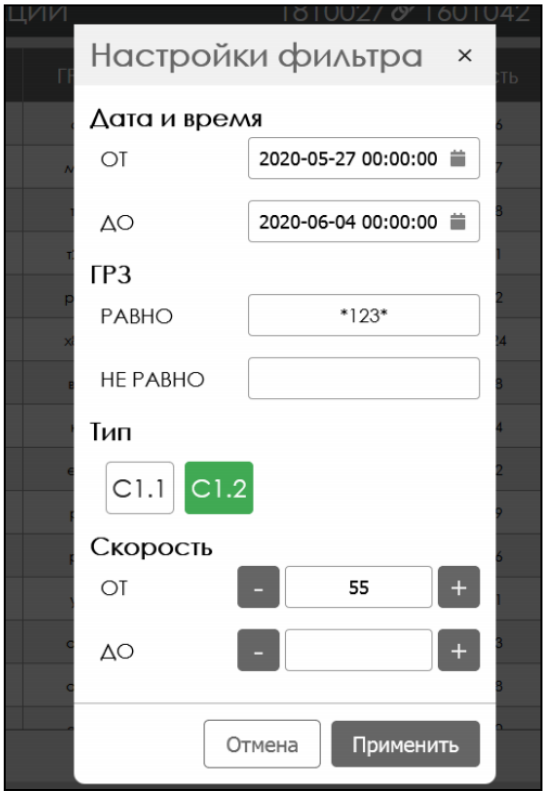


Рисунок 5.20 — Окно «Настройки фильтра» журнала

При нажатии на кнопку сохранения данных, таблица сохраняется в формате CSV и предлагается для скачивания.

Периодически производится очистка Журнала нарушений. Сохраняются последние ~125тыс. записей событий, а остальные удаляются (см. Раздел 6.9 "Автоматическая очистка БД материалов"). Данной процедурой удаляются только записи в БД, но не файлы на диске.

5.11. Сервис

В разделе «Сервис» размещены страницы с дополнительными сведениями о ПО:

- «Действия пользователей»;
- «Хранение информации».

5.11.1. Действия пользователей

На странице отображается журнал действий пользователей через веб-интерфейс, фиксируются события входа/выхода пользователей, изменения параметров и др. (рис. 5.21).

ДУЭТ · ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЕЙ 1810027 1601042

Дата и время	Пользователь	Действие	Комментарий
2020-06-05 22:52:15.515	Оператор	Вход в систему	::ffff:10.255.111.3
2020-06-05 22:52:08.125	Администратор	Выход из системы	::ffff:10.255.111.3
2020-06-05 22:33:03.813	Администратор	Вход в систему	::ffff:10.255.111.3
2020-06-05 21:54:40.181	root	Вход в систему	::ffff:10.255.111.3
2020-06-05 21:41:10.276	root	Вход в систему	::ffff:10.255.111.3
2020-06-05 21:26:22.146	Администратор	Вход в систему	::ffff:10.255.111.3
2020-06-05 21:24:35.696	root	Вход в систему	::ffff:10.255.111.3
2020-06-05 18:16:38.030	root	Изменение режима работы	контроль
2020-06-05 18:16:33.290	root	Изменение параметра	режим работы по умолчанию ("setup")
2020-06-05 18:16:33.290	root	Изменение параметра	ограничение скорости легковые (60)
2020-06-05 18:16:33.290	root	Изменение параметра	ограничение скорости грузовые (60)
2020-06-05 18:16:33.100	root	Изменение режима работы	настройка
2020-06-05 17:58:21.778	root	Изменение режима работы	контроль
2020-06-05 17:58:16.748	root	Изменение параметра	режим работы по умолчанию ("setup")
2020-06-05 17:58:16.748	root	Изменение параметра	speed_to_fixing (70)
2020-06-05 17:58:16.598	root	Изменение режима работы	настройка

Записей: 57/57

Рисунок 5.21 — Таблица действий пользователей

В таблице приведены следующие параметры:

- дата и время;
- пользователь;
- действие;
- комментарий.

Комментарии отображают информацию о совершенном действии пользователем. Например: на действие «Изменение параметра», в комментарии выводится какой параметр был изменен и новое значение параметра.

5.11.2. Хранение информации

На странице отображается объем данных на диске, занимаемый файлами нарушений (рис. 5.22).

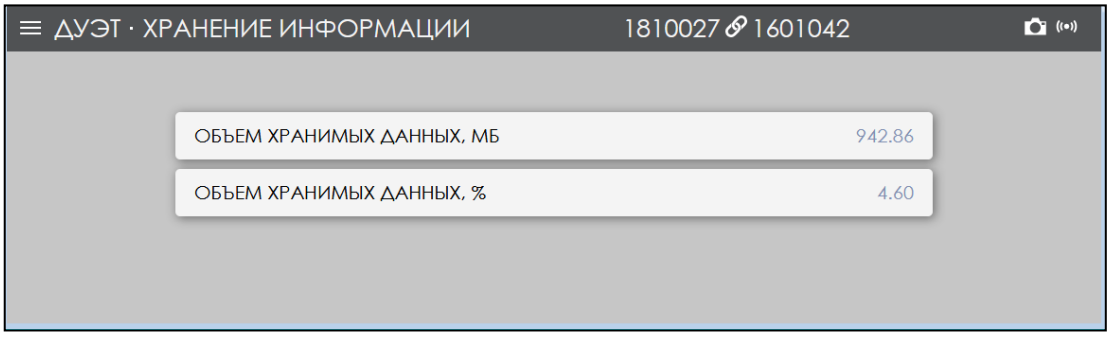


Рисунок 5.22 — Страница «Хранения информации»

5.12. Информация

На странице «Информация» можно проверить идентификационные данные ПО «ДУЭТ» и показатели работы ПО:

- Версия ПО;
- Название системы;
- Серийный номер системы;
- Серийный номер ведущего регистратора;
- Серийный номер ведомого регистратора;
- Длина контролируемого участка дороги;
- Время работы системы (часы, минуты, секунды, мс);
- Контрольная сумма метрологически значимой части ПО;
- Количество фиксаций.

Контрольная сумма метрологически значимой части программного обеспечения, должна совпадать с контрольной суммой, записанной в паспорте Системы.

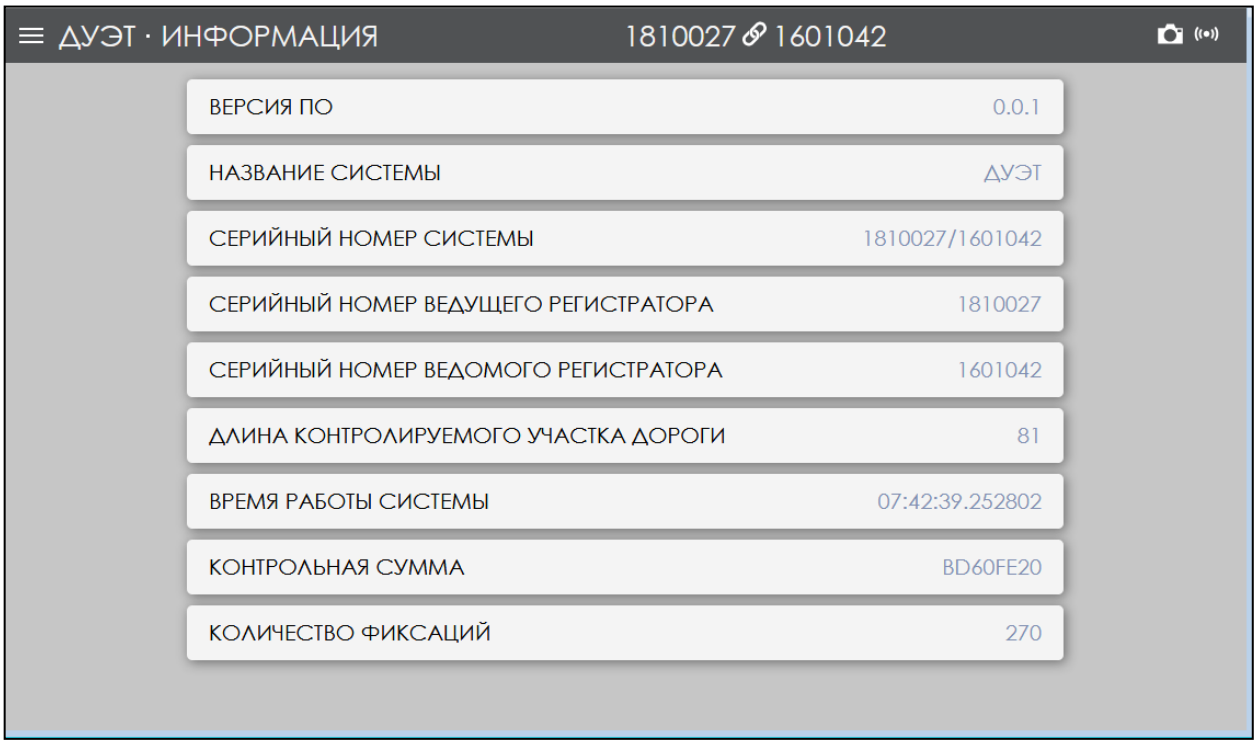


Рисунок 5.23 — Страница «Информация»

5.13. Выход

Для завершения работы в веб-интерфейсе необходимо нажать кнопку «Выход» в нижней части меню.

6. Материалы

Доступ к файлам материалов ПО «ДУЭТ» можно получить следующими способами:

- используя протокол передачи файлов FTP;
- доступ администратора, используя протокол доступа к удалённым рабочим столам (например, RDP).

6.1. Доступ через FTP

Доступ к файлам нарушений через FTP осуществляется с помощью FTP-клиента (FileZilla, WinSCP или другой).

В соответствующие поля вводятся IP-адрес устройства, на котором установлено ПО «ДУЭТ» (сервер или регистратор), логин/пароль пользователя. После подключения к устройству, открывается доступ к файлам архива материалов.

6.2. Доступ через RDP

Для подключения к удаленному рабочему столу необходимо открыть соответствующую программу-клиент RDP и ввести адрес устройства (рис. 6.1).

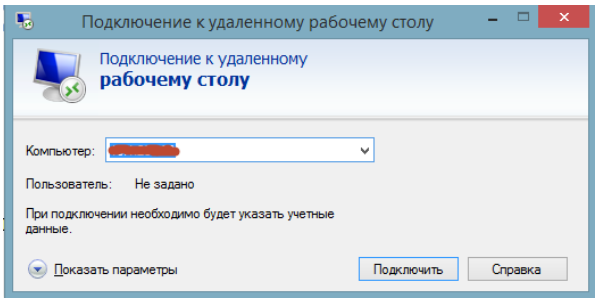


Рисунок 6.1 — Подключение к удаленному рабочему столу

Доступ к файлам материалов при подключении через RDP осуществляется следующими способами (рис. 6.2):

- через стандартную программу «Проводник»;
- через любой другой файловый менеджер (например, Free Commander, Total Commander).

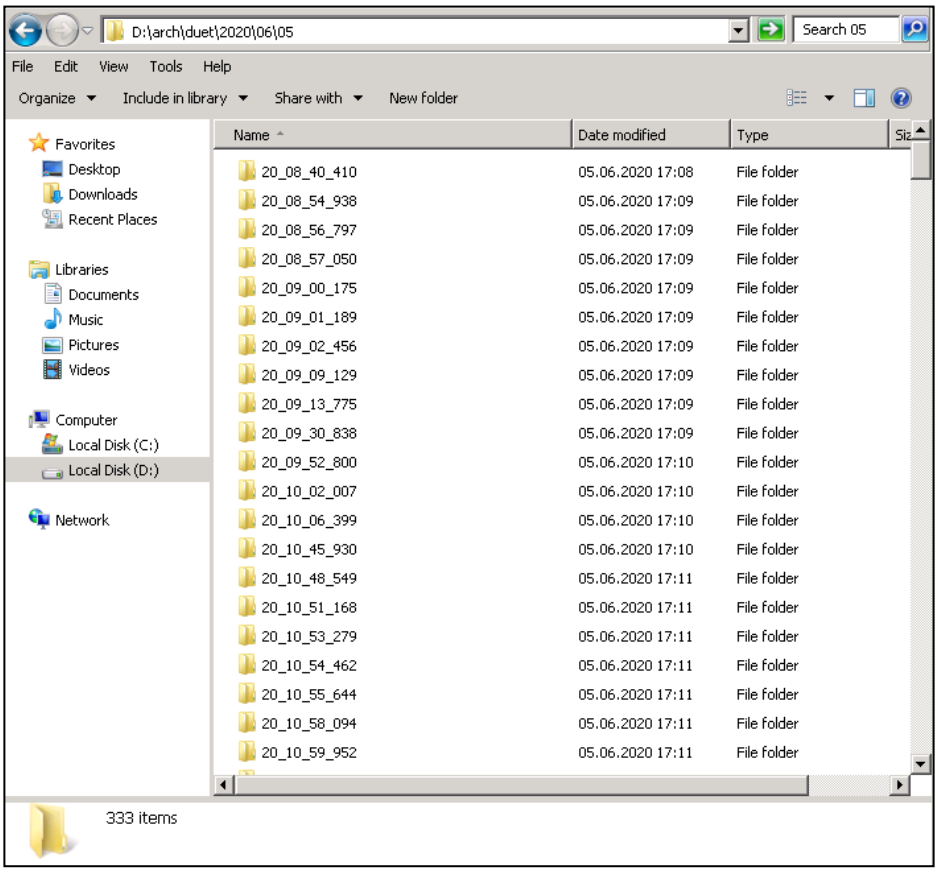


Рисунок 6.2 — Способ доступа к файлам материалов ПО «ДУЭТ» на устройстве при подключении по RDP

6.3. Состав материалов

Материалы сохраняются в каталоге, заданном в config.json — **result_directory** — (рис. 6.2), например: **D:/arch/duet**

Часть файлов при работе ПО являются временными, например исходные материалы с регистраторов. Они по умолчанию сохраняются в тот же каталог, заданный в result_directory, но если задан непустой параметр в config.json — **result_directory_tmp**, то временные файлы будут сохраняться отдельно по указанному пути (например, **D:/arch/duet_tmp**).

Структура и именование файлов зависят от протокола взаимодействия, с которым запущено ПО.

6.3.1. Протокол default

Файлы материалов находятся в подкаталогах, разделенных по времени событий (рис. 6.3). Например: **D:/arch/duet/2020/06/06/00_17_39_082/**. Все файлы одного события хранятся в одном каталоге со временем события, например "/**00_17_39_082**". Подкаталоги событий одного календарного дня хранятся в одном общем каталоге. Структура подкаталогов временных файлов (**result_directory_tmp**) полностью соответствует структуре основного каталога материалов.

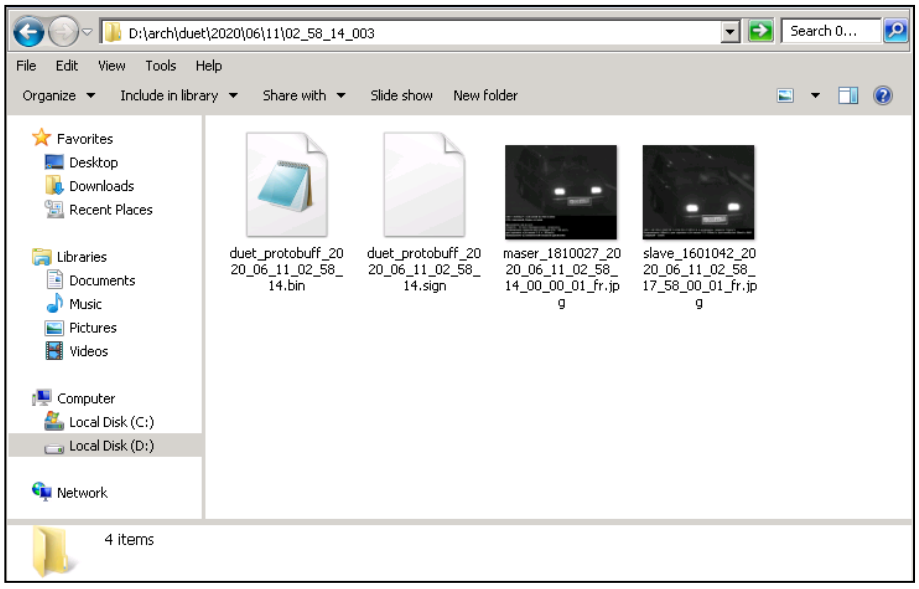


Рисунок 6.3 — Файлы нарушений

Материалы фиксаций включают в себя от 4 до 8 файлов в зависимости от настройки состава материалов в ПО (рис. 6.3-6.4, табл.6.1):

- 2 графических файла с фиксациями проезда от ведущего и ведомого регистраторов в формате jpg;
- файл материала нарушения в формате bin (protobuf);
- файл материала нарушения в формате jpg;
- текстовый файл материала нарушения в формате xml;
- цифровые подписи материалов нарушений protobuf, jpg и xml.

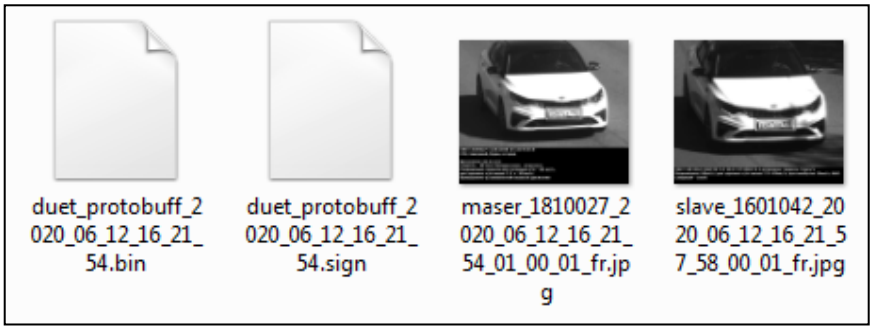


Рисунок 6.4 — Состав основных материалов фиксации

Таблица 6.1 — Файлы материалов

Основной файл в формате protobuf:	
duet_protobuff_2020_06_06_00_17_39.bin	Материал нарушения в формате protobuf **с внедренными данными и JPG изображением**
duet_protobuff_2020_06_06_00_17_39.sign	Файл цифровой подписи protobuf файла
Основной файл в формате jpg:	
duet_2020_06_06_00_17_39.jpg	Материал нарушения в виде изображения jpg **с внедренными данными в xml формате**
duet_2020_06_06_00_17_39.sign	Файл цифровой подписи jpg файла
Текстовый материал в формате xml:	
duet_2020_06_06_00_17_39_info.xml	Материал нарушения в виде текстового файла xml
duet_2020_06_06_00_17_39_info.sign	Файл цифровой подписи xml файла
Увеличенный кадр с ТС ведущего регистратора:	
maser_1810027_2020_06_06_00_17_39_08_00_01_fr	Изображение в формате JPG с **внедренными данными**
Увеличенный кадр с ТС ведомого регистратора:	
slave_1601042_2020_06_06_00_17_42_38_00_01_fr.jpg	Изображение в формате JPG с **внедренными данными**

Название файлов состоит из следующих полей:

- тип файла
  - maser — материал ведущего регистратора;
  - slave — материал ведомого регистратора;
  - duet_protobuf_*** — материал нарушения protobuf;
  - duet_*** — материал нарушения jpg;
  - duet_***_info — материал xml.
- (maser/slave) серийный номер ведущего/ведомого регистраторов, например 1711002;
- год, месяц, день;

- часы, минуты, секунды;
- (maser/slave) миллисекунды (десятки);
- (maser/slave) индивидуальный индекс (для отличия событий с одинаковым временем);
- (maser/slave) код типа события/нарушения;
- (maser/slave) суффикс типа файла (fr).

Вид изображения основного файла материала задается через файлы XML шаблонов. По умолчанию, файлы шаблонов располагаются в папке:

C:/duet/templates

Путь к папке шаблонов может быть изменён настройкой в config.json:

resultfile_templates_dir

Стандартно вместе с ПО предоставляется базовый шаблон:

ru.xml

Шаблон материалов выбирается через ВЕБ-интерфейс на странице «АДМИНИСТРИРОВАНИЕ» — «НАСТРОЙКА МАТЕРИАЛОВ» — «ВИД КОЛЛАЖА». Формат внедряемых дополнительных данных, не зависит от графического вида материала. Примеры основного файла материала приведен на рис. 6.5 (ru.xml). Подробное описание формата XML шаблонов приведено в приложении «Описание xml шаблонов фотоматериалов».

Рис. 6.5. Изображение из основного файла материала нарушения

6.3.2. Протокол samara

Файлы материалов находятся в подкаталогах, разделенных по типу и времени событий (рис. 6.6). Например: **D:/arch/duet/C1/2020/06/06/**. Все файлы событий одного календарного дня хранятся в одном каталоге. Структура подкаталогов временных файлов (**result_directory_tmp**) полностью соответствует структуре основного каталога материалов.

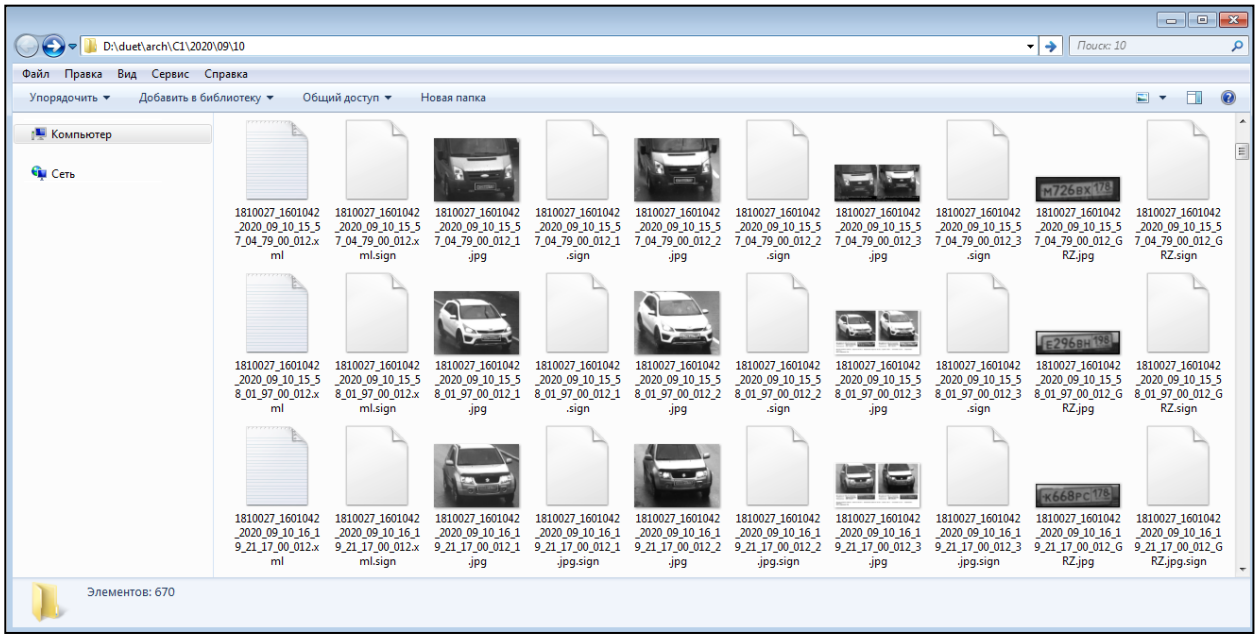


Рисунок 6.6 — Файлы нарушений

Материалы фиксаций включают в себя от 4 до 14 файлов в зависимости от настройки состава материалов в ПО (рис. 6.6-6.7, табл.6.2):

- 2 графических файла с фиксациями проезда от ведущего и ведомого регистраторов в формате jpg;
- файл материала нарушения в формате bin (protobuf);
- файлы материалов нарушения в формате jpg (1-3);
- файл фото ГРЗ нарушения в формате jpg;
- текстовый файл материала нарушения в формате xml;
- цифровые подписи материалов нарушений protobuf, jpg и xml.

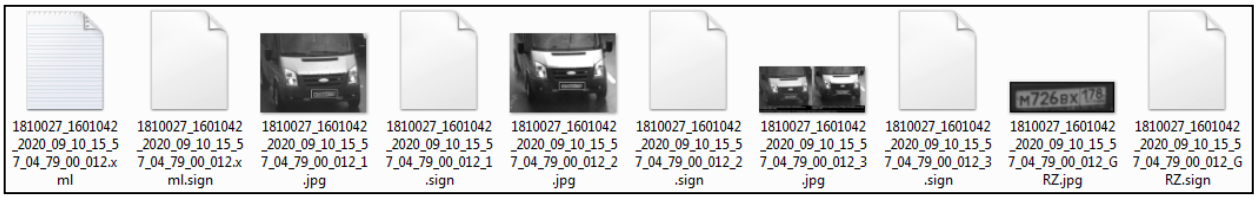


Рисунок 6.7 — Состав основных материалов фиксации

Таблица 6.2 — Файлы материалов протокола samara

Основной файл в формате protobuf	
1810027_1601042_2020_08_13_18_19_01_71_00_012.bin	Материал нарушения в формате protobuf **с внедренными данными и JPG изображением**
1810027_1601042_2020_08_13_18_19_01_71_00_012.bin.sign	Файл цифровой подписи protobuf файла
Файл фото ТС проезда с первого регистратора (jpg):	
1810027_1601042_2020_08_13_18_19_01_71_00_012_1.jpg	Фото ТС 1 в виде изображения jpg
1810027_1601042_2020_08_13_18_19_01_71_00_012_1.jpg.sign	Файл цифровой подписи jpg файла фото ТС 1
Файл фото ТС проезда со второго регистратора (jpg):	
1810027_1601042_2020_08_13_18_19_01_71_00_012_2.jpg	Фото ТС 2 в виде изображения jpg
1810027_1601042_2020_08_13_18_19_01_71_00_012_2.jpg.sign	Файл цифровой подписи jpg файла фото ТС 2
Основной файл в (jpg):	
1810027_1601042_2020_08_13_18_19_01_71_00_012_3.jpg	Материал нарушения в виде изображения jpg **с внедренными данными в XML формате**
1810027_1601042_2020_08_13_18_19_01_71_00_012_3.jpg.sign	Файл цифровой подписи jpg файла
Текстовый материал в формате xml:	
1810027_1601042_2020_08_13_18_19_01_71_00_012.xml	Материал нарушения в виде текстового файла xml
1810027_1601042_2020_08_13_18_19_01_71_00_012.xml.sign	Файл цифровой подписи xml файла
Увеличенный кадр с ТС ведущего регистратора:	
maser_1810027_2020_08_13_18_18_58_04_00_01_fr.jpg	Изображение в формате JPG

Основной файл в формате protobuf	
	с **внедренными данными**
Увеличенный кадр с ТС ведомого регистратора:	
slave_1601042_2020_08_13_18_19_01_71_00_01_fr.jpg	Изображение в формате JPG с **внедренными данными**

Название файлов состоит из следующих полей:

- тип файла
  - maser — материал ведущего регистратора;
  - slave — материал ведомого регистратора;
  - остальные — материалы нарушения системы.
- (maser/slave) серийный номер ведущего/ведомого регистраторов, например 1711002;
- (остальные) серийный номер системы (обычно — номера ведущего и ведомого регистраторов), например 1810027_1601042;
- год, месяц, день;
- часы, минуты, секунды;
- миллисекунды (десятки);
- индивидуальный индекс (для отличия событий с одинаковым временем);
- код типа события/нарушения (01 для С1.1, 012 для С1.2);
- (основной jpg) порядковый номер фотоматериала;
- (ГРЗ jpg) суффикс GRZ;
- (maser/slave) суффикс типа файла (fr).

Файлы цифровых подписей именуются как подписываемый файл добавлением к полному имени расширения ".sign".

Вид изображения основного файла материала (jpg/protobuf) задается через файлы XML шаблонов. По умолчанию, файлы шаблонов располагаются в папке:

C:/duet/templates

Путь к папке шаблонов может быть изменён настройкой в config.json:

resultfile_templates_dir

Стандартно вместе с ПО предоставляется базовый шаблон:

ru.xml

Шаблон материалов выбирается через веб-интерфейс на странице «АДМИНИСТРИРОВАНИЕ — НАСТРОЙКА МАТЕРИАЛОВ» — «ВИД КОЛЛАЖА». Формат внедряемых дополнительных данных не зависит от графического вида материала. Примеры основного файла материала приведен на рис.6.8 (ru.xml). Подробное описание формата XML шаблонов приведено в приложении «Описание xml шаблонов фотоматериалов».

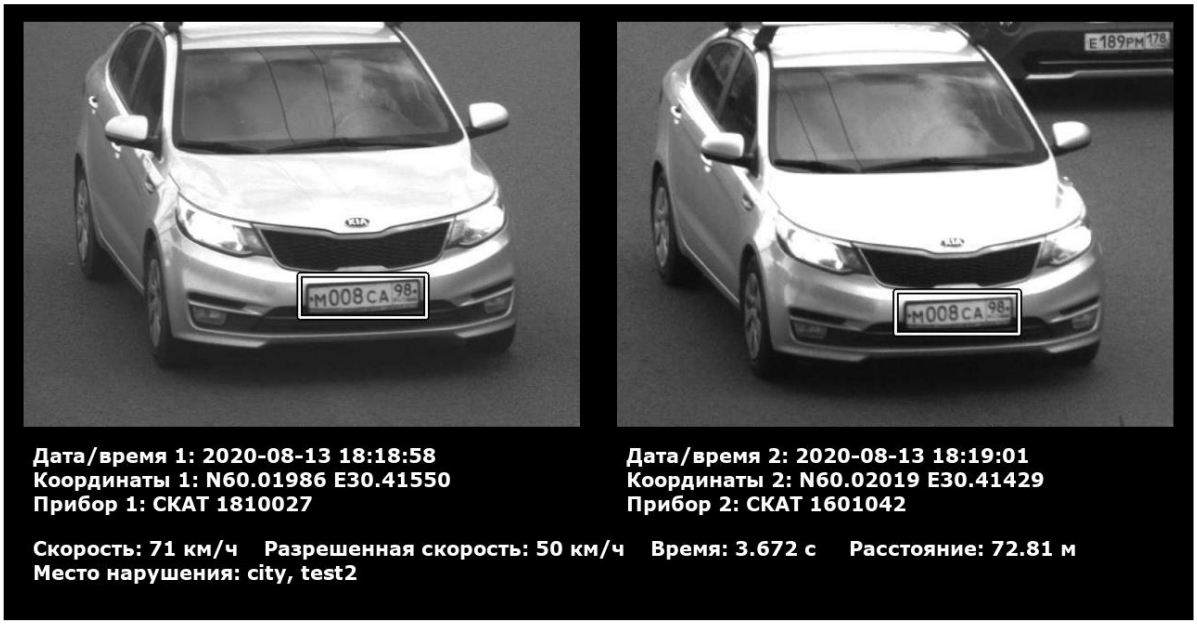


Рисунок 6.8 — Изображение из основного файла материала нарушения

6.4. Формат protobuf

Основной файл материала (duet_protobuff_2020_06_06_00_17_39.bin) представляет собой формат protobuf и содержит внедренные бинарные метаданные специальной структуры. Для извлечения информации требуются специальные утилиты. Поставляемые файлы *.proto (OutDataFormat.proto) содержат расшифровку данных структур. А также возможно с использованием данного файла согласно спецификации Protocol Buffers сгенерировать исходный код для расшифровки данных на необходимом языке программирования для упрощения процедуры интеграции.

Ниже в Листинге 6.1 приведена структура, порядок и типы данных в формате protobuf. В таблице 6.3 приведены сведения обо всех полях формата.

Листинг 6.1 — Формат материала protobuf

```
syntax = "proto2";

package duet.protobuf;

import "protobuf/bcl.proto";

message BinaryData {
  optional ImageData Image = 1;
  optional TargetData Target = 2;
  optional DeviceData Device = 3;
  optional CertificateData Certificate = 4;
  optional PlacementData Placement = 5;
  repeated DeviceData Registarators = 6;
  optional string Comment = 7;
}
message CertificateData {
  optional bcl.DateTime ValidThru = 1;
```

```
  optional string Number = 2;
  optional string Agency = 3;
  optional bcl.DateTime Date = 4;
}
message DeviceData {
  optional string Name = 1;
  optional string Serial = 2;
}
message ImageData {
  optional bytes Data = 1;
  optional string Format = 2;
  optional int32 Width = 3 [default = 0];
  optional int32 Height = 4 [default = 0];
}
message PlacementData {
  optional double Distance = 1 [default = 0];
  optional string PlaceName = 2;
  optional bool IsCity = 3 [default = false];
  optional string Street = 4;
  optional string Building = 5;
  optional string Direction = 6;
  optional int32 SpeedLimit = 7 [default = 0];
  optional int32 BugSpeedLimit = 8 [default = 0];
  optional int32 ResolveLimit = 9 [default = 0];
}
message TargetData {
  optional string Liplate = 1;
  optional double Speed = 2 [default = 0];
  optional double Distance = 3 [default = 0];
  optional bcl.TimeSpan Span = 4;
  optional bcl.DateTime StartBound = 5;
  optional bcl.DateTime StopBound = 6;
  optional double MasterDistance = 7 [default = 0];
  optional double SlaveDistance = 8 [default = 0];
  optional bool Direction = 9 [default = false];
```

Таблица 6.3 — Поля формата protobuf

Параметр/Блок	**Тип**	**Описание**
BinaryData		**Общая информационная структура**
Image	ImageData	Информация о фотоизображении
Target	TargetData	Информация о нарушителе
Device	DeviceData	Информация о Системе
Certificate	CertificateData	Информация о сертификации
Placement	PlacementData	Информация о расположении
Registarators	DeviceData (repeated)	Информация о регистраторах в составе Системы (2 шт.): - ведомый; - ведущий.
Comment	string	Комментарий. Наименование и версия ПО, например:

Параметр/Блок	Тип	Описание
	(UTF-8)	Duet 1.3.10
Image **ImageData**		**Информация о фотоизображении**
Data	bytes	Фотоизображение
Format	string (UTF-8)	Буквенное представление формата изображения. Например: ".jpeg"
Width	int32	Ширина изображения в пикселях
Height	int32	Высота изображения в пикселях
Target **TargetData**		**Информация о нарушителе**
Liplate	string (UTF-8)	Государственный регистрационный знак.
Speed	double	Рассчитанная скорость, км/ч.
Distance	double	Рассчитанная дистанция, м.
Span	bcl.TimeSpan	Рассчитанный временной интервал, до мс.
StartBound	bcl.DateTime	Время первой фиксации автомобиля, до мс.
StopBound	bcl.DateTime	Время второй фиксации автомобиля, до мс.
MasterDistance	double	Дистанция, измеренная в момент первой фиксации (ведущий регистратор), м.
SlaveDistance	double	Дистанция, измеренная в момент второй фиксации (ведомый регистратор), м.
Direction	bool	Направление движения (по ведущему регистратору): - false — встречное; - true — попутное.
Device **DeviceData**		**Информация о Системе**
Name	string (UTF-8)	Наименование системы: "ДУЭТ"
Serial	string (UTF-8)	Серийный номер. Например: "1810027/1601042"
Certificate **CertificateData**		**Информация о сертификации**
ValidThru	bcl.DateTime	Срок действия сертификата
Number	string (UTF-8)	Номер сертификата. Например: "123АБ"
Agency	string	Наименование организации, выдавшей сертификат.

Параметр/Блок	Тип	Описание
	(UTF-8)	Например: "РОСТЕСТ"
Date	bcl.DateTime	Дата выдачи
Placement **PlacementData**		**Информация о расположении**
Distance	double	Расстояние между рубежами контроля, м.
PlaceName	string (UTF-8)	Буквенное обозначение места установки системы. Например: "г. Санкт Петербург, ул. Победы"
IsCity	bool	Установка в населённом пункте или вне его: - true — да; - false — нет.
Street	string (UTF-8)	Улица.
Building	string (UTF-8)	Дом.
Direction	string (UTF-8)	Направление движения в месте установки системы по ведущей камере: - "встречное"; - "попутное"; - "любое".
SpeedLimit	int32	Скоростное ограничение (легковые ТС), км/ч.
BugSpeedLimit	int32	Скоростное ограничение для автомобилей массой более 3,5 т.
ResolveLimit	int32	Порог фиксации нарушения скоростного режима, км/ч.
Registarators[0] **DeviceData**		**Информация об устройстве** **Данные ведомого регистратора**
Name	string (UTF-8)	Название регистратора. Например: "СКАТ"
Serial	string (UTF-8)	Серийный номер. Например: "1810027"
Registarators[1] **DeviceData**		**Информация об устройстве** **Данные ведущего регистратора**
Name	string (UTF-8)	Название регистратора. Например: "КРЕЧЕТ"
Serial	string (UTF-8)	Серийный номер. Например: "1601042"

6.5. Формат xml протокола default

Текстовый файл материала (например, duet_2020_06_06_00_17_39_info.xml) представляет собой формат XML.

Ниже в Листинге 6.2 приведен пример файла. В таблице 6.4 приведены сведения обо всех полях формата.

Листинг 6.2 — Пример текстового материала формата XML протокола default


```xml
<?xml version="1.0"?>
<root version="1" software_version="1.3.9">
  <image>
    <width>1280</width>
    <height>560</height>
    <format>.jpeg</format>
  </image>
  <target>
    <liplate>o002мт|178</liplate>
    <speed>58.89</speed>
    <distance>74.30</distance>
    <span>00:00:04.542</span>
    <start_bound>2020.08.06 07:39:00.139</start_bound>
    <stop_bound>2020.08.06 07:39:04.681</stop_bound>
    <master_distance>27.80</master_distance>
    <slave_distance>34.50</slave_distance>
    <direction>встречное</direction>
  </target>
  <device>
    <name>ДУЭТ</name>
    <serial>1810027/1601042</serial>
  </device>
  <certificate>
    <agency>РОСТЕСТ</agency>
    <number>111333F</number>
    <date>02.06.2018</date>
    <valid_thru>06.06.2021</valid_thru>
  </certificate>
  <placement>
    <distance>81.00</distance>
    <place_name>Санкт-Петербург, ул.Победы</place_name>
    <is_city>true</is_city>
    <street></street>
    <building></building>
    <direction>любое</direction>
    <speed_limit>30</speed_limit>
    <resolve_limit>55</resolve_limit>
    <bug_speed_limit>50</bug_speed_limit>
  </placement>
  <registrators>
    <master>
      <name>СКАТ</name>
      <serial>1810027</serial>
    </master>
    <slave>
      <name>СКАТ</name>
```

```
        <serial>1601042</serial>
      </slave>
    </registrators>
    <comment />
</root>
```

Таблица 6.4 — Поля формата XML протокола default

Параметр/Блок	Тип	Описание
root		**Общая информационная структура** **- version — версия формата** **- software_version — версия ПО**
image		**Информация о фотоизображении**
format	string (UTF-8)	Буквенное представление формата изображения. Например: ".jpeg"
width	int	Ширина изображения в пикселях
height	int	Высота изображения в пикселях
target		**Информация о нарушителе**
liplate	string (UTF-8)	Государственный регистрационный знак.
speed	double	Рассчитанная скорость, км/ч.
distance	double	Рассчитанная дистанция, м.
span	string	Рассчитанный временной интервал, до мс. Формат: hh:mm:ss.nnn
start_bound	string	Время первой фиксации автомобиля, до мс. Формат: YYYY.MM.DD hh:mm:ss.nnn
stop_bound	string	Время второй фиксации автомобиля, до мс. Формат: YYYY.MM.DD hh:mm:ss.nnn
master_distance	double	Дистанция, измеренная в момент первой фиксации (ведущий регистратор), м.
slave_distance	double	Дистанция, измеренная в момент второй фиксации (ведомый регистратор), м.
direction	string	Направление движения ТС (по ведущему регистратору): - встречное; - попутное.
device		**Информация о Системе**
name	string (UTF-8)	Название системы: "ДУЭТ"

Параметр/Блок	Тип	Описание
serial	string (UTF-8)	Серийный номер. Например: "1810027/1601042"
certificate		**Информация о сертификации**
agency	string (UTF-8)	Наименование организации, выдавшей сертификат. Например: "РОСТЕСТ"
number	string (UTF-8)	Номер сертификата. Например: "123АБ"
date	string	Дата выдачи. Формат: DD.MM.YYYY
valid_thru	string	Срок действия сертификата. Формат: DD.MM.YYYY
placement		**Информация о расположении**
distance	double	Расстояние между рубежами контроля, м.
place_name	string (UTF-8)	Буквенное обозначение места установки системы. Например: "г. Санкт Петербург, ул. Победы"
is_city	bool	Установка в населённом пункте или вне его: - true — да; - false — нет.
street	string (UTF-8)	Улица.
building	string (UTF-8)	Номер дома.
direction	string (UTF-8)	Направление движения в месте установки системы по ведущей камере: - "встречное"; - "попутное"; - "любое".
speed_limit	int	Скоростное ограничение (легковые ТС), км/ч.
resolve_limit	int	Порог фиксации нарушения скоростного режима, км/ч.
bug_speed_limit	int	Скоростное ограничение для автомобилей массой более 3,5 т.
registrators		**Информация о регистраторах в составе Системы(2 шт.)**
master		**Информация о ведущем регистраторе**
name	string (UTF-8)	Наименование регистратора. Например: "СКАТ"
serial	string (UTF-8)	Серийный номер. Например: "1810027"

Параметр/Блок	Тип	Описание
slave		**Информация о ведомом регистраторе**
name	string (UTF-8)	Наименование регистратора. Например: "КРЕЧЕТ"
serial	string (UTF-8)	Серийный номер. Например: "1601042"
comment	**string** **(UTF-8)**	**Комментарий.**

6.6. Формат xml протокола samara

Текстовый файл материала (например, 1810027_1601042_2020_08_13_18_19_01_71_00_012.xml) представляет собой формат XML. Ниже в Листинге 6.3 приведен пример файла.

Листинг 6.3 — Пример текстового материала формата XML протокола samara


```xml
<?xml version="1.0" encoding="UTF-8"?>
<event>
  <timestamp>2020-08-13T19:26:42+03</timestamp>
  <timestamp0>2020-08-13T19:26:38+03</timestamp0>
  <timestampdb>2020-08-13T16:26:50+00</timestampdb>
  <type>C1</type>
  <path>C1/2020/08/13</path>
  <num>B104PK198</num>
  <id>1810027_1601042_2020_08_13_19_26_42_21_00_012</id>
  <tSoftVersion>1.3.5</tSoftVersion>
  <nImage1Present>1</nImage0Present>
  <nImage2Present>1</nImage0Present>
  <nImage3Present>1</nImage0Present>
  <nXmlPresent>1</nXmlPresent>
  <nGRZPresent>1</nGRZPresent>
  <nDatetime>1597346802</nDatetime>
  <nDisplaySpeed>68</nDisplaySpeed>
  <nOverSpeed>1</nOverSpeed>
  <nDirection>1</nDirection>
  <nSpeedThreshold>40</nSpeedThreshold>
  <nPDD>10.2</nPDD>
  <tRadarId>1810027/1601042</tRadarId>
  <tChannel0>1810027</tChannel0>
  <tChannel1>1601042</tChannel1>
  <tChannel>1</tChannel>
  <tSensorName0>Санкт-Петербург, ул.Победы до пл.Мужества</tSensorName0>
  <tSensorName>Санкт-Петербург, ул.Победы до пл.Мужества</tSensorName>
  <tLocationCode>28000000020</tLocationCode>
  <tRadarName>ДУЭТ</tRadarName>
  <tCertificateExpDate>2018-06-02</tCertificateExpDate>
  <tCertificateExpLimit>2021-06-06</tCertificateExpLimit>
  <tLatitude0>216051</tLatitude0>
  <tLongitude0>109490</tLongitude0>
```

```
    <tLatitude>216053</tLatitude>
    <tLongitude>109480</tLongitude>
    <nTZOffset>+03:00</nTZOffset>
    <nRectNumX1>957</nRectNumX1>
    <nRectNumY1>276</nRectNumY1>
    <nRectNumX2>1103</nRectNumX2>
    <nRectNumY2>276</nRectNumY2>
    <nRectNumX3>957</nRectNumX3>
    <nRectNumY3>320</nRectNumY3>
</event>
```

Формат полей файла XML протокола samara совпадает с полями БД xml (см. п.7.1.2).

Формат полей timestamp, timestamp0, timestampdb — ISO 8601:

YYYY-MM-DDThh:mm:ss±hh

В файле XML могут отсутствовать некоторые специальные поля, если имеют значение 0:

- event_id;
- numLat;
- nTZOffsetSystem;
- nImageXPresent.

6.7. Цифровые подписи

6.7.1. Генерация ключей

Ключи требуются для создания цифровых подписей (ЭЦП) файлов материалов в ПО «ДУЭТ» и их последующей проверки в других средствах. Ключи создаются однократно Заказчиком для своих комплексов в единой системе:

- открытый (публичный) ключ;
- закрытый (секретный, приватный) ключ.

Закрытый ключ используется для создания подписей.

Открытый ключ — для проверки подписей.

Поддерживаемые виды ЭЦП в ПО:

- RSA (хэш SHA1, размер ключа 1024);
- ГОСТ Р 34.11-2012 (Стрибог, длина закрытого ключа 256 бит, эллиптическая кривая: id-GostR3410-2001-CryptoPro-XchB-ParamSet, OID "1.2.643.2.2.36.1", ASN представление {0x06, 0x07, 0x2A, 0x85, 0x03, 0x02, 0x02, 0x24, 0x01}, RFC 4357).

Для каждого вида подписей нужны разные форматы ключей:

- RSA — ключи PEM формата;
- ГОСТ — ключи S-expression формата (SXP).

Генерация ключей RSA-SHA1 *.pem через утилиту openssl (Windows):

```
openssl genrsa -out privolvia.pem 1024
openssl rsa -in privolvia.pem -outform PEM -pubout -out pubolvia.pem
```

Генерация ключей ГОСТ *.sxp через поставляемую программу CryptConsole:

CryptConsole.exe -m gen -a gost -q pubolvia.sxp -d privolvia.sxp

Key generation successful

Created public key: pubolvia.sxp

Created private key: privolvia.sxp

Полученный закрытый ключ privolvia.pem(sxp) следует использовать в ПО «ДУЭТ» (корневая папка установки ПО под Windows), открытый ключ pubolvia.pem(sxp) — в стороннем ПО.

6.7.2. Проверка цифровых подписей

Материалы с ЭЦП RSA-SHA1 могут быть проверены при наличии файлов открытых ключей (например, pubolvia.pem файл) и средства для проверки подписи, например openssl (Windows):

openssl dgst -sha1 -verify pubolvia.pem -signature duet_protobuff_2020_06_06_00_17_39.sign duet_protobuff_2020_06_06_00_17_39.bin

Результат:

- «Verified OK» — при корректной подписи;
- «Verification Failure» — при некорректной.

Материалы с ЭЦП ГОСТ могут быть проверены при наличии файлов открытых ключей (например, pubolvia.sxp файл) и средства для проверки CryptConsole или другого, поддерживающего данный формат подписей S-expression и указанный вид ЭЦП ГОСТ:

CryptConsole.exe -m verify -i duet_protobuff_2020_06_06_00_17_39.jpg -s duet_protobuff_2020_06_06_00_17_39.sign -q pubolvia.sxp

6.7.3. Примеры ключей и подписей

Примерный вид открытого ключа формата PEM для ЭЦП RSA-SHA1 (pubolvia.pem), ОБРАЗЕЦ в Листинге 6.2.

Листинг 6.2 — Образец открытого ключа формата PEM


```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDrmw6f4E5QAD4/qb3+YRp/inhT
LZ0yTe64U2yrrLxteGUUOAx1RlmxNpei/pHIyOQzQ6+XhODYj3aW3YNR8V/YXRLt
ipqaAuX9pEN0erjr7t5GwKzfF0Dj+axb2gy+T3A/F6P5tfeUlTZC+BKQW1WwJMUn
UIIn2HZkRg65fQyCcQIDAQAB
-----END PUBLIC KEY-----
```

Примерный вид закрытого ключа формата PEM для ЭЦП RSA-SHA1 (privolvia.pem), ОБРАЗЕЦ в Листинге 6.3.

Листинг 6.3. Образец закрытого ключа формата PEM


```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDrmw6f4E5QAD4/qb3+YRp/inhTLZ0yTe64U2yrrLxteGUUOAx1
RlmxNpei/pHIyOQzQ6+XhODYj3aW3YNR8V/YXRLtipqaAuX9pEN0erjr7t5GwKzf
F0Dj+axb2gy+T3A/F6P5tfeUlTZC+BKQW1WwJMUnUIIn2HZkRg65fQyCcQIDAQAB
AoGBALgTLirk8ep5NilBMirDrRNGmVnLQozh7Ifgzf+W9qTEspxIahtaOotFLtXu
9PGdWwVedZr4+z7XX3zhVk84OXtvASnMCQsympCHh8zFiI6XRqD+JlRaN9K7lz7O
QGZ8HKYGz9AuiYIgFv+Nju4+5rdr80cvNEpBoZFJTXFwsObRAkEA9rfOj6diclNO
TglnxDJ5wkTt9XMUAcans1cDbaCJXSk8wlsvL8ugpmcC7yCMGpbriQJa+fBbgI95
wTpgQK9C3QJBAPR4Oa1+0pzHSUHgVVohAuWFptNMda5grVxtXK0AvAQTXxckzul/
VvQ9FyogQoe0yr6nx3N3KFA5LTsJAHSQcqUCQQCHrqpQ4pMHoLztucpxYfkIV/Qu
Cm20VFYiEPZ3sdQZgfZCRPd8Z5YgLYK1k8a5UMqT/NGfpwoLd9xT4hDRQhGZAkAY
LP9/MciAKpELC7VFGXig3Au4X5pPqU3bkxEUJrtxJ71Cm6g+nRIrSxNMXi5WH0Hh
gtQNjh4VeOIJMYWzAUnlAkEAm5x05ysguK4UvZTRnELGE74o64k8+IDE41nPfnnA
J/voKJ0aDHgGsZT2/KkDL/d/kLn69sIxgv5i0TdH6tsr2g==
-----END RSA PRIVATE KEY-----
```


Примерный вид подписи материала вида ЭЦП RSA-SHA1 (rif_protobuff_2020_06_06_00_17_39.sign), hex вид, 128 байт, ОБРАЗЕЦ (рис. 6.6):

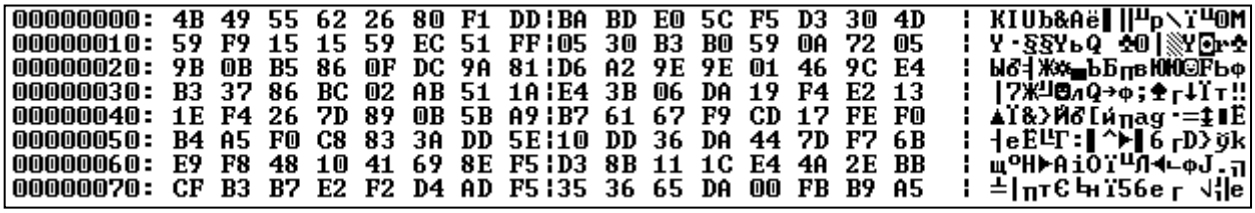


Рисунок 6.6 — Примерный вид получаемой подписи в HEX формате

Примерный вид открытого ключа формата SXP для ЭЦП ГОСТ (pubolvia.sxp), ОБРАЗЕЦ в Листинге 6.6.

Листинг 6.6 — Образец открытого ключа формата SXP


```
(public-key
 (ecc
  (curve GOST2001-CryptoPro-C)
  (q
#0476CA43DFBF2EAEE6C18FD497B9C60E4B9E595BFA0CA3804556F65B7C3D1154AA2735C23B
F3FCD4D65C9F8A5FF7C6FBBBA7410C5901181F8D491C26DAA8583694#)
  )
 )
```

Примерный вид закрытого ключа формата SXP для ЭЦП ГОСТ (privolvia.sxp), ОБРАЗЕЦ в Листинге 6.7.

Листинг 6.7 — Образец закрытого ключа формата SXP

```
(private-key
 (ecc
  (curve GOST2001-CryptoPro-C)
  (q
#0476CA43DFBF2EAEE6C18FD497B9C60E4B9E595BFA0CA3804556F65B7C3D1154AA2735C23B
F3FCD4D65C9F8A5FF7C6FBBBA7410C5901181F8D491C26DAA8583694#)
  (d #4B6A5AF6D41C809951342BA88AC34A25CBF1EF8FE67F5F8499BF720B008C6F4A#)
  )
 )
```

Примерный вид подписи материала вида ЭЦП ГОСТ (duet_protobuff_2020_06_06_00_17_39.sign), ОБРАЗЕЦ в Листинге 6.8.

Листинг 6.8 — Вид получаемой подписи по ГОСТ

```
(sig-val
 (gost
  (r #42D41C7C5FEB432EA6538AB14861F5ABD7533128F4174261527DB37DBA7B6E84#)
  (s #78743EC36847C7EA8DEE210827B505B0EF6D3D84224FB93536F85337CAB4DA68#)
  )
 )
```

6.8. Автоматическая очистка архива материалов

В процессе работы ПО «ДУЭТ» производятся периодические очистки папок архива материалов для предотвращения переполнения файлового хранилища.

Алгоритм очистки архива:

1) При каждой попытке создания файлов нового материала на диске проверяется, превышен ли объем архива материалов;
2) Контролируется суммарный объем файлов в каталоге:

   D:/arch/duet/ - параметр **result_directory** в config.json

   Также в суммарный объем входит содержимое каталога временных файлов, если он задан отдельный, например:

   D:/arch/duet_tmp/ - параметр **result_directory_tmp** в config.json

3) Если суммарный объем превышает максимальный (например, **50 ГБ**), в ПО запускается процедура очистки, планируется удаление **10 Гб** файлов. Процедура работает параллельно основной работе ПО некоторое время. Если нет — процедура очистки завершается (новый запуск с Шага 1);
4) Выполняется полное удаление очередного каталога старых файлов материалов, подготовленных в Шаге 3;
5) Возвращение к Шагу 3 для анализа нового текущего размера архива материалов.

Настраиваемые параметры через конфигурацию config.json указаны в табл.6.5.

Таблица 6.5 — Параметры очистки архива в config.json

Параметр	Значение	Описание
storage_limit_fixations	53687091200 (50 Гб)	Максимальный объем всего архива материалов. При превышении объема запускается очистка / удаление каталогов. Размер в байтах.

Параметры недоступные для изменения указаны в табл.6.6.

Таблица 6.6 — Параметры очистки архива недоступные для изменения

Параметр	Значение	Описание
Объем удаляемых данных	10 Гб	Данный объем файлов будет удалён при достижении максимума

6.9. Автоматическая очистка БД материалов

В процессе работы ПО «ДУЭТ» производится периодическая очистка БД duet_main с информацией о материалах для ускорения доступа и повышения производительности. Сервисная БД duet_service не очищается.

Алгоритм очистки:

1) При каждом старте и авторестарте ПО запускается процедура проверки БД;
2) Проверяется, сколько элементов/строк содержится в таблице duet_main.materials (самая быстро растущая таблица);
3) Если накапливается более 800 тыс. записей в таблице duet_main.materials, то выполняется очистка:
   - Удаляются последние 300 тыс. записей из duet_main.materials и связанные с ними строки из таблиц duet_main.events и duet_main.targets .
4) Далее ПО продолжает работу уже с новой уменьшенной БД;

5) В процессе работы ПО и добавлении каждых 10 тыс. новых записей в таблицу duet_main.materials производится повторная проверка достижения максимального объема (переход на Шага №3).

Параметры очистка БД закреплены в ПО и недоступны для изменения, см. табл.6.7.

Для протокола **samara** таблица xml_duet очищается симметрично с таблицей duet_main.events по значениям полям xml_duet.event_id.

Таблица 6.7 — Параметры очистки БД

Параметр	Значение	Описание
DB_MIN_ROTATED_ROWS	500000	Число актуальных записей, которые остаются в текущей рабочей БД.
DB_MAX_ROTATED_ROWS	800000	Максимальное число записей в БД main.materials, после которого запускается очистка.
DB_ROTATE_CHECK_LIMIT	10000	Число записей в БД main.materials, после, которого запускается процедура проверки объема.

7. Служебные данные

7.1. Структура БД

Ниже в таблицах 7.1-7.7 приведена структура БД ПО «ДУЭТ» протокола по умолчанию (default). Структура дополнительной БД результатов для протокола samara приведена в таблице 7.8.

7.1.1. Описание структуры БД результатов

БД **duet_main** используется в качестве основной БД для хранения информации о фиксациях нарушений и материалах.

Таблица 7.1 — Таблица targets

	Столбец	Тип	Значение
1	id	INT(11) AUTO_INCREMENT PRIMARY KEY	Уникальный id.
2	timestamp	TIMESTAMP(6)	Время записи в БД (системное время).
3	detection_time	DATETIME(6)	Время фиксации.
4	direction	TINYINT(2)	Направление движения: 1 — попутные; 0 — встречные.
5	liplate_type	CHAR(1)	-

	Столбец	Тип	Значение
6	liplate_data	VARCHAR(20)	Распознанный ГРЗ (UTF-8).

Таблица 7.2 — Таблица events

	Столбец	Тип	Значение
1	id	INT(11) AUTO_INCREMENT PRIMARY KEY	Уникальный id.
2	timestamp	TIMESTAMP(6)	Время записи в БД (системное время).
3	target_id	INT(11)	id ТС (из таблицы targets).
4	time	DATETIME(6)	Время фиксации события.
5	type	CHAR(4)	Значения событий: 1.1 — превышение порога фиксации мгновенной скорости (от регистраторов); 1.2 — превышение порога фиксации средней скорости.
6	target_mdspeed	FLOAT(12)	Скорость ТС.
7	target_mdistance	FLOAT(12)	Дистанция до ТС от ведущего регистратора.
8	target_sdistance	FLOAT(12)	Дистанция до ТС от ведомого регистратора.
9	target_mtime	DATETIME(6)	Время фиксации ведущего регистратора.
10	target_stime	DATETIME(6)	Время фиксации ведомого регистратора.
11	target_type	CHAR(1)	Всегда 0.
12	liplate_data	VARCHAR(20)	Распознанный ГРЗ (UTF-8).
13	target_distance	FLOAT(12)	Дистанция, пройденная ТС.

Таблица 7.3 — Таблица materials

	Столбец	Тип	Значение
1	id	INT(11) AUTO_INCREMENT PRIMARY KEY	Уникальный id.
2	timestamp	TIMESTAMP(6)	Время записи в БД (системное время).
3	target_id	INT(11)	id ТС (из таблицы targets).
4	event_id	INT(11)	id события (из таблицы events).

	Столбец	Тип	Значение
5	file_type	INT(11)	Тип файла: 0 — основной protobuf материал; 1 — файл кадр с увеличенным ТС ведущего регистратора; 2 — файл кадр с увеличенным ТС ведомого регистратора; 3 — цифровая подпись основного материала protobuf; 4 — основной jpg материал; 5 — цифровая подпись основного материала jpg; 6 — основной xml материал; 7 — цифровая подпись xml материала.
6	file_path	TINYTEXT	Путь к файлу
7	file_hash	CHAR(32)	-

7.1.2. Описание структуры БД результатов протокола samara

БД **region_ru_samara** используется в качестве дополнительной БД для хранения информации о фиксациях нарушений и материалах для формата отдачи XML.

Таблица 7.4 — Таблица xml_duet

	Столбец	Тип	Значение
1	event_id	INT(11) INDEX KEY	id события из таблицы events.
2	id	VARCHAR(255) INDEX KEY	Уникальный идентификатор события.
3	timestamp	DATETIME(6)	Дата и время зафиксированного события. Время проезда через второй рубеж для события С1.2 превышения средней скорости. Дата и время в локальной временной зоне регистратора.
4	timestamp0	DATETIME(6)	Дата и время зафиксированного проезда ТС через первый рубеж зоны в случае фиксации проезда (событие С0) через оба рубежа зоны или фиксации события превышения средней

	Столбец	Тип	Значение
			скорости (событие С1) на участке зоны. Всегда дата и время проезда через ведущий регистратор. Дата и время в локальной временной зоне регистратора.
5	timestampdb	DATETIME(6)	Дата и время внесения информации о зафиксированном событии в БД ПО. Дата и время в локальной временной зоне системы.
6	type	CHAR(4)	Тип зафиксированного события С0-С16 в соответствии с ГОСТ Р 57144-2016.
7	path	TINYTEXT	Путь к материалам событий на FTP-сервере.
8	num	VARCHAR(20)	ГРЗ зафиксированного ТС.
9	numLat	VARCHAR(20)	ГРЗ зафиксированного ТС, латинскими буквами.
10	tSoftVersion	VARCHAR(32)	Версия ПО. Например: 1.3.1.
11	nImage0Present	TINYINT(2)	Наличие снимка ТС во время фиксации нарушения. Проезд второго рубежа для превышения средней скорости (событие С1). (1 — присутствует, 0 — отсутствует). ВСЕГДА = 0.
12	nImage1Present	TINYINT(2)	Наличие фотоматериала с первого регистратора, снимок 1 (1 — присутствует, 0 — отсутствует или отсутствует поле).
13	nImage2Present	TINYINT(2)	Наличие фотоматериала со второго регистратора, снимок 2 (1 — присутствует, 0 — отсутствует или отсутствует поле).
14	nImage3Present	TINYINT(2)	Наличие парного снимка ТС во время фиксации нарушения с фото с двух рубежей для события превышения средней скорости (событие С1.2). (1 — присутствует, 0 — отсутствует или отсутствует поле).
15	nXmlPresent	TINYINT(2)	Наличие текстовой информации в виде XML-файла (1 — присутствует, 0 — отсутствует).
16	nGRZPresent	TINYINT(2)	Наличие снимка пластины регистрационного знака ТС в виде jpg-файла (1 — присутствует, 0 — отсутствует).

	Столбец	Тип	Значение
17	nDatetime	INT(11) UNSIGNED	Местное время (количество секунд от 01.01.1970). Время фиксации события в локальном часовом поясе.
18	nDisplaySpeed	SMALLINT(5)	Измеренная скорость (км/ч) (0 — нет измерений).
19	nOverSpeed	TINYINT(2)	Признак нарушения «превышения скорости» при значении 1, 0 или иное — нет нарушения.
20	nDirection	TINYINT(2)	Направление движения ТС. Если для проезда ТС второй рубеж ведущий регистратор — 0, если второй рубеж ведомый — 1.
21	nSpeedThreshold	SMALLINT(5)	Ограничение скоростного режима (км/ч).
22	nPDD	VARCHAR(8)	Пункт ПДД. Указание обязательное для type, равное C1 (превышение допустимой скорости). Для события с type, равное C0, отсутствует. Для остальных событий указание возможно, но не обязательно. Предоставляется в форматах X9.X9 или X3.X9.X9, где на месте X9 может находиться 1 или 2 цифровых символа. Например: "10.3".
23	tRadarId	VARCHAR(128)	Уникальный идентификатор комплекса (совокупность символов, позволяющая однозначно идентифицировать АПК). В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) уникальный идентификатор зоны. Серийный номер системы. Например: "1810027/1601042".
24	tChannel0	VARCHAR(128)	В случае фиксации двухкамерного проезда по рубежам зоны (событие CO) или превышения средней скорости на участке зоны (событие C1) уникальный идентификатор первого рубежа зоны. Идентификатор ведущего регистратора. Например: "1810027".

	Столбец	Тип	Значение
25	tChannel1	VARCHAR(128)	В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) уникальный идентификатор второго рубежа зоны. Идентификатор ведомого регистратора. Например: "1601042".
26	tChannel	TINYINT(2)	Номер видеоканала (произвольное числовое значение, задается в настройках config.json для идентификации направления фиксации нарушений в пределах одного комплекса.
27	tSensorName0	VARCHAR(128)	Текстовое описание места установки системы.
28	tSensorName	VARCHAR(128)	Текстовое описание места установки системы. Аналогично tSensorName0.
29	tLocationCode	CHAR(12)	Код по ОКАТО места установки.
30	tRadarName	VARCHAR(16)	Идентификационное наименование комплекса. Например: ДУЭТ.
31	tCertificateExpDate	DATE	Дата получения сертификата поверки. Формат ISO 8601: YYYY-MM-DD
32	tCertificateExpLimit	DATE	Срок действия сертификата поверки. Формат ISO 8601: YYYY-MM-DD
33	tLatitude0	MEDIUMINT(9)	В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) географическая широта места установки первого рубежа зоны (целое число — количество угловых секунд). Координаты ведущего регистратора.
34	tLongitude0	MEDIUMINT(9)	В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) географическая долгота места установки первого рубежа зоны (целое число —

	Столбец	Тип	Значение
			количество угловых секунд). Координаты ведущего регистратора.
35	tLatitude	MEDIUMINT(9)	Географические координаты места установки комплекса: географическая широта (целое число — количество угловых секунд, Координаты ведомого регистратора.
36	tLongitude	MEDIUMINT(9)	Географические координаты места установки комплекса: географическая долгота (целое число — количество угловых секунд. Координаты ведомого регистратора.
37	nTZOffset	CHAR(6)	Часовой пояс места установки (по настройке сдвига времени регистраторов в веб-интерфейсе и config.json complex_utc_offset). Формат: ±hh:mm
38	nTZOffsetSystem	CHAR(6)	Часовой пояс системы с ПО. Формат: ±hh:mm
39	nRectNumX1	SMALLINT(5) UNSIGNED	Координата X левого верхнего угла пластины номера на изображении.
40	nRectNumY1	SMALLINT(5) UNSIGNED	Координата Y левого верхнего угла пластины номера на изображении.
41	nRectNumX2	SMALLINT(5) UNSIGNED	Координата X правого верхнего угла пластины номера на изображении.
42	nRectNumY2	SMALLINT(5) UNSIGNED	Координата Y правого верхнего угла пластины номера на изображении.
43	nRectNumX3	SMALLINT(5) UNSIGNED	Координаты X левого нижнего угла пластины номера на изображении.
44	nRectNumY3	SMALLINT(5) UNSIGNED	Координаты Y левого нижнего угла пластины номера на изображении.

7.1.3. Описание структуры служебной БД

БД **duet_service** используется в качестве служебной БД для хранения данных о ходе работы ПО, действий пользователей, резерва данных пользователей и др.

Таблица 7.5 — Таблица info

	Столбец	Тип	Значение
1	id	INT(11) AUTO_INCREMENT PRIMARY KEY	Уникальный id
2	timestamp	TIMESTAMP(6)	Время записи в БД (системное время)
3	parameter	VARCHAR(255)	Название параметра
4	value	VARCHAR(255)	Значение параметра

Таблица 7.6 — Таблица users

	Столбец	Тип	Значение
1	id	INT(11) AUTO_INCREMENT PRIMARY KEY	Уникальный id.
2	timestamp	TIMESTAMP(6)	Время записи в БД (системное время).
3	name	VARCHAR(255)	Имя пользователя, например: - root; - Администратор; - Поверка.
4	fullname	TEXT(65535)	-
5	position	TEXT(65535)	-
6	password_hash	CHAR(32)	md5 пароля.
7	rights	VARCHAR(40)	Возможные значения прав: - root - admin - setup - verify - info - journal

Таблица 7.7 — Таблица users_actions

	Столбец	Тип	Значение
1	id	INT(11) AUTO_INCREMENT PRIMARY KEY	Уникальный id.
2	timestamp	TIMESTAMP(6)	Время записи в БД (системное время).
3	username	VARCHAR(255)	Имя пользователя.
4	action	INT(11)	Индекс действия: 10 — изменение режима работы; 101 — изменение параметра; 200 — вход в систему; 201 — ошибка входа в систему; 202 — выход из системы; 400 — добавление пользователя; 401 — изменение пользователя; 402 — удаление пользователя.
5	a	TEXT(65535)	Название параметра.
6	b	TEXT(65535)	Значение параметра.
7	c	TEXT(65535)	-
8	d	TEXT(65535)	-
9	e	TEXT(65535)	-
10	f	TEXT(65535)	-

Таблица 7.8 — Таблица vars

	Столбец	Тип	Значение
1	id	INT(11) AUTO_INCREMENT PRIMARY KEY	Уникальный id.
2	timestamp	TIMESTAMP(6)	Время записи в БД (системное время).
3	parameter	VARCHAR(255)	Название параметра.
4	value	VARCHAR(255)	Значение параметра.
5	hash	CHAR(32)	Хэш состояния хранения параметра.

7.2. Служебные протоколы и журналы работы

Обслуживание и решение проблем, возникающих в ходе работы ПО, может производиться с использованием различных формирующихся журналов.

Основные журналы и виды протоколирования событий в ПО:

- Консольные сообщения ядра ПО, от компонентов (console);
- Система протоколирования ПО;
- Журналы ОС Windows Embedded (Event Journal);
- Журналы доступа по FTP;
- Системные снимки, созданные в случае критических завершений работы ПО (crashdumps).

Для получения протоколов необходимо осуществлять удаленное подключение к рабочему столу Windows и загружать протоколы для исследования. Полный анализ проблемной ситуации осуществляется при получении указанных выше протоколов на момент события в виде файлов.

1. Консольные сообщения ядра ПО от компонентов формируются постоянно в ходе работы от загрузки и вплоть до завершения работы по какой-либо причине. На данный момент сообщения формируются в окне консоли **duet_core** (рис. 7.1).

Данные протоколы подходят для наблюдения в режиме реального времени за поведением ПО, но для удобства дальнейшего анализа требуется их скопировать. Также объем отображаемой информации ограничен, поэтому важных сообщений может уже не оказаться. Для этого есть другие уровни протоколирования, указанные далее.

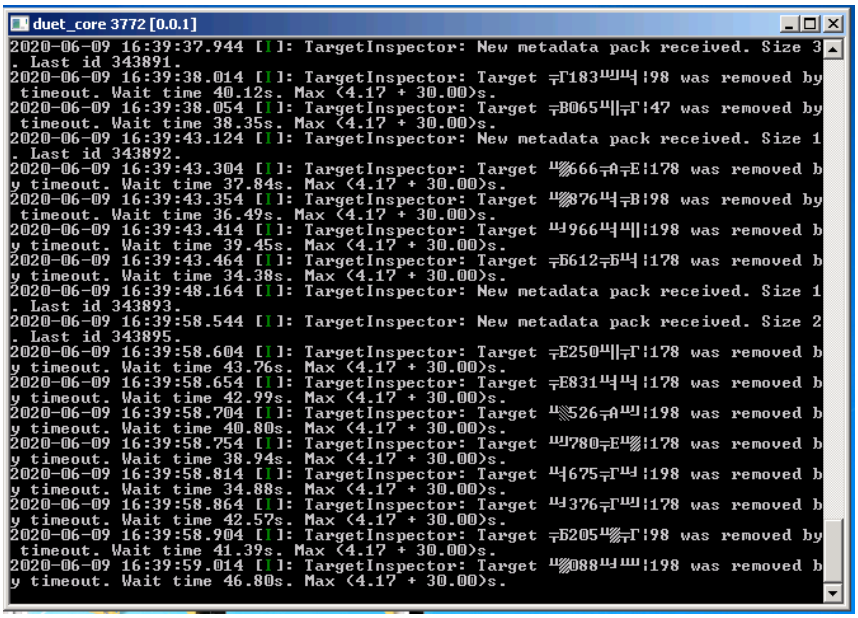


Рисунок 7.1 — Сообщения в ходе функционирования ПО в режиме реального времени

В консольных сообщениях поддерживаются уровни сообщений:

- [T] Trace — отладочные сообщения о ходе выполнения низкого уровня (нет цвета);
- [D] Debug — отладочные сообщения (нет цвета);
- [I] Info — статусные и информационные (зелёные);
- [W] Warning — предупреждения (желтый);

- [E] Error — ошибки (красный);
- [C] Critical — критические (бордово-красный).

Как правило, требуют внимания Error и Critical сообщения, но при первичной настройке требуется обращать внимание на Warning и Info, так как могут быть сообщения об использовании настроек по умолчанию и не критичных, но не выполненных пользователем настройках. Указанным цветом подкрашиваются символы уровня сообщения.

2. Система протоколирования ПО обеспечивает сохранение всех сообщений, в том числе отображаемых в консольных окнах, в текстовые файлы протоколов.

Папка расположения всех данных файлов протоколов: **C:\duet\logs**

Файлы протоколов в данном каталоге разбиты по компонентам. Состав протоколов:

- core.log — основной протокол работы приложения duet_core;
- web_server.log — протокол работы веб-сервера в составе приложения duet_core;
- supervisor.log — протокол работы программы контроля и мониторинга duet_core.

С каждым типом протокола происходит ротация файлов (автоматическое удаление) для ограничения объема и недопущения снижения производительности, в целом выполняются условия:

- объем каждого файла не более 2 Мб;
- кол-во не более 10 файлов.

Просмотр текстовых файлов протоколов возможен в любом текстовом редакторе, поддерживающим работу с UTF-8 кодировкой текста. Уровни сообщений аналогичны приведенным выше консольным сообщениям.

3. Журналы ОС Windows Embedded (Event Journal) сохраняют информацию о моментах загрузки/перезагрузки и завершения работы операционной системы.

Получение записей из данных журналов может быть осуществлено через стандартное приложение Windows: Event Viewer (eventvwr, рис. 7.2).

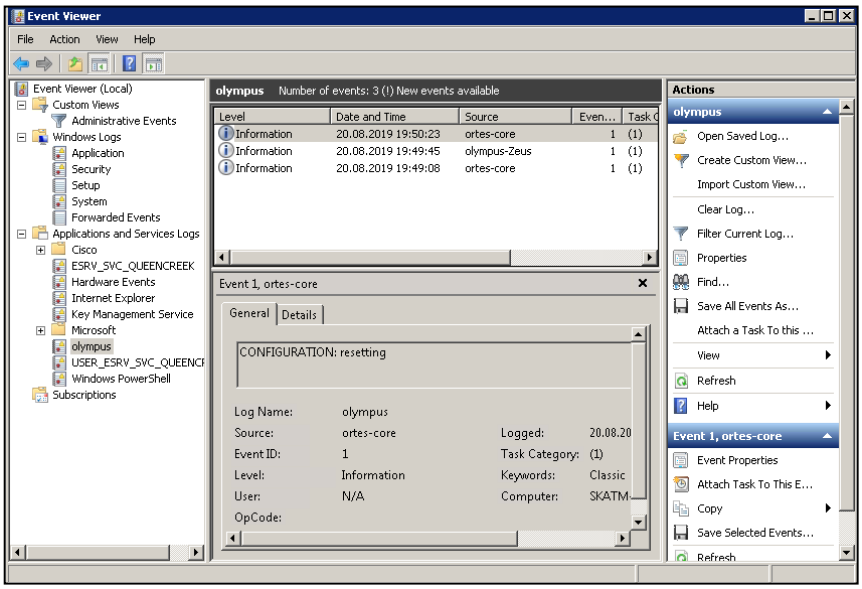


Рисунок 7.2 — Просмотр сообщений в журнале Event Journal

4. Журналы доступа по FTP используются для анализа получения доступа к загрузке материалов через FTP-клиентские приложения. Получение записей из данного журнала можно осуществить через удаленный доступ к папкам с *.log файлами FTP сервера. Журнал представляет собой операции по попыткам авторизации и загрузке файлов.

5. Системные снимки памяти процессов, созданные в случае критических завершений работы ПО (crashdumps).

Расположения файлов для анализа:

- c:\Users\USER\AppData\Local\CrashDumps\ — *.dmp системные дампы Windows (USER — имя пользователя в системе).

8. Устранение неисправностей

В данном разделе описаны типовые проблемы, возникающие при работе с ПО системы и способы их решения. Для выяснения и устранения проблем требуется доступ к веб-интерфейсу ПО, а также удалённый доступ к ОС регистратора или сервера, на котором выполняется ПО, для наблюдения за поведением ПО и получения файлов протоколов работы (п.7.2).

8.1. Проблемы и способы решения

8.1.1. Программа не запускается

При запуске ПО программа сразу закрывается или постоянно перезапускается при вызове с помощью файлов **duet_start.bat** или **duet_start_standalone.bat**.

Варианты решения:

1) Проверьте, не выполняется ли другая копия ПО в Диспетчере задач (**duet_core**). Доступен только запуск одной копии ПО одновременно. Завершите выполнение данного процесса при наличии (см. п. 4.2).

Примеры сообщений об ошибках в протоколе **core.log** и **supervisor.log**:

- `[E]: Core: Another instance of this application is running! Close it first!`
- `[C]: Core: Error while checking running instance of application (1 0)`
- `[E]: SUPERVISOR: Another instance of this application is running! Close it first!`
- `[C]: SUPERVISOR: Error while checking running instance of application (1)`

2) Проверьте файлы настроек ПО в папке .**config** (например, **C:\duet\config**\)

- наличие файлов **config.json**, **web_server.ini**, **users.json**.

Примеры сообщений об ошибках в протоколе **core.log**:

- `[C]: Core: Config loading failed!`
- `[C]: Core: Users config loading failed!`
- `[E]: Config: Can't open configuration file "config.json" for saving`
- корректность формата указанных файлов, если выполнялось редактирование вручную.

Примеры сообщений об ошибках в протоколе **core.log**:

- `[E]: Config: Can't parse json file "config.json"`
- `[E]: Config: Can't load parameter "speed_limit" from "config.json"`

3) Проверьте инициализацию и успешность подключения ПО к БД. Попытки подключения будут повторяться бесконечно, пока не будут прерваны пользователем или ОС при завершении работы программы.

- проверьте параметры доступа к БД системы, поля в настройках **config.json:**
- `"parameters_db_name"`
- `"parameters_db_pass"`
- `"parameters_db_user"`

- "result_db_name"
- "result_db_pass"
- "result_db_user"

Исправьте настройки доступа при их неверном значении, например пароль пользователя БД (см. п. 3.5.2 "Установка приложения").

Примеры сообщений об ошибках в протоколе **core.log** о подключениях БД service/results:

- [E]: DatabaseManager: connect service database failed
- [E]: DatabaseManager: connect results database failed
- [E]: DatabaseManager: connect results(web) database failed
- [C]: MySQLClient: [service] set connect timeout error, mysql_error[{}]={}
- [E]: MySQLClient: [service] set charset error, mysql_error[{}]={}
- [E]: MySQLClient: [service] connect '' 'root'@127.0.0.1:3306 error

- проверьте успешность создания и проверки структуры БД системы. При наличии подобных ошибок необходимо обратиться к разработчикам.

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: DatabaseManager: schema verification failed
- [E]: DatabaseManager: repairing database failed

- проверьте успешность создания архива БД системы. При наличии подобных ошибок необходимо обратиться к разработчикам.

Примеры сообщений об ошибках в протоколе **core.log**:

- [C]: DatabaseManager: rotation of the DB {} failed!
- [C]: DatabaseManager: rotation of the DB {},{} failed!
- [E]: DatabaseManager: database rotation failed

- проверьте успешность назначения прав доступа к БД системы. При наличии подобных ошибок необходимо обратиться к разработчикам.

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: DatabaseManager: grant access failed, mysql_error[{}]={}
- [E]: DatabaseManager: Grant access failed
- [E]: DatabaseManager: Grant standalone access failed

4) Проверьте наличие библиотеки метрологии (**avgspeed.dll**) в папке с ПО. При отсутствии файла или наличии файла с неверной контрольной суммой ПО не будет запущено. Ситуация может произойти при неполном обновлении ПО. Файл может быть перезаписан заново из дистрибутива ПО.

Примеры сообщений об ошибках в протоколе **core.log**:

- [C]: TargetInspector: CRC-32 mismatch for library 'avgspeed.dll'! Calculated: '12341BBA', Reference: '55552BAA'
- [E]: Core: Failed to init module 'TargetInspector'

8.1.2. Частые перезагрузки ПО

ПО автоматически запускается при загрузке ОС. В дальнейшем ПО контролируется супервизором и перезагружается автоматически при возникновении ошибок на старте.

Примеры сообщений об ошибках в протоколе **core.log**:

```
- [I]: SUPERVISOR: Wait for restarting app for 5 sec...
- [E]: SUPERVISOR: Failed to perform request: {}
- [W]: SUPERVISOR: No ping for more than 23 sec (max 20 sec)
- [I]: SUPERVISOR: Terminating process
- [I]: Finished with code {}
```

Варианты решения:

1) Проверьте ошибки одиночных запусков ПО аналогично приведённым выше в п.8.1.1.

2) Проверьте сторонние проблемы перезагрузки ПО - перезагрузки самого регистратора и всей операционной системы. В этом случае необходимо проверить состояние ПО самого регистратора, на котором установлена система (режим установки на регистратор).

8.1.3. Недоступен веб-интерфейс

Отображается сообщение "Страница не найдена" или "Ошибка 404" или происходит отображение ожидания загрузки на ранее открытой странице веб-интерфейса.

Варианты решения:

1) Проверьте состояние связи с устройством с установленным ПО по указанному в URL строке веб-браузера IP-адресом. Проверить связь можно с помощью системных утилит типа ping. При отсутствии связи необходимо обратиться к системному администратору для выяснения проблем сети и проверки состояния регистраторов и устройства с запущенным ПО системы.

2) Проверьте правильность указания IP-адреса и порта веб-интерфейса в URL строке веб-браузера (см. п. 5.1, по умолчанию, порт 11000).

3) Проверьте запущено и выполняется ли ПО системы на целевом регистраторе или выделенном сервере через "Диспетчер задач" ("Task Manager") непосредственно на стороне данной удалённой ОС. При отсутствии ПО в выполняемых процессах обратитесь к выяснению проблем запуска, описанных в п. 8.1.1.

4) Проверьте настройки шлюзов/роутеров и проброса настроенного порта веб-интерфейса в локальной сети. Необходимо настроить внешний порт на указанный в п. 5.1 (по умолчанию, порт 11000).

5) Проверьте наличие файла настроек веб-интерфейса **.\config\web_server.ini** (например, **C:\duet\config\web_server.ini**). При сделанных вручную изменениях в файле проверьте корректность структуры файла или восстановите настройки исходным файлом из дистрибутива.

Примеры сообщений об ошибках в протоколе **web_server.log**:

```
- [W]: WebServer: C:\duet\config\web_server.ini not found
```

- [E]: WebServer: Cannot find WebServer config file

8.1.4. Отсутствуют фиксации нарушений

ПО выполняется, но на диске не появляются файлы материалов, а также не появляются фиксации на странице "Контроль" и записи в "Журнале фиксаций" в веб-интерфейсе.

Варианты решения:

1) Проверьте текущий режим работы. Включите **режим контроля** на странице "Контроль" веб-интерфейса кнопкой "Включить".

2) Проверьте настройки подключения системы к регистраторам на странице "Параметры подключения":

- Тип регистраторов;
- IP-адрес(а);
- Имя пользователя, пароль и порт сервиса БД на регистраторе. При их отличии от стандартных на регистраторах (см. п.3.3 "Профили приборов"), изменение — только через файл настроек config.json вручную;

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: DeviceAdapterSkat: [Slave] get targets metadata query failed(#1), mysql_error(1)[2013]=Lost connection to MySQL server during query
- [I]: MySQLClient: [Slave] reconnect...
- [E]: MySQLClient: [Slave] connect 'main' 'prometheus'@192.168.0.4:3306 error, mysql_error[2003]=Can't connect to MySQL server on '192.168.0.4' (10060)
- [E]: DeviceAdapterSkat: [Slave] connect database failed
- [E]: PgSQLClient: [Slave] connect 'mary' 'prometheus'@192.168.0.23:5432 error, pgsql_error=could not connect to server: Connection refused (0x0000274D/10061) Is the server running on host "192.168.0.23" and accepting TCP/IP connections on port 5432?
- [E]: DeviceAdapterOracul: [Slave] connect database failed
- [W]: TargetInspector: Can't get slave last event index. Try again after 3 seconds.

- Имя пользователя, пароль и порт для сервиса FTP на регистраторах. Изменение порта при особых настройках сети — только через файл настроек config.json вручную.

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: FTPClient: [Slave] download: curl perform failed ({})

3) Проверьте наличие связи с регистраторами. Связь может быть недоступна временно. Состояние связи регистраторов указывается индикатором между серийными номерами в заголовочной строке веб-интерфейса (Внимание: проверка связи производится только в момент появления проездов и

попытках получения материалов от регистраторов). Ошибки связи с БД и FTP сервисами регистраторов аналогичны указанным в варианте (2) для случая неверных параметров доступа.

Проверка состояния связи также может быть осуществлена запросом получения данных о местоположении регистраторов кнопками поиска на странице "Расположение регистраторов" в веб-интерфейсе. При недоступности регистраторов запрос вернёт ошибку.

Для решения проблемы может потребоваться привлечение системного администратора для исправления настроек в сети.

4) Проверьте корректность серийных номеров ведущего и ведомого регистраторов, введенных на странице "Параметры подключения". Материалы с несоответствующими серийными номерами будут отброшены и фиксации системой будут пропущены.

Примеры сообщений об ошибках в протоколе **core.log**:

```
- [W]: TargetInspector: Bad master serial number for P012AB|178!
- [W]: TargetInspector: Bad slave serial number for P012AB|178!
```

5) Проверьте параметры "Расположения регистраторов" в веб-интерфейсе:

- Взаимная ориентация регистраторов. Проезды и нарушения не будут найдены при неверной ориентации.

6) Проверьте параметры на странице "Поверка" в веб-интерфейсе:

- Расстояние между приборами. Некорректное значение приведёт к неверно вычисляемой средней скорости, проезды и нарушения не будут найдены.

7) Проверьте порог фиксации скорости на странице "Параметры контроля". Возможно, на данном участке дороги нет превышений указанного порога скорости.

8) Проверьте наличие фиксаций проездов на самих регистраторах и наличие скорости движения ТС (производится по руководствам регистраторов, используемых в составе системы).

9) Проверьте форматы написания ГРЗ, поступающих в материалах от регистраторов, возможно, потребуется явное указание форматов вручную на странице "Параметры подключения" — "Использовать формат ГРЗ" и параметры "ГРЗ" (ведущего и ведомого).

10) Проверьте наличие ошибок чтения и разбора файлов материалов от регистраторов, возможно, используется неизвестный формат данных.

Необходимо обратиться к разработчикам и предоставить информацию о проблемном регистраторе и его форматах материалов.

Примеры сообщений об ошибках в протоколе **core.log**:

```
- [W]: TargetInspector: Empty slave file path received for x000xx|178.
- [W]: TargetInspector: Can't parse master frame for x000xx|178. Offender will be removed.
- [W]: TargetInspector: Can't find master frame for x000xx|178. Offender will be removed.
- [W]: TargetInspector: Can't parse slave frame for x000xx|178. Offender will be removed.
```

- [W]: TargetInspector: Can't find slave frame for x000xx|178. Offender will be removed.

11) Проверьте наличие и корректность файлов шаблонов xml, по которым создаются графические материалы (bin и jpg). Наименование файла текущего шаблона задается в веб-интерфейсе на странице "Настройка материалов" — "Вид коллажа" (параметр в config.json — "**resultfile_template_name**"). Расположение шаблонов по умолчанию — папка **./templates** (например, **C:\duet\templates\)** — при необходимости может быть изменено вручную в параметре в config.json — "**resultfile_templates_dir**".

При наличии ошибок о шаблоне в протоколе выполнения материалы не будут созданы. Полный список сообщений приведен в табл.8.9 "Сообщения ReportBuilder".

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: ReportBuilder: Error while opening template file C:/duet/templates/ru.xml
- [E]: ReportBuilder: Error while parsing xml structure: Start-end tags mismatch
- [E]: ReportBuilder: Error while parsing xml structure: Error parsing end element tag
- [E]: ReportBuilder: failed to parse [object_image] node
- [E]: ReportBuilder: no templates were loaded!
- [E]: ReportBuilder: unknown 'type' attribute [colge]
- [E]: ReportBuilder: attribute 'subtype' has unknown subtype [C1_3]
- [E]: ReportBuilder: 'destination' node 'x', 'y', 'width' or 'height' attributes were not found
- [E]: ReportBuilder: 'text' node 'size' attribute not found, required

12) Проверьте наличие файлов шрифтов, требуемых для шаблонов коллажей. По умолчанию требуется шрифт **verdanab.ttf**. При использовании в шаблонах других шрифтов необходимо их поместить в папку с ПО.

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: ReportBuilder: font 'verdana' loading failed

8.1.5. Долгий переход в состояние контроля

ПО выполняется, но в веб-интерфейсе долго отображается мигающий индикатор фотокамеры в заголовочной строке веб-интерфейса после запуска приложения и/или при переходе в режим Контроль.

Подготовка может длиться до 15-20 минут при запуске ПО с большим архивом файлов материалов на диске (более 30 Гб). Пути к архиву указаны в файле настроек config.json в параметрах "result_directory" и "result_directory_tmp"(при пустом значении используется "result_directory").

Максимальный объем архива файлов материалов указывается в config.json в параметре "**storage_limit_fixations**".

Варианты решения:

1) Проверьте объем каталога архива материалов. При объеме более 30 Гб уменьшите максимально допустимый размер архива параметром в config.json "**storage_limit_fixations**", но **не менее 15 Гб** в байтах (см. описание процедуры в п. 6.8 "Автоматическая очистка архива материалов"). После автоматической очистки при очередном запуске ПО новые запуски будут происходить быстрее.

2) Проверьте успешность выполнения периодического сканирования и очисток архива материалов. При появлении постоянных ошибок операций с файлами и каталогами (чтения и удаления) необходимо остановить ПО и проверить диск системными средствами и исправить ошибки файловой системы. Далее, после запуска ПО, архив файлов материалов будет корректно уменьшен в размере.

Примеры сообщений об ошибках в протоколе **core.log**:

```
- [E]: FileManager: Directory checking exists error D:/arch/duet/2020/09/20/01_05_30_702
- [E]: FileManager: Directory checking is_directory error D:/arch/duet/2020/09/20/01_05_30_702
- [E]: FileManager: Scanning results failed ({})
- [E]: FileManager: Calculating files size failed
- [E]: FileManager: Error path: "D:/arch/duet/2020/09/20/01_05_30_702/duet_protobuff_2020_09_20_01_05_30.bin"
- [E]: FileManager: check file_size error: D:/arch/duet/2020/09/20/01_05_30_702/duet_2020_09_20_01_05_30.jpg
- [E]: FileManager: error removing: D:/arch/duet/2020/08/20/01_10_30_112
- [E]: FileManager: thread scan error({})
- [E]: FileManager: thread clean error({})
- [E]: FileManager: thread scan/calc error({})
```

8.1.6. Отсутствуют файлы цифровых подписей

При формировании материалов отсутствуют файлы цифровых подписей (*.sign).

Варианты решения:

1) Проверьте выбранный тип цифровой подписи в веб-интерфейсе на странице в "Настройки материалов" в поле "Цифровая подпись" (параметр в config.json "**resultfile_sign_type**").

2) Проверьте наличие в папке с ПО (**C:/duet/**) и корректность файлов ключей, соответствующих выбранному типу цифровой подписи: **privolvia.pem** — для типа RSA, **privolvia.sxp** — для типа GOST (см. п.6.7).

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: FileSaver: sign is not inited! Sign won't be saved.
- [E]: FileSaver: error while saving sign file 'D:/arch/2020/06/16/00_39_29_396/duet_2020_06_16_00_39_29.sign'
- [E]: FileSaver: Error while writeFileAndSign
- [E]: FileSaver: Error while writeFileAndSign slave
- [E]: FileSaver: Error while writeFileAndSign master
- [E]: FileSaver: Error while writeFileAndSign GRZ
- [E]: FileSaver: Gcrypt_wrapper::init_privkey failed

3) Проверить наличие и доступность папки для сохранения файлов материалов, в config.json в параметре "result_directory" (например, D:/arch/duet).

Примеры сообщений об ошибках в протоколе **core.log**:

- [E]: FileSaver: sign is not inited! Sign won't be saved.
- [E]: FileSaver: error while saving sign file 'D:/arch/2020/06/16/00_39_29_396/duet_2020_06_16_00_39_29.sign'

При продолжении проблем из указанных выше или других обратите внимание на полный список сообщений в протоколах работы ПО (п.8.2) и обратитесь к разработчикам за решением.

8.2. Список сообщений

В таблицах 8.1-8.11 приведены основные информационные сообщения, сообщения об ошибках и предупреждения, сохраняемые в протоколах работы ПО в каталоге logs (core.log, web_server.log, supervisor.log). А также приведено описание ситуации и решением проблемы для сообщений об ошибках.

Сообщения сгруппированы по модулям ПО:

- Core — главное ядро приложения;
- Supervisor — супервизор;
- Config — модуль конфигурации и настроек;
- DatabaseManager — модуль БД системы;
- MySQLClient — модуль клиента БД типа MySQL;
- PgSQLClient — модуль клиента БД типа PostgreSQL (сообщения аналогичны модулю MySQLClient);
- FileManager — модуль индексирования и очистки архива файлов материалов;
- TargetInspector — модуль инспекции нарушений;
- DeviceAdapter* — Skat, SkatPP, Krechet,Oracul — адаптеры интерфейсов взаимодействия с регистраторами разного вида (работа с БД и FTP сервисами, чтение материалов);
- ReportBuilder — модуль построения материалов нарушений;
- FileSaver — модуль сохранения файлов материалов нарушений;

- WebServer — модуль ВЕБ-сервера для ВЕБ-интерфейса и протоколов внешнего взаимодействия (JSOB-RPC, HTTP/XML).

Типы/уровень сообщений соответствуют указанным в п.7.2.

Таблица 8.1 — Сообщения Core

№	Тип	Текст сообщения	Описание	Решение
1	INFO	Starting...	Старт приложения	
2	INFO	Core: Log level: DEBUG	Установка уровня протоколирования (DEBUG по умолчанию)	
3	INFO	1) Core: Deploy mode: standalone 2) Core: Deploy mode: module	Сигнализирует о запущенной конфигурации системы (сервер или модуль).	
4	INFO	Core: Check application instances...	Начало проверки других запущенных копий ПО.	
5	INFO	Core: Another instances not found, continue.	Другие запущенные копии ПО не обнаружены, продолжение запуска.	
6	ERROR	Core: Another instance of this application is running! Close it first!	Ошибка запуска. Обнаружены другие запущенные копии ПО.	Необходимо завершить текущие работающие процессы ПО системы duet_core через Диспетчер задач (Task manager).
7	CRITICAL	Core: Error while checking running instance of application ({} {})	см. графу 6	см. графу 6
8	INFO	Core: Init	Сигнализирует о старте инициализации компонентов системы.	
9	INFO	Core: Config path: ./config/config.json	Старт загрузки конфигурации по указанному пути.	
10	INFO	Core: Config loaded	Конфигурация успешно загружена.	
11	CRITICAL	Core: Config loading failed!	Ошибка загрузки конфигурации.	см. табл.8.3 графы 1,2,3
12	INFO	Core: Users config path: ./config/users.json	Старт загрузки конфигурации учетных записей пользователей по указанному пути.	
13	INFO	Core: Users config loaded	Конфигурация учетных записей пользователей успешно загружена.	
14	CRITICAL	Core: Users config loading failed!	Ошибка загрузки конфигурации учетных записей пользователей.	см. табл.8.3 графы 1,2,3
15	INFO	Core: Init modules	Старт инициализации модулей системы.	
16	DEBUG	Core: Initing module started: {}	Старт инициализации модуля.	
	DEBUG	Core: Initing module finished: {}	Окончание процедуры инициализации модуля.	

№	Тип	Текст сообщения	Описание	Решение
17	ERROR	Core: Failed to init module {}	Ошибка инициализации модуля.	1) См. ошибки от соответствующего модуля. 2) При отсутствии подробностей об ошибке от модуля обратитесь к разработчикам для решения проблемы.
18	INFO	Core: Start modules	Старт модулей системы.	
19	DEBUG	Core: Starting module started: {}	Старт модуля.	
20	DEBUG	Core: Starting module finished: {}	Окончание процедуры начала работы модуля.	
21	ERROR	Core: Failed to start module {}	Ошибка запуска модуля.	1) См. ошибки от соответствующего модуля. 2) При отсутствии подробностей об ошибке от модуля обратитесь к разработчикам для решения проблемы.
22	DEBUG	Core: Stopping module started: {}	Начало остановки работы модуля.	
23	DEBUG	Core: Stopping module finished: {}	Остановка модуля завершена.	
24	ERROR	Core: Failed to stop module {}	Ошибка остановки модуля.	1) См. ошибки от соответствующего модуля. 2) При отсутствии подробностей об ошибке от модуля обратитесь к разработчикам для решения проблемы.
25	INFO	Core: Set mode [setup] -> [work]	Изменение режима работы ПО. Настройка -> Контроль.	
26	ERROR	Core: Set mode failed	Ошибка изменения режима работы ПО.	1) См. другие ошибки от компонентов системы. 2) При отсутствии подробностей об ошибке обратитесь к разработчикам для решения проблемы.
27	CRITICAL	Core initialization failed	Ошибка инициализации ПО и выход.	1) См. другие ошибки от компонентов системы. 2) При отсутствии подробностей об ошибке обратитесь к разработчикам для решения проблемы.
28	INFO	Finished with code {}	Работа приложения завершена с указанным кодом.	
29	CRITICAL	Core: SIGINT({}) received, QCoreApplication::quit!	Выполнено принудительно завершение работы ПО	

№	Тип	Текст сообщения	Описание	Решение
		Handling console event({}), QCoreApplication::quit()	пользователем или операционной системой.	

Таблица 8.2 — Сообщения Supervisor

№	Тип	Текст сообщения	Описание	Решение
1	INFO	SUPERVISOR: Check application instances...	Начало проверки других запущенных копий ПО с супервизором.	
2	ERROR	SUPERVISOR: Another instance of this application is running! Close it first!	Ошибка запуска. Обнаружены другие запущенные копии ПО.	Необходимо завершить текущие работающие процессы ПО системы duet_core через Диспетчер задач (Task manager).
3	ERROR	SUPERVISOR: Error while checking running instance of application ({})	Ошибка проверки других выполняющихся копий ПО.	Необходимо обратиться к разработчикам.
4	INFO	SUPERVISOR: Watchdog ping url: 127.0.0.1:11000/jrpc	Установлен url для работы watchdog (мониторинга приложения).	
5	INFO	SUPERVISOR: Watchdog initialized to warmup={} sec, period={} sec, timeout={} sec	Установлены периоды ожидания watchdog для контроля функционирования ПО.	
6	INFO	SUPERVISOR: Starting 'duet_core.exe' with args ''	Запуск основного приложения супервизором.	
7	INFO	SUPERVISOR: Wait for restarting app for {} sec...	Ожидание супервизором перед перезапуском основного приложения.	
8	ERROR	SUPERVISOR: Failed to perform request: {}	Ошибка выполнения запроса в основное ПО.	Действий не требуется, при длительно отсутствии ответа ПО будет автоматически перезагружено.
9	WARNING	SUPERVISOR: No ping for more than {} sec (max {} sec)	Уведомление об отсутствии ответа от ПО в течение указанного времени.	
10	INFO	SUPERVISOR: Terminating process	Завершение работы и перезапуск основного процесса ПО.	
11	INFO	SUPERVISOR: Application done with code {} ({}) SUPERVISOR: Done	Завершение работы супервизора.	
12	CRITICAL	SUPERVISOR: SIGINT({}) received	Завершение работы супервизора произведено вручную или сигналом от операционной системы.	

Таблица 8.3 — Сообщения Config

№	Тип	Текст сообщения	Описание	Решение
1	ERROR	Config: Can't open configuration file "config.json" for saving	Ошибка открытия файла конфигурации.	Необходимо проверить наличие указанного файла конфигурации (config.json или users.json) в каталоге ПО ./config.
2	ERROR	Config: Can't parse json file "config.json"	Некорректный формат файла конфигурации. Вероятно, нарушена структура json формата.	Необходимо исправить структуру файла и состав полей. Исходный файл конфигурации можно взять из дистрибутива ПО.
3	ERROR	Config: Can't load parameter "speed_limit" from "config.json"	Не удалось загрузить значение параметра из файла конфигурации. Вероятно, нарушена структура json формата или значения полей изменены на недопустимые.	Необходимо исправить структуру файла и состав полей. Исходный файл конфигурации можно взять из дистрибутива ПО.
4	INFO	Config: Migrate config "config.json" from 6 to 7 version	Уведомление о миграции(обновлении) конфигурационного файла со старой на новую версию.	
5	ERROR	Config: Failed to migrate config "config.json"	Ошибка обновления файла конфигурации.	1) Старая версия файла конфигурации больше не поддерживается. Необходимо взять новый файл конфигурации из дистрибутива и выполнить настройки заново. 2) При постоянном возникновении ошибки на новых файлах настроек обратитесь к разработчикам.
6	INFO	Config: Migration OK	Обновление версии файла конфигурации успешно произведено.	
7	ERROR	Config: Can't read version of config "config.json"	Ошибка чтения версии файла конфигурации.	Файл конфигурации испорчен или имеет неизвестный формат. Необходимо взять новый файл конфигурации из дистрибутива ПО.
8	ERROR	Config: Can't migrate config to low version	Ошибка обновления файла конфигурации. Нельзя переходить с более новой версии на старую.	При необходимости перехода на более старую версию ПО необходимо вручную вносить исправления в файл конфигурации целевой версии ПО в соответствии с текущими настройками ПО. Некоторые неподдерживаемые ранее

№	Тип	Текст сообщения	Описание	Решение
				функции могут быть недоступны. Такое преобразование вручную выполнять не рекомендуется, обратитесь к разработчикам.
9	ERROR	Config: Migration step 6-7 failed	Ошибка обновления файла конфигурации с указанием проблемных шагов преобразования.	см. графу.5

Таблица 8.4 — Сообщения DatabaseManager

№	Тип	Текст сообщения	Описание	Решение
1	INFO	DatabaseManager: init started...	Начало инициализации модуля работы с БД	
2	DEBUG	DatabaseManager: try connect service database with user='root' and empty password	Уведомление о попытке подключения к БД с пустым паролем. Для систем с пустой БД и конфигурации ПО в режиме сервера.	
3	INFO	DatabaseManager: Connected to MySQL server: 5.7.29-log	Уведомление об успешном подключении к БД с отображением версии сервера БД.	
4	ERROR	DatabaseManager: connect service database failed	Ошибка подключения к БД системы	1) Проверьте параметры доступа к БД системы, поля в настройках config.json: - "parameters_db_name" - "parameters_db_pass" - "parameters_db_user" - "result_db_name" - "result_db_pass" - "result_db_user" Исправьте настройки доступа при их неверном значении, например пароль пользователя БД (см. п.3.5.2 "Установка приложения"); 2) Проверьте состояние службы сервера БД на устройстве, на котором запускается ПО; 3) Может происходить из-за резкого изменения времени на устройстве. В таком случае дополнительные действия не требуются, если подключение произошло успешно при следующей попытке.

№	Тип	Текст сообщения	Описание	Решение
5	ERROR	DatabaseManager: MySQL server info failed	Ошибка получения информации о сервере БД системы.	Неизвестный сервер БД. 1) Допускаются кратковременные появления из-за внезапной потере связи с БД, перезапуске сервиса БД; 2) При постоянном появлении необходимо обратиться к разработчикам для решения.
6	DEBUG	DatabaseManager: try reconnect service...	Уведомление о начале переподключения к БД.	
7	ERROR	DatabaseManager: reconnect service error	Ошибка переподключения к БД.	см. графу 4
8	WARNING	DatabaseManager: Service database name parameter was cleaned: {} -> {}	Уведомление о переименовании наименования сервисной БД указанной пользователем, так как имя содержит недопустимые символы.	Проверьте параметры доступа к БД системы, поля в настройках config.json, см. графу 4
9	WARNING	DatabaseManager: Main results database name parameter was cleaned: {} -> {}	Уведомление о переименовании наименования основной БД указанной пользователем, так как имя содержит недопустимые символы.	Проверьте параметры доступа к БД системы, поля в настройках config.json, см. графу 4
10	WARNING	DatabaseManager: Service database name parameter was renamed: {} -> {}	Уведомление о переименовании наименования сервисной БД указанной пользователем, так как имя совпадает с основной БД.	Проверьте параметры доступа к БД системы, поля в настройках config.json, см. графу 4
11	WARNING	DatabaseManager: Custom ({}) database name parameter was renamed: {} -> {}	Уведомление о переименовании наименования БД протокола samara, так как имя совпадает с основной БД.	Проверьте параметры доступа к БД системы, поля в настройках config.json, см. графу 4
12	ERROR	1) DatabaseManager: connect root GRANT ALL PRIVILEGES failed, mysql_error[{}]={} 2) DatabaseManager: grant access failed, mysql_error[{}]={} 3) DatabaseManager: Grant access failed 4) DatabaseManager: Grant standalone access failed	Ошибки изменения прав в БД.	Необходимо обратиться к разработчикам для устранения проблемы.
13	INFO	DatabaseManager: Checking structure	Начало создания и проверки структуры БД.	

№	Тип	Текст сообщения	Описание	Решение
14	ERROR	DatabaseManager: schema verification failed	Ошибка проверки структуры БД.	1) Необходимо обратиться к разработчикам для устранения проблемы; 2) Восстановить БД можно с удалением всех данных: подключиться к БД от администратора через внешнее приложение-клиент к БД и удалить БД, относящиеся к данному ПО (duet_main, duet_service, region_ru_samara).
15	INFO	DatabaseManager: Checking repair	Начало восстановления БД при обнаружении нарушения целостности..	
16	ERROR	DatabaseManager: repairing database failed	Ошибка восстановления БД.	1) Необходимо обратиться к разработчикам для устранения проблемы; 2) Восстановить БД можно через внешнее приложение-клиент к БД: подключиться к БД от администратора и запустить процедуру поиска ошибок и восстановления БД, относящихся к данному ПО (duet_main, duet_service, region_ru_samara).
17	CRITICAL	DatabaseManager: rotation of the DB duet_main failed!	Начало удаления старых записей фиксаций в БД.	
18	CRITICAL	DatabaseManager: rotation of the DB duet_main, duet_main_xml failed!	Ошибка удаления старых записей в БД duet_main, duet_main_xml	1) Необходимо обратиться к разработчикам для устранения проблемы; 2) Восстановить БД можно с удалением всех данных: подключиться к БД от администратора через внешнее приложение-клиент к БД и удалить БД, относящиеся к данному ПО (duet_main, duet_service, region_ru_samara).
19	ERROR	DatabaseManager: database rotation failed	Ошибка удаления старых записей в БД	см. графу 18
20	ERROR	DatabaseManager: Users config is empty. Change	Ошибка чтения учетных записей пользователей ВЕБ-интерфейса в БД.	1) Проверьте содержимое файла учетных записей пользователей

№	Тип	Текст сообщения	Описание	Решение
		users.json to new one from software distribution		"users.json". Необходимо наличие хотя бы одной учетной записи; 2) Восстановите файл "users.json" из дистрибутива ПО.
21	ERROR	DatabaseManager: Database update users failed	Ошибка чтения учетных записей пользователей ВЕБ-интерфейса в БД.	см. графу 20
22	ERROR	1) DatabaseManager: connect results database failed 2) DatabaseManager: connect results(web) database failed 3) DatabaseManager: Database connection failed	Ошибка подключения к БД системы отдельных программных клиентов и системы в целом.	см. графу 4
23	ERROR	1) DatabaseManager: escape user action user failed 2) DatabaseManager: escape user action 'a' failed 3) DatabaseManager: insert user action failed	Ошибка записи в БД данных о действиях пользователя.	При частом появлении необходимо обратиться к разработчикам для устранения проблемы.

Таблица 8.5 — Сообщения MySQLClient

№	Тип	Текст сообщения	Описание	Решение
1	INFO	MySQLClient: [CLIENT] connect " 'root'@127.0.0.1:3306 ok (5.7.29)	Уведомление об успешном подключении клиентом к БД с отображением версии сервера БД.	
2	ERROR	MySQLClient: [CLIENT] connect " 'root'@127.0.0.1:3306 error, mysql_error[1045]=Access denied for user 'root'@'localhost' (using password: NO)	Ошибка подключения клиентом к БД системы или регистратора	1) Проверьте параметры доступа к БД системы в настройках config.json, если CLIENT: results, webreslts, service. 2) Проверьте параметры доступа к БД регистраторов в настройках config.json, а также состояние связи с регистраторами, если CLIENT: master/slave. 3) Может происходить из-за резкого изменения времени на устройстве/регистраторе. В таком случае дополнительные действия не требуются, если подключение произошло успешно при следующей попытке.
3	ERROR	MySQLClient: [CLIENT] select database {} failed	Ошибка подключения клиентом к БД системы	Некорректные права пользователей БД или структура БД нарушена.

№	Тип	Текст сообщения	Описание	Решение
			или регистратора со сменой БД	Необходимо обратиться к разработчикам для устранения проблемы.
4	CRITICAL	MySQLClient: [CLIENT] set connect timeout error	Ошибка подключения клиентом к БД системы или регистратора. Истекло время ожидания.	см. графу 2
5	ERROR	MySQLClient: [CLIENT] set charset error	Ошибка запроса кодировки при подключении клиентом к БД системы или регистратора.	Необходимо обратиться к разработчикам для устранения проблемы.
6	INFO	MySQLClient: [CLIENT] reconnect...	Уведомление о начале переподключения клиента к БД.	
7	ERROR	MySQLClient: [CLIENT] exec query failed	Ошибка выполнения запроса в БД с расшифровкой подробностей.	Необходимо обратиться к разработчикам для устранения проблемы.
8	ERROR	1) MySQLClient: [CLIENT] truncate table {} failed 2) MySQLClient: [CLIENT] delete all from table {} failed	Ошибка выполнения запроса очистки таблицы в БД с расшифровкой подробностей.	Необходимо обратиться к разработчикам для устранения проблемы.
9	ERROR	MySQLClient: [CLIENT] drop table {} failed	Ошибка выполнения запроса удаления таблицы в БД с расшифровкой подробностей.	Необходимо обратиться к разработчикам для устранения проблемы.

Таблица 8.6. Сообщения FileManager

№	Тип	Текст сообщения	Описание	Решение
1	ERROR	FileManager: Invalid limits: limit={} clean={}	Ошибка инициализация заданных пределов размера файлового архива материалов.	Проверьте максимальный объем каталога файлов материалов в настройках config.json - параметр "storage_limit_fixations". Объем должен быть более 10Гб. (см. описание процедуры в п.6.8 "Автоматическая очистка архива материалов").
2	INFO	FileManager: Scanning directories...	Уведомление о запуске процедуры индексирования каталога файлов материалов.	
3	ERROR	FileManager: Directory checking exists error D:/arch/duet/2020/09/20/01_05_30_702	Ошибка проверки существования каталога на диске.	При появлении постоянных ошибок операций с файлами и каталогами (чтения и удаления) необходимо остановить ПО и проверить диск системными средствами и исправить ошибки файловой системы.

№	Тип	Текст сообщения	Описание	Решение
4	ERROR	FileManager: Directory checking is_directory error D:/arch/duet/2020/09/20/01_05_30_702	Ошибка проверки параметров на диске.	см. графу 3
5	ERROR	FileManager: Scanning results failed ({})	Ошибка этапа индексирования каталога материалов.	см. графу 3
6	ERROR	FileManager: Calculating files size failed	Ошибка этапа расчета общего объема каталога материалов.	см. графу 3
7	ERROR	FileManager: Error path: "D:/arch/duet/2020/09/20/01_05_30_702/duet_protobuff_2020_09_20_01_05_30.bin"	Ошибка обращения к указанному файлу.	см. графу 3
8	ERROR	FileManager: check file_size error: D:/arch/duet/2020/09/20/01_05_30_702/duet_2020_09_20_01_05_30.jpg	Ошибка измерения размера указанного файла.	см. графу 3
9	ERROR	FileManager: error removing: D:/arch/duet/2020/08/20/01_10_30_112	Ошибка удаления указанного файла.	см. графу 3
10	CRITICAL	FileManager: unsync files cleaner	Ошибка работы механизма автоочистки объема каталога до определенного объема из-за изменений на диске сторонними приложениями.	1) Действий не требуется. Процедура переиндексации каталога и очистки будет запущена повторно автоматически. 2) При частом появлении необходимо остановить ПО и проверить диск системными средствами и исправить ошибки файловой системы.

Таблица 8.7 — Сообщения TargetInspector

№	Тип	Текст сообщения	Описание	Решение
1	INFO	TargetInspector: Start files scanning	Сигнализирует о начале старта сканирования файлов в директориях в параметрах config.json **result_directory** и **result_directory_tmp.**	
2	CRITICAL	TargetInspector: Can't create result directory D:/arch/duet	Ошибка создания директории в файловой системе.	Возможно, в конфигурационном файле указан некорректный путь в полях **result_directory** или **result_directory_tmp**. Проверить, изменить.
3	INFO	TargetInspector: Start	Сигнализирует о начале запуска режима "Контроль" для модуля **TargetInspector**.	
4	ERROR	TargetInspector: Can't start. TargetInspector wasn't initialized correctly	Сигнализирует о неудачном запуске режима "Контроль" из-за некорректной инициализации модуля **TargetInspector.** Возможные причины: 1) Ошибка загрузки библиотеки метрологии (avgspeed).	1) Проверить наличие библиотеки avgspeed в каталоге с исполняемым файлом программы. 2) Проверить версию библиотеки (возможно некорректное значение контрольной суммы).
5	INFO	TargetInspector: Can't start. TargetInspector is running	При попытке запуска режима "Контроль" сигнализирует о том, что модуль **TargetInspector** уже находится в режиме "Контроль".	
6	INFO	TargetInspector: Init gost signer: key='privolvia.pem'	Сигнализирует об использовании указанного файла ключа для цифровой подписи файлов материалов.	
7	INFO	TargetInspector: Init ReportBuilder: template C:/duet/templates/ru.xml	Сигнализирует об использовании указанного файла XML шаблона для формирования материалов.	
8	ERROR	TargetInspector: Can't start. Master device adapter implementation [skat] is undefined	Ошибка при запуске режима "Контроль". Невозможно создать deviceadapter указанного типа для ведущего прибора.	Проверить конфигурационный файл config.json. Объект **master** поле **type** должно быть в одном из следующий значений: - skat - skatpp - oracul - krechet
9	ERROR	TargetInspector: Can't start. Slave device adapter	Ошибка при запуске режима "Контроль".	Проверить конфигурационный файл

№	Тип	Текст сообщения	Описание	Решение
		implementation [skat] is undefined	Невозможно создать deviceadapter указанного типа для ведомого прибора.	config.json. Объект **slave** поле **type** должно быть в одном из следующий значений: - skat - skatpp - oracul - krechet
10	INFO	TargetInspector: Started	Сигнализирует об успешном старте режима "Контроль"	
11	INFO	TargetInspector: Stop	Сигнализирует о начале остановки режима "Контроль" в модуле **TargetInspector**.	
12	INFO	TargetInspector: Can't stop. TargetInspector is already stopped	При попытке остановки режима "Контроль" сигнализирует о том, что данный режим уже остановлен (не активен)	
13	INFO	TargetInspector: Stopped	Сигнализирует об успешной остановке режима "Контроль" и переходе в режим "Настройки"	
14	INFO	TargetInspector: Start slave metadata downloading from event id = 1445	Сигнализирует о начале цикла скачки данных о событиях проезда из БД ведомого регистратора с указанного индекса из таблицы **events**.	
15	WARNING	TargetInspector: Can't get slave last event index. Try again after 3 seconds.	Ошибка чтения последнего индекса события проезда таблицы **events** из БД ведомого регистратора.	1) Проверить параметры сетевого подключения к БД ведомого регистратора в конфигурационном файле (IP, порт, название БД, имя пользователя, пароль); 2) Проверить сетевое соединение с ведомым регистратором; 3) Проверить корректность типа ведомого регистратора (skat/skatpp/krechet/oracul) в ВЕБ-интерфейсе или файле настроек; 4) Пустая таблица на регистраторе - ждать заполнения новыми проездами.
16	INFO	TargetInspector: Slave metadata pack received. Size 5. Last event id 1543.	Сигнализирует об успешном получении пакета данных о проездах ведомого регистратора из	

№	Тип	Текст сообщения	Описание	Решение
			БД (указанного количества записей).	
17	WARNING	TargetInspector: Target x000xx|178 was removed (undefined direction)	Предупреждение о том, что проезд ТС был отклонен от проверки на наличие нарушения средней скорости из-за неизвестного направления движения на ведомом регистраторе (не встречное и не попутное).	Проверить настройки на ведомом регистраторе, влияющие на формирование направления движения ТС.
18	WARNING	TargetInspector: Target x000xx|178 was removed (can't calculate speed - zero time span)	Ошибка расчета средней скорости из-за нулевой разницы во времени проездов транспортного средства через рубежи	1) Проверить настройку расстояния между регистраторами; 2) Проверить настройку часового пояса; 3) Проверить время фиксаций в материалах ведомого и ведущего регистраторов.
19	INFO	TargetInspector: Target x000xx|178 was removed (532.32 filtered by max speed 350.0)	Предупреждение о том, что ТС был отклонено от проверки на наличие нарушения средней скорости из-за неправдоподобно большой средней скорости.	Проверить настройку расстояния между регистраторами.
20	INFO	TargetInspector: Skip duplicate for offender x000xx|178	Сигнализирует о том, что недавно (не более 30 сек назад) для ТС с таким же ГРЗ уже сформированы материалы о нарушении средней скорости, и данный дублирующий проезд ТС будет исключён из рассмотрения на наличие нарушения.	
21	INFO	TargetInspector: New offender. Liplate x000xx|178. Speed 50.1. Limit 45.0. Fix 50.0. [m:1245/1263/C0 s:1675/1703/C0]	Сигнализирует о новом найденном нарушителе средней скорости.	
22	INFO	TargetInspector: Target x000xx|178 timed out. Wait 45.43s. Max (125.23 + 30.0)s. Latest pair 1607610320023 (165250)	Сигнализирует о том, что для проезда на ведомом регистраторе не было найдено пары на ведущем регистраторе в течение допустимого времени ожидания. Также показывает, что с таким же ГРЗ на ведущем приборе есть проезд, но этот проезд не актуален (был давно).	
23	INFO	TargetInspector: Target x000xx|178 timed out. Wait	Сигнализирует о том, что для проезда на ведомом	

№	Тип	Текст сообщения	Описание	Решение
		45.43s. Max (125.23 + 30.0)s. No pair found	регистраторе не было найдено пары на ведущем регистраторе в течении допустимого времени ожидания. С таким же ГРЗ на ведущем приборе проездов нет вообще.	
24	ERROR	TargetInspector: Target x000xx|178 was removed by metadata sql parsing error.	Сигнализирует об ошибке разбора данных о проезде, полученных из БД ведущего прибора.	Программная ошибка. Проверить структуру и формат полей БД ведущего регистратора.
25	INFO	TargetInspector: Waiting files scanning stop...	Сигнализирует о том, что модуль в режиме "Контроль", однако ожидает окончания сканирования файлов в директории с материалами, запущенного при старте ПО.	
26	INFO	TargetInspector: Files scanning done	Сигнализирует о том, что сканирование файлов материалов, запущенного при старте ПО, закончено.	
27	INFO	TargetInspector: Waiting files cleaning stop...	Сигнализирует о том, что модуль в режиме "Контроль", однако ожидает окончания процесса очистки архива материалов, запущенного при старте ПО.	
28	INFO	TargetInspector: Files cleaning done	Сигнализирует о том, что очистка файлов, запущенная при старте ПО, завершена.	
29	INFO	TargetInspector: Filemanager files scan and clean started...	Сигнализирует о запуске процесса сканирования и чистки файлов в директории материалов в режиме "Контроль".	
30	INFO	TargetInspector: Waiting files scanning in process stop...	Сигнализирует о запуске процесса сканирования файлов в директории материалов в режиме "Контроль".	
31	INFO	TargetInspector: Files scanning in process done	Сигнализирует о том, что сканирование файлов материалов закончено в режиме "Контроль".	
32	INFO	TargetInspector: Waiting files cleaning in process stop...	Сигнализирует о запуске процесса очистки файлов в директории материалов в режиме "Контроль".	
33	INFO	TargetInspector: Files cleaning in process done	Сигнализирует о том, что очистка файлов материалов закончена в режиме "Контроль".	

№	Тип	Текст сообщения	Описание	Решение
34	WARNING	TargetInspector: Can't get master material list for x000xx|178. Offender will be removed.	Ошибка получения списка файлов материалов для данного ТС из БД на ведущем регистраторе.	Возможно, на регистраторе добавился новый тип нарушения, который ПО системы не распознаёт. В случае проблем со связью, данное сообщение не должно показываться, нарушитель заносится в очередь для повторной попытки соединения и скачивания данных.
35	WARNING	TargetInspector: Can't get slave material list for x000xx|178. Offender will be removed.	Ошибка получения списка файлов материалов для данного ТС из БД на ведомом регистраторе.	Возможно, на регистраторе добавился новый тип нарушения, который ПО системы не распознаёт. В случае проблем со связью, данное сообщение не должно показываться, нарушитель заносится в очередь для повторной попытки соединения и скачивания данных.
36	INFO	TargetInspector: Skip instant speed offender x000xx|178. Write to DB	Сигнализирует о пропуске формирования материалов для нарушителя из-за наличия у него материалов С1.1 на одном из регистраторов.	Если нужно формировать С1.2 всегда, то проверить и выключить настройку "Приоритет ФМ мгновенной скорости" в ВЕБ-интерфейсе (поле **speed_instant_priority** в файле config.json: true - фильтровать, если есть С1.1, false - всегда делать С1.2).
37	ERROR	TargetInspector: writeOffenderToDB for x000xx|178 failed. Offender will be removed	Ошибка записи информации о нарушителе в БД системы. Возможные причины: 1) Ошибка записи в одну из таблиц target/events/materials	1) Проверить корректность полей в конфигурационном файле, отвечающих за работу с БД системы; 2) Проверить доступность БД, сервиса БД на устройстве с ПО.
38	WARNING	TargetInspector: Can't create result directory D:/arch/duet/2020/11/19/04_21_28_114/ for x000xx|178. Offender will be removed.	Ошибка создания каталога в файловой системе. Материалы для этого нарушителя созданы не будут.	1) Возможно, в конфигурационном файле указан некорректный путь к архиву материалов в поле **result_directory**. Проверить, изменить; 2) Проблемы с файловой системой на диске.
39	WARNING	TargetInspector: Can't create temp directory	Ошибка создания временного каталога в	1) Возможно, в конфигурационном

№	Тип	Текст сообщения	Описание	Решение
		D:/arch/duet_tmp/2020/11/19/04_21_28_114/ for x000xx|178. Offender will be removed.	файловой системе. Материалы для этого нарушителя созданы не будут.	файле указан некорректный путь к архиву временных материалов в поле **result_directory_tmp**. Проверить, изменить; 2) Проблемы с файловой системой на диске.
40	INFO	TargetInspector: Start master download D:/arch/all_fixations/2020/11/05/ 2002201_2020_11_05_00_25_21_26_00_00_fr.jpg	Сигнализирует о начале передачи указанного файла материалов из ведущего регистратора.	
41	INFO	TargetInspector: Finish master download D:/arch/all_fixations/2020/11/05/ 2002201_2020_11_05_00_25_21_26_00_00_fr.jpg	Сигнализирует об успешном завершении передачи указанного файла материалов из ведущего регистратора.	
42	WARNING	TargetInspector: Empty master file path received for x000xx|178.	Предупреждение о том, что из БД ведущего регистратора пришел пустой путь к файлу материалов. Следовательно, такой файл не может быть передан. Нарушение будет пропущено.	
43	INFO	TargetInspector: Start slave download D:/arch/all_fixations/2020/11/05/ 1902201_2020_11_05_00_25_23_20_00_00_fr.jpg	Сигнализирует о начале передачи указанного файла материалов из ведомого регистратора.	
44	INFO	TargetInspector: Finish slave download D:/arch/all_fixations/2020/11/05/ 1902201_2020_11_05_00_25_23_20_00_00_fr.jpg	Сигнализирует об успешном завершении передачи указанного файла материалов из ведомого регистратора.	
45	WARNING	TargetInspector: Empty slave file path received for x000xx|178.	Предупреждение о том, что из БД ведомого регистратора пришел пустой путь к файлу материалов. Следовательно, такой файл не может быть передан. Нарушение будет пропущено.	
46	WARNING	TargetInspector: Can't parse master frame for x000xx|178. Offender will be removed.	Ошибка разбора файла материалов с ведущего регистратора. Варианты ошибки: 1) Изменился формат файла, поэтому он не может	1) Необходимо обратиться к разработчикам для анализа причины ошибки и выяснения особого формата данных

№	Тип	Текст сообщения	Описание	Решение
			быть корректно разобран; 2) Неправильно указан тип ведущего регистратора в настройках. Материалы для этого нарушителя созданы не будут.	материалов от регистратора; 2) Проверить корректность настройки типа ведущего регистратора в ВЕБ-интерфейсе на странице "Параметры подключения" (поле **type** в секции **master** в файле настроек config.json)
47	WARNING	TargetInspector: Can't find master frame for x000xx|178. Offender will be removed.	Не удалось загрузить файл материалов с ведущего регистратора (тип файла = 2 "_fr"). Материалы для этого нарушителя созданы не будут.	1) Проверить корректность значений соответствующих полей в конфигурационном файле секции ведущего регистратора - **master** (параметры доступа к сервисам регистратора: FTP, путь fixation_dir и т.п.); 2) Проверить сетевое соединение с ведущим регистратором; 3) Проверить корректность типа ведущего регистратора (skat/skatpp/krechet/oracul); 4) Файл материала отсутствует на регистраторе. Проверить настройки на ведущем регистраторе, влияющие на формирование материалов, включить материалы с увеличенными ТС (_fr).
48	WARNING	TargetInspector: Bad master serial number for x000xx|178!	Предупреждение о несоответствии серийного номера ведущего регистратора, полученного из файла материала с регистратора, с серийным номером в настройках регистратора в системе - ВЕБ-интерфейс "Параметры подключения" (поле serial в конфигурационном файле в секции **master).**	Изменить в настройках серийный номер ведущего регистратора для соответствия серийному номеру в материалах.
49	WARNING	TargetInspector: Can't parse slave frame for x000xx|178. Offender will be removed.	Ошибка разбора файла материала с ведомого регистратора. Варианты ошибки: 1) Изменился формат	1) Необходимо обратиться к разработчикам для анализа причины ошибки и выяснения особого

№	Тип	Текст сообщения	Описание	Решение
			файла, поэтому он не может быть корректно разобран; 2) Неправильно указан тип ведомого регистратора в настройках. Материалы для этого нарушителя созданы не будут.	формата данных материалов от регистратора; 2) Проверить корректность настройки типа ведомого регистратора в ВЕБ-интерфейсе на странице "Параметры подключения" (поле **type** в секции **slave** в файле настроек config.json)
50	WARNING	TargetInspector: Can't find slave frame for x000xx|178. Offender will be removed.	Не удалось загрузить файл материалов с ведомого регистратора (тип файла = 2 "_fr"). Материалы для этого нарушителя созданы не будут.	1) Проверить корректность значений соответствующих полей в конфигурационном файле секции ведомого регистратора - **slave** (параметры доступа к сервисам регистратора: FTP, путь fixation_dir и т.п.); 2) Проверить сетевое соединение с ведомым регистратором; 3) Проверить корректность типа ведомого регистратора (skat/skatpp/krechet/oracul); 4) Файл материала отсутствует на регистраторе. Проверить настройки на ведомом регистраторе, влияющие на формирование материалов, включить материалы с увеличенными ТС (_fr).
51	WARNING	TargetInspector: Bad slave serial number for x000xx|178!	Предупреждение о несоответствии серийного номера ведомого регистратора, полученного из файла материала с регистратора, с серийным номером в настройках регистратора в системе - ВЕБ-интерфейс "Параметры подключения" (поле serial в конфигурационном файле в секции **slave).**	Изменить в настройках серийный номер ведомого регистратора для соответствия серийному номеру в материалах.
52	ERROR	TargetInspector: buildAndSaveReport for x000xx|178 failed. Offender will be removed	Ошибка создания материала для указанного нарушителя.	1) Проверить или заменить файл шаблона; 2) Проверить путь к файлу шаблона и его

№	Тип	Текст сообщения	Описание	Решение
			Материалы для этого нарушителя созданы не будут. Возможные основные причины: 1) Некорректный файл шаблона материалов; 2) Некорректный путь к файлу шаблона материалов.	название в настройках ("Вид коллажа").
53	ERROR	TargetInspector: Can't prepare report - master frame list is empty	Ошибка создания материалов. Отсутствует исходное изображение с ведущего регистратора.	Необходимо обратиться к разработчикам для анализа причины ошибки и выяснения особого формата данных материалов от регистратора.
54	ERROR	TargetInspector: Can't prepare report - master frame is empty	Ошибка создания материалов. Отсутствует исходное изображение с ведущего регистратора.	Необходимо обратиться к разработчикам для анализа причины ошибки и выяснения особого формата данных материалов от регистратора.
55	ERROR	TargetInspector: Can't prepare report - slave frame list is empty	Ошибка создания материалов. Отсутствует исходное изображение с ведомого регистратора.	Необходимо обратиться к разработчикам для анализа причины ошибки и выяснения особого формата данных материалов от регистратора.
56	ERROR	TargetInspector: Can't prepare report - slave frame is empty	Ошибка создания материалов. Отсутствует исходное изображение с ведомого регистратора.	Необходимо обратиться к разработчикам для анализа причины ошибки и выяснения особого формата данных материалов от регистратора.
57	ERROR	TargetInspector: Report builder hasn't been initialized correctly	Ошибка инициализации построителя материалов. Возможные основные причины: 1) Некорректный файл шаблона материалов; 2) Некорректный путь к файлу шаблона материалов.	1) Проверить или заменить файл шаблона; 2) Проверить путь к файлу шаблона и его название в настройках ("Вид коллажа").
58	ERROR	TargetInspector: Failed to prepare report context	Ошибка подготовки данных для построителя материалов.	см. графы 53, 54, 55, 56
59	ERROR	TargetInspector: Report builder failed to build report	Ошибка формирования файла материалов. Возможные основные причины: 1) Некорректный файл шаблона материалов; 2) Некорректный путь к файлу шаблона материалов.	1) Проверить или заменить файл шаблона; 2) Проверить путь к файлу шаблона и его название в настройках ("Вид коллажа").

№	Тип	Текст сообщения	Описание	Решение
60	WARNING	TargetInspector: Failed to compute master liplate collage position	Предупреждение о том, что не удалось вычислить координаты номерной пластины на изображении с ведущего регистратора на коллаже. В графическом материале не будет размещена область с ГРЗ от ведущего регистратора.	Необходимо обратиться к разработчикам для анализа причины ошибки и выяснения особого формата данных материалов от регистратора.
61	ERROR	TargetInspector: Failed to save report to disk.	Ошибка создания файлов материалов.	см. ошибки в табл. 8.9 ReportBuilder
62	INFO	TargetInspector: DB rotation started	Сигнализирует о начале ротации БД (удаление старых записей).	
63	INFO	TargetInspector: DB rotation finished	Сигнализирует о завершении ротации БД (удаление старых записей).	
64	ERROR	TargetInspector: db_write_target failed	Ошибка записи информации о нарушителе в БД системы в таблицу targets.	1) Проверить корректность полей в конфигурационном файле, отвечающих за работу с БД системы; 2) Проверить доступность БД, сервиса БД на устройстве с ПО.
65	ERROR	TargetInspector: db_write_event failed	Ошибка записи информации о нарушителе в БД системы в таблицу events.	1) Проверить корректность полей в конфигурационном файле, отвечающих за работу с БД системы; 2) Проверить доступность БД, сервиса БД на устройстве с ПО.
66	ERROR	TargetInspector: db_write_material failed	Ошибка записи информации о нарушителе в БД системы в таблицу materials.	1) Проверить корректность полей в конфигурационном файле, отвечающих за работу с БД системы; 2) Проверить доступность БД, сервиса БД на устройстве с ПО.
67	ERROR	TargetInspector: db_write_event_xml failed	Ошибка записи информации о нарушителе в БД системы в таблицу xml протокола samara.	1) Проверить корректность полей в конфигурационном файле, отвечающих за работу с БД системы; 2) Проверить доступность БД, сервиса БД на устройстве с ПО.
68	INFO	TargetInspector: Loading library 'avgspeed.dll' ...	Сигнализирует о начале процесса загрузки библиотеки метрологии.	
69	CRITICAL	TargetInspector: Failed to open library 'avgspeed.dll' for CRC-32 checking	Ошибка открытия файла библиотеки метрологии.	Проверить наличие файла библиотеки в директории

№	Тип	Текст сообщения	Описание	Решение
				с исполняемым файлом ПО.
70	INFO	TargetInspector: Calculated checksum for library 'avgspeed.dll'. CRC-32: 'BD60FE02 '	Показывает вычисленную контрольную сумму (CRC) сумму библиотеки метрологии.	
71	CRITICAL	TargetInspector: CRC-32 mismatch for library 'avgspeed.dll'! Calculated: ' BD60FE02', Reference: 'AA10AA20'	Ошибка - контрольная сумма библиотеки метрологии не совпадает с ожидаемой.	Заменить файл библиотеки на корректный.
72	CRITICAL	TargetInspector: Failed to load library 'avgspeed.dll ' ({} : {})	Ошибка загрузки библиотеки метрологии.	Проверить наличие файла библиотеки в директории с исполняемым файлом ПО. Заменить файл библиотеки на корректный.
73	CRITICAL	TargetInspector: Error while importing avgspeed methods: {}	Ошибка регистрации функций библиотеки метрологии.	Заменить файл библиотеки на корректный.
74	INFO	TargetInspector: Load library 'avgspeed.dll' OK	Сообщение об успешной загрузке библиотеки метрологии.	

Таблица 8.8 — Сообщения DeviceAdapter

№	Тип	Текст сообщения	Описание	Решение
1	ERROR	DeviceAdapter***: [Slave] connect database failed	Ошибка подключения к БД регистратора (Slave - ведомого, Master - ведущего).	1) Проверить параметры сетевого подключения к БД ведущего/ведомого регистратора в конфигурационном файле (IP, порт, название БД, имя пользователя, пароль); 2) Проверить сетевое соединение с ведущим/ведомым регистратором; 3) Проверить корректность типа ведущего/ведомого регистратора (skat/skatpp/krechet/oracul) в ВЕБ-интерфейсе или файле настроек.
2	ERROR	DeviceAdapter***: [Slave] get last event id query failed	Ошибка получения индекса последнего события из БД регистратора.	см. графу 1
3	ERROR	DeviceAdapter***: [Slave] get targets metadata query failed	Ошибка получения данных о нескольких ТС из БД регистратора.	см. графу 1

№	Тип	Текст сообщения	Описание	Решение
4	ERROR	DeviceAdapter***: [Slave] get target materials query failed	Ошибка получения данных о материалах на проезд ТС из БД регистратора.	см. графу 1
5	ERROR	DeviceAdapter***: [Slave] find target metadata query failed	Ошибка поиска данных о ТС в БД регистратора.	см. графу 1
6	ERROR	DeviceAdapter***: [Slave] get last event materials query failed	Ошибка получения данных материалов последнего события из БД регистратора.	см. графу 1
7	ERROR	DeviceAdapter***: [Slave] get last targets metadata query failed	Ошибка получения данных о нескольких последних ТС из БД регистратора.	см. графу 1

Таблица 8.9 — Сообщения ReportBuilder

№	Тип	Текст сообщения	Описание	Решение
1	ERROR	ReportBuilder: Error while opening template file C:/duet/templates/ru.xml	Ошибка открытия файла шаблона.	1) Проверить путь к файлу шаблона и его название в настройках ("Вид коллажа").
2	ERROR	1) ReportBuilder: Error while parsing xml structure: Start-end tags mismatch 2) ReportBuilder: Error while parsing xml structure: Error parsing end element tag	Ошибка разбора файла шаблона.	Проверить и исправить файл шаблона в соответствии с синтаксисом xml-формата.
3	ERROR	ReportBuilder: failed to parse [object_image] node	Ошибка разбора указанной сущности в файле шаблона.	Проверить и исправить файл шаблона в соответствии с синтаксисом xml-формата.
4	ERROR	ReportBuilder: no templates were loaded!	Ошибка загрузки шаблона - в xml файле нет ни одной сущности template.	Исправить файл шаблона.
5	ERROR	ReportBuilder: no default template with collage type and empty subtype found!	Ошибка загрузки шаблона - в xml файле ни одной сущности template с атрибутом type=collage не найдено (и без атрибута subtype).	Исправить файл шаблона - добавить сущность template с типом collage.
6	ERROR	ReportBuilder: attribute 'type' required	Ошибка загрузки шаблона - в xml файле содержится сущность template без обязательного атрибута type.	Проверить и добавить во все сущности template атрибут type (collage / web).
7	ERROR	ReportBuilder: unknown 'type' attribute [colge]	Ошибка загрузки шаблона - в xml файле содержится сущность template с атрибутом type, значение которого не соответствует разрешенным (допустимы: collage, web).	Исправить файл шаблона - заменить значение атрибута type на одно из значений: collage , web.

№	Тип	Текст сообщения	Описание	Решение
8	ERROR	ReportBuilder: attribute 'subtype' has unknown subtype [C1_3]	Ошибка загрузки шаблона - в xml файле содержится сущность template с атрибутом subtype, значение которого не соответствует разрешенным (допустимо: C1_2).	Исправить файл шаблона - заменить значение атрибута subtype на одно из используемых значений (C1_2).
9	ERROR	ReportBuilder: 'canvas' node 'width' or 'height' attributes were not found	Ошибка разбора файла шаблона - у сущности canvas не найдены атрибуты width и height.	Исправить файл шаблона.
10	ERROR	ReportBuilder: canvas area is invalid, w=0, h=200	Ошибка разбора файла шаблона - у сущности canvas один из атрибутов (или оба) width и height равен нулю.	Исправить файл шаблона.
11	ERROR	ReportBuilder: invalid id [C1_3_text]	Ошибка разбора файла шаблона - у сущности string атрибут id имеет некорректное значение (допустимо: C1_2_text).	Исправить файл шаблона.
12	ERROR	ReportBuilder: frameType 'image_fr_labeled_D3' with hash '100435732' is invalid	Ошибка разбора файла шаблона - у сущности object_image source некорректный атрибут type. Допустимые type: - image_fr_D1 - image_fr_D2 - image_fr_master - image_fr_slave - image_fr_labeled_D1 - image_fr_labeled_D2 - image_fr_labeled_master - image_fr_labeled_slave - image_grz_D1 - image_grz_D2 - image_grz_master - image_grz_slave - image_grz_roi_D1 - image_grz_roi_D2 - image_grz_roi_master - image_grz_roi_slave	Исправить файл шаблона.
13	ERROR	ReportBuilder: 'destination' node 'x', 'y', 'width' or 'height' attributes were not found	Ошибка разбора файла шаблона - у сущности destination некорректные координаты.	Исправить файл шаблона.
14	ERROR	ReportBuilder: destination of node 'image_fr_labeled_D1' x0=10,y0=10,x1=900,y1=700 is out of canvas borders 800,600	Ошибка разбора файла шаблона - у указанной сущности destination координаты за пределами изображения.	Исправить файл шаблона.
15	ERROR	ReportBuilder: font 'verdana' loading failed	Ошибка разбора файла шаблона - ошибка загрузки	Проверить название шрифта в файле шаблона и наличие

№	Тип	Текст сообщения	Описание	Решение
			шрифта для сущности object_text.	соответствующего файла шрифта в каталоге ПО.
16	ERROR	ReportBuilder: 'text' node 'size' attribute not found, required	Ошибка разбора файла шаблона - у сущности object_text отсутствует обязательный атрибут size.	Исправить файл шаблона.

Таблица 8.10 — Сообщения FileSaver

№	Тип	Текст сообщения	Описание	Решение
1	ERROR	FileSaver: writeFile: file directory not exists. Must be created before.	Ошибка сохранения файла материала на диск.	1) Возможно, в конфигурационном файле указан некорректный путь к архиву материалов в поле **result_directory**. Проверить, изменить; 2) Проблемы с файловой системой на диске.
2	ERROR	FileSaver: Failed to writeFile 'D:/arch/duet/2020/11/19/04_21_28_114/duet_2020_11_19_04_21_24.jpg'	Ошибка сохранения указанного файла материала на диск.	1) Возможно, в конфигурационном файле указан некорректный путь к архиву материалов в поле **result_directory**. Проверить, изменить; 2) Проблемы с файловой системой на диске.
3	ERROR	FileSaver: sign is not inited! Sign won't be saved.	Ошибка инициализации модуля цифровых подписей, цифровая подпись не может быть создана.	Проверить наличие файла ключа для цифровых подписей, а также соответствующую ему настройку типа цифровой подписи в настройках ПО. 1) GOST: - privolvia.sxp 2) RSA: - privolvia.pem
4	ERROR	FileSaver: error while saving sign file D:/arch/duet/2020/11/19/04_21_28_114/duet_2020_11_19_04_21_24.sign'	Ошибка сохранения файла цифровой подписи материала на диск.	Возможно, в конфигурационном файле указан некорректный путь в полях **result_directory**. Проверить, изменить.
5	CRITICAL	FileSaver: Error while composeFilename	Невозможно создать уникальное имя для файла материала. Подобная проблема может возникнуть, когда файлов с одинаковым именем в одном каталоге больше 99. Имя файла формируется из даты, времени и типа нарушения, которые могут	Возможно, от регистратора приходит одинаковое время и тип события для всех проездов. Проблема в материалах регистратора.

№	Тип	Текст сообщения	Описание	Решение
			оказаться одинаковы для нескольких одновременно проезжающих ТС.	
6	CRITICAL	FileSaver: Error creation unique filename, duplicated (index=99): '1908013_1908007_2020_12_12_20_55_30_05_99_00'	см. графу 5	см. графу 5
7	ERROR	FileSaver: Error while generateAndSaveSamaraAddMats	Ошибка записи на диск файлов материалов по xml протоколу samara.	1) Возможно, в конфигурационном файле указан некорректный путь к архиву материалов в поле **result_directory**. Проверить, изменить; 2) Проблемы с файловой системой на диске.
8	ERROR	FileSaver: Error while writeFileAndSign	Ошибка записи на диск файлов материалов.	см. графы 2, 3
9	ERROR	FileSaver: Unsupported textfile format {1}. Not implemented yet	Ошибка создания текстового файла материала. В настройках указан неподдерживаемый формат текстовых файлов. Доступен только формат xml.	В ВЕБ-интерфейсе выбрать "Формат текстового файла" XML (в конфигурационном файле в соответствующем поле **resultfile_textfile_format** поставить значение xml).
10	ERROR	FileSaver: Error while writeFileAndSign slave	Ошибка записи на диск изображения проезда с ведомого регистратора (для протокола Самары).	см. графы 2, 3
11	ERROR	FileSaver: Error while writeFileAndSign master	Ошибка записи на диск изображения проезда с ведущего регистратора (для протокола Самары).	см. графы 2, 3
12	ERROR	FileSaver: Error while writeFileAndSign GRZ	Ошибка записи на диск изображения ГРЗ с ведущего регистратора (для xml протокола samara).	1) См. графы 2, 3 2) В файле "*_fr.jpg" с регистратора нет информации о координатах номерной пластины на изображении или файлы материалов некорректные. Необходимо выяснение проблемы формата материалов на регистраторе.
13	ERROR	FileSaver: Gcrypt_wrapper::init_privkey failed	Ошибка инициализации цифровой подписи.	см. графу 3

Таблица 8.11 — Сообщения WebServer

№	Тип	Текст сообщения	Описание	Решение
1	INFO	WebServer: create	Начало инициализации ВЕБ-сервера	
2	INFO	WebServer: WebServer logger inited!	Уведомление об успешной инициализации протоколирования ВЕБ-сервера.	
3	DEBUG	WebServer: Log path: ./logs/web_server.log	Уведомление о путях к файлам протоколов ВЕБ-сервера.	
4	INFO	WebServer: Using config file C:/duet/config/web_server.ini	Уведомление об успешной загрузке файла настроек ВЕБ-сервера web_server.ini.	
5	WARNING	WebServer: C:\duet\config\web_server.ini not found	Ошибка загрузки файла настроек ВЕБ-вервера. Файл не найден.	Проверьте наличие файла настроек ВЕБ-интерфейса .\config\web_server.ini (например, C:\duet\config\web_server.ini). При сделанных вручную изменениях в файле проверьте корректность структуры файла или восстановите настройки исходным файлом из дистрибутива.
6	ERROR	WebServer: Cannot find WebServer config file	Ошибка загрузки файла настроек ВЕБ-вервера.	см. графу 5

ПРИЛОЖЕНИЕ А

Описание веб-протоколов

Версия ПО: 1.3.4+

В данном приложении приводится API веб-протоколов взаимодействия, используемых в программном обеспечении «ДУЭТ», «РИФ» и «ОРАКУЛ-КУД»:

- JSON-RPC по HTTP — основной протокол API взаимодействия;
- HTTP — протокол для файлов ВЕБ интерфейса и файлов материалов.

А1. Веб-сервер и протоколы

Обработку запросов по всем протоколам производит встроенный в ПО веб-сервер.

Параметры подключения:

- Адрес: **http://<адрес устройства>**
- Порт: **11000**
- Точки доступа для протоколов (см. табл. 1.1).

Таблица А1.1 — Точки доступа протоколов

/	HTTP GET/POST запросы файлов ВЕБ-интерфейса (html/js/css и др.). Примеры URL: http://192.168.0.100:11000/ http://192.168.0.100:11000/index.html
/jrpc	JSON-RPC запросы по HTTP. Пример URL: http://192.168.0.100:11000/jrpc
/files	HTTP GET/POST запросы файлов материалов нарушений. Пример URL: http://192.168.0.100:11000/files
/get*	HTTP GET/POST запросы XML результатов по HTTP. (только при включенном протоколе samara: result_protocol=samara) Пример URL: http://192.168.0.100:11000/getstatus

Адрес устройства может быть IP адресом или доменным именем.

Точка доступа запросов веб-интерфейса ("/") используется непосредственно для работы в браузере пользователем. Поддерживаются современные браузеры: Google Chrome, Mozilla Firefox, Opera, Safari и другие. Версии не ранее 2019 г. выпуска.

Остальные точки доступа предоставляют специальные API для взаимодействия с ПО со специальным телом запроса. Они описаны далее.

Конфигурация веб-сервера ПО находится в файле: **web_server.ini**

Стандартное расположение файла при установке:

- Windows: ./config/web_server.ini (относительно папки ПО)
- Linux: /etc/duet/config/web_server.ini

ВНИМАНИЕ! Изменение настроек в файле конфигурации может привести к неработоспособности веб-интерфейса или ПО в целом!

Изменение конфигурации обычно не требуется, но может быть использовано для настройки адреса и порта веб-сервера, а также тонкой настройки задержек, кэширования и др.

Изменение порта веб-сервера в **web_server.ini**:

```
[listener]
port=11000
```

При изменении порта в настройках web_server.ini требуется синхронно изменить порт в параметрах запуска программы **supervisor**, например, изменение на порт 9999 (по умолчанию 11000):

- Файл duet_start.bat:

```
@echo off
start duet_supervisor --target duet_core.exe -p 9999
```

- Файл duet_start_standalone.bat

```
@echo off
start duet_supervisor --target duet_core.exe -p 9999 -- -m s
```

A2. JSON-RPC API

Точка доступа: **http://<адрес устройства>:11000/jrpc**

Протокол JSON-RPC с использованием транспорта HTTP используется для запросов управления и мониторинга ПО. Является основным протоколам для взаимодействия веб-интерфейса с основным процессом ПО.

Протокол соответствует базовой спецификации: JSON-RPC 2.0.

Ссылка: https://www.jsonrpc.org/specification

A2.1. Основы JSON-RPC

В данном подразделе приведены базовые принципы JSON-RPC.

Request object

Объект запроса отправляется клиентом на сервер.

Формат запроса: JSON объект.

Поля запроса см. в табл.A2.1.

Таблица A2.1 — Поля запроса JSON-RPC

jsonrpc	Строка версии JSON-RPC протокола. Всегда "2.0".
method	Строка, содержащая имя метода

params	Строка, содержащая параметры вызова метода в виде структуры (массив, объект) или одиночного значения. Опциональное поле.
id	Идентификатор запроса. Задаётся клиентом, обычно строка/число, возрастающее значение, позволяющее идентифицировать запрос. Не должно быть null. веб-сервер ПО возвращает значение в поле id назад в ответе.

Response object

На RPC запрос сервер посылает RPC Ответ (Response), за исключением запросов-уведомлений (Notifications). Уведомительные запросы не используются в данном ПО.

Формат ответа: JSON объект.

Поля ответа см. в табл.А2.2.

Таблица А2.2 — Поля ответа JSON-RPC

jsonrpc	Строка версии JSON-RPC протокола. Всегда "2.0".
result	Содержит ответ. Тип значения зависит от конкретного метода запроса. Обязательно присутствует в ответе, за исключением ответов с описанием ошибки.
error	Содержит информацию об ошибке. Используется только в ответах с возвращением ошибки. Тип значения — объект.
id	Идентификатор. Значение, соответствует значению в запросе. При ошибке формата поля id в запросе, в ответе выставляется в null.

Далее введены вспомогательные обозначения:

- "-->" — запрос;
- "<--" — ответ.

Пример JSON-RPC запрос/ответ:

--> {"jsonrpc": "2.0", "method": "sub", "params": [42, 23], "id": 1}

<-- {"jsonrpc": "2.0", "result": 19, "id": 1}

Пример JSON-RPC запрос/ответ с именованными параметрами:

--> {"jsonrpc": "2.0", "method": "sub", "params": {"p1": 23, "p2": 42}, "id": 3}

<-- {"jsonrpc": "2.0", "result": 19, "id": 3}

ВНИМАНИЕ! Массивы запросов (Batch requests) не поддерживаются!

HTTP transport

Содержимое JSON-RPC запроса/ответа вкладывается в HTTP запрос/ответ. Используется кодировка UTF-8 для текста запроса и ответа.

Примерный вид основных HTTP заголовков для передачи запроса JSON-RPC:

Host: 127.0.0.1:11000

Accept: application/json

Content-Type: application/json; charset=utf-8

Content-Length: 148

...

После заголовков следует тело JSON-RPC запроса.

Примерный вид основных HTTP заголовков для передачи ответа JSON-RPC:

HTTP/1.1 200 OK

Server: DUET_SERVER JSON-RPC 2.0

Date: Mon, 25 May 2020 10:56:54 GMT

Access-Control-Allow-Origin: http://192.168.0.4

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Connection: keep-alive

Content-Length: 159

Content-Type: application/json; charset=utf-8

Keep-Alive: timeout=100, max=20000

Pragma: no-cache

Set-Cookie: session_id=M73QFA55L95dNyH63laxPT2kg==; Comment=Identifies the user; Max-Age=900;

...

После заголовков следует тело JSON-RPC ответа.

При успешном выполнении выставляется код HTTP ответа 200 (успех) и вкладывается корректное содержимое JSON-RPC ответа.

А2.2. Ошибки в JSON-RPC и HTTP

В реализованном протоколе взаимодействия JSON-RPC по HTTP существует два уровня ошибок:

1. Уровень ошибок HTTP протокола.

Выставляется код HTTP ответа с кодом ошибки и, опционально, содержимое JSON-RPC ответа с ошибкой;

2. Уровень ошибок JSON-RPC протокола.

При этом выставляется код HTTP ответа 200 (успех), но содержимое JSON-RPC ответа содержит данные об ошибке.

Если при выполнении RPC запроса произошла ошибка, в ответе выставляется поле error с JSON объектом с описанием ошибки.

Поля объекта error приведены в таблице А2.3.

Таблица А2.3 — Поля объекта error ответа JSON-RPC

code	Число, идентифицирующее тип ошибки.
message	Строка, содержащая краткое описание ошибки.
data	Простое значение или массив/объект с подробной информацией об ошибки. Может отсутствовать.

Коды ошибок (code) от -32768 до -32000 соответствуют предопределенным базовым ошибкам в протоколе (см. табл. А2.4).

Таблица А2.4 — Стандартные коды ошибок JSON-RPC

code	**message**	**Описание**
-32700	Parse error	Неверный формат JSON запроса был принят сервером. Ошибка возникает при разборе JSON сообщения.
-32600	Invalid Request	Отправленный JSON является неверным объектом запроса.
-32601	Method not found	Метод, указанный в JSON запросе, неизвестен или недоступен.
-32602	Invalid params	Неверные параметры запроса.
-32603	Internal error	Внутренняя системная ошибка JSON-RPC сервера.
от -32000 до -32099	Server error	Зарезервированные в протоколе коды ошибок. Не используются в ПО.

Пример JSON-RPC запроса и ответ с ошибкой несуществующего метода:

--> {"jsonrpc": "2.0", "method": "foobar", "id": 1}

<-- {"jsonrpc": "2.0", "error": {"code": -32601, "message": "Method not found"},
"id": 1}

Пример JSON-RPC запроса и ответ с ошибкой некорректного формата JSON объекта запроса:

--> {"jsonrpc": "2.0", "method": "foobar, "params": "bar", "baz]

<-- {"jsonrpc": "2.0", "error": {"code": -32700, "message": "Parse error"}, "id": null}

Пример JSON-RPC запроса и ответ с ошибкой некорректного JSON запроса:

--> {"jsonrpc": "2.0", "method": 1, "params": "bar"}

<-- {"jsonrpc": "2.0", "error": {"code": -32600, "message": "Invalid Request"},
"id": null}

Пример JSON-RPC запроса и ответ с ошибкой на неподдерживаемые batch запросы:

--> [
{"jsonrpc": "2.0", "method": "sum", "params": [1,2,4], "id": 1},
{"jsonrpc": "2.0", "method"
]

<-- {"jsonrpc": "2.0", "error": {"code": -32700, "message": "Parse error"}, "id": null}

Дополнительные ошибки определены самим ПО (см. табл. А2.5).

Таблица А2.5 — Дополнительные коды ошибок JSON-RPC

code	message	Описание
-31001	No rights to execute this method	Нет прав пользователя на выполнение метода запроса. Требуется выполнение от имени привилегированного пользователя или пользователя с определенной ролью.
-31000	Method internal error: TEXT	Внутренняя ошибка выполнения метода. Опционально присутствует описание ошибки (TEXT).
-30004	Not found	Не найдено. Для запросов, возвращающих информацию о какой-либо сущности, которая не найдена.
-30003	Access denied	Доступ запрещен. Запрещено выполнение операции, вызванной методом в данных условиях.
-30001	Already exists	Создаваемая сущность уже существует.
-30000	Method returns false	Метод вернул ошибку. Произошла внутренняя ошибка при выполнении метода.

Пример JSON-RPC запроса и ответ с ошибкой на ненайденные данные о нарушителе:

--> {"jsonrpc": "2.0", "method": "get_target_data", "params": {"id": 999999}, "id": 4}

<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 4}

Используемые коды ответов HTTP на ошибки приведены в табл. А2.6:

Таблица А2.6 — Коды ответов статуса HTTP

HTTP code	message	Описание
401	Unauthorized	Неавторизованный запрос. Например: - некорректная сессия (не существует, истекло время сессии); - запрос выполнения защищенных методов без указания session_id (идентификатора сессии).
403	Forbidden	Запрещено. Например: - верная сессия, но недостаточно прав пользователя на выполнение метода JSON-RPC, дополнительно содержит тело JSON-RPC ответа с ошибкой "No Rights" (-31001).
400	Bad Request	Некорректный запрос. Например: - неверный формат полей запроса JSON-RPC (id, поля params метода и др. ошибки разбора запроса); - неизвестный метод в запросе JSON-RPC, дополнительно содержит тело JSON-RPC ответа с ошибкой "Method not found" (-32601)
500	Internal Server Error	Системная ошибка. При выполнении запроса возникли проблемы в ПО.

А2.3. Аутентификация через JSON-RPC

Аутентификация пользователя производится через специальные JSON-RPC запросы. Она необходима, в том числе, для работы в веб-интерфейсе и выполняется веб-приложением при вводе данных (имени и пароля) пользователем на странице «Вход».

Аутентификация реализована через получение идентификатора сессии через специальное значение cookie в HTTP заголовке и его использование при дальнейших запросах. Некоторые JSON-RPC запросы позволяют работать без идентификатора сессии, т.н. **открытые**(public) запросы, другие запросы — **закрытые**(private) — требуют отправки актуального идентификатора сессии.

Значения заголовков HTTP при JSON-RPC запросе входа, выполнении действий и выходе пользователя:

login — получить идентификатор сессии, ответ:

<-- HTTP response header:

Set-Cookie: session_id=N3QF5L95dNyH63laxPT2qg==;

some_method — какой-то метод с использованием идентификатора сессии, запрос:

--> HTTP request header:

Cookie: session_id=N3QF5L95dNyH63laxPT2qg==;

logout — выйти, отменить идентификатор сессии, запрос:

--> HTTP request header:

Cookie: session_id=N3QF5L95dNyH63laxPT2qg==;

Подробное описание методов JSON-RPC для аутентификации приведено далее.

А2.4. Пример работы по протоколу JSON-RPC

Можно использовать для проверки запросов в Windows и Linux консольную утилиту **curl** (https://curl.haxx.se/download.html). В примерах использован локальный адрес 127.0.0.1:11000.

1.Аутентификация login() и получение session_id:

```
curl -i -v -H "Accept: application/json" -H "Content-Type: application/json; charset=utf-8" --data
"@login.json" http://127.0.0.1:11000/jrpc

HTTP/1.1 200 OK

Set-Cookie: session_id=79nCKyHu3uqXOMce91CO5A==; Comment=Identifies the user; Max-
Age=900; Path=/; HttpOnly; Version=1
```

Тело запроса JSON-RPC — login.json (UTF-8):

```
{"jsonrpc": "2.0", "method": "login", "params": {
   "username": "Администратор",
   "password": "346aa765d6174a2626cf99"
}, "id": 3}
```

Тело ответа JSON-RPC:

```
{
   "id": 3,
   "jsonrpc": "2.0",
   "result": {
      "fullname": "",
      "rights": "admin",
      "session_id": "79nCKyHu3uqXOMce91CO5A==",
      "username": "Администратор"
   }
```

```
}
```

Запрос успешно выполнен (HTTP/1.1 200 OK), получено значение session_id=79nCKyHu3uqXOMce91CO5A== в HTTP заголовке ответа и в теле запроса.

2. Открытый запрос get_ver() без session_id:

```
curl -i -v -H "Accept: application/json" -H "Content-Type: application/json; charset=utf-8" --data
"@get_ver.json" http://127.0.0.1:11000/jrpc
```

Тело запроса JSON-RPC — get_ver.json (UTF-8):

```
{"jsonrpc": "2.0", "method": "get_ver", "params": {}, "id": 4}
```

Тело ответа JSON-RPC:

```
{
    "id": 4,
    "jsonrpc": "2.0",
    "result": {
        "build": "",
        "company": "Olvia",
        "copyright": "Copyright © Olvia 2020",
        "crc": "BD60FE02",
        "product": "Duet",
        "version": "1.1.2",
        "version_full": "1.1.2"
    }
}
```

3. Закрытый запрос get_state() с session_id:

```
curl -i -v -H "Accept: application/json" -H "Content-Type: application/json; charset=utf-8" --
cookie "session_id=79nCKyHu3uqXOMce91CO5A==" --data "@get_state.json"
http://127.0.0.1:11000/jrpc
```

Тело запроса JSON-RPC — get_state.json (UTF-8):

```
{"jsonrpc":"2.0", "method": "get_state", "id": 5}
```

Тело ответа JSON-RPC:

```
{
    "id": 5,
```

"jsonrpc": "2.0",
"result": {
"fixation_count": 0,
"inspection": false,
"last_pushed_id": 0,
"mode": "setup",
"paired": false,
"running": false,
"uptime": "00:29:27.687718",
"used_percent": 0.12993839103728533,
"used_size": 26.611382484436035
}
}

А2.5. Методы JSON-RPC

Полный список JSON-RPC методов:

- login
- logout
- get_state
- set_params
- get_params
- set_mode
- get_ver
- get_serial_number
- get_target_data
- get_db_size
- get_db_time_minmax
- get_event_item
- get_events_list_info
- get_events_list
- get_event_info
- get_target_info
- get_users
- get_user
- add_user
- modify_user

- remove_user
- get_users_actions
- get_users_actions_count
- get_report_templates_list

А2.5.1. Метод login

Аутентификация пользователя и создание новой сессии. Всегда создаёт новую сессию пользователя и возвращает новое значение session_id. Если в Cookie передано значение текущей сессии (session_id), старая сессия будет удалена.

Доступ: открытый.

Параметры запроса:

- username — имя пользователя, string;
- password — MD5 хэш от пароля пользователя, string.

Результат запроса — объект данных пользователя:

- username — имя пользователя, string;
- fullname — полное имя пользователя, string (если задано, иначе — пусто);
- rights — уровень прав пользователя (root, admin, verify, setup, info, journal).

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "login", "params": {
        "username": "Администратор",
        "password": "000bd17e96000aadafa84a2a7b46ddb"
    }, "id": 3}
<-- {"jsonrpc": "2.0", "result": {
        "username": "admin",
        "fullname": "bla bla bla",
        "rights": "info"
    }, "id": 3}
```

Заголовок HTTP ответа:

```
Set-Cookie: session_id=N3QF5L95dNyH63laxPT2qg==; Comment=Identifies the user; Max-Age=900; Path=/; HttpOnly; Version=1
```

Пример запрос/ответ — ошибка:

- неверный пользователь/пароль, ошибка аутентификации:

```
--> {"jsonrpc": "2.0", "method": "login", "params": {
        "username": "Поверка",
        "password": "0031bd17e96000aadafa00a2a7b46ddb"
    }, "id": 3}
```

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 3}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "login", "params": {"user": "Поверка"}, "id": 3}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 3}

- внутренняя ошибка:

--> {"jsonrpc": "2.0", "method": "login", "params": {"user": "Поверка"}, "id": 3}
<-- {"jsonrpc": "2.0", "error": {"code": -32603, "message": "Internal error"}, "id": 3}

Имена пользователей по умолчанию:

- Администратор;
- Поверка;
- Настройка;
- Оператор;
- Журнал.

А2.5.2. Метод logout

Завершение текущей сессии пользователя, удаление текущей сессии. Завершается сессия переданная в session_id в cookie.

Доступ: закрытый.

Параметры запроса: нет.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "logout", "id": 4}
<-- {"jsonrpc": "2.0", "result": true, "id": 4}

А2.5.3. Метод get_state

Получение состояние ПО.

Доступ: закрытый.

Параметры запроса: нет.

Результат запроса — объект данных состояния:

- uptime — время работы в формате "hh:mm:ss.nnnnnn" (до микросекунд), string;
- fixation_count — число фиксаций с начала работы ПО, uint64;
- last_pushed_id — последний id нарушителя, uint64 (null, если нет материалов в БД);
- used_size — размер архива файлов материалов, float (MB);
- used_percent — процент заполненности архива файлов материалов, float (%);
- paired — наличие связи с регистраторами, bool (true/false — есть/нет);
- running — стартовала ли инспекция, но может быть в подготовке, bool (true/false);
- inspection — инспектирование нарушений работает, bool (true/false);
- mode — режим работы, string (work — контроль, setup — настройка).

Пример запрос/ответ — успех:

--> {"jsonrpc":"2.0", "method": "get_state", "id": 5}

<-- {"jsonrpc":"2.0", "result": {

"uptime": "07:55:36.3578987",

"fixation_count": 4226,

"last_pushed_id": 180562,

"used_size": 30437.19,

"used_percent": 50.73,

"paired": true,

"running": true,

"inspection": true,

"mode": "work"

}, "id":5}

А2.5.4. Метод set_params

Настройка параметров. Необходимо устанавливать структуру параметров в соответствии со структурой в config.json. Можно указывать только изменяемые параметры, но если изменяется параметр второго уровня, то требуется полностью задать все соседние параметры на этом уровне.

Доступ: закрытый.

Параметры запроса: объект с параметрами в виде ключ-значение.

Результат запроса — true/false.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "set_params", "params": {"parameters_db_name": "servicedb", "place_in_city": true}, "id": 1}

<-- {"jsonrpc": "2.0", "result": true, "id": 1}

- пропуск неизвестных параметров:

--> {"jsonrpc": "2.0", "method": "set_params", "params": {"par": "service"}, "id": 1}

<-- {"jsonrpc": "2.0", "result": true, "id": 1}

- пропуск пустого набора параметров:

--> {"jsonrpc": "2.0", "method": "set_params", "params": {}, "id": 1}

<-- {"jsonrpc": "2.0", "result": true, "id": 1}

Пример запрос/ответ — ошибка:

- неверный формат параметров, объект ключ-значение:

--> {"jsonrpc": "2.0", "method": "set_params", "params": "hello", "id": 1}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 1}

- внутренняя ошибка:

--> {"jsonrpc": "2.0", "method": "set_params", "params": {"master": {"abc" : "123"}}, "id": 1}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 1}

А2.5.5. Метод get_params

Получение текущих параметров. Можно запросить определенный набор параметров верхнего уровня, параметры нижнего уровня возвращаются для узла верхнего в полном составе.

Доступ: закрытый.

Параметры запроса: объект с ключами — требуемыми параметрами (значения — null) или пустой объект для получения всех параметров.

Результат запроса — объект с запрошенными или всеми параметрами.

К параметрам config.json возвращаются дополнительные параметры ПО, например конфигурация модуль/сервер (deploy_mode).

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_params", "params": {}, "id": 4}
<-- {"jsonrpc": "2.0", "result": {
"complex_default_mode": "setup",
...
"master": {
"fixations_dir": "D:/arch/all_fixations",
...
"type": "skat"
},
...
}, "id": 4}

--> {"jsonrpc": "2.0", "method": "get_params", "params": {"parameters_db_name": null, "master": null}, "id": 5}
<-- {"jsonrpc": "2.0", "result": {"parameters_db_name": "duet_service", "master": {"fixations_dir": "D:/arch/all_fixations", ...}}, "id": 5}

Пример запрос/ответ — ошибка:

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "get_params", "params": {"par": "service"}, "id": 6}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 6}

А2.5.6. Метод set_mode

Установка режима работы. Старт/стоп фиксации.

Доступ: закрытый.

Параметры запроса:

- mode — требуемый режим работы, string (work — контроль, setup — настройка).

Результат запроса — true/false.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "set_mode", "params": {"mode": "setup"}, "id": 4}

<-- {"jsonrpc": "2.0", "result": true, "id": 4}

Пример запрос/ответ — ошибка:

- не удалось переключить режим ПО:

--> {"jsonrpc": "2.0", "method": "set_mode", "params": {"mode": "work"}, "id": 9}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 9}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "set_mode", "params": {"m": "service"}, "id": 9}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 9}

А2.5.7. Метод get_ver

Получение данных о версии ПО.

Доступ: открытый.

Параметры запроса: нет.

Результат запроса — объект с информацией о ПО:

- build — номер сборки, не используется, см. version, string;
- company — название компании производителя, string;
- copyright — правообладатель, string;
- crc — контрольная сумма метрологически значимой части ПО, string;
- product — наименование ПО, string;
- version — версия ПО, string;
- version_full — полная версия ПО с номером сборки, string.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_ver", "id": 4}

<-- {"jsonrpc": "2.0", "result": {
"build": "",
"company": "Olvia",
"copyright": "Copyright © Olvia 2020",
"crc": "BD60FE02",
"product": "Duet",
"version": "1.1.2",
"version_full": "1.1.2-15-g64830"
}, "id": 4}

А2.5.8. Метод get_serial_number

Получение серийного номера регистраторов и самой системы.

Доступ: открытый.

Параметры запроса — массив значений (строк), варианты:

1) [] — запрос серийного номера системы;
2) [1] — запрос серийного номера ведущего регистратора;
3) [2] — запрос серийного номера ведомого регистратора;
4) [1, 2] — запрос серийных номеров обоих регистраторов.

Результат запроса — массив с результатами, варианты:

1) элемент, серийный номер системы, string;
2) элемент, серийный номер ведущего регистратора, string;
3) элемент, серийный номер ведомого регистратора, string;
4) два элемента, серийные номера обоих регистраторов, string.

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "get_serial_number", "params": [1], "id": 4}
<-- {"jsonrpc": "2.0", "result": ["121231"], "id": 4}

--> {"jsonrpc": "2.0", "method": "get_serial_number", "params": [1,2], "id": 4}
<-- {"jsonrpc": "2.0", "result": ["121231","145411"], "id": 4}

--> {"jsonrpc": "2.0", "method": "get_serial_number", "params": [], "id": 4}
<-- {"jsonrpc": "2.0", "result": ["121231/145411"], "id": 4}
```

Пример запрос/ответ — ошибка:

- неверный набор параметров:

```
--> {"jsonrpc": "2.0", "method": "get_serial_number", "params": [3], "id": 5}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 5}
```

А2.5.9. Метод get_target_data

Получение основной информации о нарушителе из БД.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- id — id нарушителя, uint64.

Результат запроса — объект со значениями:

- time — время фиксации нарушения в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- speed — средняя скорость (км/ч), float;
- distance — пройденное ТС расстояние (м), float;
- liplate — номерная пластина (ГРЗ), string;
- event_id — id события, uint64;

- target_type — идентификатор типа ТС (не используется — "0"), string.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_target_data", "params": {"id": 107192}, "id": 4}

<-- {"jsonrpc": "2.0", "result": {
"time": "2020-04-15 13:35:28.848000",
"speed": 51.0,
"distance": 270.4728,
"liplate": a993pp|178",
"event_id": 328627,
"target_type": "0"
}, "id": 4}

Пример запрос/ответ — ошибка:

- не найден:

--> {"jsonrpc": "2.0", "method": "get_target_data", "params": {"id": 9999999}, "id": 4}

<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 4}

A2.5.10. Метод get_db_size

Получение размера БД материалов — кол-во записей таблицы материалов (самая быстро наполняемая таблица).

Доступ: закрытый.

Параметры запроса: нет.

Результат запроса — число записей, uint64.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_db_size", "id": 4}

<-- {"jsonrpc": "2.0", "result": 250074, "id": 4}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_db_size", "id": 4}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 4}

A2.5.11. Метод get_db_time_minmax

Получение временного диапазона хранимых материалов в БД.

Доступ: закрытый.

Параметры запроса: нет.

Результат запроса — объект со значениями:

- min — минимальное время фиксации нарушения в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string (null при отсутствии);

- max — максимальное время фиксации нарушения в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string (null при отсутствии).

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "get_db_time_minmax", "id": 6}
<-- {"jsonrpc": "2.0", "result": {
        "min": "2020-03-24 15:15:13.091000",
        "max": "2020-04-23 12:55:53.649000"
    }, "id": 6}
```

- нет записей в БД:

```
--> {"jsonrpc": "2.0", "method": "get_db_time_minmax", "id": 6}
<-- {"jsonrpc": "2.0", "result": {
        "min": null,
        "max": null
    }, "id": 6}
```

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

```
--> {"jsonrpc": "2.0", "method": "get_db_time_minmax", "id": 6}
<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 6}
```

A2.5.12. Метод get_event_item

Получение данных события нарушения из БД по id события.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- id — id события нарушения, uint64.

Результат запроса — объект со значениями:

- id — id события нарушения, uint64;
- timestamp — время фиксации нарушения в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_id — id нарушителя, uint64;
- time — время записи события нарушения в БД в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- type — тип события нарушения ТС, string:
  - "1.1" — нарушение мгновенной скорости (только в режиме учёта нарушений С1.1 с регистраторов вместо С1.2);
  - "1.2" — нарушение средней скорости.
- target_mdspeed — средняя скорость (км/ч), float;
- target_mdistance — расстояние до ТС от ведущего регистратора (м), float;
- target_sdistance— расстояние до ТС от ведомого регистратора (м), float;

- target_mtime — время фиксации нарушения на ведущем регистраторе в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_stime — время фиксации нарушения на ведомом регистраторе в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_type — идентификатор типа ТС (не используется — "0"), string;
- liplate_data — номерная пластина (ГРЗ), string;
- target_distance — расстояние, пройденное ТС (м), float;

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_event_item", "params": {"id": 328709}, "id": 12}

<-- {"jsonrpc": "2.0", "result": {
"id": 328709,
"timestamp":"2020-03-18 10:45:30.105000",
"target_id": 149154,
"time":"2020-03-18 10:45:24.149000",
"type":"0",
"target_mdspeed": 59.0,
"target_mdistance": 32.0,
"target_sdistance": 12.2,
"target_mtime":"2020-03-18 10:45:24.149000",
"target_stime":"2020-03-18 10:45:31.111000",
"target_type":"0",
"liplate_data":"к689ау|799",
"target_distance": 78.0,
...
}, "id": 12}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_event_item", "params": {"id": 328709}, "id": 9}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 9}

- не найден:

--> {"jsonrpc": "2.0", "method": "get_event_item", "params": {"id": 999999999}, "id": 9}

<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 9}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "get_event_item", "params": {}, "id": 9}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 9}

А2.5.13. Метод get_events_list_info

Получение информации о списке нарушений из БД с учетом заданных правил фильтрации.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- filter — объект фильтруемых полей с правилами фильтрации, ключ — имя поля в БД событий, значение — объект правила фильтрации.

Доступные правила фильтрации аналогичны указанным в запросе **get_events_list**.

Результат запроса — объект со значениями:

- rows_count — количество записей, uint64.

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "get_events_list_info", "params": {
        "filter": {
            "liplate_data": {
                "like": "*178*"
            }
        }
    }, "id": 9}
<-- {"jsonrpc": "2.0", "result": {"rows_count":"125037"}, "id": 9}
```

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

```
--> {"jsonrpc": "2.0", "method": "get_events_list_info", "params": {
        "filter": {
            "liplate_data": {
                "like": "*178*"
            }
        }
    }, "id": 9}
<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 9}
```

- неверный набор параметров, ГРЗ не может быть массивом:

```
--> {"jsonrpc": "2.0", "method": "get_events_list_info", "params": {
        "filter": {
            "liplate_data": []
        }
    }, "id": 9}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 9}
```

A2.5.14. Метод get_events_list

Получение списка событий нарушений из БД с учетом заданных правил фильтрации.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- offset — смещение от начала списка всех найденных записей, для возврата только начиная с номера по порядку, uint64;
- limit — максимальное количество возвращаемых записей списка, uint64;
- order — наименование поля записей в БД для сортировки, string;
- desc — сортировка списка, true — по убыванию, false — по возрастанию;
- columns — наименование возвращаемых полей записей из БД (из таблицы events), массив строк, доступные значения — string:
  - id — id события нарушения (всегда добавляется в результат выдачи);
  - timestamp — время записи события нарушения в БД;
  - target_id — id нарушителя;
  - time — время фиксации нарушения;
  - type — тип события нарушения ТС;
  - target_mdspeed — средняя скорость;
  - target_mdistance — расстояние до ТС от ведущего регистратора;
  - target_sdistance — расстояние до ТС от ведомого регистратора;
  - target_mtime — время фиксации нарушения на ведущем регистраторе;
  - target_stime — время фиксации нарушения на ведомом регистраторе;
  - target_type — идентификатор типа ТС;
  - liplate_data — номерная пластина (ГРЗ);
  - target_distance — расстояние, пройденное ТС.
- filter — объект фильтруемых полей с правилами фильтрации, ключ — имя поля в БД событий, значение — объект правила фильтрации.

Правила фильтрации — объект (имя фильтра — значение):

- like — фильтр поиска по строке, подобен "LIKE %value%";
- notlike — фильтр поиска по строке, подобен "NOT LIKE %value%";
- equal — проверка на точное равенство значения;
- from — проверка значения на больше или равно;
- to — проверка значения на меньше или равно;
- in — проверка значения на соответствие одному из списка, подобен "IN ('1', 'abc',...)".

Все правила фильтрации по всем указанным в фильтре полям складываются по условию "И".

Результат запроса — массив с объектами записей.

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "get_events_list", "params": {
        "offset": 0,
        "desc": true,
        "order": "time",
        "filter": {
            "liplate_data": {
```

```
            "like": "*178*"
        },
        "time": {
            "from": "2020-03-17 12:00:00.000000"
        },
        "target_mtime": {
            "to": "2020-03-19 12:30:10.000000"
        },
        "target_mdspeed": {
            "from": 55.5
        }
    },
    "limit": 200,
    "columns": [
        "time",
        "liplate_data",
        "type",
        "target_mdspeed"
    ]
}, "id": 11}
<-- {"jsonrpc": "2.0", "result": [
    {"0":"2020-03-17 12:01:53.649000", "1":"c157e|178", "2":"0", "3": "60.1", "id": "328715"},
    {"0":"2020-03-17 12:02:54.149000", "1":"c267b|178", "2":"0", "3": "60.0", "id": "328714"},
    {"0":"2020-03-17 14:55:31.399000", "1":"b137e|178", "2":"0", "3": "57.0", "id": "328712"},
    {"0":"2020-03-17 15:20:30.899000", "1":"a621a|178", "2":"0", "3": "56.2", "id": "328713"},
    {"0":"2020-03-18 09:03:29.024000", "1":"b777e|178", "2":"0", "3": "59.0", "id": "328711"},
    {"0":"2020-03-18 10:35:27.774000", "1":"a734e|178", "2":"0", "3": "58.0", "id": "328710"},
    {"0":"2020-03-18 10:45:24.149000", "1":"c731e|178", "2":"0", "3": "59.5", "id": "328709"},
    {"0":"2020-03-18 11:30:22.899000", "1":"b777e|178", "2":"0", "3": "61.1", "id": "328708"},
    {"0":"2020-03-19 11:40:21.149000", "1":"h711t|178", "2":"0", "3": "62.0", "id": "328706"}
], "id": 11}

<-- {"jsonrpc": "2.0", "result": [], "id": 11}
```


Пример запрос/ответ — ошибка:

- ошибка выполнения метода:


```
--> {"jsonrpc": "2.0", "method": "get_events_list", "params": {...}, "id": 11}
<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 11}
```


- неверный набор параметров:

```
--> {"jsonrpc": "2.0", "method": "get_events_list", "params": {
        "filter": {
            "liplate_data": []
        }
    }, "id": 9}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 9}
```

А2.5.15. Метод get_event_info

Получение полных данных о событии нарушения с информацией о материалах из БД по id события.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- id — id события нарушения, uint64.

Результат запроса — объект со значениями:

- target_id — id нарушителя, uint64;
- time — время фиксации нарушения в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- type — тип события нарушения ТС, string:
  - "1.1" — нарушение мгновенной скорости (только в режиме учёта нарушений С1.1 с регистраторов вместо С1.2);
  - "1.2" — нарушение средней скорости.
- timestamp — время записи события нарушения в БД в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_mdspeed — средняя скорость (км/ч), float;
- target_mdistance — расстояние до ТС от ведущего регистратора (м), float;
- target_sdistance — расстояние до ТС от ведомого регистратора (м), float;
- target_mtime — время фиксации нарушения на ведущем регистраторе в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_stime — время фиксации нарушения на ведомом регистраторе в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_type — идентификатор типа ТС (не используется — "0"), string;
- liplate_data — номерная пластина (ГРЗ), string;
- target_distance — расстояние, пройденное ТС (м), float;
- materials — массив объектов данных материалов, сформированных для события, поля каждого объекта:
  - id — id материала нарушения, uint64.
  - file_type — идентификатор типа материала, int:
    - 0 — protobuf основной фотоматериал (.bin);
    - 1 — материал ведущего регистратора (maser_***_fr.jpg);

- 2 — материал ведомого регистратора (slave_***_fr.jpg);
- 3 — цифровая подпись protobuf основного материала (.sign);
- 4 — jpeg основной фотоматериал (.jpg);
- 5 — цифровая подпись jpeg основного фотоматериала (.sign);
- 6 — xml материал события (.xml);
- 7 — цифровая подпись xml материала события (.sign);

- has_material — формировался ли материал или только запись в БД, true/false.
- file_exists — существует ли файл материала на диске, true/false.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_event_info", "params": {"id":328715}, "id": 15}

<-- {"jsonrpc": "2.0", "result": {
"target_id": 317740,
"time": "2020-04-23 12:55:53.649000",
"type": "0",
"target_mdspeed": 60.1,
"target_mdistance": 36.4532,
"target_sdistance": 35.2121,
"target_mtime": "2020-04-23 12:55:53.649000",
"target_stime": "2020-04-23 12:55:50.649000",
"target_type": "0",
"liplate_data":"к759ав|198",
"materials":[
{
"id": 657431,
"file_type": 0,
"has_material": true,
"file_exists": true
},
{
"id": 657432,
"file_type": 1,
"has_material": true,
"file_exists": true
},
{
"id": 657433,
"file_type": 2,
"has_material": true,

"file_exists": true
},
{
"id": 657434,
"file_type": 3,
"has_material": true,
"file_exists": true
}
],
}, "id": 15}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_event_info", "params": {"id": 328709}, "id": 9}
<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 9}

- не найдено:

--> {"jsonrpc": "2.0", "method": "get_event_info", "params": {"id":999999}, "id": 15}
<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 9}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "get_event_info", "params": {"idd":3453}, "id": 8}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 8}

А2.5.16. Метод get_target_info

Получение полных данных о нарушителе со всеми его событиями нарушений и с информацией о материалах из БД по id нарушителя.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- id — id нарушителя, uint64.

Результат запроса — объект со значениями:

- detection_time — время события нарушения в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- direction — направление движения ТС по отношению к ведущему регистратору (1 — попутное, 0 — встречное), uint64;
- liplate_data — номерная пластина (ГРЗ), string;
- events — массив объектов данных событий, зафиксированных для ТС, поля каждого объекта:
  - event_id — id события нарушения, uint64;
  - target_mdistance — расстояние до ТС от ведущего регистратора (м), float;
  - target_mdspeed — средняя скорость (км/ч), float;

- target_mtime — время фиксации нарушения на ведущем регистратора в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_sdistance — расстояние до ТС от ведомого регистратора (м), float;
- target_stime — время фиксации нарушения на ведомом регистратора в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- target_type — идентификатор типа ТС (не используется — "0"), string;
- time — время записи события нарушения в БД в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- type — тип события нарушения ТС, string:
  - "1.1" — нарушение мгновенной скорости (только в режиме учёта нарушений С1.1 с регистраторов вместо С1.2);
  - "1.2" — нарушение средней скорости.
- target_distance — расстояние, пройденное ТС (м), float;
- materials — массив объектов данных материалов, сформированных для события, поля каждого объекта:
  - id — id материала нарушения, uint64.
  - file_type — идентификатор типа материала, int:
    - 0 — protobuf основной фотоматериал (.bin);
    - 1 — материал ведущего регистратора (maser_***_fr.jpg);
    - 2 — материал ведомого регистратора (slave_***_fr.jpg);
    - 3 — цифровая подпись protobuf основного материала (.sign);
    - 4 — jpeg основной фотоматериал (.jpg);
    - 5 — цифровая подпись jpeg основного фотоматериала (.sign);
    - 6 — xml материал события (.xml);
    - 7 — цифровая подпись xml материала события (.sign).
  - has_material — формировался ли материал или только запись в БД, true/false;
  - file_exists — существует ли файл материала на диске, true/false.

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "get_target_info", "params": {"id":400}, "id": 15}
<-- {"jsonrpc": "2.0", "result": {
      "detection_time": "2020-05-21 11:58:56.836000",
      "direction": 1,
      "events": [
         {
            "event_id": 400,
            "materials": [
               {
                  "file_exists": true,
```

"file_type": 0,
"has_material": true,
"id": 1597
},
{
"file_exists": true,
"file_type": 1,
"has_material": true,
"id": 1598
},
{
"file_exists": true,
"file_type": 2,
"has_material": true,
"id": 1599
},
{
"file_exists": false,
"file_type": 3,
"has_material": true,
"id": 1600
}
],
"target_mdistance": 56.8,
"target_mdspeed": 45.3,
"target_mtime": "2020-05-21 11:58:56.836000",
"target_sdistance": 52.8,
"target_stime": "2020-05-21 11:59:21.836000",
"target_type": "0",
"time": "2020-05-21 11:58:56.836000",
"type": "0",
"target_sdistance": 78.2
}
],
"liplate_data": "d023c|705"
}, "id": 15
}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_target_info", "params": {"id": 328709}, "id": 9}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 9}

- не найдено:

--> {"jsonrpc": "2.0", "method": "get_target_info", "params": {"id": 999999}, "id": 15}

<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 15}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "get_target_info", "params": {"idd": 3453}, "id": 8}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 8}

А2.5.17. Метод get_users

Получение списка пользователей. Системные пользователи не возвращаются (root).

Доступ: закрытый.

Параметры запроса: нет.

Результат запроса — массив с объектами данных пользователей:

- name — имя пользователя, string;
- fullname — полное имя пользователя, string (если задано, иначе — пусто);
- rights — уровень прав пользователя (root, admin, verify, setup, info, journal).

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_users", "id": 22}


```
<-- {"jsonrpc": "2.0", "result": [
        {
            "name": "Администратор",
            "fullname": "",
            "rights": "admin"
        },
        {
            "name": "Поверка",
            "fullname": "Иван Иванович",
            "rights": "verify"
        },
        {
            "name": "Журнал",
            "fullname": "",
            "rights": "journal"
        }
    ], "id": 22}
```


Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_users", "id": 23}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 23}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "get_users", "id": 23}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 23}

А2.5.18. Метод get_user

Получение данных пользователя по имени.

Доступ: закрытый.

Параметры запроса:

- name — имя пользователя, string;

При отсутствии параметров или при пустом объекте параметров возвращаются данные о текущем пользователе.

Результат запроса — массив с объектами данных пользователей:

- name — имя пользователя, string;
- fullname — полное имя пользователя, string (если задано, иначе — пусто);
- rights — уровень прав пользователя (root, admin, verify, setup, info, journal).

Пример запрос/ответ — успех:

- данные текущего пользователя:

--> {"jsonrpc": "2.0", "method": "get_user", "params": {}, "id": 22}

<-- {"jsonrpc": "2.0", "result": {
"name": "Поверка",
"fullname": "Иван Иванович",
"rights": "verify"
}, "id": 22}

- данные указанного пользователя:

--> {"jsonrpc": "2.0", "method": "get_user", "params": {"name": "Журнал"}, "id": 22}

<-- {"jsonrpc": "2.0", "result": {
"name": "Журнал",
"fullname": "Петрович",
"rights": "journal"
}, "id": 22}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_user", "id": 23}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 23}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "get_user", "id": 23}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 23}

А2.5.19. Метод add_user

Добавление пользователя в систему. Добавлять пользователя могут только пользователи с правами admin или root. Запрещено добавление root пользователей.

Доступ: закрытый.

Параметры запроса — объект с данными пользователя:

- name — имя пользователя, string;
- password_hash — MD5 хэш от пароля пользователя, string.
- fullname — полное имя пользователя, string (не обязательное);
- rights — уровень прав пользователя (root, admin, verify, setup, info, journal), не обязательное, по умолчанию — info.

Результат запроса: true/false.

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "add_user", "params": {
        "name": "user1",
        "password_hash": "a10b425c3512f236a",
        "fullname": "bla bla bla",
        "rights": "info"
    }, "id": 25}
<-- {"jsonrpc": "2.0", "result": true, "id": 25}
```

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "add_user", "params": {...}, "id": 25}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 25}

- пользователь уже существует:

--> {"jsonrpc": "2.0", "method": "add_user", "params": {...}, "id": 26}

<-- {"jsonrpc": "2.0", "error": {"code": -30001, "message": "Already exists"}, "id": 26}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "add_user", "params": {"name": "user1", "password": "1234"}, "id": 27}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 27}

- нет доступа:

--> {"jsonrpc": "2.0", "method": "add_user", "params": {...}, "id": 28}

<-- {"jsonrpc": "2.0", "error": {"code": -30003, "message": "Access denied"}, "id": 28}

А2.5.20. Метод modify_user

Редактирование существующего пользователя по name.

- Пользователь должен существовать в системе по имени;
- Редактировать других пользователей могут только пользователи с правами admin или root;
- Никто не может устанавливать права root пользователя;
- Запрещено изменение пользователем своего уровня прав.

Доступ: закрытый.

Параметры запроса — объект с данными пользователя:

- name — текущее имя пользователя, string;
- newname — новое имя пользователя, string (не обязательное);
- password_hash — MD5 хэш от пароля пользователя, string (не обязательное);
- fullname — полное имя пользователя, string (не обязательное);
- rights — уровень прав пользователя (root, admin, verify, setup, info, journal), по умолчанию — info (не обязательное).

Результат запроса: true/false.

Пример запрос/ответ — успех:

```
--> {"jsonrpc": "2.0", "method": "modify_user", "params": {
        "name": "user1",
        "newname": "user11",
        "password_hash": "321",
        "fullname": "Ivan Sergeevich",
        "rights": "info"
    }, "id": 28}
<-- {"jsonrpc": "2.0", "result": true, "id": 28}
```

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

```
--> {"jsonrpc": "2.0", "method": "modify_user", "params": {...}, "id": 28}
<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 28}
```

- пользователь не найден:

```
--> {"jsonrpc": "2.0", "method": "modify_user", "params": {...}, "id": 28}
<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 28}
```

- неверный набор параметров:

```
--> {"jsonrpc": "2.0", "method": "modify_user", "params": {"password_hash":
"120012461f34c376a"}, "id": 29}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 29}
```

- нет доступа:

--> {"jsonrpc": "2.0", "method": "modify_user", "params": {...}, "id": 30}

<-- {"jsonrpc": "2.0", "error": {"code": -30003, "message": "Access denied"}, "id": 30}

А2.5.21. Метод remove_user

Удаление существующего пользователя по name.

- Пользователь должен существовать в системе по имени;
- Удалять других пользователей могут только пользователи с правами admin или root;
- Никто не может удалить своего пользователя;
- Запрещено удаление root пользователя.

Доступ: закрытый.

Параметры запроса — объект с данными пользователя:

- name — имя пользователя, string.

Результат запроса: true/false.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "remove_user", "params": {"name": "user1"}, "id": 29}

<-- {"jsonrpc": "2.0", "result": true, "id": 29}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "remove_user", "params": {...}, "id": 29}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 29}

- пользователь не найден:

--> {"jsonrpc": "2.0", "method": "remove_user", "params": {"name": "xxx"}, "id": 29}

<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 29}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "remove_user", "params": {}, "id": 29}

<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 29}

- нет доступа:

--> {"jsonrpc": "2.0", "method": "modify_user", "params": {...}, "id": 30}

<-- {"jsonrpc": "2.0", "error": {"code": -30003, "message": "Access denied"}, "id": 30}

А2.5.22. Метод get_users_actions

Получение списка действий пользователей.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- offset — смещение от начала списка всех найденных записей, для возврата только начиная с номера по порядку, uint64;
- limit — максимальное количество возвращаемых записей списка, uint64;

- order — наименование поля записей в БД для сортировки, string;
- desc — сортировка списка, true — по убыванию, false — по возрастанию.

Результат запроса — массив с объектами событий пользователей:

- id — id действия, uint64;
- tm — время действия в формате "YYYY-MM-DD hh:mm:ss.nnnnnn", string;
- us — имя пользователя, string;
- ac — код действия, string;
- va — данные действия, string;
- vb — данные действия, string;
- vc — данные действия, string;
- vd — данные действия, string;
- ve — данные действия, string;
- vf — данные действия, string.

Формат и заполнение данных действия зависит от кода действия.

Коды действий:

- 10 — изменение режима работы ПО:
  - va — название режима (work/setup), string.
- 101 — изменение параметра:
  - va — название параметра (например, speed_limit), string;
  - vb — новое значение параметра, string, формат — escaped json, например "{\"speed_limit\":30.2}".
- 200 — вход в систему:
  - va — ip адрес пользователя, string.
- 201 — ошибка входа в систему:
  - va — ip адрес пользователя, string.
- 202 — выход из системы:
  - va — ip адрес пользователя, string.
- 400 — добавление пользователя:
  - va — новое имя пользователя (например, user1), string;
  - vb — новые права пользователя (например, verify), string.
- 401 — изменение пользователя:
  - va — новое имя пользователя (например, user1), string;
  - vb — новые права пользователя (например, verify), string.
- 402 — удаление пользователя:
  - va — имя пользователя (например, user1), string;
  - vb — права пользователя (например, verify), string.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_users_actions", "params": {

"order": "tm",
"desc": true,
"limit": 100,
"offset": 0
}, "id": 31}
<-- {"jsonrpc": "2.0", "result": [
{"id":340, "tm":"2020-03-26 13:07:36", "us":"Администратор", "ac":10, "va":"setup", "vb":null, "vc":null, "vd":null, "ve":null, "vf":null},
{"id":335, "tm":"2020-03-26 13:02:35", "us":"Администратор", "ac":101, "va":"speed_limit", "vb":"{\"speed_limit\":30.2}", "vc":null, "vd":null, "ve":null, "vf":null},
{"id":335, "tm":"2020-03-26 13:02:35", "us":"Администратор", "ac":101, "va":"speed_limit_truck", "vb":"{\"speed_limit_truck\":30.0}", "vc":null, "vd":null, "ve":null, "vf":null},
{"id":232, "tm":"2020-03-19 04:05:17", "us":"Поверка", "ac":200, "va":"181.65.241.45", "vb":null, "vc":null, "vd":null, "ve":null, "vf":null},
{"id":259, "tm":"2020-03-23 15:27:35", "us":"Поверка", "ac":201, "va":"84.52.101.111", "vb":null, "vc":null, "vd":null, "ve":null, "vf":null},
{"id":294, "tm":"2020-03-26 11:08:15", "us":"Администратор", "ac":202, "va":"84.52.101.121", "vb":null, "vc":null, "vd":null, "ve":null, "vf":null},
{"id":241, "tm":"2020-03-19 09:35:46", "us":"Администратор", "ac":10, "va":"work", "vb":null, "vc":null, "vd":null, "ve":null, "vf":null},
{"id":212, "tm":"2020-03-17 01:47:12", "us":"root", "ac":101, "va":"slave", "vb":"{\"slave\":{\"fixations_dir\":\"blabla\",\"ftp_pass\":\"abcabc\",...}}", "vc":null, "vd":null, "ve":null, "vf":null}
], "id": 31}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_users_actions", "params": {...}, "id": 40}
<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 40}

- неверный набор параметров:

--> {"jsonrpc": "2.0", "method": "get_users_actions", "params": {"offset": "hello"}, "id": 40}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 40}

А2.5.22. Метод get_users_actions_count

Получение количества записей действий пользователей.

Доступ: закрытый.

Параметры запроса: нет.

Результат запроса: uint64.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_users_actions_count", "id": 41}

<-- {"jsonrpc": "2.0", "result": 254, "id": 41}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_users_actions_count", "id": 42}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 42}

А2.5.23. Метод get_report_templates_list

Получение списка с информацией о шаблонах материалов.

Доступ: закрытый.

Параметры запроса: нет.

Результат запроса — массив с объектами со значениями:

- file_name — имя файла шаблона, string;
- file_path — полный путь к файлу шаблона, string;
- default — является ли шаблон встроенным в ПО, true/false (не используется);
- selected — является ли шаблон текущим выбранным для материалов в ПО, true/false.

Пример запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_report_templates_list", "id": 45}

<-- {"jsonrpc": "2.0", "result": [

{"file_name":"ru.xml", "file_path":"/mnt/targets/templates/ru.xml", "default":true, "selected":true},

{"file_name":"ru_samara.xml", "file_path":"/mnt/targets/templates/ru_samara.xml", "default":false, "selected":false}

], "id": 45}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода:

--> {"jsonrpc": "2.0", "method": "get_report_templates_list", "id": 46}

<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 46}

А2.5.24. Метод get_device_info

Получение данных от регистраторов. Используются данные из готовых материалов на регистраторах, для этого должны быть фиксации на регистраторах.

Доступ: закрытый.

Параметры запроса — объект со значениями:

- num — индекс регистратора (1 — ведущий, 2 — ведомый), uint64;
- serial — серийный номер регистратора, например "1620254", string (не обязательное);

- timeout — период времени в мс, в течение которого ждать ответа, uint64 (по умолчанию — 10000).

Требуется указать один из параметров — num или serial.

Результат запроса — объект со значениями:

- device_name — название регистратора, string (например, "СКАТ");
- gns_lat — широта из координат установки регистратора, float (например, 60.01986);
- gns_lat_prefix — северная/южная широта из координат установки регистратора, string ("N", "S" или "" — не известно);
- gns_lon — долгота из координат установки регистратора, float (например, 30.4155);
- gns_lon_prefix — восточная/западная долгота из координат установки регистратора, string ("E", "W" или "" — не известно);
- place_direction — описание направления движения на регистраторе, string (например, "в город");
- place_location — место расположения регистратора, string (например, "СПб, Северный, будка");
- serial_number — серийный номер регистратора, string (например, "1620254");
- time_offset_ms — сдвиг временной зоны относительно GMT в мс, uint64 (например, 10800000, соответствует +03:00);

Примеры запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_device_info", "params": {"num": 1}, "id": 48}

<-- {"jsonrpc":"2.0","result":{
"device_name":"СКАТ",
"gns_lat":60.01986,
"gns_lat_prefix":"N",
"gns_lon":30.4155,
"gns_lon_prefix":"E",
"place_direction":"в город",
"place_location":"СПб, Северный, дом.24",
"serial_number":"1810027",
"time_offset_ms":10800000
}, "id":48}

--> {"jsonrpc": "2.0", "method": "get_device_info", "params": {"serial": "1601042"}, "id": 49}

<-- {"jsonrpc":"2.0","result":{
"device_name":"СКАТ",
"gns_lat":60.02019,
"gns_lat_prefix":"N",
"gns_lon":30.41429,

"gns_lon_prefix":"",
"place_direction":"в сторону Гражданского пр.",
"place_location":"Северный — дом.19",
"serial_number":"1601042",
"time_offset_ms":10800000
}, "id":49}

Пример запрос/ответ — ошибка:

- ошибка выполнения метода (ошибка соединения или время истекло):

--> {"jsonrpc": "2.0", "method": "get_device_info", "params": {"num": 1, "timeout": 10}, "id": 50}
<-- {"jsonrpc": "2.0", "error": {"code": -30000, "message": "Method returns false"}, "id": 50}

- не найдено (нет материалов на регистраторе):

--> {"jsonrpc": "2.0", "method": "get_device_info", "params": {"num": 1}, "id": 50}
<-- {"jsonrpc": "2.0", "error": {"code": -30004, "message": "Not found"}, "id": 50}

- неверный набор параметров (номер):

--> {"jsonrpc": "2.0", "method": "get_device_info", "params": [3], "id": 50}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 50}

- неверный набор параметров (серийный номер):

--> {"jsonrpc": "2.0", "method": "get_device_info", "params": {"num": 0}, "id": 50}
<-- {"jsonrpc": "2.0", "error": {"code": -32602, "message": "Invalid params"}, "id": 50}

- внутренняя ошибка:

--> {"jsonrpc": "2.0", "method": "get_device_info", "params": {"num": 1}, "id": 50}
<-- {"jsonrpc": "2.0", "error": {"code": -32603, "message": "Internal error"}, "id": 50}

А2.5.25. Метод get_selfpath

Получение текущего пути запуска ПО.

Доступ: закрытый.

Параметры запроса — нет.

Результат запроса — строка, путь в файловой системе, например "C:\Duet\bin".

Примеры запрос/ответ — успех:

--> {"jsonrpc": "2.0", "method": "get_selfpath", "id": 51}
<-- {"jsonrpc": "2.0", "result": "C:\Duet\bin", "id": 51}

Пример запрос/ответ — ошибка:

- внутренняя ошибка:

--> {"jsonrpc": "2.0", "method": "get_selfpath", "id": 51}
<-- {"jsonrpc": "2.0", "error": {"code": -32603, "message": "Internal error"}, "id": 51}

А3. HTTP files API

Точка доступа: **http://<адрес устройства>:11000/files**

Протокол HTTP используется для запросов файлов материалов нарушений и других. Является вспомогательным протоколам для взаимодействия веб-интерфейса с основным процессом ПО.

А3.1. Запросы

Варианты запросов:

- GET, параметры в url;
- POST, параметры в теле запроса.

Название запроса указывается в параметрах в виде:

?r=REQUESTNAME

Далее в url могут следовать другие параметры запроса. Состав параметров зависит от вида запроса.

Возвращаемые коды результата базово соответствуют используемым в REST API.

Успех:

- 200 — OK.

Ошибки:

- 400 — Bad Request;
- 401 — Unauthorized;
- 403 — Forbidden;
- 404 — Not Found;
- 500 — Internal Server Error.

А3.2. Ошибки в HTTP

В реализованном протоколе взаимодействия по HTTP для /files выставляется код HTTP ответа с кодом ошибки и, опционально, содержимое JSON ответа с ошибкой.

Используемые коды ответов HTTP на ошибки приведены в табл. А3.1:

Таблица А3.1 — Коды ответов статуса HTTP

HTTP code	message	Описание
400	Bad Request	Некорректный запрос. Например: - неверный формат параметров в url или теле запроса; - неизвестное название запроса.
401	Unauthorized	Неавторизованный запрос. Например: - некорректная сессия (не существует, истекло время сессии);

HTTP code	message	Описание
		- запрос выполнения защищенных методов без указания session_id (идентификатора сессии).
403	Forbidden	Запрещено. Например: - верная сессия, но недостаточно прав пользователя на выполнение запроса.
404	Not Found	Запрашиваемый элемент не найден, например, файл на диске.
500	Internal Server Error	Системная ошибка. При выполнении запроса возникли проблемы в ПО.

А3.3. Аутентификация

Аутентификация требуется для защищенных (закрытых) запросов и производится через JSON-RPC запросы аутентификации. Далее полученное значение session_id используется в cookie в заголовках каждого HTTP запроса (см. п. А2.3).

Запросы HTTP по доступу делятся на две категории:

- открытые (public) — позволяют работать без идентификатора сессии;
- закрытые (private) — требуют отправки актуального идентификатора сессии в заголовках.

А3.4. Запросы HTTP

Полный список HTTP запросов:

- get_events_list_csv;
- get_event_target_file;
- get_target_image.

А3.4.1. Запрос get_events_list_csv

Получение списка событий нарушений из БД с учетом заданных правил фильтрации в виде файла формата CSV.

Доступ: закрытый.

Параметры запроса — объект JSON в теле запроса со значениями (аналогично JSON-RPC get_events_list):

- offset — смещение от начала списка всех найденных записей, для возврата только начиная с номера по порядку, uint64;
- limit — максимальное количество возвращаемых записей списка, uint64;
- order — наименование поля записей в БД для сортировки, string;
- desc — сортировка списка, true — по убыванию, false — по возрастанию;

- columns — наименование возвращаемых полей записей из БД (из таблицы events), массив строк, доступные значения — string:
  - id — id события нарушения (всегда добавляется в результат выдачи);
  - timestamp — время записи события нарушения в БД;
  - target_id — id нарушителя;
  - time — время фиксации нарушения;
  - type — тип события нарушения ТС;
  - target_mdspeed — средняя скорость;
  - target_mdistance — расстояние до ТС от ведущего регистратора;
  - target_sdistance — расстояние до ТС от ведомого регистратора;
  - target_mtime — время фиксации нарушения на ведущем регистраторе;
  - target_stime — время фиксации нарушения на ведомом регистраторе;
  - target_type — идентификатор типа ТС;
  - liplate_data — номерная пластина (ГРЗ);
  - target_distance — расстояние, пройденное ТС.
- filter — объект фильтруемых полей с правилами фильтрации, ключ — имя поля в БД событий, значение — объект правила фильтрации.

Правила фильтрации — объект (имя фильтра — значение):

- like — фильтр поиска по строке, подобен "LIKE %value%";
- notlike — фильтр поиска по строке, подобен "NOT LIKE %value%";
- equal — проверка на точное равенство значения;
- from — проверка значения на больше или равно;
- to — проверка значения на меньше или равно;
- in — проверка значения на соответствие одному из списка, подобен "IN ('1', 'abc',...)".

Все правила фильтрации по всем указанным в фильтре полям складываются по условию "И".

Результат запроса — файл с названием events.csv.

Пример запрос/ответ — успех:

```
--> POST ?r=get_events_list_csv
  {
    "offset": 0,
    "desc": true,
    "order": "time",
    "filter": {
      "liplate_data": {
        "like": "*178*"
      },
      "target_mtime": {
        "from": "2020-03-17 12:00:00.000000"
```

```
            },
            "target_mtime": {
                "to": "2020-03-19 12:30:10.000000"
            },
            "target_mdspeed": {
                "from": 55.5
            }
        },
        "limit": 200,
        "columns": [
            "time",
            "liplate_data",
            "type",
            "target_mdspeed"
        ]
    }
<--
    HTTP/1.1 200 OK
    ...
    Server: DUET_SERVER
    ...
    Content-Type: text/csv; charset=utf-8
    Content-Disposition: attachment; filename=events.csv
    Content-Length: 123
    Content-Transfer-Encoding: binary
    DATA
```


Пример запрос/ответ — ошибка:

- ошибка выполнения метода или некорректного запроса:


```
<--
    HTTP/1.0 400 Bad Request
    Content-Type: text/csv; charset=utf-8
```


- внутренняя системная ошибка:


```
<--
    HTTP/1.0 500 Internal Server Error
```


А3.4.2. Запрос get_event_target_file

Получение файла материала на событие в виде изображения или файла для скачивания. При невозможности получения файла в виде изображения для несовместимых форматов файлов будет возвращена ошибка некорректного запроса.

Доступ: закрытый.

Параметры запроса в url:

- e — id события нарушения (event_id), uint64;
- ft — идентификатор типа файла материала (filetype), доступность зависит от настроек ПО, int:
  - 0 — protobuf основной фотоматериал (.bin);
  - 1 — материал ведущего регистратора (maser_***_fr.jpg);
  - 2 — материал ведомого регистратора (slave_***_fr.jpg);
  - 3 — цифровая подпись protobuf основного материала (.sign);
  - 4 — jpeg основной фотоматериал (.jpg);
  - 5 — цифровая подпись jpeg основного фотоматериала (.sign);
  - 6 — xml материал события (.xml);
  - 7 — цифровая подпись xml материала события (.sign).
- dl — запрос на скачивание файла (1) или показ(0) картинки, int.
- thumb — запрос на показ уменьшенного изображения из материала (0 — полное, 1 — уменьшенное), int.

Результат запроса (разные заголовки HTTP в зависимости от запроса):

1) image/jpeg изображение (dl=0, thumb любое);
2) application/octet-stream данные файла для скачивания (dl=1), имя скачиваемого файла зависит от запрашиваемого типа материала и параметров:
   - filename="thumb-FILENAME";
   - filename="frame-FILENAME";
   - filename="FILENAME".

Пример запрос/ответ — успех:

- изображение из ptotobuf:

```
--> GET ?r=get_event_target_file&e=3436&ft=0&dl=0
<--
    HTTP/1.1 200 OK
    ...
    Server: DUET_SERVER
    ...
    Content-Type: image/jpeg
    Transfer-Encoding: chunked
    Content-Disposition: attachment;
    DATA
```

- скачать уменьшенное изображение из ptotobuf в виде файла:

--> GET ?r=get_event_target_file&e=3436&ft=0&dl=1&thumb=1

<--

HTTP/1.1 200 OK

...

Server: DUET_SERVER

...

Content-Type: application/octet-stream

Content-Length: SIZE

Content-Disposition: attachment; filename="thumb-duet_protobuff_2020_03_17_12_33_24.jpg"

Content-Transfer-Encoding: binary

Cache-Control: must-revalidate, post-check=0, pre-check=0

DATA

Пример запрос/ответ — ошибка:

- ошибка выполнения метода или некорректного запроса:

<--

HTTP/1.0 400 Bad Request

- внутренняя системная ошибка:

<--

HTTP/1.0 500 Internal Server Error

А3.4.2. Запрос get_target_image

Получение изображения из основного материала нарушения по id нарушителя.

Доступ: закрытый.

Параметры запроса в url:

- id — id нарушителя, uint64;
- size — запрос на получение уменьшенного изображения из материала ("small" — уменьшенное, параметр отсутствует — полное), string.

Изображение пересылается от первого найденного основного материала с картинкой (из protobuf или jpg).

Результат запроса — image/jpeg изображение.

Пример запрос/ответ — успех:

--> POST ?r=get_target_image&id=180620&size=small

<--

HTTP/1.1 200 OK

...

Server: DUET_SERVER

...

Content-Type: image/jpeg

DATA

Пример запрос/ответ — ошибка:

- ошибка выполнения метода или некорректного запроса:

<--

HTTP/1.0 400 Bad Request

- внутренняя системная ошибка:

<--

HTTP/1.0 500 Internal Server Error

- не найдено:

<--

HTTP/1.0 404 Not Found

A4. HTTP XML API

Точка доступа: **http://<адрес устройства>:11000/get***

Протокол HTTP используется для запросов информации о нарушениях и о состоянии системы, результат возвращается в формате XML. Является обязательным для работы по протоколу обмена данными **samara**. Работает только при включенном протоколе в config.json: **result_protocol=samara**

A4.1. Запросы

Варианты запросов:

- GET, параметры в url;

Название запроса указывается в параметрах в виде пути:

/get*

Далее в url могут следовать параметры запроса. Состав параметров зависит от вида запроса.

Возвращаемые коды результата базово соответствуют используемым в REST API.

Успех:

- 200 — OK.

Ошибки:

- 400 — Bad Request;
- 401 — Unauthorized;
- 403 — Forbidden;
- 404 — Not Found;
- 500 — Internal Server Error.

А4.2. Ошибки в HTTP XML

В реализованном протоколе взаимодействия по HTTP для /get* выставляется код HTTP ответа с кодом ошибки.

Используемые коды ответов HTTP на ошибки приведены в табл. А4.1:

Таблица А4.1 — Коды ответов статуса HTTP

HTTP code	message	Описание
400	Bad Request	Некорректный запрос. Например: - неверный формат параметров в url или теле запроса; - неизвестное название запроса.
401	Unauthorized	Неавторизованный запрос. Например: - некорректная сессия (не существует, истекло время сессии); - запрос выполнения защищенных методов без указания session_id (идентификатора сессии).
403	Forbidden	Запрещено. Например: - верная сессия, но недостаточно прав пользователя на выполнение запроса.
404	Not Found	Запрашиваемый элемент не найден, например, файл на диске.
500	Internal Server Error	Системная ошибка. При выполнении запроса возникли проблемы в ПО.

А4.3. Аутентификация

Аутентификация требуется только для защищенных (закрытых) запросов и производится через JSON-RPC запросы аутентификации. Далее полученное значение session_id используется в cookie в заголовках каждого HTTP запроса (см. п.2.3).

Запросы HTTP по доступу делятся на две категории:

- открытые (public) — позволяют работать без идентификатора сессии;
- закрытые (private) — требуют отправки актуального идентификатора сессии в заголовках.

А4.4. Запросы HTTP

Полный список HTTP запросов:

- geteventdata;
- geteventlist;
- getstatus.

А4.4.1. Запрос geteventdata

Получение данных событий из БД в формате XML с учетом заданных правил фильтрации.

Доступ: открытый.

Формат запроса:

/geteventdata?t1=<дата и время начала интервала поиска по дате и времени события>&t2=<дата и время окончания интервала поиска по дате и времени события>&tdb1=<дата и время начала интервала поиска по дате и времени попадания записи о событии в БД ПО>&tdb2=<дата и время окончания интервала поиска по дате и времени попадания записи о событии в БД ПО>&type=<тип события>&num=<маска для поиска ГРЗ>&cnt=<кол-во записей>

Параметры запроса приведены в таблице А4.2.

Таблица А4.2 — Параметры запроса событий XML протокола

Параметр	**Значение**	**Параметр присутствует в запросе**	**Параметр отсутствует в запросе**
t1	Дата и время	Выбираются события, зафиксированные не ранее указанных в параметре даты и времени.	Выбираются события, начиная с самого раннего, информация о котором имеется в системе на момент запроса.
t2	Дата и время	Выбираются события, зафиксированные не позже указанных в параметре даты и времени.	Выбираются события, зафиксированные до текущей даты и времени.
tdb1	Дата и время	Выбираются события, внесенные в БД ПО не ранее указанных в параметре даты и времени.	Выбираются события, начиная с самого раннего, информация о внесении которого в БД ПО имеется в системе на момент запроса.
tdb2	Дата и время	Выбираются события, внесенные в БД ПО не позже указанных в параметре даты и времени.	Выбираются события, внесенные в БД ПО до текущей даты и времени.
type	Список типов событий через запятую или префикс VIOL	Выбираются только те события, тип которых указан в списке через запятую в параметре. Если параметр задан в виде префикса VIOL, то из выдачи исключаются события с кодом C0, т.е. из выдачи исключаются проезда без совершения АП.	События выбираются без учета их типа.

Параметр	Значение	Параметр присутствует в запросе	Параметр отсутствует в запросе
num	ГРЗ в тестовом виде с указанием "*" на месте не проверяемых символов	События выбираются с учетом посимвольного совпадения ГРЗ со значением параметра. При поиске не проверяются символы ГРЗ, для которых указана "*" в значении параметра. Таким образом, может быть произведен поиск по полностью известному ГРЗ, так и по ГРЗ, для которого не все символы известны.	События выбираются без учета их ГРЗ.
cnt	Максимальное количество возвращаемых записей	Сформированный ответ содержит указанное количество записей или менее, если записей удовлетворяющих параметрам запроса меньше.	Количество выбираемых записей ограничено числом 2000.

Формат указания даты и времени в запросе соответствует ISO 8601:

YYYY-MM-DDThh:mm:ss±hh

Список типов событий может содержать значения C0-C16 в соответствии с ГОСТ Р 57144-2016: C0, C1, C1.1, C1.2, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16.

Буква "C" в указании кода события — латинская заглавная.

При указании шаблона или значения ГРЗ для обозначения букв русского алфавита, совпадающих по написанию с латинскими, а именно: A, B, C, E, H, K, M, O, P, T, X, У, используются только латинские заглавные буквы.

Результат запроса — XML форматированный список данных событий. Последовательность выдачи данных о событии (порядок следования составных тегов event) — по возрастанию даты-времени внесения информации о событии в ПО. Следующий ключ сортировки — дата-время самого события.

Поля XML ответа приведены в таблице А4.3.

Таблица А4.3 — Поля ответа данных событий XML протокола

	Имя тега	Значение
1	id	Уникальный идентификатор события.
2	timestamp	Дата и время зафиксированного события. Время проезда через второй рубеж для события C1.2 превышения средней скорости. Формат ISO 8601: YYYY-MM-DDThh:mm:ss±hh

	Имя тега	Значение
3	timestamp0	Дата и время зафиксированного проезда транспортного средства через первый рубеж зоны в случае фиксации проезда (событие C0) через оба рубежа зоны или фиксации события превышения средней скорости (событие C1) на участке зоны. Всегда дата и время проезда через ведущий регистратор. Формат ISO 8601: YYYY-MM-DDThh:mm:ss±hh
4	timestampdb	Дата и время внесения информации о зафиксированном событии в БД ПО. Формат ISO 8601: YYYY-MM-DDThh:mm:ss±hh
5	type	Тип зафиксированного события C0-C16 в соответствии с ГОСТ Р 57144-2016.
6	path	Путь к материалам событий на FTP-сервере.
7	num	ГРЗ зафиксированного ТС.
8	tSoftVersion	Версия программного обеспечения. Например: 1.3.1.
9	nImage0Present	Наличие снимка ТС во время фиксации нарушения. Проезд второго рубежа для превышения средней скорости (событие C1). (1 — присутствует, 0 — отсутствует или отсутствует поле).
10	nImage1Present	Наличие фотоматериалов, снимок 1 (1 — присутствует, 0 — отсутствует или отсутствует поле).
11	nImage2Present	Наличие фотоматериалов, снимок 2 (1 — присутствует, 0 — отсутствует или отсутствует поле).
12	nXmlPresent	Наличие текстовой информации в виде XML-файла (1 — присутствует, 0 — отсутствует или отсутствует поле).
13	nGRZPresent	Наличие снимка пластины регистрационного знака ТС в виде jpg-файла (1 — присутствует, 0 — отсутствует или отсутствует поле).
14	nDatetime	Местное время (количество секунд от 01.01.1970). Время фиксации события в локальном часовом поясе.
15	nDisplaySpeed	Измеренная скорость (км/ч) (0 — нет измерений).
16	nOverSpeed	Признак нарушения "превышения скорости" при значении 1, 0 или иное — нет нарушения.
17	nDirection	Направление движения ТС. Если для проезда ТС второй рубеж — ведущий регистратор — 0, если второй рубеж ведомый — 1.
18	nSpeedThreshold	Ограничение скоростного режима (км/ч).

	Имя тега	Значение
19	nPDD	Пункт ПДД. Указание обязательное для type, равное C1 (превышение допустимой скорости). Для события с type, равное C0, отсутствует. Для остальных событий указание возможно, но не обязательно. Предоставляется в форматах X9.X9 или X3.X9.X9, где на месте X9 может находиться 1 или 2 цифровых символа. Например: "10.3".
20	tRadarId	Уникальный идентификатор комплекса (совокупность символов, позволяющая однозначно идентифицировать АПК). В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) уникальный идентификатор зоны. Серийный номер системы. Например: "1810027/1601042"
21	tChannel0	В случае фиксации двухкамерного проезда по рубежам зоны (событие CO) или превышения средней скорости на участке зоны (событие C1) уникальный идентификатор первого рубежа зоны. Идентификатор ведущего регистратора. Например: "1810027"
22	tChannel1	В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) уникальный идентификатор второго рубежа зоны. Идентификатор ведомого регистратора. Например: "1601042"
23	tChannel	Номер видеоканала (произвольное числовое значение, задается в настройках config.json для идентификации направления фиксации нарушений в пределах одного комплекса.
24	tSensorName0	Текстовое описание места установки системы.
25	tSensorName	Текстовое описание места установки системы. Аналогично tSensorName0.
26	tLocationCode	Код по ОКАТО места установки.
27	tRadarName	Идентификационное наименование комплекса. Например: ДУЭТ.
28	tCertificateExpDate	Дата получения сертификата поверки. Формат ISO 8601: YYYY-MM-DD
29	tCertificateExpLimit	Срок действия сертификата поверки. Формат ISO 8601: YYYY-MM-DD

	Имя тега	Значение
30	tLatitude0	В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) географическая широта места установки первого рубежа зоны (целое число — количество угловых секунд). Координаты ведущего регистратора.
31	tLongitude0	В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) географическая долгота места установки первого рубежа зоны (целое число — количество угловых секунд). Координаты ведущего регистратора.
32	tLatitude	Географические координаты места установки комплекса: географическая широта (целое число — количество угловых секунд, Координаты ведомого регистратора.
33	tLongitude	Географические координаты места установки комплекса: географическая долгота (целое число — количество угловых секунд. Координаты ведомого регистратора.
34	nTZOffset	Часовой пояс места установки (по настройке сдвига времени регистраторов в ВЕБ-интерфейсе и config.json complex_utc_offset). Формат ±hh:mm
35	nRectNumX1	Координата X левого верхнего угла пластины номера на изображении.
36	nRectNumY1	Координата Y левого верхнего угла пластины номера на изображении.
37	nRectNumX2	Координата X правого верхнего угла пластины номера на изображении.
38	nRectNumY2	Координата Y правого верхнего угла пластины номера на изображении.
39	nRectNumX3	Координаты X левого нижнего угла пластины номера на изображении.
40	nRectNumY3	Координаты Y левого нижнего угла пластины номера на изображении.

Примеры запросов:

/geteventdata?tdb1=2019-05-24T15:03:21-04&type=VIOL

/geteventdata?t2=2018-05-24T15:03:21-03&type=C0&num=C0*4C*177&cnt=50

/geteventdata?t1=2018-05-23T18:31:42-03&type=C0,C11&num=C0*4C*177

/geteventdata?t1=2018-05-23T18:31:42-03&t2=2018-05-24T15:03:21-03&cnt=1000

/geteventdata?t1=2018-05-23T18:31:42-03&cnt=1000

/geteventdata?t1=2018-05-23T18:31:42-03&t2=2018-05-24T15:03:21-03&type=C0,C2,C3&num=C0*4C*177

Пример ответа:


```
<?xml version="1.0" encoding="UTF-8"?>
<events>
  <event>
    <timestamp>2020-08-13T19:26:42+03</timestamp>
    <timestamp0>2020-08-13T19:26:38+03</timestamp0>
    <timestampdb>2020-08-13T16:26:50+00</timestampdb>
    <type>C1</type>
    <path>C1/2020/08/13</path>
    <num>B104PK198</num>
    <id>1810027_1601042_2020_08_13_19_26_42_21_00_012</id>
    <tSoftVersion>1.3.2</tSoftVersion>
    <nImage0Present>1</nImage0Present>
    <nXmlPresent>1</nXmlPresent>
    <nGRZPresent>0</nGRZPresent>
    <nDatetime>1597346802</nDatetime>
    <nDisplaySpeed>68</nDisplaySpeed>
    <nOverSpeed>1</nOverSpeed>
    <nDirection>1</nDirection>
    <nSpeedThreshold>40</nSpeedThreshold>
    <nPDD>10.2</nPDD>
    <tRadarId>1810027/1601042</tRadarId>
    <tChannel0>1810027</tChannel0>
    <tChannel1>1601042</tChannel1>
    <tChannel>1</tChannel>
    <tSensorName0>Санкт-Петербург, ул.Победы</tSensorName0>
    <tSensorName>Санкт-Петербург, пл.Мужества</tSensorName>
    <tLocationCode>28000000020</tLocationCode>
    <tRadarName>ДУЭТ</tRadarName>
    <tCertificateExpDate>2018-06-02</tCertificateExpDate>
    <tCertificateExpLimit>2021-06-06</tCertificateExpLimit>
    <tLatitude0>216051</tLatitude0>
    <tLongitude0>109490</tLongitude0>
    <tLatitude>216053</tLatitude>
    <tLongitude>109480</tLongitude>
```

```
        <nTZOffset>+03:00</nTZOffset>
        <nRectNumX1>957</nRectNumX1>
        <nRectNumY1>276</nRectNumY1>
        <nRectNumX2>1103</nRectNumX2>
        <nRectNumY2>276</nRectNumY2>
        <nRectNumX3>957</nRectNumX3>
        <nRectNumY3>320</nRectNumY3>
    </event>
    <event>
        ...
    </event>
    <event>
        ...
    </event>
</events>
```


4.4.2. Запрос geteventlist

Получение списка событий из БД в формате XML с учетом заданных правил фильтрации. Список событий представляет собой минимальную информацию по событиям, удовлетворяющим параметрам запроса (полная текстовая информация возвращается на соответствующий запрос данных событий).

Доступ: открытый.

Формат запроса:

/geteventlist?t1=<дата и время начала интервала поиска по дате и времени события>&t2=<дата и время окончания интервала поиска по дате и времени события>&tdb1=<дата и время начала интервала поиска по дате и времени попадания записи о событии в БД ПО>&tdb2=<дата и время окончания интервала поиска по дате и времени попадания записи о событии в БД ПО>&type=<тип события>&num=<маска для поиска ГРЗ>&cnt=<кол-во записей>

Параметры запроса аналогичны приведенным в таблице А4.2.

Результат запроса — XML форматированный список событий с данными в атрибутах. Последовательность выдачи обобщенных сведений о событии в списке (порядок следования составных тегов event) — по возрастанию даты-времени внесения информации о событии в ПО. Следующий ключ сортировки — дата-время самого события.

Атрибуты XML ответа приведены в таблице А4.4.

Таблица А4.4 — Поля ответа данных событий XML протокола

	Имя атрибута	Значение
1	id	Уникальный идентификатор события.
2	timestamp	Дата и время зафиксированного события. Время проезда через второй рубеж для события С1.2 превышения средней скорости. Формат ISO 8601: YYYY-MM-DDThh:mm:ss±hh
3	timestampdb	Дата и время внесения информации о зафиксированном событии в БД ПО. Формат ISO 8601: YYYY-MM-DDThh:mm:ss±hh
4	type	Тип зафиксированного события С0-С16 в соответствии с ГОСТ Р 57144-2016.
5	path	Путь к материалам событий на FTP-сервере.
6	num	ГРЗ зафиксированного ТС.

Примеры запросов:

/geteventlist?tdb1=2019-05-24T15:03:21-04&type=VIOL

/geteventlist?t2=2018-05-24T15:03:21-03&type=C0&num=C0*4C*177&cnt=50

/geteventlist?t1=2018-05-23T18:31:42-03&type=C0,C11&num=C0*4C*177

/geteventlist?t1=2018-05-23T18:31:42-03&t2=2018-05-24T15:03:21-03&cnt=1000

/geteventlist?t1=2018-05-23T18:31:42-03&cnt=1000

/geteventlist?t1=2018-05-23T18:31:42-03&t2=2018-05-24T15:03:21-03&type=C0,C2,C3 &num=C0*4C*177

Пример ответа:


```
<?xml version="1.0" encoding="UTF-8"?>
<events>
   <event timestamp="2020-08-04T02:30:49+03" timestampdb="2020-08-04T02:29:06+03"
num="e920ак198" type="C1" path="C1/2020/08/04"
id="1810027_1601042_2020_08_04_02_30_45_51_00_012" />
   <event timestamp="2020-08-04T02:31:40+03" timestampdb="2020-08-04T02:29:54+03"
num="p789тс98" type="C1" path="C1/2020/08/04"
id="1810027_1601042_2020_08_04_02_31_36_87_00_012" />
   <event timestamp="2020-08-04T02:35:27+03" timestampdb="2020-08-04T02:33:41+03"
num="c785кс178" type="C1" path="C1/2020/08/04"
id="1810027_1601042_2020_08_04_02_35_23_67_00_012" />
   <event timestamp="2020-08-04T02:36:18+03" timestampdb="2020-08-04T02:34:33+03"
num="a707ов198" type="C1" path="C1/2020/08/04"
id="1810027_1601042_2020_08_04_02_36_14_60_00_012" />
```

</events>

А4.4.3. Запрос getstatus

Запрос данных о текущем состоянии и настройках комплекса в формате XML.

Доступ: открытый.

Формат запроса:

/getstatus

Параметров запроса нет.

Результат запроса — XML форматированный список параметров в тегах. Теги XML ответа приведены в таблице А4.5.

Таблица А4.5 — Поля ответа данных состояния XML протокола

	Имя тега	Значение
1	timestamp	Текущая дата и время комплекса. Формат ISO 8601: YYYY-MM-DDThh:mm:ss±hh
2	tSVersion	Версия программного обеспечения. Например: 1.3.1.
3	nDatetime	Местное время (количество секунд от 01.01.1970) в локальном часовом поясе комплекса.
4	nWrongTurn	Признак фиксации нарушения "запрещенный манёвр" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
5	nWrongTurnPosition	Полный аналог nWrongTurn. Допускается двоякое наименование тега.
6	nOverSpeed	Признак нарушения "превышения скорости" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
7	nStopLine	Признак фиксации нарушения "пересечение стоп-линии" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
8	nRedLight	Признак фиксации нарушения "проезд на красный свет" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
9	nParkALine	Признак фиксации нарушения "остановка на полосе для маршрутных ТС" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
10	tStayRestrict	Признак фиксации нарушения "запрещенная парковка" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.

	Имя тега	Значение
11	nWrongDirection	Признак фиксации нарушения "движение во встречном направлении" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
12	nWrongLine	Признак фиксации нарушения "нарушение требований разметки" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
13	nTramways	Признак фиксации нарушения "движение по трамвайным путям" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
14	tPCross	Признак фиксации нарушения "непропуск пешехода" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
15	tPPrior	Полный аналог tPCross. Допускается двоякое наименование тега.
16	nNoLeftOnlyRightForward	Признак фиксации нарушения "поворот налево запрещен" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
17	nNoLeftRightOnlyForward	Признак фиксации нарушения "движение только прямо" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
18	nNoUTurnOnlyRightForward	Признак фиксации нарушения "движение прямо и направо" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
19	nNoUTurnOnlyForward	Признак фиксации нарушения "движение прямо, разворот запрещен" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
20	nProhibitorySignMove	Признак фиксации нарушения "движение во встречном направлении по дороге с односторонним движением" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
21	nMoveOnRedPereezd	Признак фиксации нарушения правил проезда ЖД переезда при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
22	nNoLight	Признак фиксации нарушения "выключенные внешние световые приборы" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.

	Имя тега	Значение
23	nNoForward	Признак фиксации нарушения "неправильное положение ТС на проезжей части, движение по обочине" при значении 1. При отсутствии фиксации нарушения тэг может отсутствовать.
24	tRadarId	Уникальный идентификатор комплекса (совокупность символов, позволяющая однозначно идентифицировать АПК). Серийный номер системы. Например: "1810027/1601042"
25	tChannel0	В случае фиксации двухкамерного проезда по рубежам зоны (событие С0) или превышения средней скорости на участке зоны (событие С1) уникальный идентификатор первого рубежа зоны. Идентификатор ведущего регистратора. Например: "1810027"
26	tChannel1	В случае фиксации двухкамерного проезда по рубежам зоны (событие С0) или превышения средней скорости на участке зоны (событие С1) уникальный идентификатор второго рубежа зоны. Идентификатор ведомого регистратора. Например: "1601042"
27	tChannel	Номер видеоканала (произвольное числовое значение, задается в настройках config.json для идентификации направления фиксации нарушений в пределах одного комплекса.
28	tSensorName0	Текстовое описание места установки системы.
29	tSensorName	Текстовое описание места установки системы. Аналогично tSensorName0.
30	tLocationCode	Код по ОКАТО места установки.
31	tRadarName	Идентификационное наименование комплекса. Например: ДУЭТ
32	tCertificateExpDate	Дата получения сертификата поверки. Формат ISO 8601: YYYY-MM-DD
33	tCertificateExpLimit	Срок действия сертификата поверки. Формат ISO 8601: YYYY-MM-DD
34	tLatitude0	В случае фиксации двухкамерного проезда по рубежам зоны (событие С0) или превышения средней скорости на участке зоны (событие С1) географическая широта места установки первого рубежа зоны (целое число — количество угловых секунд). Координаты ведущего регистратора.

	Имя тега	Значение
35	tLongitude0	В случае фиксации двухкамерного проезда по рубежам зоны (событие C0) или превышения средней скорости на участке зоны (событие C1) географическая долгота места установки первого рубежа зоны (целое число — количество угловых секунд). Координаты ведущего регистратора.
36	tLatitude	Географические координаты места установки комплекса: географическая широта (целое число — количество угловых секунд, Координаты ведомого регистратора.
37	tLongitude	Географические координаты места установки комплекса: географическая долгота (целое число — количество угловых секунд. Координаты ведомого регистратора.
38	nTZOffset	Часовой пояс места установки (по настройке сдвига времени регистраторов в ВЕБ-интерфейсе и config.json complex_utc_offset). Формат ±hh:mm
39	nZoneCount	Число зон контроля (целое число). Всегда 1.
40	ctimestamp	Дата и время последнего зафиксированного события. Формат ISO 8601: YYYY-MM-DDThh:mm:ss±hh
41	cpuload	% загрузки процессора (целое число 0-100)
42	freediskspace	% свободного дискового пространства (целое число 0-100)
43	uptime	время работы с последнего включения (целое число секунд)
44	freememory	% свободной оперативной памяти (целое число 0-100)

Пример ответа:


```
<?xml version="1.0" encoding="UTF-8"?>
<status>
  <timestamp>2020-08-11T04:22:38+03</timestamp>
  <tSVersion>1.3.1</tSVersion>
  <nDatetime>1597119758</nDatetime>
  <nWrongTurn>0</nWrongTurn>
  <nWrongTurnPosition>0</nWrongTurnPosition>
  <nOverSpeed>1</nOverSpeed>
  <nStopLine>0</nStopLine>
  <nRedLight>0</nRedLight>
```

<nParkALine>0</nParkALine>
<tStayRestrict>0</tStayRestrict>
<nWrongDirection>0</nWrongDirection>
<nWrongLine>0</nWrongLine>
<nTramways>0</nTramways>
<tPCross>0</tPCross>
<tPPrior>0</tPPrior>
<nNoLeftOnlyRightForward>0</nNoLeftOnlyRightForward>
<nNoLeftRightOnlyForward>0</nNoLeftRightOnlyForward>
<nNoUTurnOnlyRightForward>0</nNoUTurnOnlyRightForward>
<nNoUTurnOnlyForward>0</nNoUTurnOnlyForward>
<nProhibitorySignMove>0</nProhibitorySignMove>
<nMoveOnRedPereezd>0</nMoveOnRedPereezd>
<nNoLight>0</nNoLight>
<nNoForward>0</nNoForward>
<tRadarId>1810027/1601042</tRadarId>
<tChannel0>1810027</tChannel0>
<tChannel1>1601042</tChannel1>
<tChannel>1</tChannel>
<tSensorName0>Санкт-Петербург, ул.Победы</tSensorName0>
<tSensorName>Санкт-Петербург, пл.Мужества</tSensorName>
<tLocationCode>28000000020</tLocationCode>
<tRadarName>ДУЭТ</tRadarName>
<tCertificateExpDate>2018-06-02</tCertificateExpDate>
<tCertificateExpLimit>2021-06-06</tCertificateExpLimit>
<tLatitude0>216051</tLatitude0>
<tLongitude0>109490</tLongitude0>
<tLatitude>216053</tLatitude>
<tLongitude>109480</tLongitude>
<nTZOffset>+03:00</nTZOffset>
<nZoneCount>1</nZoneCount>
<ctimestamp>2020-08-06T10:20:47+03</ctimestamp>
<cpuload>0</cpuload>
<freediskspace>27</freediskspace>
<uptime>16</uptime>
<freememory>58</freememory>
</status>

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Инв. № подп

Ли	Изм.	№ докум.	Подп.	Дат

БКЮФ.201219.029РП

Лист

173

ПРИЛОЖЕНИЕ Б

Описание XML шаблонов фотоматериалов «ДУЭТ»

Версия XML 1.0.5

Б1. Описание

ПО «ДУЭТ» использует для составления основного фотоматериала специальные шаблоны, составленные с помощью языка разметки XML. В шаблоне описываются области текста и изображений, в которые ПО будет автоматически подставлять данные о нарушении.

Поддерживаемая версия ПО: **1.3.3+**

Б2. Формат шаблона

Шаблон редактируется в кодировке UTF-8.

В файле XML шаблона могут входить описания для разных фотоматериалов одновременно (например, для разных типов нарушений). На данный момент в ПО поддерживается только один вариант подвид шаблона:

- основной материал (коллаж);

Пример шаблона "ru.xml":

```
<?xml version="1.0"?>
<template type="collage" version="1.0.5">
    <canvas width="1280" height="560" background_color="0xFFFFFFFF">
        <object_image name="image_fr_labeled_D1">
            <source        type="image_fr_labeled_D1"        crop_top="0"
crop_bottom="0"/>
            <destination    x="0"    y="0"    width="639"    height="539"
insert_type="fill"/>
        </object_image>
        <object_image name="image_fr_labeled_D2">
            <source        type="image_fr_labeled_D2"        crop_top="0"
crop_bottom="0"/>
            <destination    x="640"    y="0"    width="640"    height="539"
insert_type="fill"/>
        </object_image>
        <object_text name="footer">
            <text         size="12"         background_color="0x000000FF"
color="0xFFFFFFFF"
                crop_if_long="false">ДУЭТ       ${serial_D1}/${serial_D2}
C${violation_code}   Измеренное расстояние — ${pass_distance} м   Время
проезда  —  ${pass_time}  с    Скорость  —  ${pass_average_speed}  км/ч
Разрешенная — ${allowed_speed} км/ч</text>
            <destination x="0" y="540" width="1280" height="20"/>
        </object_text>
    </canvas>
</template>
```

Шаблон состоит из блоков типа template с обязательными атрибутами:

- type — задает тип шаблона: "collage";
- version — указывает версию шаблона, в текущем примере — "1.0.5".

Понятия и обозначения:

- первый/второй регистратор (D1/D2) — указание на использование фото или параметра от первого или второго регистратора по порядку проезда ТС, постоянно меняются из-за разных направлений движения ТС;
- ведущий/ведомый регистратор (master/slave) — указание на использование фото или параметра от ведущего или ведомого регистратора по настройке ПО, постоянный порядок, изменяется только настройками роли регистраторов.

Большинство переменных и типов могут быть указаны с суффиксами D1/D2/ master/slave для получения требуемых данных. Далее введено обозначение для указания на любой подставляемый вариант суффикса с переменной, например для "name":

name_DEVICE — для name_D1 , name_D2 , name_master , name_slave.

Б2.1. Структура блока template

Содержимое:

- **<canvas>** — блок полотна

Пример:

```
<template type="collage" version="1.0.5">  </template>
```

Б2.2. Блок canvas

Полотно описывает положение и содержание всех <object_*> графических блоков на шаблоне.

Атрибуты:

- **width** — задает ширину результирующего изображения в пикселях;
- **height** — задает высоту результирующего изображения в пикселях;
- **background_color** — (необязательный) задает фон заливки для результирующего изображения в формате 0xRRGGBBAA, где компонента RR, GG, BB — задают яркость в оттенках серого, AA — не используются. По умолчанию: 0x000000FF.

Вложенные блоки:

- <object_image> ;
- <object_text> .

Пример:

```
<canvas width="3234" height="3828" background_color="0x000000FF">
</canvas>
```

Б2.3. Блок object_image

Блок описывает изображение, которое необходимо отобразить в указываемых координатах полотна шаблона. Используется для разных вариантов изображений из фотоматериалов регистраторов.

Пример — изображение кадра с первого регистратора:

```
<object_image name="image_fr_D1">
    <source type="image_fr_D1" crop_top="0" crop_bottom="0"/>
    <destination      x="20"      y="10"      width="600"      height="450"
insert_type="fit"/>
</object_image>
```

Атрибуты:

- **name** — (необязательный) задает пользовательское имя блока, используется при протоколировании подготовки шаблона и для удобства опознавания областей шаблона.

Вложенные блоки:

- **<source>** — описывает параметры используемого исходного изображения, которое будет использовано при отрисовке;
- **<destination>** — описывает положение на полотне шаблона, куда будет отрисовано исходное изображение из **<source>**.

Б2.3.1. Блок source

Параметры используемого исходного изображения, которое будет использовано при отрисовке.

Атрибуты:

- **type** — тип изображения.

Возможные значения:

- **image_fr_DEVICE** — фото от регистратора, без подписи;
- **image_fr_labeled_DEVICE** — фото от регистратора полное с подписью;
- **crop_top**, **crop_bottom**, **crop_left**, **crop_right** — (необязательные) задают параметры отсечения исходного изображения до его отрисовки.

Пример: при задании crop_top = " 300", crop_bottom = " 300", будет высчитана зона x = 0, y = 300, width = 2448, height = 1450 (2050-300-300).

Б2.3.2. Блок destination

Параметры положения на полотне шаблона, куда будет отрисовано исходное изображение из **<source>**.

Атрибуты:

- **x**,

  y,

  width,

  height — задают область отрисовки в пикселях на полотне;
- **insert_type** — (необязательный) задает тип вставки исходного изображения в указанную область целевого изображения. Значение по умолчанию — fill.

Возможные значения:

- **fill** — изображение будет вписано, соотношение сторон не сохраняются;

- **fit** — изображение будет вписано по большей стороне в центр целевой области с сохранением соотношения сторон, пустые области не будут закрашены;
- **crop** — изображение будет вписано по меньшей стороне с сохранением соотношения сторон, часть изображения будет отрезана.

Б2.4. Блок object_text

Описывает блок текста.

Пример:

```
<object_text name="footer">
    <text size="12" background_color="0x000000FF" color="0xFFFFFFFF"
        crop_if_long="false">ДУЭТ                ${serial_D1}/${serial_D2}
C${violation_code}  Измеренное расстояние — ${pass_distance} м  Время
проезда — ${pass_time} с  Скорость — ${pass_average_speed} км/ч
Разрешенная — ${allowed_speed} км/ч</text>
    <destination x="0" y="540" width="1280" height="20"/>
</object_text>
```

Атрибуты:

- **name** — (необязательный) задает пользовательское имя блока, используется при протоколировании подготовки шаблона и для удобства опознавания областей шаблона.

Вложенные блоки:

- **<text>** — текст, который должен быть отображен;
- **<destination>** — описывает положение на полотне шаблона, куда будет отрисован текст.

Б2.4.1. Блок text

Описывает блок текста.

Атрибуты:

- **size** — задает размер шрифта в пикселях;
- **background_color** — (необязательный) задает цвет фона в формате 0xRRGGBBAA, где компонента RR, GG, BB — задает яркость в оттенках серого, AA — не используются. По умолчанию равен 0x000000FF;
- **color** — (необязательный) задает цвет текста в формате 0xRRGGBBAA, где компонента RR — задает яркость в оттенках серого, AA,GG,BB — не используются. По умолчанию равен 0xFFFFFFFF;
- **crop_if_long** — (необязательный) отключает перенос по словам на следующую строку при достижении правой границы целевой области. Значения: true/false — не переносить/переносить;
- **font** — (необязательный) шрифт, путь к шрифту формата TTF. По умолчанию, значение "C:\Windows\Fonts\verdanab.ttf". Поиск шрифтов без указания пути в порядке:
  - ./ - папка установки ПО;
  - ./font/ - подпапка font от папки ПО;
  - C:/Windows/Fonts/ ;

- {CURRENT_TEMPLATE_PATH}/ — папка текущего шаблона.

Текст может содержать переменные, формат **${var_name}**, где var_name — имя переменной. Список доступных системных переменных приведен ниже.

Б2.4.2. Блок destination

Блок описывает положение на полотне шаблона, куда будет отрисован текст.

Атрибуты:

- **x**,
- **y**,
- **width**,
- **height** — задают область отрисовки в пикселях на полотне.

ВНИМАНИЕ! Текст не может выйти за пределы области, заданной в destination. При превышении часть текста будет отрезана.

Б2.4.3. Переменные в тексте

Переменные для размещения в блоке текста <object_text>.

Формат написания: **${var_name}**,

где var_name — имя переменной.

Список доступных переменных с примерами:

- **serial_DEVICE** — серийный номер регистратора (например, 1900001);
- **type_DEVICE** — название типа регистратора (например, СКАТ, ОРАКУЛ и т.п.);
- **location_latitude_prefix_DEVICE** — префикс широты из координат регистратора — "N";
- **location_latitude_DEVICE** — широта из координат регистратора. Формула ${location_latitude_prefix_DEVICE }${location_latitude_value_DEVICE }. Формат "N%08.5f" или "N%09.5f" в зависимости от величины: N60.00053;
- **location_latitude_value_DEVICE** — значение широты из координат регистратора. Формат "%08.5f": 60.00053;
- **location_longitude_prefix_DEVICE** — префикс долготы из координат регистратора — "E";
- **location_longitude_DEVICE** — долгота из координат регистратора. Формула ${location_longitude_prefix_DEVICE }${location_longitude_value_DEVICE }. Формат "E%09.5f" или "E%08.5f" в зависимости от величины: E30.21585;
- **location_longitude_value_DEVICE** — значение долготы из координат регистратора. Формат "%09.5f": 30.21585;
- **target_grz_DEVICE** — исходный текст номерной пластины автомобиля (ГРЗ): p166ку|177;

- **target_grz_alphanum_DEVICE** — текст номерной пластины автомобиля (ГРЗ) без разделительных символов: р166ку177;
- **target_grz_upper_DEVICE** — текст номерной пластины автомобиля (ГРЗ) в верхнем регистре: Р166КУ|177;
- **target_grz_alphanum_upper_DEVICE** — текст номерной пластины автомобиля (ГРЗ) без разделительных символов в верхнем регистре: Р166КУ177;
- **target_grz_lat_DEVICE** — текст номерной пластины автомобиля (ГРЗ) латинскими буквами в верхнем регистре: P166KY|177;
- **target_grz_alphanum_lat_DEVICE** — текст номерной пластины автомобиля (ГРЗ) без разделительных символов латинскими буквами в верхнем регистре: P166KY177;
- **target_direction_DEVICE** — направление проезда ТС: Встречное;
- **target_speed_DEVICE** — скорость автомобиля (км/ч): 16;
- **target_time_DEVICE** — время регистрации проезда ТС;
- **incoming_description** — текстовое описание направления дороги, в котором направлен ведущий регистратор системы (из настроек ПО): "на Тихорецкий пр.";
- **outcoming_description** — текстовое описание направления дороги, обратное тому, в котором направлен ведущий регистратор системы (из настроек ПО): "на пр. Непокорённых";
- **direction_description** — текстовое описание направления, в котором проехало ТС по отношению к ведущему регистратору системы. Используется одно из значений ${incoming_description} или ${outcoming_description}. Пустая строка — при отсутствующем направлении: "на Тихорецкий пр.";
- **place_location** — адрес расположения системы: "Санкт-Петербург, ул. Политехническая 21";
- **place_location_and_direction** — адрес расположения системы с направлением движения через запятую: "Санкт-Петербург, ул. Политехническая 21, на Тихорецкий пр.";
- **allowed_speed_auto** — разрешенная скорость для автомобилей с типом зафиксированного ТС в полосе движения транспортного средства, для регистраторов без классификации соответствует **allowed_speed**: 60;
- **allowed_speed** — разрешенная скорость для легковых автомобилей в полосе движения транспортного средства: 60;
- **allowed_speed_truck** — разрешенная скорость для грузовых автомобилей в полосе движения транспортного средства: 55;
- **pass_distance** — пройденное ТС расстояние между регистраторами (м, формат "%.2f"): 4100.04;
- **pass_time** — пройденное ТС расстояние между регистраторами (м, формат "%.3f"): 32.28;
- **pass_average_speed** — средняя скорость автомобиля (км/ч): 87;
- **violation_code** — текстовый код нарушения: "1.2";

Пример полученного фотоматериала с использованием шаблона ru.xml приведен на рис. Б1.

Рисунок Б1 — Пример результата материала по шаблону ru.xml

ПРИЛОЖЕНИЕ В

Описание алгоритма расчета контрольной суммы

метрологической части ПО «ДУЭТ»

Версия ПО: 1.2+

В данном приложении приводится подробное описание алгоритма, который используется для расчета контрольной суммы метрологически значимой части ПО «ДУЭТ», а также описание процедуры проверки контрольной суммы в ПО.

В1. Библиотека метрологической части ПО

В качестве метрологически значимой части ПО выступает библиотека: **avgspeed.**

Именование файла библиотеки под разные операционные системы (далее — ОС):

- Windows: avgspeed.dll
- Linux: **avgspeed.so**

Библиотека распространяется в дистрибутиве ПО под соответствующую ОС.

Несанкционированная модификация библиотеки метрологически значимой части, т.е. проверка целостности ПО, проверяется расчетом контрольной суммы библиотеки с помощью алгоритма циклического кода CRC-32 и сравнением ее с номинальным значением.

Процесс проверки контрольной суммы в ПО в начале работы:

1. Чтение данных файла библиотеки **avgspeed** с диска;
2. Вычисление контрольной суммы CRC-32 библиотеки **avgspeed** ;
3. Сравнение вычисленного значения контрольной суммы с номинальным, заложенным в ПО;
4. Если вычисленное значение контрольной суммы **соответствует** номинальному, то производится:

- вывод сообщений об успешной проверке на экран;
- запись сообщений об успешной проверке в файл протокола работы (core.log):

```
2020-07-08 19:10:58.691 [I]: TargetInspector: Calculated checksum for library 'avgspeed.dll'. CRC-32: 'BD60FE02'
2020-07-08 19:10:58.711 [I]: TargetInspector: Load library 'avgspeed.dll' OK
```

- продолжение работы ПО.

5. Если вычисленное значение контрольной суммы **отличается** от номинального, то производится:

- вывод сообщения об ошибке (тип C — Critical) на экран;
- запись сообщения об ошибке (тип C — Critical) в файл протокола работы (core.log):

```
2020-07-10 15:52:24.414 [I]: TargetInspector: Calculated checksum for library 'avgspeed.dll'. CRC-32: 'FAC7F2AF'
```

```
2020-07-10 15:52:24.414 [C]: TargetInspector: CRC-32 mismatch for library
'avgspeed.dll'! Calculated: 'FAC7F2AF', Reference: 'BD60FE02'
```

- завершение работы ПО.

В2. Алгоритм расчета контрольной суммы

В качестве алгоритма вычисления контрольной суммы используется алгоритм вычисления CRC (циклический избыточный код).

В2.1. Параметры алгоритма

Вид: CRC-32.

Согласно спецификации алгоритма CRC по методике Ross N. Williams, используется алгоритм CRC-32 со следующими параметрами:

- Степень многочлена (WIDTH):
  32
- Полином (POLY):
  0x4C11DB7
  g(X) = x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10 + x^8 + x^7 + x^5 + x^4 +
  + x^2 + x + 1
- Начальное (стартовое) слово (INIT):
  0xFFFFFFFF
- Начало и направление вычислений (порядок обработки битов, REFIN):
  True — с младшего бита (LSB-first)
- Порядок битов регистра при входе на элемент XOR (REFOUT):
  True — инвертируется
- Число, с которым складывается по модулю 2 полученный результат (XOROUT):
  0xFFFFFFFF
- Проверочное значение CRC для строки «123456789» (CHECK):
  0xCBF43926

Данный вид CRC-32 рекомендуется IEEE, также именуется как: CRC-32-IEEE802.3, CRC-32/ADCCP, PKZIP, CRC-32/ISO-HDLC, CRC-32/V-42, CRC-32/XZ.

В2.2. Алгоритм

Ниже приведено описание простой версии алгоритма.

Используется входная последовательность байт I.

1. В 32-битный регистр REM заносится значение INIT (32 бита);
2. Берется 1 байт входной последовательности I и помещается в младшие разряды регистра IR (32 бита);

3. Если REFIN=TRUE, тогда младший байт IR отражается младшими байтами вперед (LSB-first):

IR = reflect(IR, 8)

иначе остается без изменений;

4. Младший байт IR сдвигаем в сторону старших разрядов:

IR = IR << (WIDTH — 8)

5. Полученное значение складываем по модулю 2 с REM:

REM = REM ^ IR

6. Для каждого бита из старшего байта REM выполняем:

Если старший бит REM = 1, то сдвигаем REM на 1 бит в сторону старших разрядов и складываем по модулю 2 с POLY:

REM = (REM << 1) ^ POLY

иначе просто сдвигаем REM на 1 бит в сторону старших разрядов:

REM = REM << 1

7. Возвращаемся на Шаг 2 и повторяем описанный алгоритм для каждого байта исходной последовательности I;

8. После этого производим финальное сложение по модулю 2:

Если REFOUT=TRUE, тогда отражается REM младшими байтами вперед (LSB-first) и складываем по модулю 2 с XOROUT:

REM = XOROUT ^ reflect(REM, WIDTH)

иначе просто складываем по модулю 2 REM с XOROUT:

REM = XOROUT ^ REM

9. После данных шагов REM содержит итоговую контрольную сумму CRC-32.

В2.3. Пример исходного кода

Пример исходного кода расчёта CRC-32 на языке C/C++ вычисления для входной последовательности "123456789" представлен в Листинге В1.

Листинг В1 — Исходный код расчёта CRC-32 на языке C/C++

```
#include <cstdio>

#define CHECK 0xCBF43926
#define BITMASK(X) (1L << (X))
#define MASK32 0xFFFFFFFFL

struct cm_t
{
      int cm_width;                       /* Parameter: Width in bits [8,32].
*/
      unsigned long cm_poly;        /* Parameter: The algorithm's polynomial.
*/
      unsigned long cm_init;        /* Parameter: Initial register value. */
      bool cm_refin;                      /* Parameter: Reflect input bytes? */
      bool cm_refot;                      /* Parameter: Reflect output CRC? */
```

```
	unsigned long cm_xorot;	/* Parameter: XOR this to output CRC. */
	unsigned long cm_reg;		/* Context: Context during execution.
*/
};

// Example: reflect(0x3e23L,3) == 0x3e26
unsigned long reflect(unsigned long v, int b)
{
	unsigned long t = v;
	for (int i = 0; i < b; i++)
	{
		if (t & 1L)
			v |= BITMASK((b-1)-i);
		else
			v &= ~BITMASK((b-1)-i);
		t >>= 1;
	}
	return v;
}

// Returns a longword whose value is (2^p_cm->cm_width)-1
// The trick is to do this portably (e.g. without doing <<32)
unsigned long widmask(cm_t* p_cm)
{
	return (((1L << (p_cm->cm_width – 1)) – 1L) << 1) | 1L;
}

void cm_nxt(cm_t* p_cm, unsigned long ch)
{
	unsigned long topbit = BITMASK(p_cm->cm_width-1);

	if (p_cm->cm_refin)
		ch = reflect(ch, 8);
	p_cm->cm_reg ^= (ch << (p_cm->cm_width – 8));

	for (int i = 0; i < 8; i++)
	{
		if (p_cm->cm_reg & topbit)
			p_cm->cm_reg = (p_cm->cm_reg << 1) ^ p_cm->cm_poly;
		else
			p_cm->cm_reg <<= 1;
		p_cm->cm_reg &= widmask(p_cm);
	}
}

void cm_blk(cm_t* p_cm, unsigned char* blk_adr, unsigned long blk_len)
{
	while (blk_len--)
		cm_nxt(p_cm, *blk_adr++);
}

void cm_crc(cm_t* p_cm)
{
	if (p_cm->cm_refot)
		p_cm->cm_reg  =  p_cm->cm_xorot  ^  reflect(p_cm->cm_reg,  p_cm-
>cm_width);
```

```
	else
		p_cm->cm_reg = p_cm->cm_xorot ^ p_cm->cm_reg;
}

int main()
{
	cm_t model{};
	model.cm_width = 32;
	model.cm_poly = 0x4C11DB7;
	model.cm_init = 0xFFFFFFFF;
	model.cm_refin = true;
	model.cm_refot = true;
	model.cm_xorot = 0xFFFFFFFF;

	model.cm_reg = model.cm_init;

	unsigned char data[9] = { '1', '2', '3', '4', '5', '6', '7', '8', '9' };

	cm_blk(&model, data, 9);
	cm_crc(&model);

	printf("CRC-32 = %lx\n\n", model.cm_reg);

	if(model.cm_reg == CHECK)
		printf("CHECK OK\n");

	return 0;
}
```

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий номер сопроводительного документа	Подпись	Дата
	измененных	заменен-ных	новых	анну-лирован-ных					