

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭЛИКОНТ-КС

Руководство администратора

АДИГ.00024.32

Версия 2.8

Листов 52

ВВЕДЕНИЕ

Настоящий документ содержит сведения об установке и сопровождении Эликонт-КС. Данный документ является эксплуатационным документом и предназначен для администратора Эликонт-КС.

Данное Руководство предполагается использовать совместно с Руководством пользователя, которое содержит информацию о диагностике, журналировании событий и необходимые сведения о программе.

Исключительные права на программное обеспечение Эликонт-КС и все его компоненты принадлежат АО «ЭЛАРА».

АО «ЭЛАРА» оставляет за собой право изменять информацию в настоящем Руководстве без предварительного уведомления пользователей.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	2
1 НАЗНАЧЕНИЕ ПО	5
2 СОСТАВ ПО И СХЕМЫ РАЗВЕРТЫВАНИЯ	5
3 ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ .7	
3.1 Требования к настройке TCP-портов компьютера	7
3.1.1 Сервисные порты для внешних подключений.....	7
3.1.2 Сервисные порты для служебных целей	7
3.1.3 Изменение сервисных портов.....	8
3.1.4 Прикладные порты	9
3.2 Требования к аппаратному и программному обеспечению.....	10
4 КОНТРОЛЬ ЦЕЛОСТНОСТИ ПО ЭЛИКОНТ-КС В ОС ASTRA LINUX.....	11
5 УСТАНОВКА И ЗАПУСК ПО.....	13
5.1 Установка Эликонт- КС в ОС MS Windows	13
5.1.1 Восстановление Эликонт-КС в MS Windows.....	16
5.1.2 Запуск Коммуникационного ядра в ОС MS Windows.....	18
5.1.3 Запуск Конфигуратора в ОС MS Windows	18
5.2 Установка Эликонт-КС в ОС Linux	18
5.2.1 Запуск Коммуникационного ядра в ОС Linux	19
5.2.2 Запуск Конфигуратора в Linux.....	20
5.3 Удаление Эликонт-КС.....	20
6 НАСТРОЙКА ПОДКЛЮЧЕНИЯ К DOCKER КОНТЕЙНЕРУ	21
6.1 Настройка и запуск Docker контейнера	21
6.1.1 Загрузка образа из сетевого хранилища	21
6.1.2 Загрузка образа из локального хранилища	21
6.1.3 Создание и запуск контейнера из образа.....	22
6.2 «Проброс» аппаратного (USB) ключа в Docker Desktop (Windows).....	25
6.3 Настройка подключения Docker контейнера по протоколу SNMP для Windows	27
6.4 Проверка подключения Конфигуратора к Docker контейнеру.....	27
6.4.1 Проверка подключения	27
6.4.2 Проверка «проброса» ключа.....	28
6.5 Основные команды для управления контейнером	28
7 АКТИВАЦИЯ ПРОГРАММНОГО КЛЮЧА	29
7.1 Активация лицензии по файлу запроса	29
7.2 Обновление лицензии по файлу запроса	32
7.3 Перенос лицензии на другой компьютер	32
7.4 Просмотр журнала событий мастера	33
8 СОПРОВОЖДЕНИЕ ПО.....	34

8.1	Синхронизация системного времени	34
8.2	Администрирование в Web-интерфейсе КС	34
8.2.1	Общие сведения	34
8.2.2	Сброс конфигурации КС	35
8.2.3	Настройка пользователей.....	36
8.2.4	Просмотр сведений об интерфейсе.....	37
8.3	Настройка клиентского подключения OPC UA в сети с компьютерами с ОС Linux без организации домена	37
8.4	Обновление версий приложений	40
8.5	Описание конфигурационных файлов	40
8.6	Сведения об архивировании сигналов	41
ПРИЛОЖЕНИЕ А		42
	Описание файла статической конфигурации КС для ОС Windows и Linux.....	42
ПРИЛОЖЕНИЕ В.....		48
	Структура БД архива PostgreSQL	48
	Структура БД архива SQLite.....	49
ПРИЛОЖЕНИЕ С		51
	Полезные ссылки.....	51
	Подключение к Web-интерфейсу.....	51
	Просмотр версии ПО	51
	Просмотр лицензии	51
	Диагностика ПК с установленным Коммуникационным ядром.....	51
	Просмотр логов.....	51
	Размещение архивной БД SQLite.....	51
	Размещение конфигурационных файлов КС	52

1 НАЗНАЧЕНИЕ ПО

Эликонт-КС - коммуникационное программное обеспечение, предназначенное для построения различных систем сбора, хранения и передачи данных.

2 СОСТАВ ПО И СХЕМЫ РАЗВЕРТЫВАНИЯ

Эликонт-КС состоит из приложений: Коммуникационное ядро и Конфигуратор.

Коммуникационное ядро (далее КС) обеспечивает обмен данными от источников данных к приёмникам по ряду протоколов. *Конфигуратор* - приложение для настройки, управления и отображения данных Коммуникационного ядра. Конфигуратор загружает проект из локального файла, такой проект называется *локальным*.

Примечание - Проект - описание конфигурации Коммуникационного ядра (или нескольких Коммуникационных ядер). Проект содержит конфигурации всех устройств системы.

Модульная архитектура КС позволяет разворачивать системы разного масштаба: от простых шлюзов на несколько десятков сигналов, до сложных, распределенных систем с десятками тысяч изменений в секунду.

На Рисунках 1 и 2 показаны варианты развертывания Эликонт-КС. При распределенной установке Коммуникационное ядро и Конфигуратор устанавливаются на разные компьютеры. При минимальной установке оба приложения устанавливаются на один компьютер.

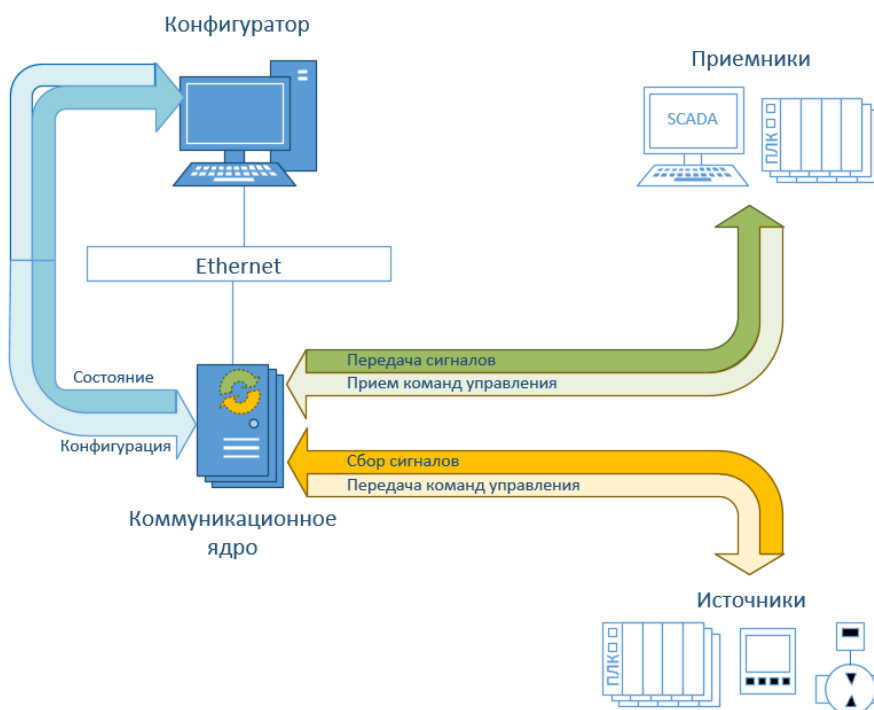


Рисунок 1 – Распределенная установка

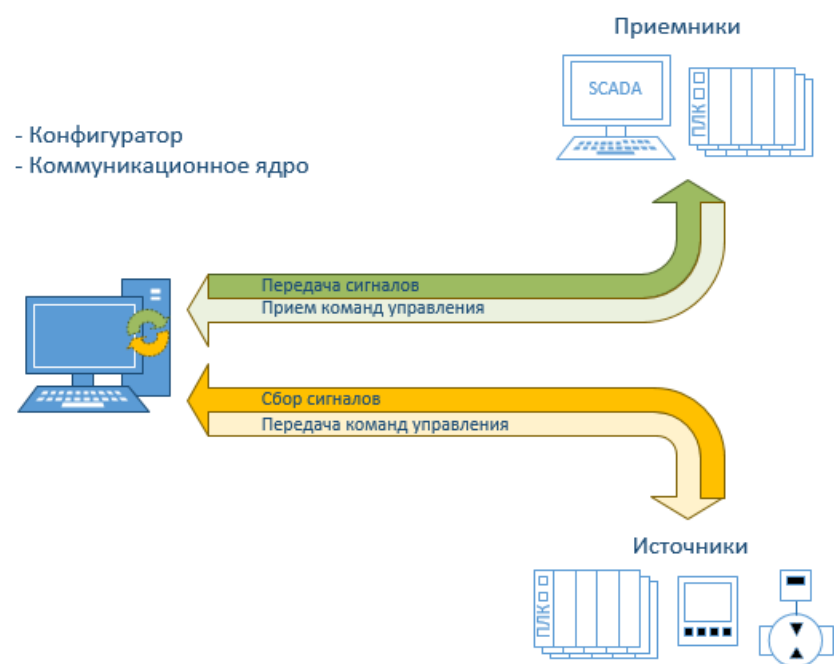


Рисунок 2 – Минимальная установка

3 ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

3.1 Требования к настройке TCP-портов компьютера

На компьютере с установленным Коммуникационным ядром используются:

- сервисные порты для внешних подключений и служебных целей,
- прикладные порты для работы серверных каналов.

Важно! Сервисные и прикладные порты должны быть всегда открыты.

3.1.1 Сервисные порты для внешних подключений

Список используемых портов приведен в Таблице 1. Все они доступны для изменения в файле *static_config*. При изменении портов 8080 и 9039, соответствующие изменения должны быть внесены и в Конфигураторе, см. раздел 3.1.3.

Таблица 1 – Сервисные порты для внешних подключений

Порт	Назначение порта	Изменение порта
1	2	3
5401	Подключение к WEB-серверу	В элементе <Monitor> <Services> файла <i>static_config</i>
8080	Загрузка конфигурации	В КС (Конфигуратор) и в элементе <API Gateway> файла <i>static_config</i>
8089	Формирование лог-сообщений	В элементе <Logger> файла <i>static_config</i>
9039	Передача оперативных данных в Конфигуратор в режиме «Исполнение»	В КС (Конфигуратор) и в элементе <Channels> файла <i>static_config</i>

3.1.2 Сервисные порты для служебных целей

Список используемых портов приведен в Таблице 2. Все они доступны для изменения в файле *static_config*, за исключением порта 9040. См. раздел 3.1.3.

Таблица 2 – Сервисные порты для служебных целей

Порт	Назначение порта	Изменение порта
1	2	3
4242 4343 8081 8084 8085 9084	Внутреннее использование в КС	В файле <i>static_config</i>

Порт	Назначение порта	Изменение порта
1	2	3
9085		
8083	Сервис архива	
8086	Сервис лицензирования	
9040	Внутреннее использование в КС	Не подлежит изменению

3.1.3 Изменение сервисных портов

Для изменения портов выполнить действия:

1 изменить порты 8080, 9039 в режиме «Конфигурация» на уровне КС в Конфигураторе:

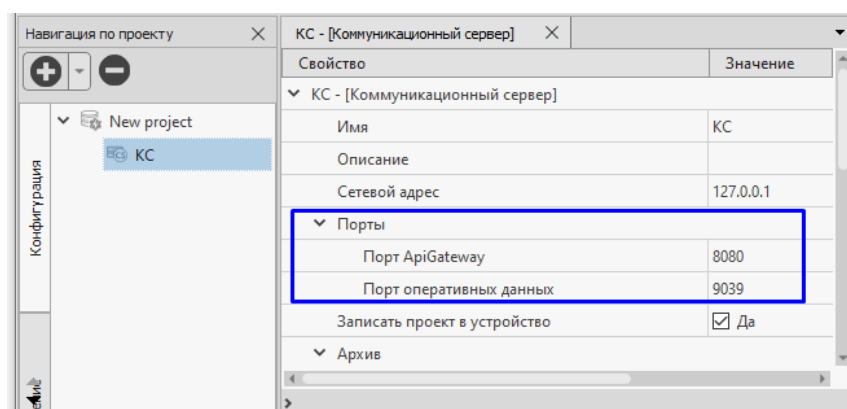


Рисунок 3 – Порты для КС

2 остановить службу ElaraCS;

3 открыть системный файл *static_config* (см. Примечание, далее) от имени администратора (например, в приложении Notepad++). Изменить порты в указанных элементах, согласно Таблице 1 и Таблице 2.

Важно! Сервисный порт 8085 указан в файле *static_config* в нескольких элементах. Необходимо убедиться, что выполнена сквозная замена порта по всему файлу.

При изменении порта 8085, необходимо заменить атрибут запуска службы ElaraCS. Для этого в командной строке от имени администратора выполнить команду (пример для Windows):

```
sc config ElaraCS binPath= "C:\Program Files\ELARA\EliCont-CS\services\sm_host.exe
127.0.0.1:8033 \"C:\Program Files\ELARA\EliCont-CS\services\pcfg_host.exe\" \"C:\Program
Files\ELARA\EliCont-CS\config\static_config_win.xml\" \"C:\Program Files\ELARA\EliCont-
CS\config\user_config.xml\" -s=ElaraCS -a=15"
```

где в данном примере:

C:\Program Files\ELARA – место установки Коммуникационного ядра,

127.0.0.1:8033 – атрибут запуска службы (вместо 8085 - порт 8033).

Примечание - Системный конфигурационный файл *static_config* находится на компьютере с установленным Коммуникационным ядром. Папка с файлом: <место установки программы>\EliCont-CS\config. Имя файла:

OC Windows	<i>static_config_win.xml</i>
OC Linux	<i>static_config_lin.xml</i>

Описание файла приведено в разделе 8.5.

- 4 запустить службу ElaraCS;
- 5 проверить работоспособность программы. Порядок действий:
 - а) загрузить конфигурацию в Коммуникационное ядро (КС -> пункт контекстного меню «Загрузить конфигурацию»). Должно появиться уведомление, что конфигурация загружена успешно;
 - б) подключиться к КС в режиме «Исполнение». Убедиться, что сигналы/команды обновляются.;
 - с) открыть лог-файл КС (см. раздел «Просмотр логов»). Убедиться в отсутствии сообщений об ошибках.

3.1.4 Прикладные порты

Прикладные порты используются для обмена данными по протоколам телемеханики, см. Таблицу 3. Используемые порты настраиваются в конфигурации КС на уровне канала в Конфигураторе.

Таблица 3 – Прикладные порты

Порт	Протоколы для установки соединения	Изменение порта
1	2	3
102	МЭК 61850 MMS, S7comm	Для канала (в Конфигураторе)
161	SNMP	
502	Modbus TCP	
1883	MQTT	
2404	МЭК 60870-5-104	
9600 (UDP)	Omron FINS	
48010	OPC UA	

3.2 Требования к аппаратному и программному обеспечению

Минимальные требования к аппаратному и программному обеспечению при установке Эликонт-КС приведены в Таблице 4.

Таблица 4 – Перечень требований

Технические характеристики	Коммуникационное ядро					Конфигуратор				
ОС	Windows 10	Astra Linux 1.8	Debian 11	РОСА Хром 12	РЕД ОС	Windows 10	Astra Linux 1.8	Debian 11	РОСА Хром 12	РЕД ОС
Аппаратная платформа (процессор)	x86, x64	i386 AMD 64	i386 AMD64 ARM ARM64	i386 x86_64	i386 x86_64	x86, x64	AMD64		x86_64	
Процессор	От 1 ГГц					От 2 ГГц				
Оперативная память	512 Мб					От 4 Гб				
Свободное пространство ПЗУ	1 Гб					5 Гб				
Браузер	Firefox 78 и выше, Google Chrome 113 и выше					-				
Видеокарта	Встроенная					Встроенная				

Для архивирования значений сигналов в сетевую БД используется приложение PostgreSQL, версия - не ниже 16.4.

4 КОНТРОЛЬ ЦЕЛОСТНОСТИ ПО ЭЛИКОНТ-КС В ОС ASTRA LINUX

Контроль целостности файлов необходим для повышения защищенности операционной системы и выполняется в режиме замкнутой программной среды (далее ЗПС).

Режим ЗПС обеспечивается проверкой электронной подписи файлов (ЭП) с помощью невыгружаемого ядра Astra Linux (модулем `digsig_verif`). Исполнение файлов с электронными подписями, не прошедшими проверку, не производится. Этот режим гарантирует обнаружение в исполняемом коде приложения изменений, несанкционированных производителем ПО.

Проверка целостности ПО Эликонт-КС применяется к исполняемым файлам формата ELF и файлам совместно используемых библиотек (.SO). В специальном поле исполняемых файлов встроена ЭП. Для активации механизма проверки ЭП необходимо выполнить следующие действия от имени администратора:

- 1 в конфигурационном файле ОС: `/etc/digsig/digsig` установить параметр:
`DIGSIG_ELF_MODE=1` (для Astra Linux 1.7).

«1» — штатный режим проверки встроенной ЭП. Проверка встроенной ЭП включена. Выполнение файлов, имеющих неверную встроенную ЭП или не имеющих встроенную ЭП запрещается.

- 2 скопировать публичный ключ `elara_pub.key` в каталог дополнительных ключей `/etc/digsig/keys` с помощью команды:

```
sudo cp /<каталог>/elara_pub.key /etc/digsig/keys/
```

Ключ предоставляется АО «НПО РусБИТех».

- 3 для того, чтобы изменения, сделанные в каталоге `/etc/digsig` (в том числе, изменения, сделанные в конфигурационном файле), активировались, выполнить команду:

```
sudo update-initramfs -u -k all
```

- 4 перезагрузить компьютер;

- 5 проверить запуск Конфигуратора и загрузку конфигурации проекта в Коммуникационное ядро.

В случае ошибки при проверке ЭП файлов Конфигуратора, при запуске Конфигуратора появляется всплывающее уведомление Системы защиты ОС (Рисунок 4), сообщение об ошибке сохраняется в журнале событий ОС (Рисунок 5).

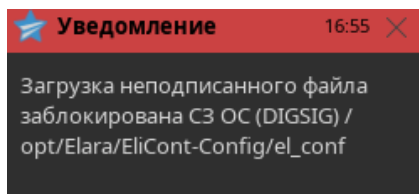


Рисунок 4 – Уведомление о блокировке загрузки файла

Время регистрации события	2025-01-29T16:56:48+03:00
Группа	elicont
Имя события	Загрузка неподписанного файла заблокирована СЗ ОС (DIGSIG)

Рисунок 5- Запись в журнале событий ОС о блокировке

В случае ошибки при проверке ЭП файлов Коммуникационного ядра его запуск не выполняется (Рисунок 6), сообщение об ошибке сохраняется в журнале событий ОС.

```

root@astra-48462:~# systemctl status ElaraCS
* ElaraCS.service - ElaraCS application
   Loaded: loaded (/lib/systemd/system/ElaraCS.service; enabled; preset: enabled)
   Active: failed (Result: signal) since Wed 2025-01-29 16:56:48 MSK; 18s ago
     Duration: 2.023s
   Process: 3171 ExecStart=/opt/Elara/EliCont-CS/services/sm_host 127.0.0.1:8085 /opt/Elara/EliCont-CS/servi>
   Main PID: 3171 (code=killed, signal=SEGV)
      CPU: 2.023s

янв 29 16:56:46 astra-48462 systemd[1]: Started ElaraCS.service - ElaraCS application.
янв 29 16:56:48 astra-48462 systemd[1]: ElaraCS.service: Main process exited, code=killed, status=11/SEGV
янв 29 16:56:48 astra-48462 systemd[1]: ElaraCS.service: Failed with result 'signal'.
янв 29 16:56:48 astra-48462 systemd[1]: ElaraCS.service: Consumed 2.023s CPU time.

```

Рисунок 6 - Служба ElaraCS была остановлена

5 УСТАНОВКА И ЗАПУСК ПО

5.1 Установка Эликонт- КС в ОС MS Windows

Установка Эликонт-КС в среде MS Windows должна производиться пользователем с правами локального администратора.

Установка выполняется в следующей последовательности:

- 1 запустить из дистрибутива установочный файл:

EliCont-CS -<разрядность процессора> <язык> <версия>.msi

Например, EliCont-CS-x86-ru-2.0.0.msi

- 2 в окне установки нажать кнопку «Далее» (Рисунок 7);

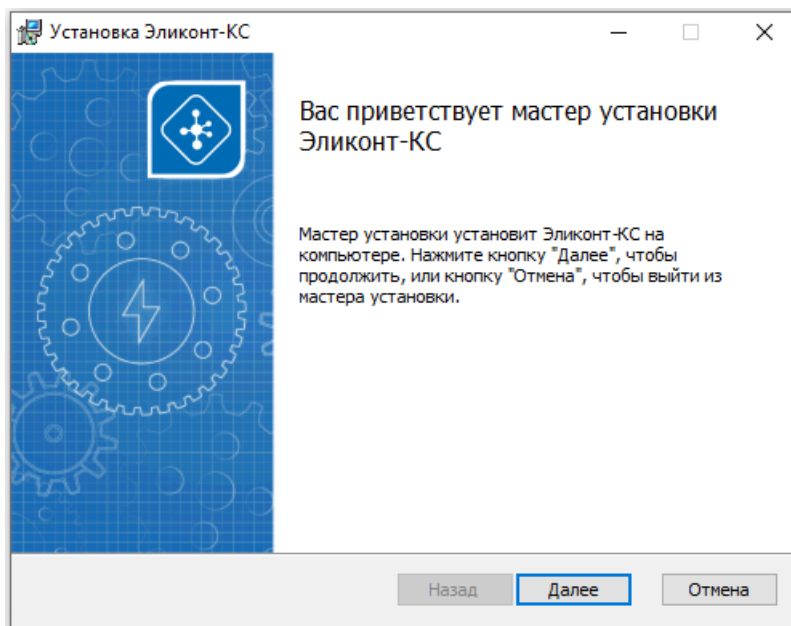


Рисунок 7 – Окно установки

- 3 во втором окне установки принять условия лицензионного соглашения;
- 4 в третьем окне указать путь к месту установки (Рисунок 8), по умолчанию :

<место установки> \ELARA

Нажать кнопку «Далее»;

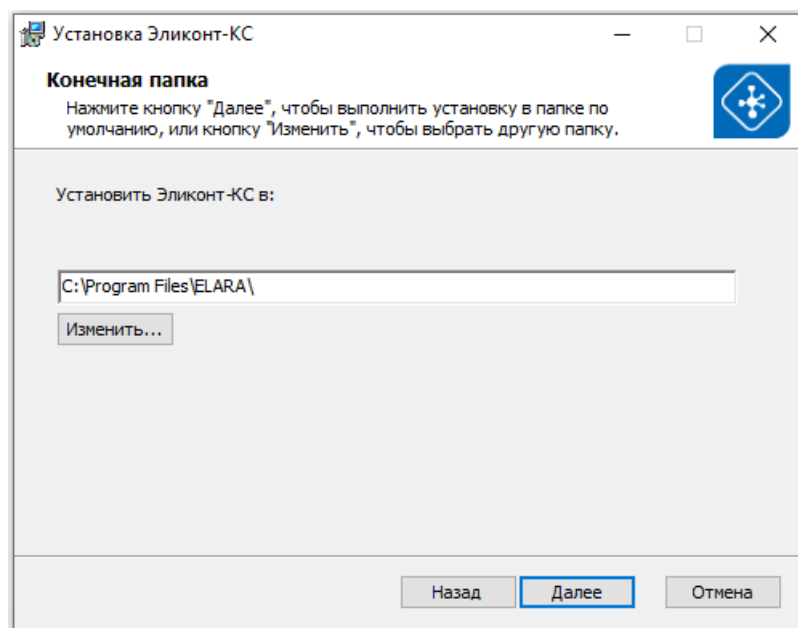


Рисунок 8 – Путь к месту установки программы

5 в следующем окне указать конфигурацию установки компонентов (Рисунок 9) и нажать кнопку «Установить».

Слева от каждого компонента в окне отображается значок . Для отмены установки компонента щелкнуть мышью по этому значку (или нажать клавишу Пробел) и в появившемся меню выбрать пункт:  Компонент будет полностью недоступен.

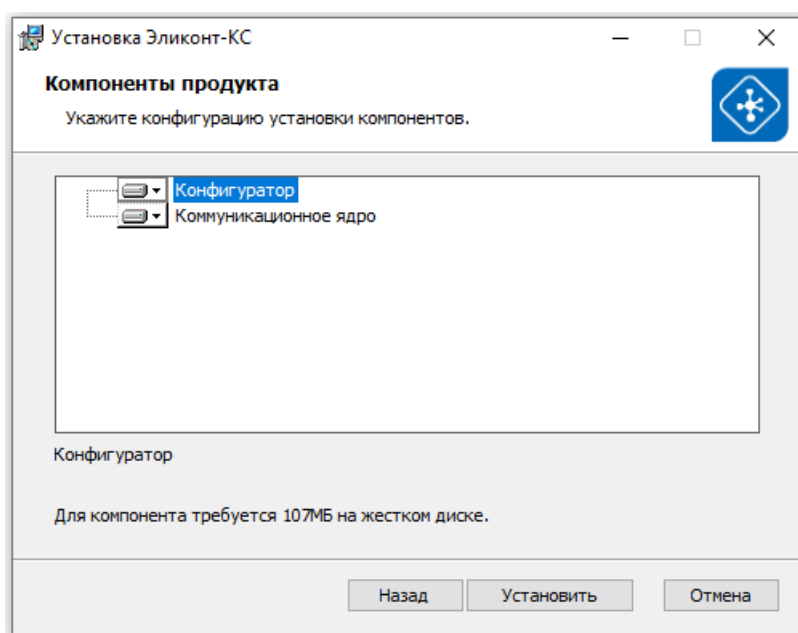


Рисунок 9 – Выбор конфигурации установки

6 в следующем окне (Рисунок 10) подтвердить начало установки. Начинается процесс установки (Рисунок 11);

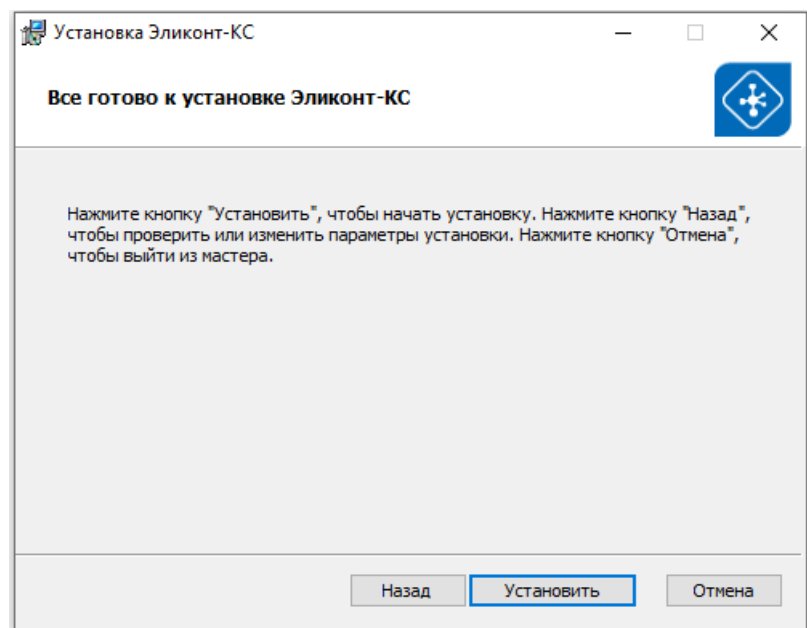


Рисунок 10 - Начать установку

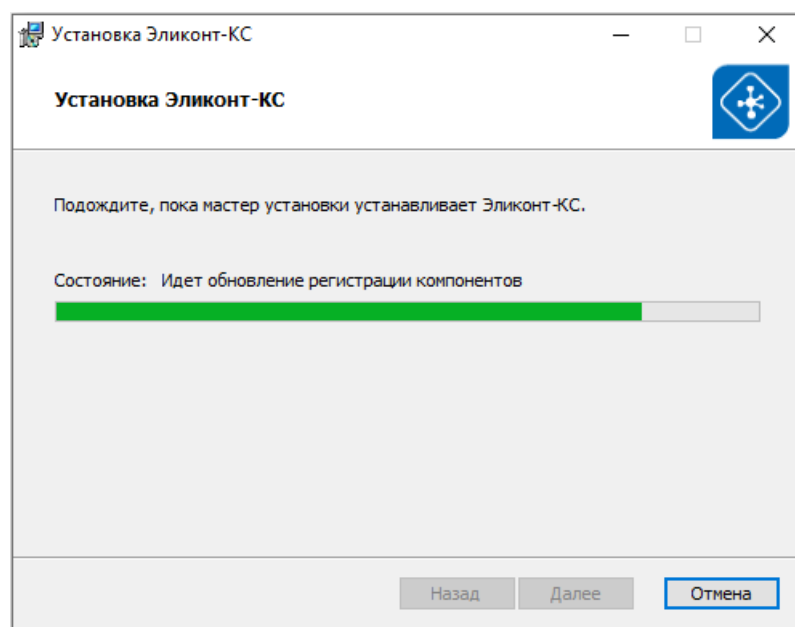


Рисунок 11 – Состояние установки

7 в заключительном окне установки нажать кнопку «Готово» (Рисунок 12). Процесс установки закончен.

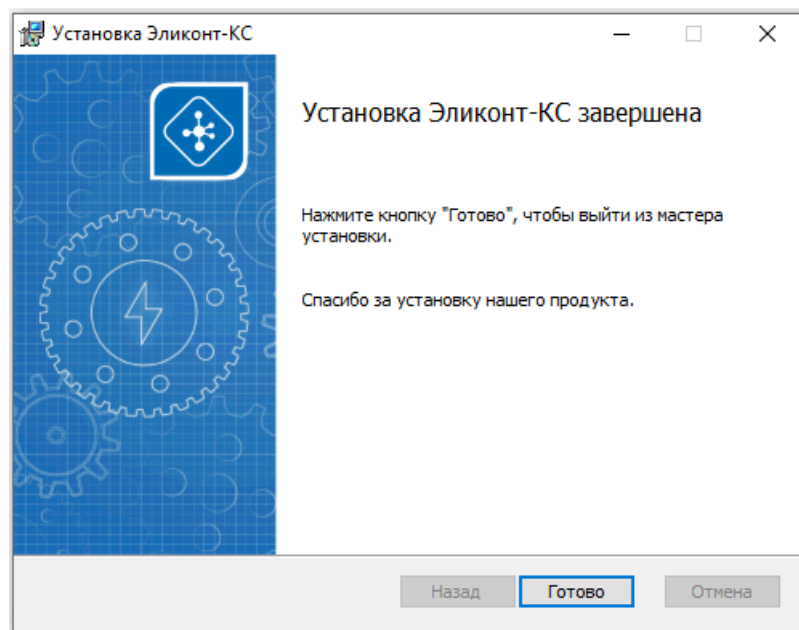


Рисунок 12 – Заключительное окно

В результате установки Коммуникационного ядра создается и автоматически запускается служба ElaraCS (Рисунок 13).

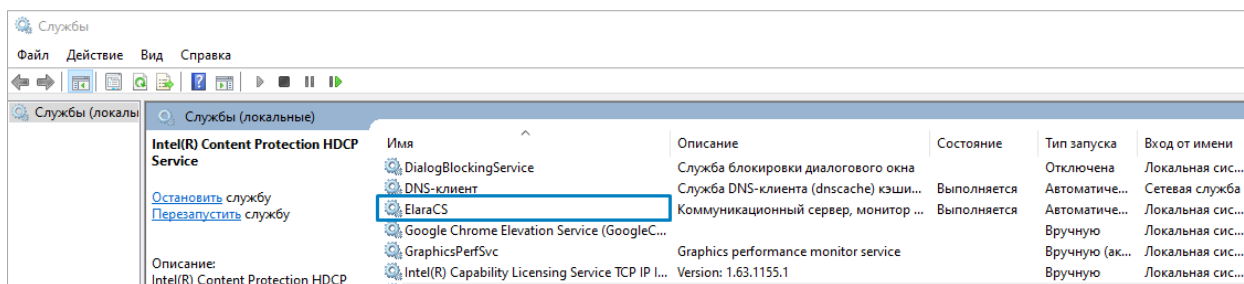


Рисунок 13 – Служба после установки Эликонт-КС

8 если на компьютер требуется установить недостающий компонент, необходимо запустить дистрибутив и в окне «Изменение, восстановление или удаление установки» нажать кнопку «Изменить» и указать компонент для установки.

5.1.1 Восстановление Эликонт-КС в MS Windows

Восстановление Эликонт-КС выполняется для исправления ошибок при последней установке путем восстановления отсутствующих и поврежденных файлов, ярлыков и записей реестра.

Для выполнения восстановления приложения:

- 1 запустить из дистрибутива установочный файл с названием:

EliCont-CS -<разрядность процессора> <язык> <версия>.msi

Например, EliCont-CS-x86 -ru-2.0.0.msi

- 2 в окне установки нажать кнопку «Далее» (Рисунок 14);

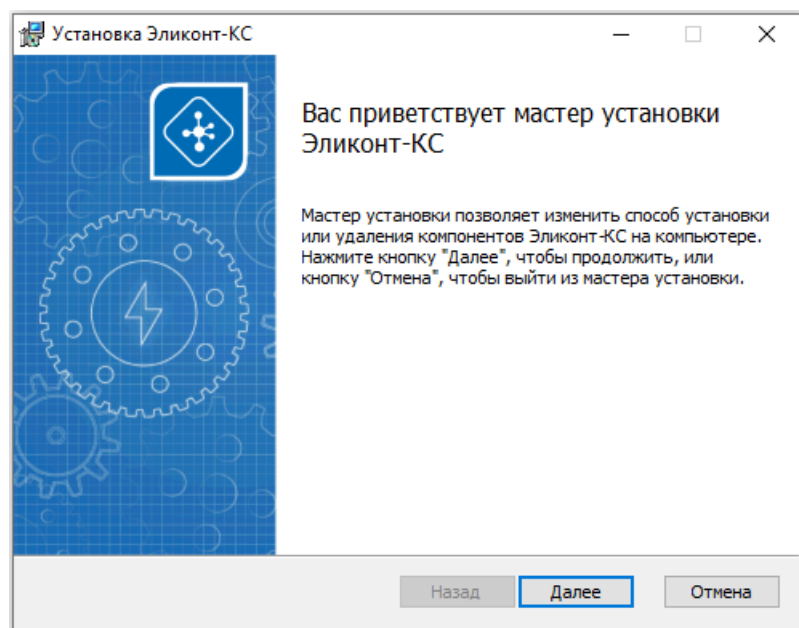


Рисунок 14 – Окно установки

- 3 в следующем окне нажать кнопку «Восстановить» (см. Рисунок 15);

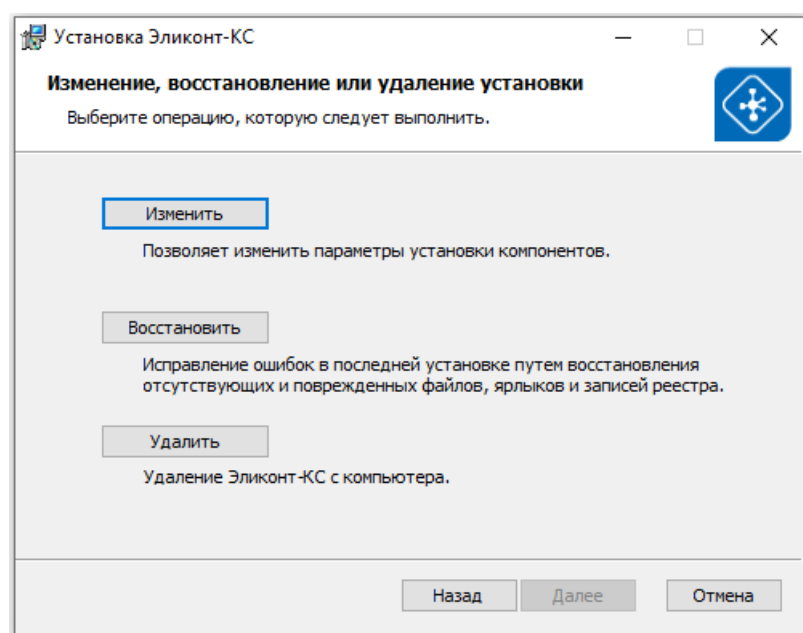


Рисунок 15 – Восстановить программу

- 4 в следующем окне нажать кнопку «Восстановить» для начала восстановления установленного Эликонт-КС (Рисунок 16);

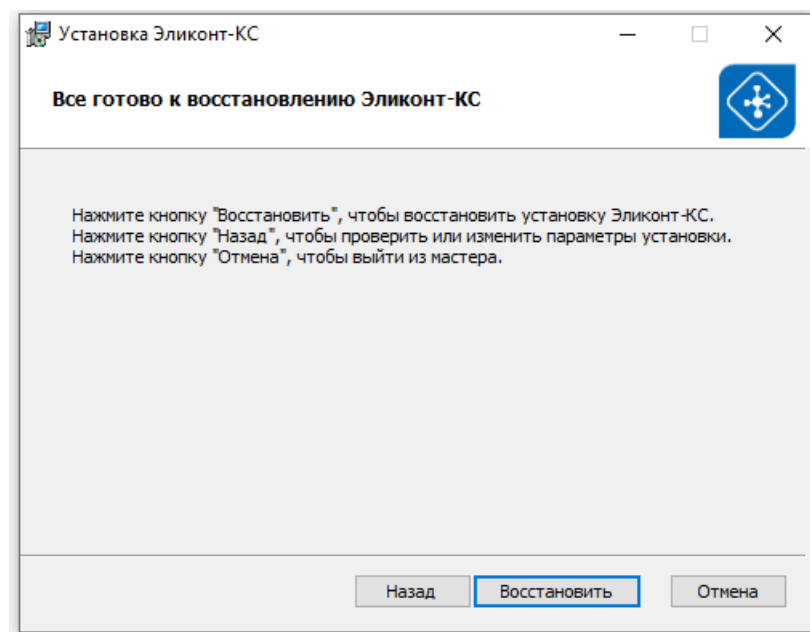


Рисунок 16 – Готовность к восстановлению программы

5 в заключительном окне нажать кнопку «Готово».

5.1.2 Запуск Коммуникационного ядра в ОС MS Windows

После установки Коммуникационного ядра служба ElaraCS запускается автоматически.

Пользователь с правами локального администратора может запускать и останавливать службу Коммуникационного ядра вручную, с использованием стандартных инструментов Windows в окне Диспетчер задач.

5.1.3 Запуск Конфигуратора в ОС MS Windows

Для запуска Конфигуратора в системе Windows:

- в главном меню системы выбрать Пуск – Элара Эликонт-КС – Конфигуратор, или
- запустить с помощью ярлыка на рабочем столе компьютера .

5.2 Установка Эликонт-КС в ОС Linux

Установка Коммуникационного ядра и Конфигуратора в Linux: Debian, Astra, РОСА Хром и РЕД ОС, производится под привилегированным пользователем («root»), или командой «sudo» (имеющей возможность выполнять действия от имени пользователя «root»).

Порядок действий следующий:

- 1 для установки Коммуникационного ядра авторизоваться в ОС с правами непривилегированного пользователя, открыть терминал и ввести команду:
 - в ОС Astra Linux и Debian:

sudo dpkg -i <путь до установочного файла>/<название файла>.deb

Например,

```
sudo dpkg -i /home/user/Desktop/Configurator/elicont-cs_2.3.3_amd64.deb
```

– в ОС РОСА Хром:

```
sudo rpm -i <путь до установочного файла>/<название файла>.rpm
```

Например,

```
sudo rpm -i elicont-cs-2.5.0-1097.i386.rpm
```

– в РЕД ОС:

```
sudo rpm -i <путь до установочного файла>/<название файла>.rpm
```

Например,

```
sudo rpm -i elicont-cs-redos-2.5.0-1096.i386.rpm
```

В результате, распаковывается архив «elicont-cs» и выполняется замена предыдущего пакета (если был ранее установлен).

Коммуникационное ядро разворачивается по пути: <место установки>/Elara/elicont-cs с автоматическим запуском службы ElaraCS.

2 для установки Конфигуратора авторизоваться в ОС с правами непривилегированного пользователя, открыть терминал и выполнить команду:

– в ОС Astra Linux и Debian:

```
sudo dpkg -i /< путь до установочного файла >.deb
```

Например,

```
sudo dpkg -i /home/user/Desktop/Configurator/ elicont-config_2.3.3_amd64.deb
```

– в ОС РОСА Хром и РЕД ОС:

```
sudo rpm -i <путь до установочного файла>/<название файла>.rpm
```

Например,

```
sudo rpm -i elicont-config-2.5.0-1097.x86_64.rpm
```

Конфигуратор разворачивается по пути: <место установки>/elicont-config.

3 при обновлении, изменении программы используется режим «Обновление». При этом выполняется установка Конфигуратора «поверх» существующего.

5.2.1 Запуск Коммуникационного ядра в ОС Linux

Для запуска службы в терминале ввести команду:

```
sudo systemctl start ElaraCS
```

5.2.1.1 Проверка статуса службы

Для отображения службы КС в терминале ввести команду:

```
sudo systemctl status ElaraCS
```

5.2.1.2 Остановка службы

Для остановки службы необходимо в терминале ввести команду:

```
sudo systemctl stop ElaraCS
```

5.2.2 Запуск Конфигуратора в Linux

Для запуска Конфигуратора в Linux запустить файл

```
<место установки>/EliCont-Config/el_conf
```

5.3 Удаление Эликонт-КС

Удаление приложений Эликонт-КС в Windows выполняется стандартными средствами ОС.

В Linux следует авторизоваться в ОС с правами непривилегированного пользователя и в терминале ввести команды:

- в ОС Astra Linux и Debian для удаления Коммуникационного ядра и Конфигуратора:

```
sudo dpkg --purge elicont-cs
```

```
sudo dpkg --purge elicont-config
```

(ключ «purge» – для удаления пакета вместе с файлами настроек).

- в ОС РОСА Хром для удаления Коммуникационного ядра:

```
sudo rpm -e elicont-cs
```

- в РЕД ОС для удаления Коммуникационного ядра и Конфигуратора:

```
sudo dnf remove elicont-cs
```

```
sudo dnf remove elicont-config
```

6 НАСТРОЙКА ПОДКЛЮЧЕНИЯ К DOCKER КОНТЕЙНЕРУ

Docker контейнер (в дистрибутиве архивный файл *.TAR) строится на основе ОС Astra Linux и содержит Коммуникационное ядро.

Для подготовки к работе с Docker контейнером выполнить шаги:

- настройка и запуск Docker контейнера (раздел 6.1);
- «проброс» лицензионного ключа (раздел 6.2);
- установка и запуск Конфигуратора (раздел 5);
- проверка подключения к Docker контейнеру (раздел 6.4);
- подключение по протоколу SNMP для Windows (раздел 6.3).

Для работы с Docker контейнером необходимо установить приложение Docker версии 4.28 или выше и, дополнительно, для Windows – приложение WSL2 (для запуска Linux).

Для «проброса» аппаратных ключей для Windows необходимо установить проект usbipd-win 4.2.0 и выше, см. <https://learn.microsoft.com/ru-ru/windows/wsl/connect-usb>

Важно! При работе с Docker контейнером (а также в процессе настройки) служба ElaraCS должна быть остановлена. Приложение Docker - запущено.

6.1 Настройка и запуск Docker контейнера

Порядок действий администратора:

- 1 в Windows открыть командную оболочку PowerShell от имени администратора. В Linux открыть терминал от имени пользователя root;
- 2 загрузить образ из контейнера, размещенного в сетевом (раздел 6.1.1) или локальном хранилище (раздел 6.1.2);
- 3 создать и запустить контейнер из образа (раздел 6.1.3).

6.1.1 Загрузка образа из сетевого хранилища

Загрузить образ с помощью команды:

Windows `docker pull <Путь до файла образа>:<Имя образа>`

Linux

Windows (пример):

`docker pull my_server\cs_product:2.6.0`

Linux (пример):

`docker pull my server/library/cs_product:2.6.0`

6.1.2 Загрузка образа из локального хранилища

Загрузить образ из файла tar:

Windows `docker load -i "<буква диска>:\<путь к образу>\<имя образа>"`

пример:

```
docker load -i "C:\User\docker\downloads\cs_product.tar"
```

Linux

```
docker load -i "/<путь к образу>/<имя образа>"
```

пример:

```
docker load -i "/user/downloads/cs_product.tar"
```

6.1.3 Создание и запуск контейнера из образа

Для создания контейнера необходим ID образа. Создание и запуск могут быть выполнены вручную (раздел 6.1.3.2) или через compose-файл (раздел 6.1.3.3).

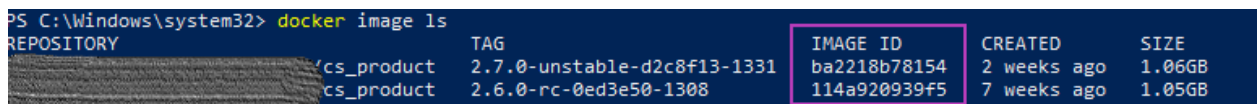
6.1.3.1 Получение ID образа

Получить список образов:

Windows `docker image ls`

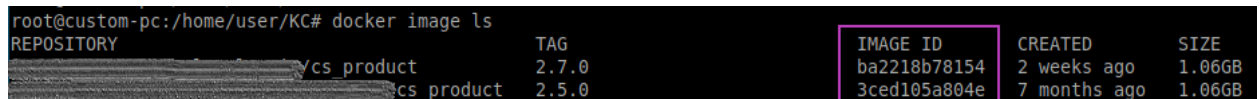
Linux

Получить ID образа в Windows:



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
cs_product	2.7.0-unstable-d2c8f13-1331	ba2218b78154	2 weeks ago	1.06GB
cs_product	2.6.0-rc-0ed3e50-1308	114a920939f5	7 weeks ago	1.05GB

ID образа в Linux:



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ycs product	2.7.0	ba2218b78154	2 weeks ago	1.06GB
cs product	2.5.0	3ced105a804e	7 months ago	1.06GB

6.1.3.2 Создание и запуск контейнера вручную

Создать и запустить:

Windows `docker run --name=<имя контейнера> --restart=always -p <порт>:<порт> -d --privileged -v /dev/bus/usb <image ID>`

пример:

```
docker run --name=CS --restart=always -p 5401:5401 -p 8080:8080 -p 8089:8089 -p 9039:9039 -d --privileged -v /dev/bus/usb ba2218b78154
```

Linux

```
docker run --name=<имя контейнера> --restart=always -p <порт>:<порт> -d --privileged -v /dev/bus/usb:/dev/bus/usb --device /dev/ttyS0 <image ID>
```

пример:

```
docker run --name=CS --restart=always -p 5401:5401 -p 8080:8080 -p
8089:8089 -p 9039:9039 -d --privileged -v /dev/bus/usb:/dev/bus/usb --
device /dev/ttyS0 --device/dev/ttyS1 ba2218b78154
```

В команде используются ключи:

- a) *name* – пользовательское имя для создаваемого контейнера;
- b) *restart=always* – позволяет запускать контейнер, если приложение Docker перезапущено;
- c) *p* – запускает контейнер в фоновом режиме;
- d) *privileged* - дает контейнеру права доступа к периферийным устройствам хост-машины;
- e) *v* - используется для подключения томов, папок и устройств хост-машины к контейнеру;
- f) *device*- используется для подключения COM-портов хост-машины в контейнер;
- g) *p* - используется для перенаправления портов хост-машины на порты контейнера. «Проброс» USB-портов необходим для работы с аппаратным лицензионным ключом;
- h) *<image ID>* - ID образа контейнера (см. раздел 0).

Таким образом, в примере передача аппаратного ключа с Linux хост-машины в контейнер производится с помощью ключей: *--privileged -v /dev/bus/usb:/dev/bus/usb*

При запуске контейнера должны быть «проброшены»:

- a) **служебные** порты 5401, 8080, 8089, 9039. Данные порты «пробрасываются» «напрямую», например, 8089:8089. В то же время порты 5401 и 8080 могут быть «проброшены» и на другие, например, 5400:5401. Назначение служебных портов:
 - 5401 – для подключения клиентов к веб-серверу Коммуникационного ядра;
 - 8080 – для сервиса API шлюза и загрузки конфигурации из Конфигуратора в Коммуникационное ядро;
- b) 8089 – для сервиса логирования;
- c) 9039 – для подключения Конфигуратора с целью мониторинга в режиме «Исполнение».
- d) порты **для серверных протоколов**. Порты «пробрасываются» напрямую, например, 102:102.

Важно! Для всех серверных протоколов в контейнере указывается IP-адрес 0.0.0.0. При других адресах серверный протокол «отвечать» не будет!

«Пробрасывание» портов для клиентских протоколов не требуется.

6.1.3.3 Создание контейнера через compose-файл. Запуск контейнера

Порядок действий администратора:

- 1 создать файл в текстовом редакторе с именем `docker-compose.yml` или `docker-compose.yaml`. Наполнить файл (см. примеры).

Пример файла для Windows:

```
version: '3.8'
services:
  elara_cs:
    image: 2031e2d6b24b
    container_name: CS
    restart: always
    privileged: true
    ports:
      - "9401:5401"
      - "9080:8080"
      - "9089:8089"
      - "9039:9039"
    devices:
      - /dev/bus/usb:/dev/bus/usb
    volumes:
      - C:\docker\cs\docker_volume:/opt/Elara/
      - EliCont-CS/config
```

Пример файла для Linux:

```
version: '3.8'
services:
  elara_cs:
    image: 2031e2d6b24b
    container_name: CS
    restart: always
    privileged: true
    ports:
      - "5401:5401"
      - "8080:8080"
      - "8089:8089"
      - "9039:9039"
    devices:
      - "/dev/ttyS0:/dev/ttyS0"
      - "/dev/ttyS1:/dev/ttyS1"
    volumes:
      - "/dev/bus/usb:/dev/bus/usb"
      - "/docker_volume:/opt/Elara/EliCont-
      CS/config"
```

Создание файла для Linux выполняется с помощью команды:

```
Linux      nano docker-compose.yml
           CTRL+O сохранить файл в приложении nano
```

Compose-файл содержит поля:

- a) `elara_cs` - имя группы, в которой будет создан контейнер;
- b) `image: <...>` - ID образа для создания контейнера;
- c) `container_name: <...>` - имя контейнера;
- d) `restart: always` - перезапуск при загрузке контейнера;
- e) `privileged: true` - предоставление доступа к периферийным устройствам компьютера;
- f) `ports <...>` – «пробрасываемые» порты;
- g) `devices: <...>` - «проброшенные» устройства;

- h) `volumes:/opt/Elara/EliCont-CS/config` - `"/docker_volume` - копирование содержимого папки `config` между контейнером и хостом.

Передача аппаратного ключа с Linux хост-машины в контейнер производится с помощью ключей: `--privileged -v /dev/bus/usb:/dev/bus/usb`

2 запустить контейнер.

Перейти в каталог с compose-файлом и выполнить запуск контейнера:

Windows `cd <путь к контейнеру>`

Linux `docker-compose up -d`

или

`docker-compose -f "<путь к compose файлу>" up -d`

3 добавить запись в файл `hosts` для корректной работы внешних OPC клиентов с OPC сервером Коммуникационного ядра. Порядок действий:

a) отобразить все созданные контейнеры:

Windows `docker ps -a`

Linux

b) зайти внутрь контейнера:

Windows `docker exec -ti <Id_контейнера>/bin/bash`

Linux

c) добавить запись в файл:

Windows `hosts: echo "IP Имя_Устройства" >> /etc/hosts`

Linux

d) проверить файл:

Windows `docker ps -a`

Linux `cat /etc/hosts`

6.2 «Проброс» аппаратного (USB) ключа в Docker Desktop (Windows)

В текущей версии программы рекомендуется использовать только аппаратный ключ.

Передача аппаратного ключа с Linux хост-машины в контейнер производится при создании контейнера из образа с помощью ключей: `--privileged -v /dev/bus/usb:/dev/bus/usb`

В ОС Windows «проброс» USB-устройства в Docker Desktop выполняется с использованием WSL2.

Для «проброса» ключа выполнить шаги:

- 1 установить и перезагрузить usbipd-win 4.2.0 и выше;
- 2 после перезагрузки открыть CMD или Powershell с правами администратора;
- 3 вывести список подключенных USB-устройств:

Windows `usbipd list`

Найти в списке значение <BUSID> требуемого устройства для использования в следующих шагах: 4 и 5:

```
Connected:
BUSID  VID:PID  DEVICE                                STATE
1-7    046d:c31c USB-устройство ввода                 Not shared
1-8    046d:c077 USB-устройство ввода                 Not shared
1-10   0a89:00c2 Guardant Sign                        Shared
```

- 4 подготовить устройство к подключению:

Windows `usbipd bind --busid=<BUSID>`

- 5 передать в контейнер usb-лицензию:

Windows `usbipd attach --wsl --busid=<BUSID>`

- 6 создать и запустить контейнер (раздел 6.1) или перезапустить созданный контейнер:

Windows `docker ps -a`
`docker restart <ID контейнера>`

USB-устройство не должно использоваться другой программой в процессе «проброса» в контейнер. Все необходимые программы и процессы, использующие USB-устройство, должны быть завершены.

- 7 проверить, что лицензия передана в контейнер.

Проверить, что устройство подсоединено:

Windows `usbipd list`

Подсоединенные устройства имеют статус Attached:

```
Connected:
BUSID  VID:PID  DEVICE                                STATE
1-7    046d:c31c USB-устройство ввода                 Not shared
1-8    046d:c077 USB-устройство ввода                 Not shared
1-10   0a89:00c2 Guardant Sign                        Attached
```

Проверить, что Коммуникационное ядро обнаружило лицензию, см. раздел 6.4.2.

6.3 Настройка подключения Docker контейнера по протоколу SNMP для Windows

Если адресное пространство контейнера и ОС Windows хоста пересекаются, необходимо настроить подключение по SNMP. Для этого следует добавить поля "bip" и "default-address-pools" в файл:

<папка пользователя>\.docker\daemon.json.

```
{
  "bip": "10.10.0.1/16",
  "builder": {
    "gc": {
      "defaultKeepStorage": "20GB",
      "enabled": true
    }
  },
  "default-address-pools": [
    {
      "base": "10.0.0.0/8",
      "size": 16
    }
  ],
  "experimental": false
}
```

После добавления полей перезапустить компьютер (или только Docker Desktop Service). Подключение по SNMP выполнится корректно.

6.4 Проверка подключения Конфигуратора к Docker контейнеру

6.4.1 Проверка подключения

Для проверки подключения выполнить шаги:

- 1 создать проект в Конфигураторе. Для КС ввести сетевой адрес компьютера, где запущен Docker контейнер;
- 2 добавить один протокол и один сигнал;
- 3 загрузить конфигурацию. Появится уведомление об успешной загрузке;
- 4 в адресной строке браузера ввести: <IP-адрес компьютера>:<Порт>, где указать адрес и порт компьютера, где запущен Docker контейнер. Откроется Web-интерфейс КС. Перейти на страницу «Исполнение». Убедиться, что отображаются корректные данные.

6.4.2 Проверка «проброса» ключа

В Web-интерфейсе КС (см. раздел 6.4.1) перейти на страницу «Лицензия». Убедиться, что ключ обнаружен.

Информация о лицензии	
Ключ	Обнаружен
Идентификатор ключа	#31794501
Режим	Нормальный
Ограничения	Клиентские пр CLIENT.OMD

6.5 Основные команды для управления контейнером

Команды для использования при работе с Docker контейнером в Windows и Linux.

Получить ID контейнера	<code>docker ps -a</code>
Запустить контейнер	<code>docker start <ID контейнера></code>
Перезапустить контейнер	<code>docker restart <ID контейнера></code>
Остановить контейнер	<code>stop <ID контейнера></code>
Остановить контейнер, находясь в директории с compose-файлом	<code>docker-compose down</code>
Остановить контейнер, указав compose файла	<code>docker-compose -f "<путь к compose файлу>" down</code>
Удалить контейнер	<code>docker rm <ID контейнера></code>
Удалить образ контейнера	<code>docker rmi <ID образа контейнера></code>

7 АКТИВАЦИЯ ПРОГРАММНОГО КЛЮЧА

Для активации программной лицензии необходим серийный номер ключа. Серийный номер предоставляется по запросу в ИнЦ Элара. Номер имеет формат XXXXXX-XXXXXX-XXXXXX-XXXXXX.

Активация выполняется с помощью утилиты Мастер лицензий Guardant (приложение license_wizard). Если данная утилита отсутствует в поставке Эликонт-КС, то ее следует запросить у представителя ИнЦ Элара. В запросе необходимо указать ОС (Windows/ Linux), в среде которой будет работать приложение.

7.1 Активация лицензии по файлу запроса

Для получения файла запроса на лицензию:

- запустить Мастер лицензий Guardant на компьютере, Рисунок 17;

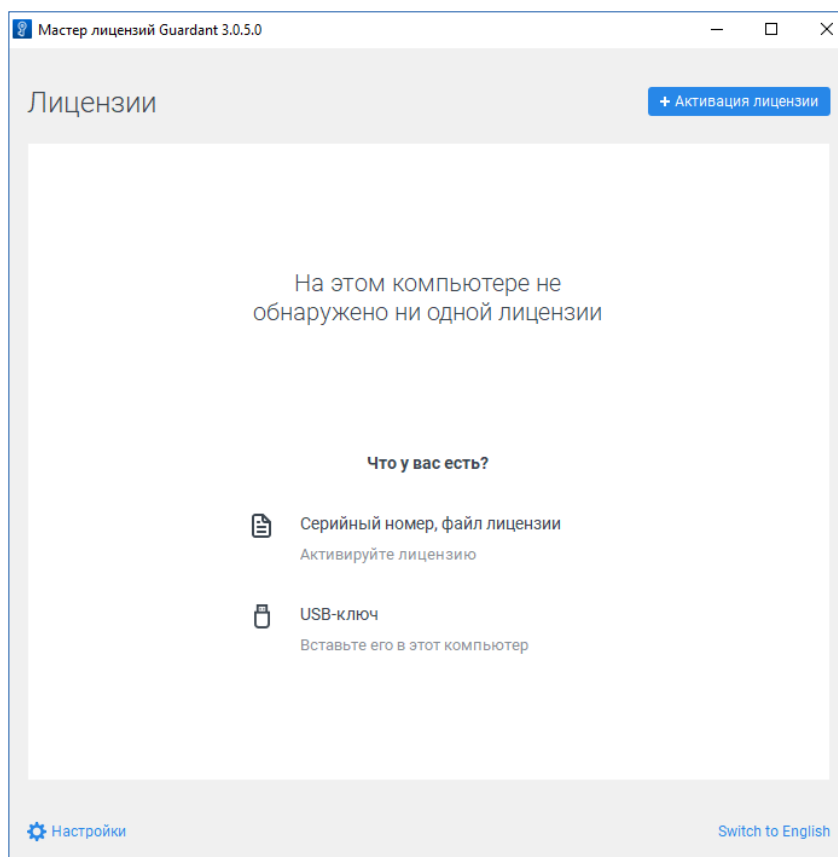


Рисунок 17 – Начальное окно Мастера лицензий

- в окне мастера нажать кнопку «Активация лицензии»;
- нажать кнопку «На этом»;
- щелкнуть по ссылке «Оффлайн активация»;
- выбрать вкладку «Новая лицензия» и нажать кнопку «Сохранить», Рисунок 18. Сохранить файл запроса в файловой системе. Закрыть Мастер лицензий.
- передать файл запроса в ИнЦ Элара для получения файла лицензии.

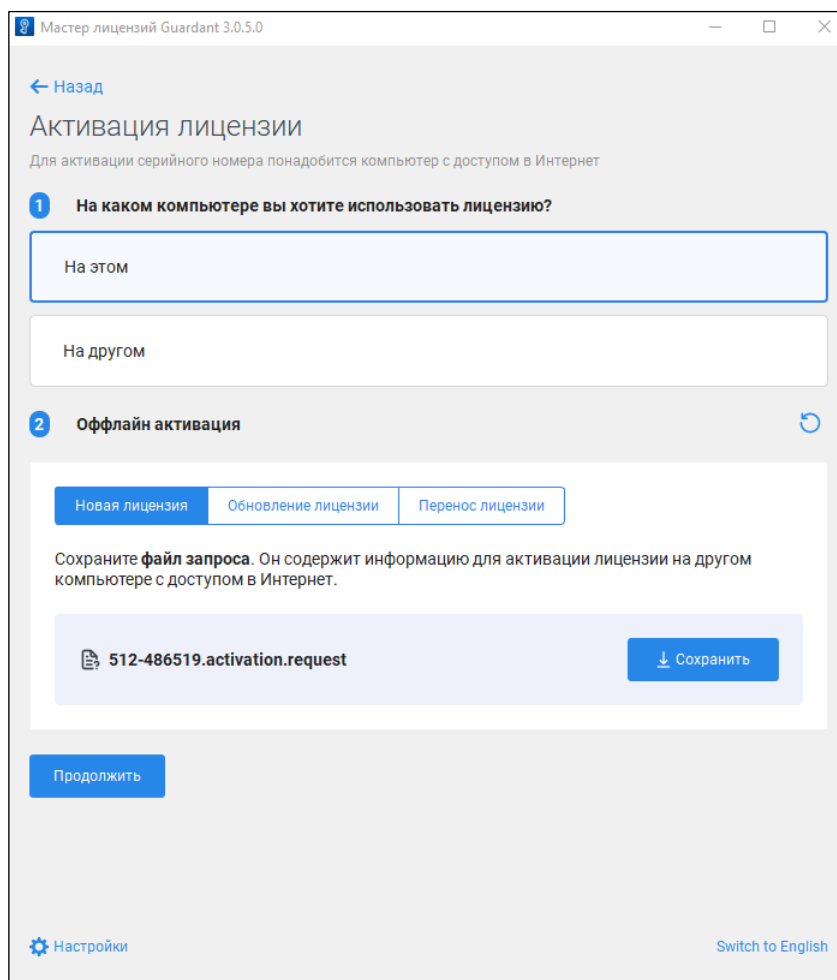


Рисунок 18 – Запрос для активации лицензии

Для активации лицензии, см. Рисунок 19:

- запустить Мастер лицензий на компьютере, где был получен запрос на лицензию;
- нажать кнопку «Активация лицензии»;
- нажать кнопку «На этом»;
- щелкнуть по ссылке «Оффлайн активация»;
- нажать кнопку «Продолжить»;
- нажать кнопку «Продолжить, у меня есть файл лицензии»;
- нажать кнопку «Выбрать файл»;
- в окне проводника указать файл лицензии и нажать кнопку «Открыть».

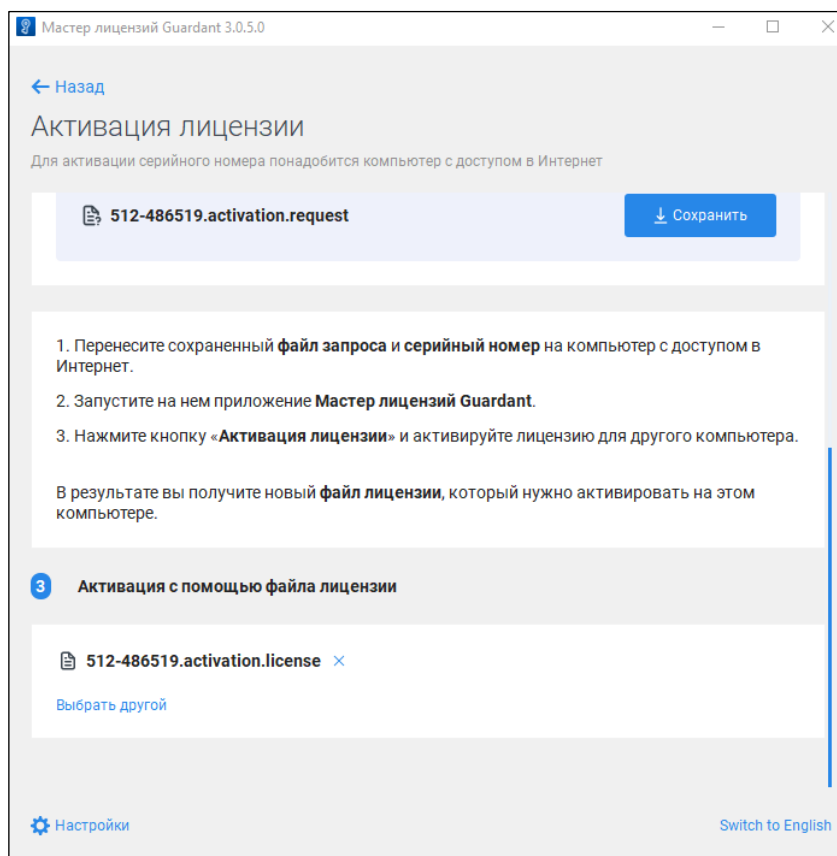


Рисунок 19 – Активация с помощью файла лицензии

В результате, программная лицензия активирована офлайн. В главном окне мастера отображается ID лицензии и перечень доступных функций (данные активированной лицензии появляются сразу, после выбора файла лицензии).

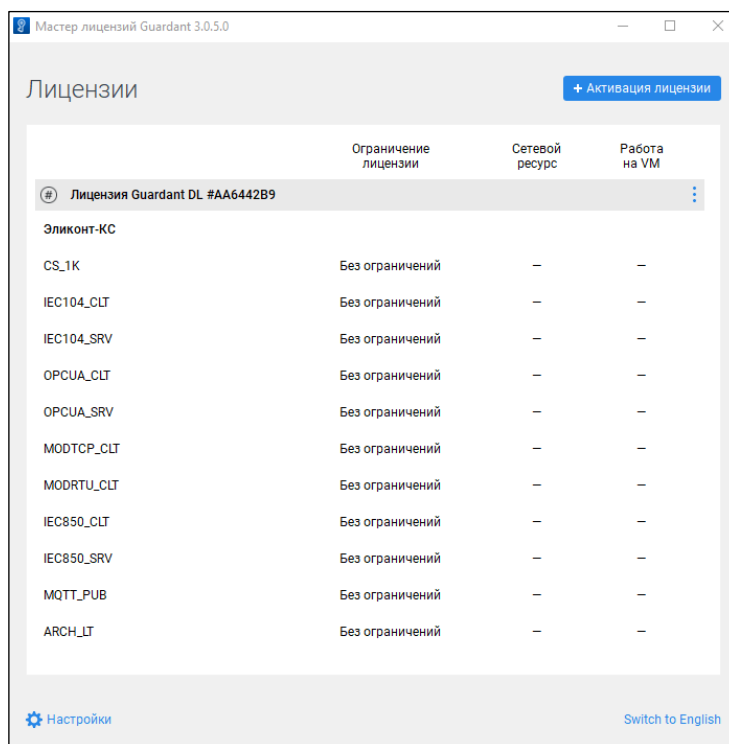


Рисунок 20 - Активированная лицензия

7.2 Обновление лицензии по файлу запроса

Для получения файла запроса на обновление лицензии:

- запустить Мастер лицензий Guardant;
- в строке с лицензией нажать кнопку  и выбрать пункт «Оффлайн обновление»;
- нажать кнопку «На этом»;
- перейти на вкладку «Обновление лицензии»;
- нажать кнопку «Сохранить»;
- сохранить файл запроса на компьютере;
- закрыть Мастер лицензий;
- передать файл запроса в ИнЦ Элара для получения файла обновления лицензии.

Для активации обновления:

- запустить Мастер лицензий Guardant на том компьютере, где используется лицензия;
- нажать кнопку «Активация лицензии»;
- перейти на вкладку «Обновление лицензии»;
- нажать кнопку «Выбрать файл» и указать файл лицензии;
- нажать кнопку «Обновление лицензии».

7.3 Перенос лицензии на другой компьютер

После переноса с данного компьютера лицензия на нем станет неактивной.

Для переноса лицензии на другой компьютер:

- запустить Мастер лицензий Guardant;
- в строке с лицензией нажать кнопку  и выбрать пункт «Перенести на другой компьютер»;
- нажать кнопку «Сохранить». В результате загрузится файл лицензии;
- перенести этот файл на другой компьютер;
- активировать лицензию на компьютере.

7.4 Просмотр журнала событий мастера

Журнал событий — это текстовый файл, в котором фиксируются записи о всех операциях и ошибках при работе с мастером.

Для открытия папки с лог-файлами:

- запустить Мастер лицензий Guardan;
- в левом нижнем углу нажать кнопку «Настройки»;
- щёлкнуть по ссылке «Перейти в журнал».

8 СОПРОВОЖДЕНИЕ ПО

8.1 Синхронизация системного времени

Синхронизация системного времени КС производится средствами операционной системы. Для этого необходимо настроить в ОС доступ к источнику точного времени и другие параметры синхронизации.

Если системное время изменяется более, чем на 5 секунд, КС может произвести перезапуск клиентских каналов для обеспечения стабильного сбора данных. В таких случаях, на время синхронизации рекомендуется остановить службу ElaraCS.

8.2 Администрирование в Web-интерфейсе КС

8.2.1 Общие сведения

Для подключения к Web-интерфейсу в адресной строке браузера ввести:

<IP-адрес компьютера>:5401, где

<IP-адрес компьютера>:5401 - IP-адрес и порт компьютера, на котором установлено Коммуникационное ядро. На локальном компьютере – «localhost:5401».

На странице авторизации ввести логин и пароль: по умолчанию для администратора - admin, admin, и для пользователя - guest, guest.

После авторизации открывается Главная страница Web-интерфейса КС (Рисунок 21).

Панель навигации интерфейса описаны в Таблице 5. Администратор имеет доступ ко всем страницам. Пользователь имеет доступ к страницам: Главная, Лицензия, Исполнение, Диагностика.

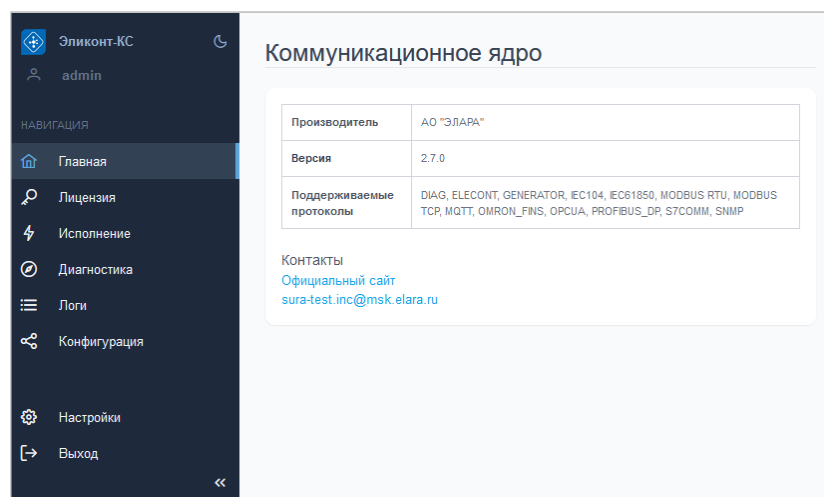


Рисунок 21 – Главная страница

Таблица 5 – Панель навигации

Пункты меню	Состав страницы	Примечание
Главная	Сведения о производителе, контакты, версия Коммуникационного ядра, поддерживаемые протоколы	Версия Коммуникационного ядра, Конфигуратора и системной базы – см. Конфигуратор, меню строки «Справка – Версия ПО»
Лицензия	Информация о лицензии	Описано в Руководстве пользователя
Исполнение	Оперативные (текущие) данные сигналов	Описано в Руководстве пользователя
Диагностика	Аппаратные и программные показатели компьютера с установленным Коммуникационным ядром	Описано в Руководстве пользователя
Логи	Ссылка для загрузки лог-файла CS.log	Страница доступна только администратору
Конфигурация	Кнопка для сброса текущей конфигурации	Страница доступна только администратору. См. раздел 8.2.2
Настройки	- Список пользователей. - Сведения об интерфейсе. - Локальное время в формате ГГГГ.ММ.ДД ЧЧ:ММ:СС	Страница доступна только администратору. См. разделы 8.2.3, 8.2.4
Выход		Завершение сеанса работы пользователя

8.2.2 Сброс конфигурации КС

Сброс текущей конфигурации (файл *user_config*) и загрузка конфигурации по умолчанию выполняются на странице «Конфигурация» с помощью кнопки «Выполнить».

При включении флажка «Жесткий сброс» и нажатия кнопки «Выполнить», будет выполнен сброс конфигурации и перезапуск службы ElaraCS.

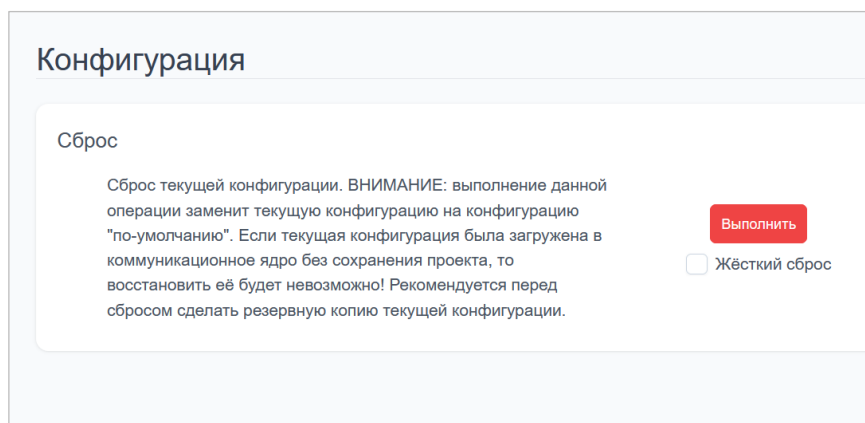


Рисунок 22 – Сброс конфигурации

8.2.3 Настройка пользователей

Настройка пользователей выполняется на вкладке «Пользователи» страницы «Настройки», см. Рисунок 23.

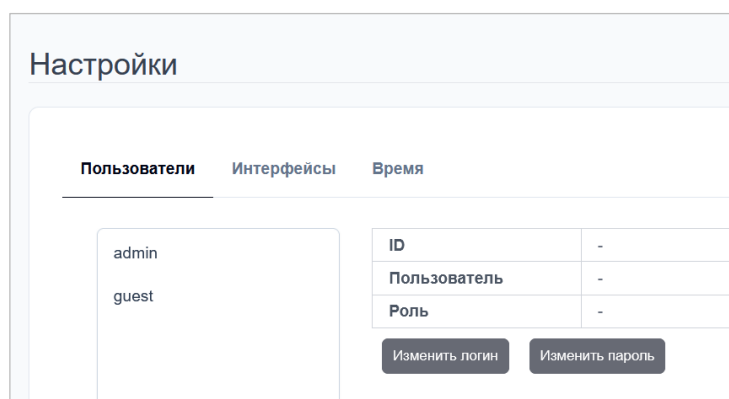


Рисунок 23 – Список пользователей по умолчанию

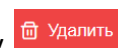
Для создания пользователя, в нижней части страницы нажать кнопку «Новый», расположенную ниже списка. В окне создания пользователя ввести:



- *логин*: может содержать латинские буквы и цифры;
- *пароль*: не менее 5-ти символов; могут быть использованы символы латинского алфавита, (заглавные, строчные), цифры, спецсимволы;
- *роль*: выбор из списка: Администратор, Пользователь.

Для изменения логина, пароля и роли выбрать пользователя в списке, см. Рисунок 23, и внести изменения.

Для удаления выбрать в списке пользователя и нажать кнопку «Удалить», расположенную ниже списка.



Удаление администратором собственной учетной записи невозможно.

8.2.4 Просмотр сведений об интерфейсе

На странице «Настройки» на вкладке «Интерфейсы» имеются сведения о параметрах сети.

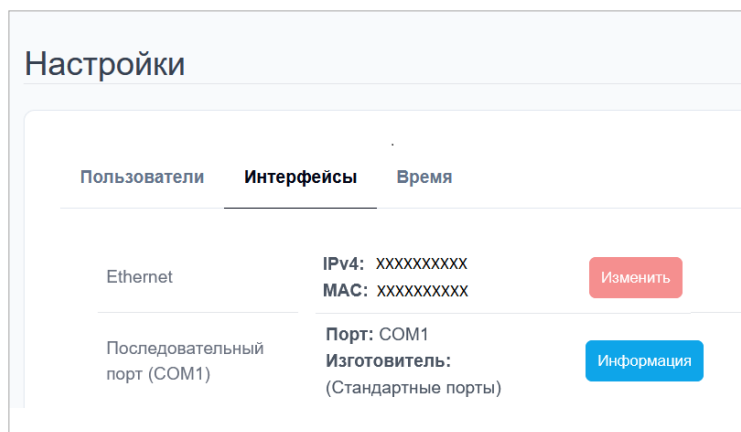


Рисунок 24 - Интерфейсы

8.3 Настройка клиентского подключения OPC UA в сети с компьютерами с ОС Linux без организации домена

При построении сети нижнего уровня систем автоматизированного управления, как правило, не используется доменная система, что может приводить к проблемам при установлении связи по протоколу OPC UA между устройствами с ОС Linux.

Для настройки сбора данных по протоколу OPC UA в Эликонт-КС требуется указывать URL сервера, с которым должно быть установлено соединение. В состав URL входит имя компьютера и TCP-порт сервера OPC UA.

Если Эликонт-КС установлен на компьютере с ОС Linux (Debian, Astra или др.), не включенном в домен, то без соответствующей настройки не может выполняться сопоставление имени компьютера, на котором установлен Сервер OPC UA (источник данных, к которому необходимо подключиться) с его IP-адресом. Это приводит к отказу в установке соединения.

При настройке безопасного соединения: данные не поступали в Эликонт-КС при следующей конфигурации сети:

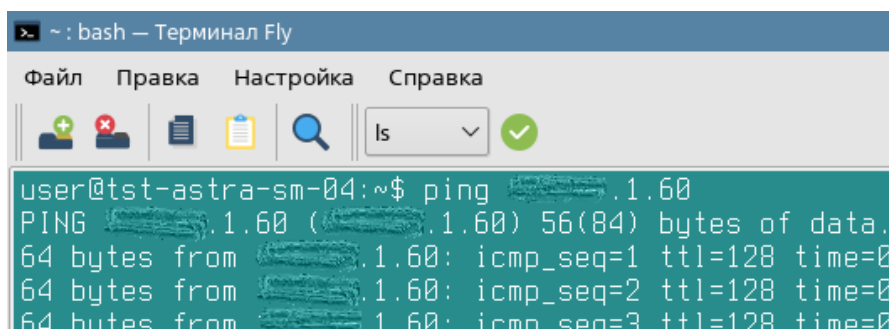
- Эликонт-КС является клиентом OPC UA на компьютере с ОС Astra Linux;
- при этом сервер OPC UA (источник данных) установлен на компьютер под ОС Windows. Все устройства находятся в одной подсети, но не включены в общий домен;
- межсетевые экраны отключены.

Рекомендуемым способом обеспечить сопоставление имени компьютера с его IP-адресом является внесение соответствующей информации в hosts-файл. Изменение сетевой информации требует корректировки файла сетевых данных вручную.

- выяснить, может ли компьютер с установленным Коммуникационным ядром Эликонт-КС сопоставить имя компьютера - источника данных с его IP-адресом. Для этого в терминале следует выполнить две команды (Рисунок 25):

а) `ping <ip_address>`, где

`<ip_address>` - IP-адрес компьютера - источника данных.



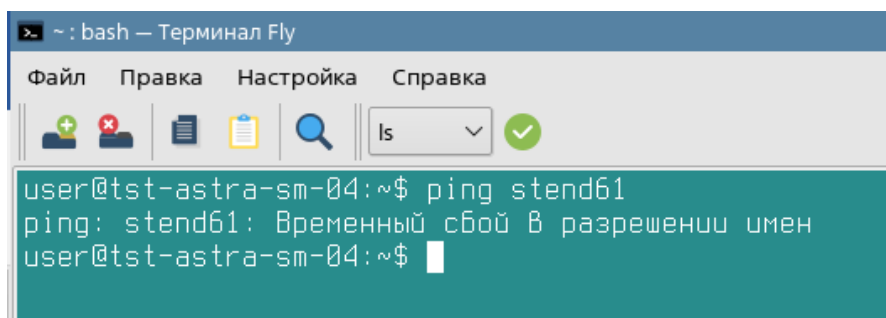
```
user@tst-astra-sm-04:~$ ping 1.60
PING 1.60 (1.60) 56(84) bytes of data.
64 bytes from 1.60: icmp_seq=1 ttl=128 time=0
64 bytes from 1.60: icmp_seq=2 ttl=128 time=0
64 bytes from 1.60: icmp_seq=3 ttl=128 time=0
```

Рисунок 25 – Проверка наличия связи

Наличие ICMP-ответов показывает, что связь установлена. Отсутствие ICMP-ответов свидетельствует о необходимости настройки сети.

б) `ping <computer_name>` (Рисунок 26), где

`<computer_name>` - имя компьютера - источника данных.



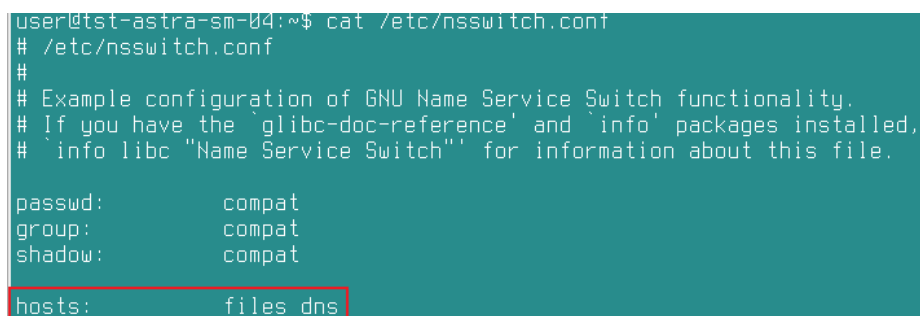
```
user@tst-astra-sm-04:~$ ping stend61
ping: stend61: Временный сбой в разрешении имен
user@tst-astra-sm-04:~$
```

Рисунок 26 – Проверка сопоставления

Если результатом выполнения команды является сбой, значит компьютер с установленным Эликонт-КС не может сопоставить имя компьютера - источника данных с его IP-адресом. Для сопоставления требуется выполнить действия, описанные далее.

- 1 убедиться в правильности порядка разрешения доменных имен. Для этого необходимо выполнить команду (Рисунок 27):

`cat /etc/nsswitch.conf`



```
user@tst-astra-sm-04:~$ cat /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc-reference' and 'info' packages installed,
# 'info libc "Name Service Switch"' for information about this file.

passwd:          compat
group:           compat
shadow:          compat
hosts:           files dns
```

Рисунок 27 – Проверка порядка разрешения доменных имен

Порядок «files dns» указывает, что на первом шаге разрешения доменного имени ОС обратится к файлу `hosts`, а на втором – к службе DNS. Опция `files` должна присутствовать, т.к. сопоставление доменного имени и IP-адреса указывается в файле `hosts`. В случае отсутствия опции, ее следует добавить.

2 добавить в файл `hosts` соответствующее сопоставление. Для этого:

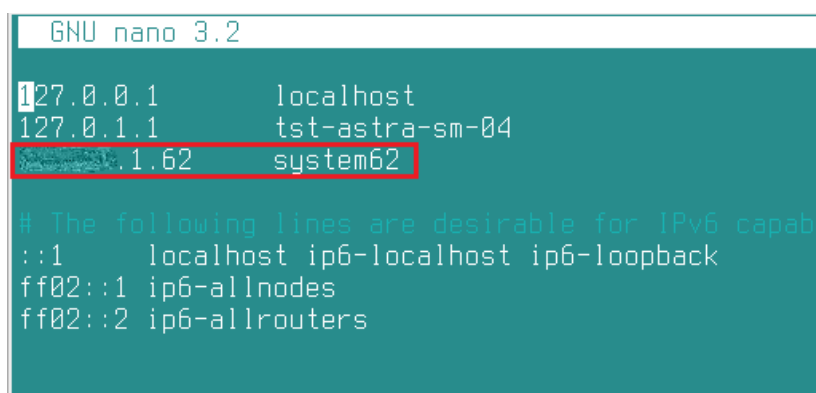
– выполнить команду:

`sudo <editor> /etc/hosts`, где в качестве `<editor>` указать любой из доступных текстовых редакторов, например, встроенный `nano`; `/etc/hosts` – путь к файлу `hosts` в ОС Linux; `sudo` – действие от имени `superuser`. См. Рисунок 28.

Примечание - для выполнения команды требуется пароль `root`.

`sudo nano /etc/hosts`

– добавить соответствие IP-адреса и имени компьютера – источника данных,



```
GNU nano 3.2
127.0.0.1    localhost
127.0.1.1    tst-astra-sm-04
192.168.1.62 system62
# The following lines are desirable for IPv6 capable
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

Рисунок 28 – Добавление сопоставления в файл `hosts`

– сохранить файл: для `nano` нажать сочетание клавиш `Ctrl+O` и подтвердить, нажав `Enter`,

– выйти из редактора, нажав клавиши `Ctrl+X`.

3 в настройках соединения клиентского канала OPC UA в проекте, созданном в Эликонт-КС, указать URL с именем источника данных, Рисунок 29.

4 добавить сигнал (-ы) из Адресного пространства.

5 загрузить конфигурацию в Коммуникационное ядро, подключиться к КС. Убедиться, что канал связи активен, данные поступают, Рисунок 30.

Клиентское подключение OPC UA в сети с компьютером с ОС Linux без организации домена настроено.

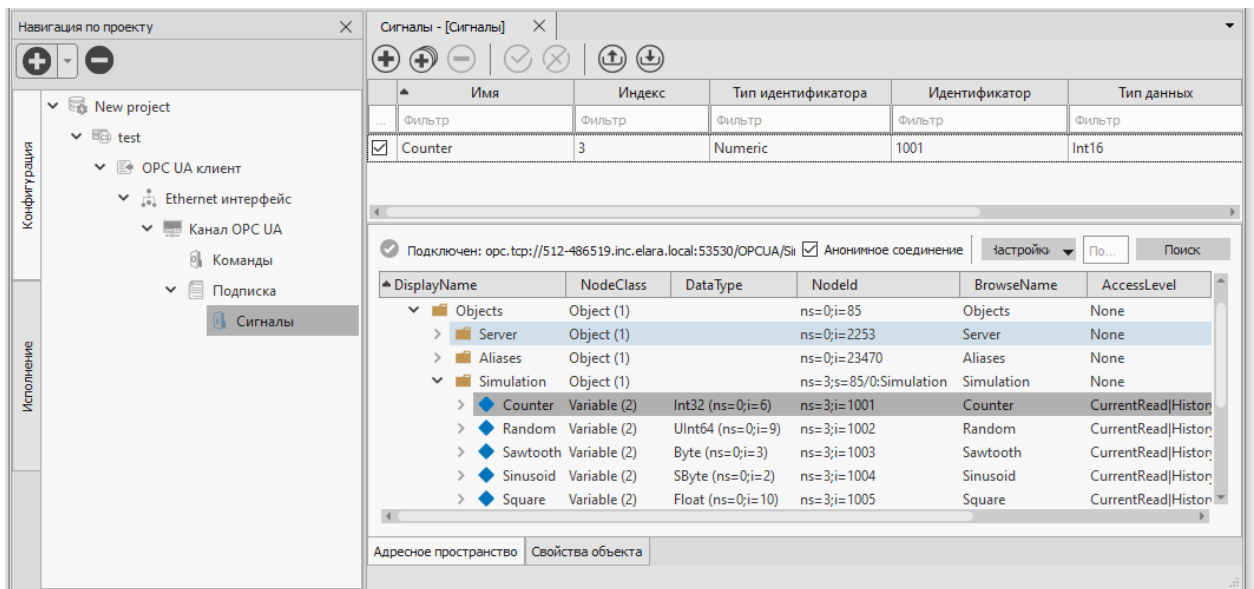


Рисунок 29 – Выбор сигнала

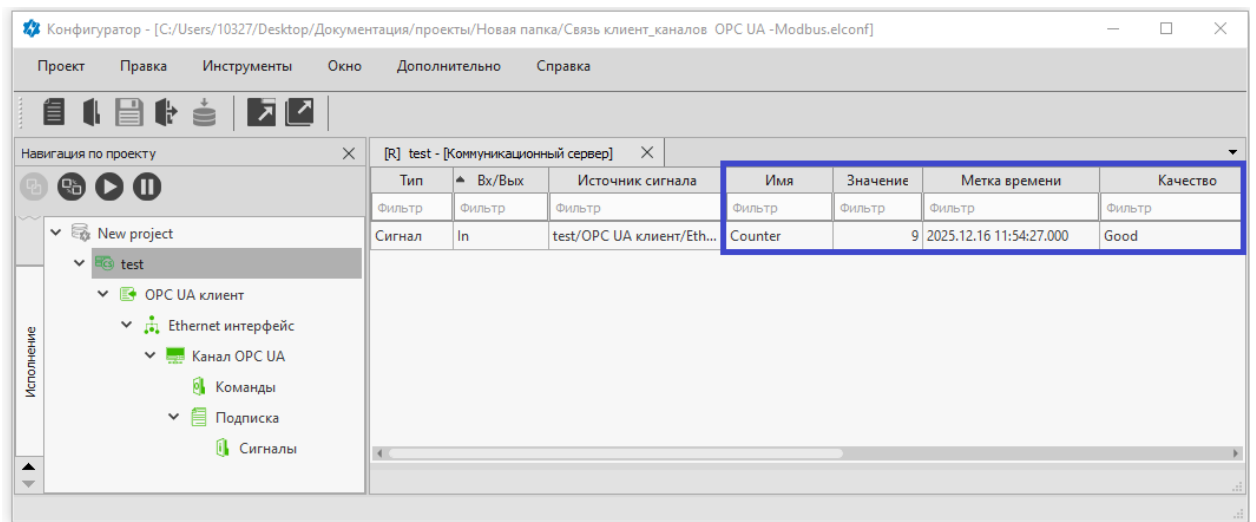


Рисунок 30 – Сигнал, принимаемый КС

8.4 Обновление версий приложений

Обновление версий производится путём удаления старой версии приложения и установкой новой версии.

8.5 Описание конфигурационных файлов

user_config.xml – конфигурационный файл, который содержит настройки каналов передачи данных, команд и сигналов. Файл обновляется после успешного выполнения команды «Загрузить конфигурацию».

static_config_win.xml (*static_config_lin*) – системный файл со служебной информацией для MS Windows (Linux). Файл содержит статическую часть конфигурации: настройку сервисов и Коммуникационного Сервера. Описание системного файла конфигурации приведено в Приложении А.

Конфигурационные файлы КС размещаются на компьютере с установленным Коммуникационным ядром в папке:

OS Windows <место установки>\ELARA\EliCont-CS\config\

OS Linux <место установки>/elicont-cs/config/

Перед внесением изменений в файлы рекомендуется сделать их резервные копии. После внесения изменений следует перезапустить службу ElaraCS.

Ошибка редактирования файлов может привести к потере работоспособности КС.

8.6 Сведения об архивировании сигналов

Для хранения архивных данных сигналов могут быть использованы следующие СУБД:

- SQLite – внутренняя СУБД;
- PostgreSQL – внешняя СУБД.

Для записи архивных данных в СУБД SQLite предварительная подготовка не требуется. В Конфигураторе нужно включить встроенное архивирование и указать объем БД и время хранения данных. Архивные данные будут записываться в файл ... cs_archive.db. Прочитать данные можно с помощью утилиты SQLite Studio.

Для архивирования значений сигналов в БД PostgreSQL необходимо предварительно установить СУБД PostgreSQL, сделать ее доступной по сети, создать базу данных, а также настроить необходимые права для учетной записи, от которой будет производиться архивирование. В Конфигураторе для КС требуется включить архивирование в PostgreSQL, указать IP адрес и порт компьютера, где находится БД, а также имя БД, логин и пароль для подключения.

Структура БД архива PostgreSQL и SQLite описана в Приложении В.

Важное замечание. Если БД архива PostgreSQL была создана в версии Эликонт-КС 2.6, необходимо перенести все данные в новую БД с учетом изменения формата поля timestamp таблицы «event» из «integer» в «timestamp without time zone». Для преобразования может быть использована функция to_timestamp (см. документацию PostgreSQL). После переноса данных подключить новую БД.

Описание файла статической конфигурации КС для ОС Windows и Linux

(static_config_win.xml, static_config_lin.xml)

```

<?xml version="1.0" encoding="UTF-8"?>
- <StaticConfiguration>
  + <Config>
  + <Registry>
  + <License>
    <!--Redundancy> <GrpcServer> <IpAddr>127.0.0.1</IpAddr> <Port>8082</Port>
    <Transport>TCP</Transport> </GrpcServer> <Logging> <Logger> <Type>file</Type>
    <Path>./ServiceRedundancy.log</Path> <Severity>DEBUG</Severity> </Logger> </Logging>
    <DefaultActivity type="PASSIVE"/> </Redundancy-->
  - <Archive>
    + <GrpcServer>
      <!--Logging> <Logger> <Type>file</Type> <Path>./ServiceArchive.log</Path>
      <Severity>DEBUG</Severity> </Logger> </Logging-->
    </Archive>
  - <Monitor>
    + <Services>
      <!--Logging> <Logger> <Type>file</Type> <Path>./ServiceMonitor.log</Path>
      <Severity>DEBUG</Severity> </Logger> </Logging-->
    </Monitor>
  - <CommunicationServer>
    + <Settings>
    + <Libraries>
    </CommunicationServer>
  - <ApiGateway>
    <DataFolder>./</DataFolder>
    + <ActivitySwitcher>
      <!--Logging> <Logger> <Type>file</Type> <Path>./ApiGateway.log</Path>
      <Severity>DEBUG</Severity> </Logger> </Logging-->
    </ApiGateway>
  - <ServiceLogger>
    + <GrpcServer>
    + <LoggerManaging>
    </ServiceLogger>
</StaticConfiguration>

```

Элементы XML	Атрибут	Значение	Описание
<StaticConfiguration>			Статическая конфигурация (корневой элемент)
<Config>			Раздел настройки сервиса конфигурации
<GrpcServer>			Настройка подключения по GRPC
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса
	Port	8084	Порт

Элементы XML	Атрибут	Значение	Описание
	Transport	TCP	Протокол передачи данных
<Ping>	interval	1000	Период подтверждения своей работоспособности
<Registry>			Раздел настройки сервиса регистрации
<GrpcServer>			Настройка подключения по GRPC
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса
	Port	8085	Порт
	Transport	TCP	Протокол передачи данных
<Settings>			
	UseEnchanced-Logging	0 или 1	Включение (выключение) режима расширенного логирования
	MaxSkippedPings	6	Количество пропущенных подтверждений для определения неработоспособности сервиса
<License>			Раздел настройки сервиса лицензии
<GrpcServer>			Настройка подключения по GRPC
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса
	Port	8086	Порт
	Transport	TCP	Протокол передачи данных

Элементы XML	Атрибут	Значение	Описание
<KeyType>	type	GUARDANT	Тип лицензионного ключа
<Archive>			Раздел настройки сервиса архивирования
<GrpcServer>			Настройка подключения по GRPC
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса
	Port	8083	Порт
	Transport	TCP	Протокол передачи данных
<Monitor>			Раздел настройки сервиса мониторинга
<Services>			Процессы, контролируемые сервисом мониторинга
<Service>			Настройки для запуска процессов
	Reg_attempts	15	Количество попыток обнаружения запуска процесса
	parameters	127.0.0.1:8085	Параметры командной строки процесса
	name		Регистрационное имя процесса
	Path	./***.exe	Путь к исполняемому файлу процесса
<CommunicationServer>			Раздел настройки Коммуникационного ядра

Элементы XML	Атрибут	Значение	Описание
<Settings>			Настройки коммуникационного ядра
<Strategy>	type	Multithread или Cyclic	Метод обработки каналов (многоканальный или последовательный)
	cycle_time	10000	Период циклической обработки, мксек
<GrpcServer>			Настройка подключения по GRPC
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса
	Port	8081	Порт
	Transport	TCP	Протокол передачи данных
<Libraries>			Библиотеки протоколов
<Lib>			Настройка библиотек протоколов
	Path	../plugins/***.dll или ../plugins/***.so	Путь к файлу библиотеки протокола
	<Protocol>		Параметры протокола
	type		Тип протокола
	cmd_attributes		Атрибуты команд
	sig_attributes		Атрибуты сигналов
	<Settings>		Настройки передачи данных
	OpenSSLStore		Путь к библиотеке OpenSSL

Элементы XML	Атрибут	Значение	Описание
	<SignalRetranslationReasons>		
	ValueChanged	0 или 1	Передача по изменению значения
	QualityChanged	0 или 1	Передача по изменению качества
<Channels>			Служебные каналы передачи данных
<Channel>			Параметры служебного канала
	type	ELECONT	Тип канала
	name		Имя канала
	id		Идентификатор канала
	guid		GUID канала
<Settings>			Настройки служебного канала
	Client	0 или 1	клиент или сервер
	IP	0.0.0.0	IP -адрес сетевого интерфейса
	Port	9039, 9041, 9042	Порт
Controller	type	NO_BUFFERING	Буферизация данных
<ApiGateway>			Раздел настройки сервиса ApiGateway
<GrpcServer>			Настройка подключения по GRPC
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса

Элементы XML	Атрибут	Значение	Описание
	Port	8080	Порт
	Transport	TCP	Протокол передачи данных
<ServiceLogger>			Раздел настройки сервиса логирования
<GrpcServer>			Настройка подключения по GRPC
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса
	Port	8089	Порт
	Transport	TCP	Протокол передачи данных
<LoggerManaging>			Настройка управления логированием
	IpAddr	127.0.0.1	IP -адрес сетевого интерфейса
	Port	4343	Порт
	Transport	TCP	Протокол передачи данных

Структура БД архива PostgreSQL

Таблица описаний объектов (description)

Поле	Тип данных	Описание
objectid	text	Идентификатор объекта (GUID)
name	text	Имя объекта
objectdescription	text	Описание объекта
class	integer	Класс объекта (канал, сигнал и т.п.)
type	integer	Тип объекта внутри класса

Таблица событий (event)

Поле	Тип данных	Описание
id	integer	Первичный ключ
objectid	text	Идентификатор (GUID)
timestamp	timestamp without time zone	Метка времени без указания часового пояса
value	text	Значение сигнала
valuequality	integer	Качество значения

Таблица качества значений сигналов (quality)

Поле	Тип данных	Описание
valuequality	integer	Идентификатор качества значения
stringquality	text	Расшифровка идентификатора качества

Таблица иерархии объектов (tree)

Поле	Тип данных	Описание
objectid	text	Идентификатор объекта (GUID)
parentid	text	Идентификатор родителя объекта (GUID)

Структура БД архива SQLite

Таблица описаний объектов (Description)

Поле	Тип данных	Описание
ObjectID	TEXT	Идентификатор объекта (GUID)
Name	TEXT	Имя объекта
ObjectDescription	TEXT	Описание объекта
Class	INT	Класс объекта (канал, сигнал и т.п.)
Type	INT	Тип объекта внутри класса

Таблица интервалов (Interval)*

Поле	Тип данных	Описание
ID	INT	Порядковый номер интервала
LeftBound	INT	Идентификатор первой записи интервала
RifghtBound	INT	Идентификатор последней записи интервала

Таблица событий (Event)

Поле	Тип данных	Описание
ID	INT	Первичный ключ
ObjectID	TEXT	Идентификатор (GUID)
TimeStamp	INT	Метка времени (в формате Unix)
Value	TEXT	Значение
ValueQuality	TEXT	Качество значения

Таблица качества значений сигналов (Quality)

Поле	Тип данных	Описание
ValueQuality	INT	Идентификатор качества значения
StringQuality	TEXT	Расшифровка идентификатора качества

Таблица иерархии объектов (Tree)

Поле	Тип данных	Описание
------	------------	----------

ObjectID	TEXT	Идентификатор объекта (GUID)
ParentID	TEXT	Идентификатор родителя объекта (GUID)

Примечание - *В конфигурации для КС пользователь настраивает параметр «Время хранения данных, в днях», т.е. период ротации базы данных по времени. Спустя указанное время в таблице Interval появляется строка с порядковым номером интервала, ID первого сигнала и ID последнего сигнала, которые были записаны в течение этого интервала. После того, как пройдет еще один такой же интервал из таблицы Event удаляются записи в соответствии с указанными в Interval идентификаторами сигналов. Эта информация используется для удаления старых записей из таблицы Event.

Полезные ссылки

Подключение к Web-интерфейсу

<IP-адрес компьютера>:5401, где

<IP-адрес компьютера>:5401 - IP-адрес и порт компьютера, на котором установлено Коммуникационное ядро. На локальном компьютере – «localhost:5401».

Логин и пароль: по умолчанию для администратора - admin, admin, для пользователя - guest, guest.

Просмотр версии ПО

- 1 версия Коммуникационного ядра доступна в Web-интерфейсе;
- 2 версия ПО доступна в Окне конфигуратора: в строке меню пункт «Справка – Версия ПО». Отображается версия КС, Конфигуратора и системной базы.

Просмотр лицензии

Доступен в Web-интерфейсе.

Диагностика ПК с установленным Коммуникационным ядром

Доступна в Web-интерфейсе

Просмотр логов

Лог-файл Коммуникационного ядра CS.log можно скачать в Web-интерфейсе.

Лог-файл Коммуникационного ядра CS.log находится в папке ПК с установленным Коммуникационным ядром:

OS Windows <место установки>\ELARA\EliCont-CS\logs

OS Linux <место установки>/elicont-cs/logs

Лог-файл Конфигуратора el_conf.log находится в папке ПК с установленным Конфигуратором:

OS Windows <папка пользователя>\.EliCont-Config\logs

OS Linux /home/user/.EliCont-Config/logs

Размещение архивной БД SQLite

Файл cs_archive.db на ПК с установленным Коммуникационным ядром находится в папке:

OC Windows	<место установки>\ELARA\EliCont-CS\services
OC Linux	<место установки>/elicont-cs/services

Размещение конфигурационных файлов КС

Системный и пользовательский конфигурационные файлы на компьютере с установленным КС:

OC Windows	<место установки>\ELARA\EliCont-CS\config\static_config_win.xml
	<место установки>\ELARA\EliCont-CS\config\user_config.xml
OC Linux	<место установки>/elicont-cs/config/static_config_lin.xml
	<место установки>/elicont-cs/user_config.xml