

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭЛИКОНТ-КС

Описание программы

АДИГ.00024.13

Версия 2.8

Листов 33

ВВЕДЕНИЕ

В настоящем документе представлено описание структуры и работы Эликонт-КС. Данный документ предназначен для пользователей, работающих с Эликонт-КС, а также для ответственных лиц, заинтересованных в ознакомлении с продуктом.

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

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

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	2
1 НАЗНАЧЕНИЕ ПО	4
2 СОСТАВ ПО И СХЕМЫ РАЗВЕРТЫВАНИЯ	6
3 ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	8
4 ОПИСАНИЕ ФУНКЦИЙ КС	9
4.1 Сбор и передача сигналов	9
4.1.1 Время обработки потока данных в КС	12
4.1.2 Производительность обработки данных	13
4.2 Прием и передача команд управления	13
4.3 Ретрансляция сигналов между клиентскими каналами	16
4.4 Преобразование типов данных	18
4.5 Преобразование флагов качества	19
4.6 Преобразование меток времени сигналов из/в UTC	19
4.7 Присвоение метки времени сигналам	19
4.8 Инверсия дискретных сигналов	20
4.9 Масштабирование аналоговых сигналов	20
4.10 Контроль актуальности принимаемого сигнала для OPC UA и МЭК 60870-5-104	20
4.11 Контроль границ диапазона принимаемых сигналов	20
4.12 Передача аналоговых сигналов по апертуре	20
4.13 Буферизация данных для МЭК 60870-5-104 и МЭК 61850 MMS	21
4.14 Резервирование каналов OPC UA	21
4.15 Генерация сигналов	23
4.16 Интерфейс взаимодействия с пользовательским приложением	23
4.17 Самодиагностика	25
4.18 Архивирование сигналов	27
5 КОНТРОЛЬ ЦЕЛОСТНОСТИ ПО ЭЛИКОНТ-КС В ОС ASTRA LINUX	28
6 РЕАЛИЗАЦИЯ ПРОТОКОЛОВ	29
6.1 МЭК 61850-8-1 MMS	29
6.2 МЭК 60870-5-104	29
6.3 Profibus DP	29
7 СВЕДЕНИЯ О ЛИЦЕНЗИРОВАНИИ	31
СОКРАЩЕНИЯ, ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	33

1 НАЗНАЧЕНИЕ ПО

Эликонт-КС - прикладное программное обеспечение, предназначенное для построения систем сбора, хранения и передачи данных. Использование Эликонт-КС позволяет решать задачи по сопряжению устройств и обмену данными между системами, работающими по различным промышленным протоколам. Эликонт-КС может быть использован в системах автоматизации объектов энергетики, производства, транспортной инфраструктуры, ритейла, а также «умных» городов и зданий.

Функции, выполняемые Эликонт-КС:

- сбор и передача данных по промышленным протоколам, см. Таблицу 1;
- буферизация данных;
- конвертация типов данных;
- ретрансляция данных в вышестоящие системы управления;
- прием и передача команд дистанционного управления.

Таблица 1 - Поддерживаемые протоколы

Протоколы	Наименование
Сбор данных	OPC UA, Modbus RTU/TCP, S7comm, FINS (Omron), МЭК 61850-8-1 MMS, МЭК 60870-5-104, MQTT, SNMP, PROFIBUS DP, User channel*
Передача данных	OPC UA, Modbus RTU/TCP, МЭК 61850-8-1 MMS, МЭК 60870-5-104, MQTT, User channel
Примечание - *Интерфейс для подключения устройств и систем, которые ведут обмен данными по проприетарным (пользовательским) протоколам.	

В сочетании с аппаратными средствами Эликонт- КС выступает в роли:

- сервера сбора и передачи информации для применения на объектах энергетики в составе систем СОТИ АССО, ССПИ, АСМ РЗА и др. См. Рисунок 1;
- универсального шлюза промышленных протоколов для применения на объектах энергетики и производства в составе различных систем АСУ;
- шлюза для промышленного интернета вещей (англ. Industrial Internet of Things (IIoT)).

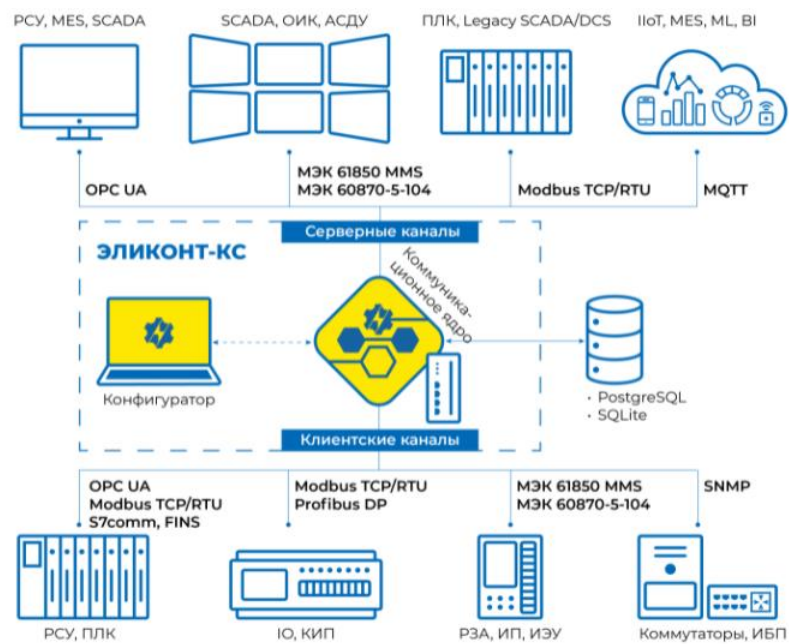


Рисунок 1 – Схема взаимодействия Эликонт-КС с другими устройствами

2 СОСТАВ ПО И СХЕМЫ РАЗВЕРТЫВАНИЯ

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

Коммуникационное ядро (далее КС или Коммуникационный сервер) обеспечивает обмен данными между источниками данных и приемниками по ряду протоколов.

Конфигуратор - приложение для настройки, управления и отображения данных Коммуникационного ядра.

Масштабируемая архитектура КС позволяет создавать как простые шлюзы на несколько десятков сигналов, так и сложные распределенные системы с десятками тысяч сигналов.

Различные варианты развертывания КС, представлены на Рисунках 2 и 3. При распределенной установке Коммуникационное ядро и Конфигуратор устанавливаются на отдельные компьютеры. При локальной установке оба приложения устанавливаются на один компьютер.

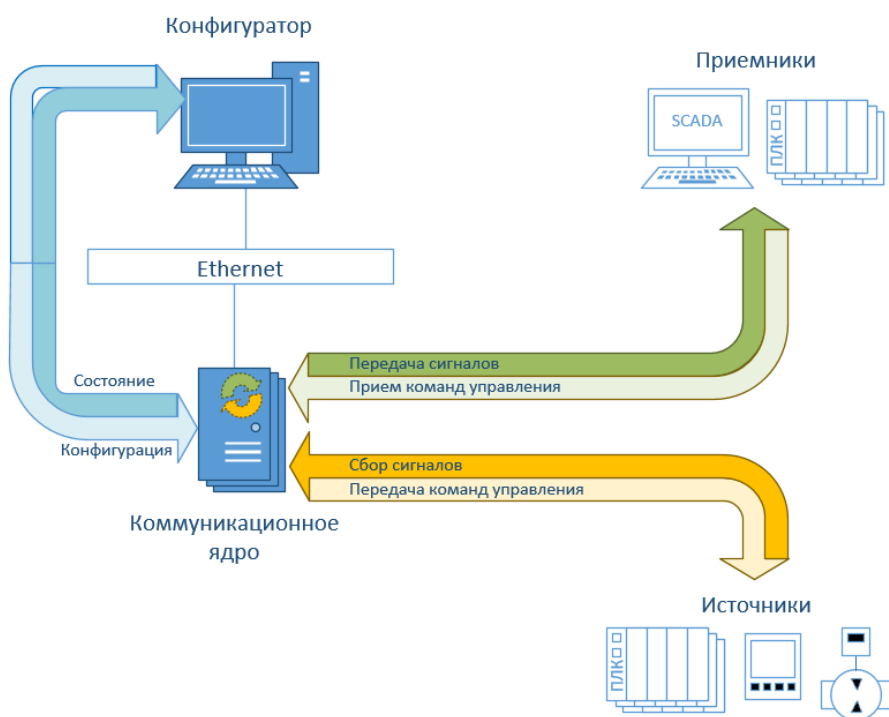


Рисунок 2 – Распределенная установка

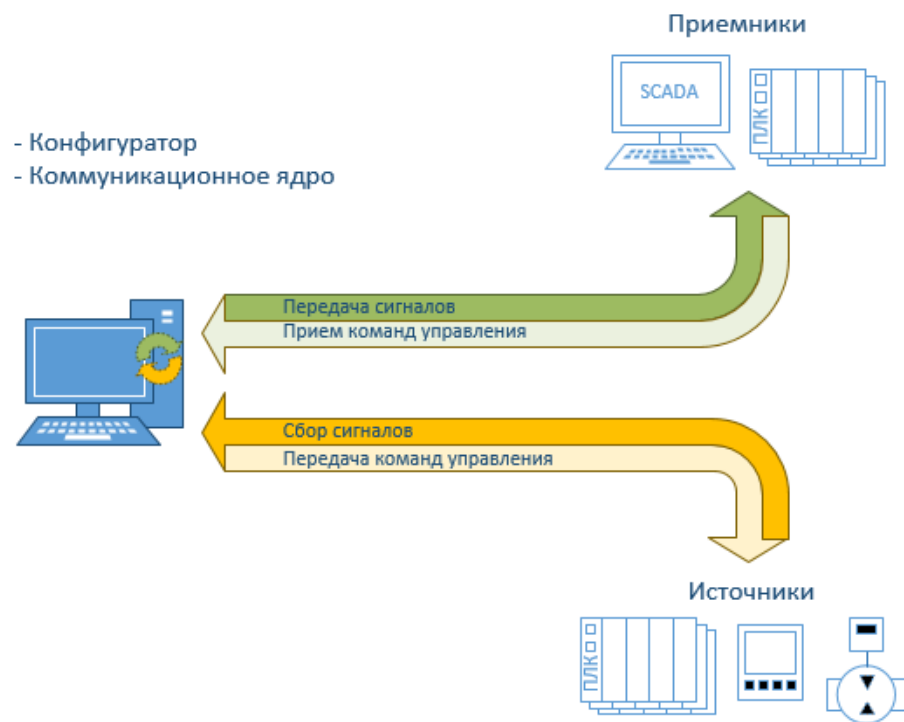


Рисунок 3 – Локальная установка

3 ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Минимальные требования к аппаратному и программному обеспечению приведены в Таблице 2.

Таблица 2 – Минимальные требования

Технические характеристики	Коммуникационное ядро					Конфигуратор				
Операционная система	Windows 10	Astra Linux 1.8	Debian 11	ПОСА «ХРОМ» 12	РЕД ОС	Windows 10	Astra Linux 1.8	Debian 11	ПОСА 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				

4 ОПИСАНИЕ ФУНКЦИЙ КС

4.1 Сбор и передача сигналов

КС выполняет сбор и передачу аналоговых и дискретных сигналов по различным протоколам, см. Таблицу 1.

КС собирает сигналы с внешних устройств по клиентским протоколам и размещает полученные данные в свою внутреннюю информационную модель. Для каждого сигнала во внутренней модели хранится принятое значение, метка времени и качество. Если метка времени и качество не принимаются по данному протоколу (например, по Modbus), то эти параметры в момент приема добавляются к сигналам. Поступившие во внутреннюю модель данные публикуются на серверных протоколах и становятся доступными для получения внешними клиентами.

Перечень поддерживаемых типов данных при сборе и передаче сигналов приведен в Таблице 3.

Таблица 3 – Типы данных сигналов

Типы данных внутренней информационной модели КС	Клиентские каналы (сбор/чтение сигналов)									Серверные каналы (передача/публикация сигналов)				
	OPC UA	MMS (CDC)	104 (ASDU)	Modbus (FUNC)	S7comm	FINS	MQTT sub	SNMP	Profibus DP	OPC UA	MMS (CDC)	104 (ASDU)	Modbus (FUNC)	MQTT pub
BOOLEAN	+	+(SPS/SPC)	+(1/30)	+(1/2/3/4)	+	+	+	-	+	+	+(SPS/SPC)	+(1/30)	+(1/2)	+
INT8U	Byte	-	-	-	+	-	Byte	-	+	Byte	-	-	-	+
INT8	SByte	-	+(5/32)	-	+	-	SByte	-	+	SByte	-	+(5/32)	-	+
INT16	+	-	+(11/35)	+(3/4)	+	+	+	-	+	+	-	+(11/35)	+(3/4)	+
INT16U	+	-	-	+(3/4)	+	+	+	-	+	+	-	-	+(3/4)	+
INT32	+	+(INC/INS)	+(15/37)	+(3/4)	+	+	+	+	+	+	+(INC/INS)	+(15/37)	+(3/4)	+
INT32U	+	-	-	+(3/4)	+	+	+	Counter32, Unsigned32, Gauge32, TimeTicks	+	+	-	-	+(3/4)	+
INT64	+	-	-	-	-	-	+	-	-	+	-	-	-	+

Типы данных внутренней информационной модели КС	Клиентские каналы (сбор/чтение сигналов)									Серверные каналы (передача/публикация сигналов)				
	OPC UA	MMS (CDC)	104 (ASDU)	Modbus (FUNC)	S7comm	FINS	MQTT sub	SNMP	Profibus DP	OPC UA	MMS (CDC)	104 (ASDU)	Modbus (FUNC)	MQTT pub
INT64U	+	-	-	-	-	-	+	Counter64, Gauge64	-	+	-	-	-	+
FLOAT32	Float	+(MV/CMV)	Float (13/36, 9/34)	+(3/4)	+	+	Float	-	+	Float	Float (MV/CMV)	+(13/36, 9/34)	+(3/4)	+
FLOAT64	Double	-	-	-	-	-	Double	-	-	Double	-	-	-	+
VISIBLE_STRING_255	String	-	-	String (3/4)	-	String	String	-	-	String	-	-	-	+
BITFIELD_2	+	CODED ENUM (DPC)	+(3/31)	-	-	-	-	-	-	-	CODED ENUM (DPC)	+(3, 31)	-	+

4.1.1 *Время обработки потока данных в КС*

В КС протоколы реализованы в виде отдельных плагинов. Последовательность приема/передачи обработки сигналов показана на Рисунке 4. Клиентский плагин отвечает за преобразование принятых данных в формат внутренней информационной модели КС (модуль распаковки). Ядро выполняет дополнительную обработку сигналов (модуль вычислений), временное хранение сигналов в промежутке времени, когда модуль ретрансляции занят и не может их принять (модуль аккумуляции), а также выполняет координацию работы плагинов (модуль ретрансляции). Серверный плагин отвечает за преобразование данных в формат протокола для передачи по серверному каналу (модуль упаковки).

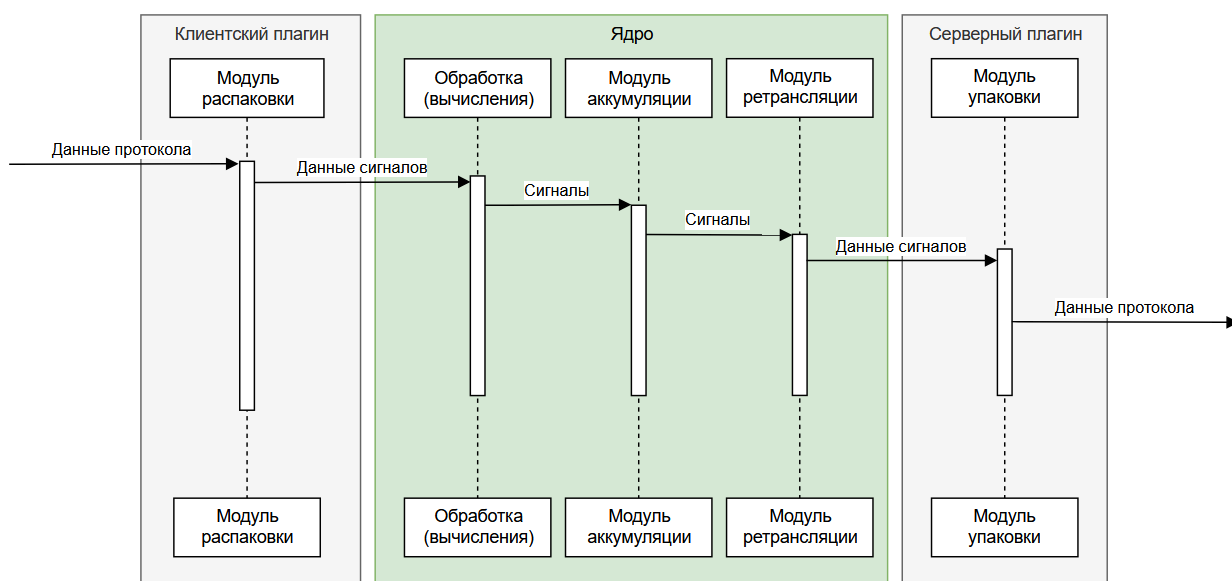


Рисунок 4 – Последовательность прохождения сигналов по протоколу в КС

Таким образом, время прохождения сигналов $t_{КС}$ вычисляется по формуле:

$$t_{КС} = t_{кл_пр} + t_{выч} + t_{акк} + t_{серв_пр}, \text{ где}$$

$t_{кл_пр}$ – время преобразования сигналов во внутренний формат КС, добавление метки времени и качества при их отсутствии.

$t_{выч}$ – время дополнительной обработки сигналов с учетом апертуры, инверсии, масштабирования из настроек серверного сигнала в КС.

$t_{акк}$ – время ожидания сигналов в очереди модуля аккумуляции.

$t_{серв_пр}$ – время преобразования сигналов из внутреннего формата КС в формат протокола.

Таким образом, время прохождения сигналов в КС складывается из временных затрат на преобразование данных в протокольных плагинах, обработку сигналов в ядре КС, ожидания в очереди перед передачей в серверный плагин.

4.1.2 Производительность обработки данных

В клиентском или серверном плагине обработка сигналов (т.е. потока данных) ведется независимо. В плагинах обработка для различных протоколов выполняется параллельно при многопоточной стратегии обработки каналов (Multithread) или последовательно при последовательной (Cyclic) стратегии.

Многопоточная стратегия обеспечивает наилучшее быстродействие КС по сравнению с последовательной. Рекомендуется использовать многопоточную стратегию обработки каналов. Последовательную стратегию целесообразно использовать в системах с низкой аппаратной производительностью.

Стратегия характеризуется *Временем цикла* (Cycle time), которое задается в файле статической конфигурации и по умолчанию равно 10 мс. Это время определяет поведение «сторожевого» таймера watchdog. Для последовательной стратегии обработки при превышении времени цикла таймер будет останавливать обработку сигналов. Для многопоточной стратегии при превышении времени цикла таймер соответствующего канала будет выдавать предупреждение в log-файл.

Время цикла при многопоточной стратегии – пороговое значение времени для каждого из потоков. Если указанное время цикла превышено для одного или нескольких потоков, то их таймеры генерируют ошибки обмена. Обмен данными в этих потоках (и в каналах) прекращается. По остальным потокам (каналам) обмен продолжается.

Время цикла при последовательной стратегии – суммарное время обработки всех потоков каналов. При срабатывании таймера обмен данными прекращается по всем потокам и генерируется ошибка обмена.

4.2 Прием и передача команд управления

КС выполняет прием и передачу команд управления. Перечень поддерживаемых типов данных при приеме и передаче команд приведен в Таблице 4.

Таблица 4 – Типы данных команд

Типы данных информационной модели Эликонт- КС	Серверные каналы (прием/чтение команд)				Клиентские каналы (передача/запись команд)						
	OPC UA	MMS	104	Modbus	OPC UA	MMS	104	Modbus	S7comm	FINS	Profibus DP
BOOLEAN	+	+(SPC/DPC)	+(45)	+(5, 15)	+	+	+(45)	+(5)	+	+	+
INT8U	Byte	-	-	-	Byte	-	-	-	-	-	UInt8
INT8	SByte	-	-	-	SByte	-	-	-	-	-	Int8
INT16	+	-	+(49)	+(6)	+	-	+(49)	+(6)	+	+	+
INT16U	+	-	-	+(6)	+	-	-	+(6)	+	+	+
INT32	+	+(INC)	-	+(16)*	+	-	-	+(16)	+	+	+
INT32U	+	-	-	+(16)	+	-	-	+(16)	+	+	+
INT64	+	-	-	-	+	-	-	-	-	-	-
INT64U	+	-	-	-	+	-	-	-	-	-	-

Типы данных информационной модели Эликонт- КС	Серверные каналы (прием/чтение команд)				Клиентские каналы (передача/запись команд)						
	OPC UA	MMS	104	Modbus	OPC UA	MMS	104	Modbus	S7comm	FINS	Profibus DP
FLOAT32	Float	-	+	+	Float	-	+	+	+	+	+
			(48/50)	(16)			(48/50)	(16)			
FLOAT64	Double	-	-	-	Double	-	-	-	-	-	-
VISIBLE_STRING_2 55	String	-	-	-	String	-	-	-	-	String	-
BITFIELD_2	-	-	+	-	-	-	+	+	-	-	-
			(46/47)				(46/47)	(15)			

Для протоколов МЭК 61870-5-104 и МЭК 61850 MMS поддерживаются следующие сценарии управления:

- прямая передача команд;
- передача команд с предварительным выбором.

Внимание! Поступление команд через один серверный канал от нескольких внешних клиентов воспринимается КС как прием команд от единого устройства.

При приеме команд управления по серверным каналам МЭК 60870-5-104 проверяется:

- последовательность команд. Если в серверном канале КС содержится команда с предварительным выбором, то поступление ASDU с командой исполнения считается ошибкой. В этом случае КС регистрирует поступление ошибочного ASDU, не передает команду дальше и не формирует ответного сообщения источнику команды;
- доступность приемника. Если КС не может передать приемнику принятую команду по причине отсутствия связи или неработоспособности приемника, то КС формирует ответное сообщение источнику с причиной передачи <48>;
- невыполнение команд приемником. Если КС передал приемнику команду управления и получил в ответ сообщение о невозможности выполнения команды, то КС формирует ответное сообщение приемнику с причиной передачи <49>.

При приеме команд управления по серверному каналу OPC UA не проверяется доступность или отказ приемника выполнить команду. Качество «узла на запись» всегда имеет значение «Good». Для контроля выполнения команды ведущее устройство должно контролировать изменение значения соответствующего «узла на чтение».

Команда Modbus передается с помощью процедуры записи переменных в регистры ведомого устройства. В качестве подтверждения активации команды управления рассматривается положительный ответ на процедуру записи. Отрицательный ответ с ошибкой в ответ на процедуру записи рассматривается, как отказ выполнить команду управления. Для команд Modbus можно настроить подтверждение их выполнения, для этого указываются регистры для считывания состояния объекта управления. Подтверждение выполнения команды по умолчанию выключено.

КС может контролировать факт выполнения команды за отведенное время. Если в настройках команды задано свойство «Время ожидания», то КС производит чтение состояния выполнения команды управления в заранее заданных регистрах адресного пространства ведомого устройства за указанное время. Если совместно с временем ожидания включена настройка «Ждать завершения», то КС контролирует время до подтверждения выполнения команды и поступления ответного сообщения. Если свойство «Время ожидания» установлено в 0, то проверка по времени выполнения команд не проводится.

4.3 Ретрансляция сигналов между клиентскими каналами

Данная функция обеспечивает горизонтальное межсерверное взаимодействие и используется, когда необходимо передать сигнал с сервера одного устройства на сервер

другого устройства. При ретрансляции происходит запись сигнала одного клиентского канала в другой клиентский канал. Ретрансляция выполняется в клиентских каналах: OPC UA, Modbus RTU/TCP, S7comm, FINS (Omron), Profibus DP. Тип данных сигналов при ретрансляции должен быть одинаков.

На Рисунке 5 показано, что для взаимодействия устройств («Устройство 1» и «Устройство 2») используется КС с настроенным клиентским каналом для приема сигнала («Клиент 1») и клиентским каналом, на котором ретранслируется данный сигнал в команду («Клиент 2»). После ретрансляции сигнал поступает на сервер «Устройства 2».

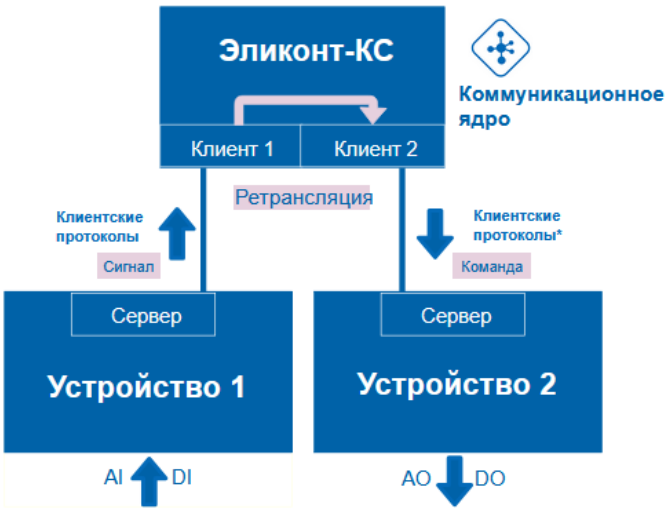


Рисунок 5 – Ретрансляция сигнала

В левой части рисунка указаны клиентские протоколы для сбора данных (чтения сигналов). Полный перечень протоколов сбора данных приведен в Таблице 1.

В правой части рисунка указаны клиентские протоколы для ретрансляции сигналов в команды управления (записи сигналов). Список этих протоколов приведен в начале раздела. Сведения о типах данных, применяемых при операциях чтения и записи, систематизированы в Таблице 5.

Таблица 5 – Типы данных при ретрансляции сигналов

Типы данных информационной модели Эликонт-КС	Клиентские каналы (ретрансляция/запись сигналов)				
	OPC UA	Modbus (FUNC)	S7comm	FINS	Profibus DP
BOOLEAN	+	+(5)	+	+	+
INT8U	Byte	-	-	-	+
INT8	SByte	-	-	-	+

Типы данных информационной модели Эликонт-КС	Клиентские каналы (ретрансляция/запись сигналов)				
	OPC UA	Modbus (FUNC)	S7comm	FINS	Profibus DP
INT16	+	+ (6)	+	+	+
INT16U	+	+ (6)	+	+	+
INT32	+	+ (16)	+	+	+
INT32U	+	+ (16)	+	+	+
INT64	+	-	-	-	-
INT64U	+	-	-	-	-
FLOAT32	Float	+ (16)	+	+	+
FLOAT64	Double	-	-	-	-
VISIBLE_STRING_255	String	-	-	String	-
BITFIELD_2	-	+ (15)	-	-	-

Важно! Команды, которые настроены на ретрансляцию, становятся недоступными для использования на серверных каналах.

4.4 Преобразование типов данных

КС позволяет преобразовывать (конвертировать) типы данных передаваемых сигналов. Поддерживается преобразование между целочисленными типами данных (int8, int16, int32 и др.) и между типами данных с плавающей точкой (float32, float64). Возможно повышающее и понижающее преобразование. При понижающим преобразованием (например, из int32 в int16), происходит отсечение верхней границы значений и потеря этих данных. Правила преобразования типов данных описаны в Приложении А Руководства пользователя.

4.5 Преобразование флагов качества

При передаче сигнала по серверным каналам КС конвертирует флаги качества одного протокола во флаги качества другого протокола.

При поступлении сигналов по клиентским каналам полученное качество преобразуется в качество внутренней модели. При передаче сигналов качество из внутренней модели преобразуется в протокольное качество. Исключением является передача сигналов из протокола в протокол одного типа (например, из OPC UA в OPC UA).

4.6 Преобразование меток времени сигналов из/в UTC

КС преобразует часовой пояс при передаче сигналов между устройствами с различными настройками часовых поясов (UTC, Local). Преобразование часового пояса, заданного на канале, выполняется согласно правилам в Таблице 6.

Таблица 6 – Часовые пояса

На внешнем источнике	Настройка КС		На внешнем приемнике	В КС (режим «Исполнение»)
	Клиентский канал	Серверный канал	(выход)	
(вход)				
UTC	UTC	UTC	UTC	Local
UTC	Local	UTC	UTC – TimeShift*	UTC
UTC	UTC	Local	Local	Local
UTC	Local	Local	UTC	UTC
Local	UTC	UTC	Local	Local + TimeShift
Local	Local	UTC	UTC	Local
Local	UTC	Local	Local + TimeShift	Local + TimeShift
Local	Local	Local	Local	Local

*TimeShift – сдвиг локального времени в часах относительно всемирного координированного времени (UTC).

Пример преобразования UTC -> Local (см. выделенную строку в таблице): на источнике данных установлено системное время в формате UTC. Клиентский канал настраивается на прием данных с метками времени в UTC. Серверный канал настраивается на работу в локальном часовом поясе (Local), куда данные и передаются. Сигналы поступают с внешнего устройства (UTC), в проект КС, где они проходит через настроенный канал. Метка времени на приемнике будет отображаться в соответствии с локальным часовым поясом, демонстрируя смещение из UTC в Local.

4.7 Присвоение метки времени сигналам

В КС метка времени - неотъемлемая часть сигнала. В случае, когда при приеме пакета информации метка времени отсутствует, КС формирует метку в момент приема сигналов.

Это правило выполняется при поступлении сигналов по протоколам, допускающим передачу данных без метки времени.

4.8 Инверсия дискретных сигналов

Для дискретных сигналов, имеющих значения: «0» и «1», КС выполняет инверсию: значение «1» преобразуется в «0», а значение «0» - в «1». По умолчанию – функция не используется (коэффициент равен «1»).

4.9 Масштабирование аналоговых сигналов

Для масштабирования значения передаваемого аналогового сигнала задается множитель - масштабирующий коэффициент. По умолчанию – функция не используется (коэффициент равен «1»).

4.10 Контроль актуальности принимаемого сигнала для OPC UA и МЭК 60870-5-104

Данная функция предназначена для контроля регулярного обновления сигнала.

Сигнал поступает на КС спорадически. Для контроля регулярности его обновления пользователь может задать для клиентского канала протокола OPC UA и МЭК 60870-5-104 свойство «Время актуальности сигналов», т.е. интервал времени, в течение которого сигнал должен обновиться. Если КС определит, что сигнал ни разу не обновился в течение этого времени, то на канале передачи данных для него выставляется признак качества «Неактуальный». Это означает, что полученное значение сигнала было достоверным, но не обновлялось дольше, чем положено. По умолчанию - функция отключена («Время актуальности сигналов» = 0).

4.11 Контроль границ диапазона принимаемых сигналов

КС позволяет проверить достоверность значений принимаемых аналоговых сигналов по средствам контроля границ диапазона сигналов. При выходе значения сигнала за границы диапазона, на канале передачи данных у этого сигнала выставляется признак качества – переполнение.

Важно! При указании границ диапазона равных нулю, функция деактивирована.

Внимание!!! Данная функция приводит к изменению качества сигналов, принимаемых от источника сигналов. Функция является опциональной и используется по необходимости.

4.12 Передача аналоговых сигналов по апертуре

Передача аналоговых сигналов возможна с учетом заданной апертury. Апертюра представляет собой изменение значения аналогового сигнала, выраженное в процентах и настраиваемое на серверном канале.

Если значение апертury равно нулю, любое принятое изменение значения сигнала будет передаваться по серверному каналу.

4.13 Буферизация данных для МЭК 60870-5-104 и МЭК 61850 MMS

Данная функция обеспечивает сохранность информации при кратковременных перебоях связи для серверных протоколов МЭК 60870-5-104 и МЭК 61850 MMS. В случае обрыва соединения выполняется накопление передаваемой информации. После восстановления связи накопленные данные отправляются клиенту. Если буфер заполнен, то новые данные перезаписывают «старые».

Для протокола МЭК 60870-5-104 буферизация настраивается для канала и по умолчанию включена. Сигналы/команды выстраиваются в очереди - пополняют буфер, если изменяется их значение или качество. Дискретные сигналы используют общий буфер, а аналоговые сигналы имеют индивидуальный буфер, см. Рисунок 6. Для дискретных/аналоговых сигналов настраивается свойство «Размер буфера ТС»/«Размер буфера ТИ».

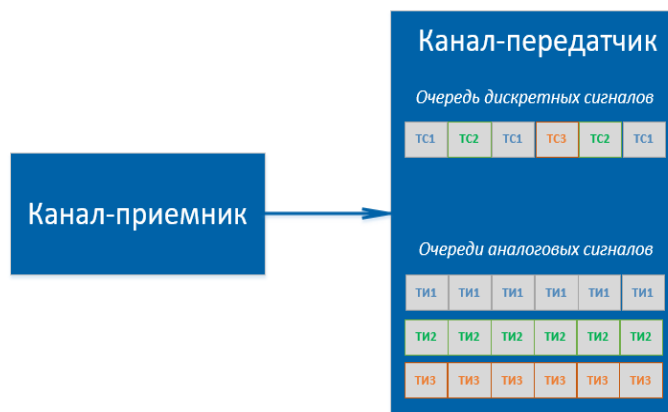


Рисунок 6 – Буферизация сигналов

Для протокола МЭК 61850 MMS буферизация настраивается для отчета «Report» и по умолчанию отключена.

При восстановлении соединения КС выполняет передачу данных из буфера в фоновом режиме. Это означает, что приоритет передачи данных из буфера ниже приоритета передачи значений реального времени, т.е. спонтанных изменений данных, ответов на общих опрос, циклической передачи данных и т.п. Передача данных из буфера выполняется прикладными блоками данных с меткой времени CP56Время2a и причиной передачи <3> := спорадически spont.

4.14 Резервирование каналов OPC UA

Данная функция повышает стабильность обмена данными за счет использования двух клиентских каналов. При резервировании выполняется автоматическое переключение с недоступного канала на резервный. В КС реализован режим «горячего» резервирования (Hot). Это означает, что КС подключается сразу к двум устройствам и получает от них данные.

На Рисунке 7 показан пример обмена данными по протоколу OPC UA при работе с резервированными каналами. КС собирает данные с внешних резервных серверов, данные

поступают в КС одновременно. Предполагается, что устройства содержат одинаковый набор данных.

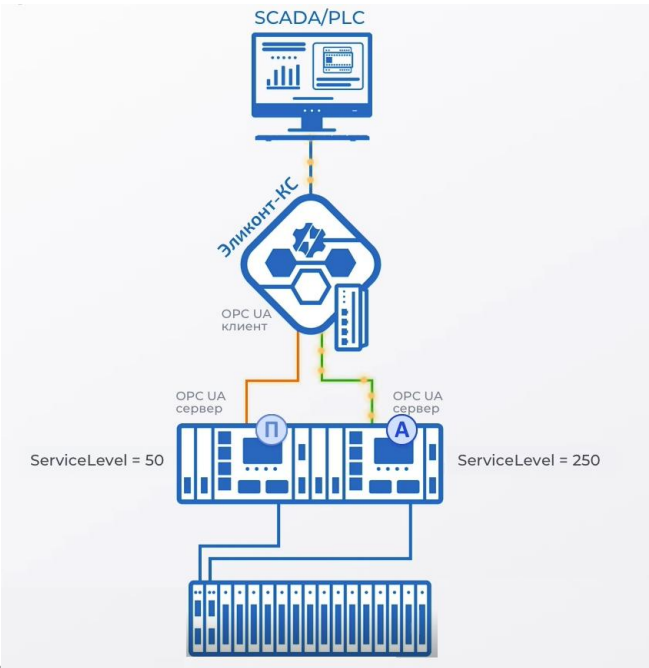


Рисунок 7 –Пример обмена данными при резервировании

КС использует значение параметра узла ServiceLevel, чтобы определить устройство с самым высоким уровнем обслуживания и использовать его как приоритетный источник данных.

Примечание - ServiceLevel – это параметр от 0 до 255, указывающий на способность сервера обеспечивать клиента данными. Значения 0-199 – данные не доступны или частично не доступны. Значения 200-255 – клиент может получить всю информацию от сервера.

В качестве активного источника данных для сигналов выбирается активный канал с параметром ServiceLevel ≥ 200 , у которого этот параметр больше. Если ServiceLevel одинаковый, то выбирается канал, по которому связь была установлена раньше. Другой канал считается резервным (пассивным) и данные с него игнорируются, пока основной канал активен.

Правила обмена данными с резервированным каналом:

1. КС устанавливает одновременно два соединения с настройками подключения для основного и резервного устройства. КС выбирает соединение, которое будет использовать как активное, и его данные транслирует в серверный канал.

№	Состояние соединения	Описание
1	Один канал активный, другой – пассивный. [A P] или [P A]	КС установил соединение по обоим каналам и выбрал один канал как активный. КС ведет обмен данными по обоим каналам, но передает данные в серверный канал только от активного канала.

№	Состояние соединения	Описание
Примечание: [A] – канал активный; [P] – канал пассивный.		

2. при отказе активного соединения КС начинает передачу оперативных данных из другого канала (пассивный канал становится активным). КС периодически пытается восстановить связь по недоступному каналу. При восстановлении соединения КС продолжает работать с данными из текущего активного канала. См. пункт 1.

№	Состояние соединения	Описание
2	Один канал активный, другой – не доступен [– A] или [A –]	КС установил соединение по одному каналу и передает его данные в серверный канал. КС пытается установить соединение по другому каналу.
Примечание: [A] – канал активный; [–] – канал недоступен.		

При изменении активности канала ведущее устройство может получить данные с «устаревшими» метками времени от ведомого устройства (например, если системное время одного устройства отстает от другого).

3. при потере всех соединений КС пытается восстановить потерянные каналы связи.

№	Состояние соединения	Описание
3	Оба канала не доступны [– –]	КС не имеет соединения ни с одним устройством

4.15 Генерация сигналов

Генератор сигналов представляет собой источник сигналов всех числовых типов данных, поддерживаемых в Коммуникационном ядре. Сгенерированные сигналы передаются по выбранным серверным каналам.

Функция генерации сигналов защищена лицензионными ограничениями. Сигналы генератора учитываются в лицензии как сигналы клиентского канала.

4.16 Интерфейс взаимодействия с пользовательским приложением

КС предоставляет интерфейс для подключения устройств и систем, ведущих обмен данными по *проприетарным* (англ. – proprietary, собственный) протоколам.

Для подключения таких устройств и систем необходимо разработать пользовательское приложение для обмена данными с этими устройствами и передачи их в КС. См. Рисунок 8. Данные можно «читать» из КС и передавать во внешнюю систему.

Информационный обмен между пользовательским приложением и КС реализован на базе технологии gRPC (актуальный перечень языков доступен на сайте <https://grpc.io/docs/languages/>).

Интерфейс взаимодействия с КС определяется методами работы с сигналами и командами, описанными в файле *elecont.proto*, см. электронную справку (Руководство пользователя раздел «Разработка приложения»).

Приложение самостоятельно формирует параметры обмена для сигналов и команд (значение, метку времени и др.).

Для настройки обмена данных в Конфигураторе необходимо:

1. создать конфигурацию, включающую клиентский и серверный каналы протоколов «User channel» клиент и сервер;
2. для сигнала/команды заполнить параметр «Данные пользователя». Этот параметр приложение может использовать для идентификации элементов данных.

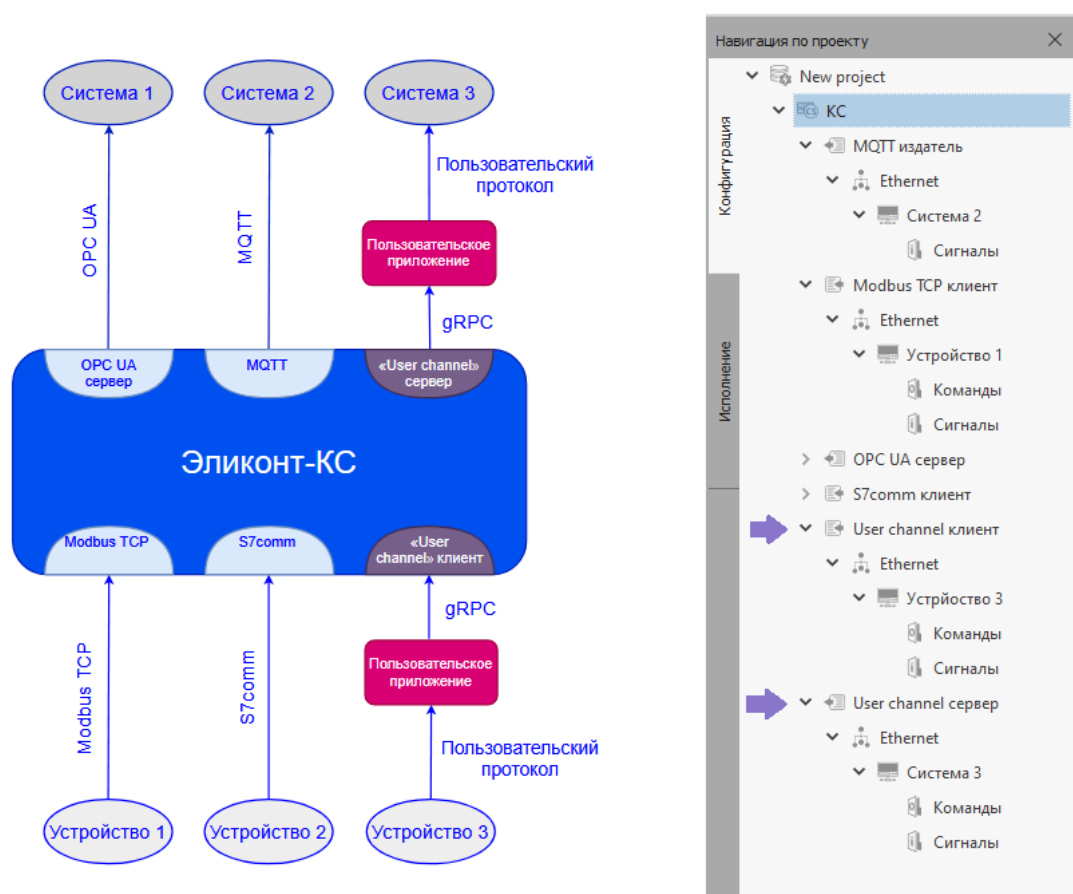


Рисунок 8 – Схема и конфигурация для обмена данных (пример)

Предоставляемый интерфейс управления расширяет возможности взаимодействия пользовательского приложения с системой, обеспечивая доступ к сервисным функциям КС.

Управление КС реализуется через взаимодействие с сервисом API Gateway на базе технологии gRPC. Контракт взаимодействия с сервисом (методы и структуры данных) содержится в proto-файле *api_gateway.proto*, см. электронную справку (Руководство пользователя раздел «Разработка приложения»).

Использование в сторонних скриптах методов, определенных в proto-файле, позволяет обеспечить:

- конфигурирование устройства;
- получение состояния устройства;
- управление информационными каналами устройства;
- получение информации об устройстве, включая лицензионные ограничения;
- управление трассировкой информационных каналов;
- получение журналов сообщений;
- получение состояния БД архива.

4.17 Самодиагностика

КС предоставляет следующие данные о своей работе:

1. Журнал событий.

КС записывает все события, происходящие в системе в хронологическом порядке в файл. Доступ к файлу можно получить в Web-интерфейсе, или найти файл в каталоге установки программы. Уровень детализации ведения журнала настраивается пользователем.

В Конфигураторе также можно просмотреть последние события из журнала.

2. Аппаратные и программные показатели КС

Аппаратные показатели компьютера с установленным КС можно получить в Web-интерфейсе КС (см. Рисунок 9). Там же можно получить сведения о лицензии, использовании протоколов и количестве сигналов в конфигурации (см. Рисунок 10).

3. Информация о состоянии каналов приема-передачи данных

В КС формируются статусные сигналы для диагностики состояния каналов приема-передачи данных Эликонт-КС. Эти сигналы пользователь может использовать в системах «верхнего» уровня для мониторинга. Значение сигнала «1» соответствует состоянию канала «есть соединение», значение «0» - «нет соединения».

4. Трассировка для протоколов Modbus и Profibus DP.

Отображение «сырых данных» обмена с внешним устройством от КС предназначено для отладки и выполняется в Конфигураторе в окне трассировки.

5. Диагностика канала и устройств Profibus DP.

Диагностические сообщения содержат протокольную информацию о состоянии и ошибках обмена данными с расшифровкой. Сообщения доступны в Конфигураторе и дублируются в лог-файле. См. Рисунок 11.

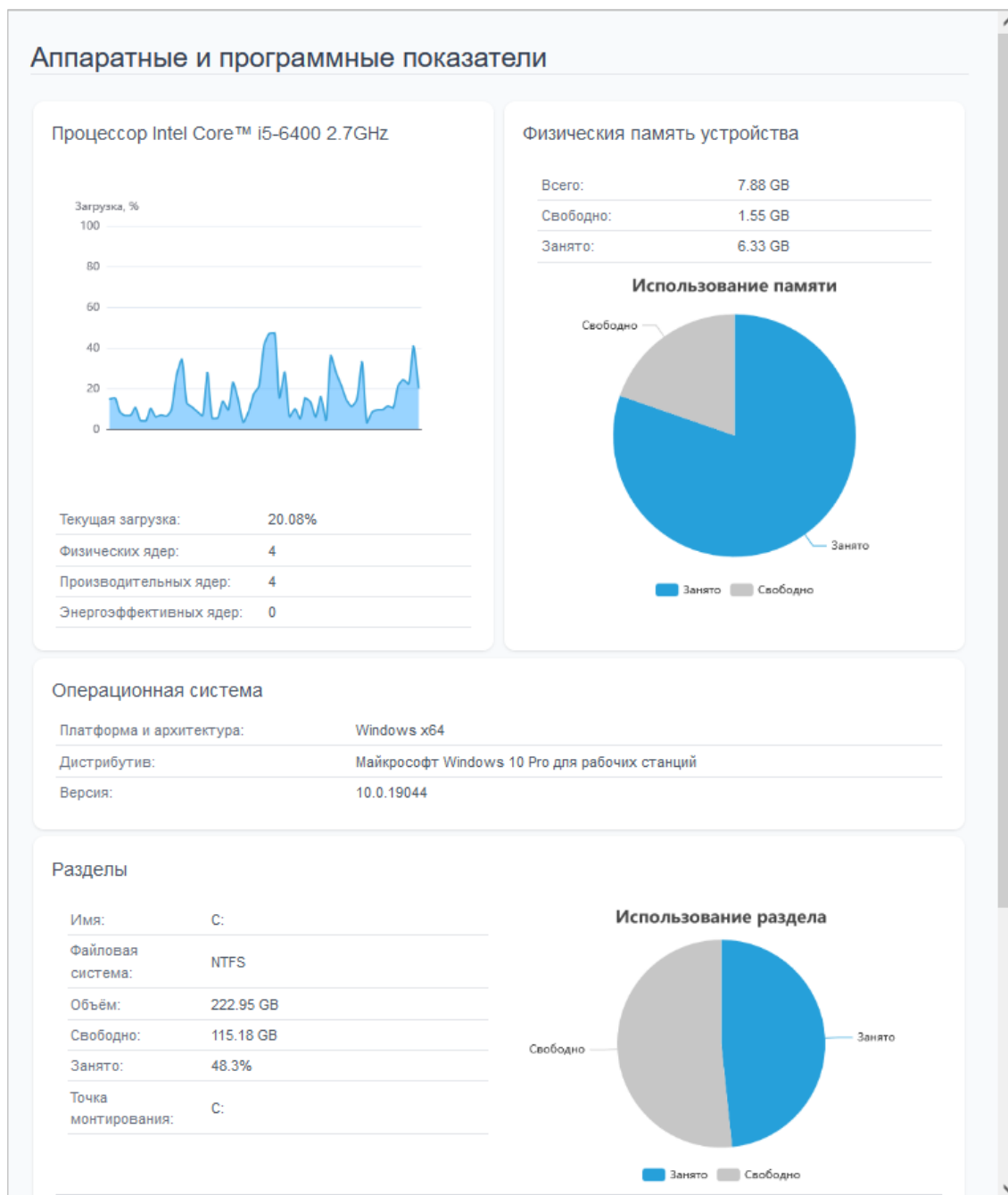


Рисунок 9 – Мониторинг показателей КС

Информация о лицензии	
Ключ	Обнаружен
Идентификатор ключа	#3f7948b4
Режим	Нормальный
Ограничения	Клиентские протоколы
	GENERATOR CLIENT, IEC104 CLIENT, IEC61850 CLIENT, MODBUS RTU CLIENT, MODBUS TCP CLIENT, MQTT CLIENT, OMRON_FINS CLIENT, OPCUA CLIENT, PROFIBUS_DP CLIENT, S7COMM CLIENT, SNMP CLIENT
	Серверные протоколы
	IEC104 SERVER, IEC61850 SERVER, MODBUS RTU SERVER, MODBUS TCP SERVER, MQTT SERVER, OPCUA SERVER
Функции	ARCHIVE_POSTGRESQL, ARCHIVE_SQLITE
	Максимальное количество сигналов
	10000
Конфигурация	Клиентские протоколы
	GENERATOR CLIENT, IEC104 CLIENT, PROFIBUS_DP CLIENT
	Серверные протоколы
	Функции
Количество сигналов	106
	Оставшееся время
	Не ограничено

Рисунок 10 – Сведения о лицензии и текущей конфигурации

```

06.06.2025 11:29:00.643 [I] <CommServer> Channel "Станция" (id = 0): Diagnostics. Extended block: 44 01 40 80 83 80 a2 84
c2 bb 85 47 3c 08 82 00 00 1b 00 0c 00
06.06.2025 11:29:00.643 [I] <CommServer> Channel "Станция" (id = 0): Diagnostics. Identifier Related Data: 44 01 40 80
06.06.2025 11:29:00.643 [I] <CommServer> Channel "Станция" (id = 0): Identifier-related event: The module installed in slot
1 has an event(s).
06.06.2025 11:29:00.643 [I] <CommServer> Channel "Станция" (id = 0): Identifier-related event: The module installed in slot
15 has an event(s).
06.06.2025 11:29:00.643 [I] <CommServer> Channel "Станция" (id = 0): Identifier-related event: The module installed in slot
24 has an event(s).

```

Рисунок 11- Пример диагностического блока Identifier Related Data для Profibus DP

4.18 Архивирование сигналов

Функция архивирования обеспечивает запись сигналов во внутреннюю базу данных SQLite и внешнюю базу данных PostgreSQL. Запись выбранных сигналов выполняется при изменении их значения или качества (т.е. спорадически).

Сигналы архивируются на клиентских и (или) на серверных каналах. На серверном канале сигналы записываются с учетом обработки. Для обращения к архивированным данным могут быть использованы стандартные механизмы доступа.

Функция архивирования защищена лицензионными ограничениями.

Для архивирования в БД SQLite настраивается время хранения данных и максимальный объем БД. При записи и хранении сигналов в БД выполняются правила:

1. время хранения сигнала отсчитывается от момента записи сигнала в архивную БД;
2. удаление данных выполняется при превышении времени хранения или при переполнении БД.

5 КОНТРОЛЬ ЦЕЛОСТНОСТИ ПО ЭЛИКОНТ-КС В ОС ASTRA LINUX

Контроль целостности файлов используется для обнаружения несанкционированного изменения их содержимого. В ОС Astra Linux контроль выполняется в режиме замкнутой программной среды (ЗПС). ЗПС обеспечивает доступ пользователей и процессов только к санкционированным программам и предотвращает несанкционированное изменение исполняемых файлов.

При запуске ПО проверяется встроенная в файлы электронная подпись. Если файл не прошел проверку, его загрузка блокируется. Это гарантирует невозможность внесения несанкционированных изменений, способных повлиять на работу системы. Таким образом, контроль целостности является важной мерой безопасности, обеспечивающей стабильность и защиту ПО.

6 РЕАЛИЗАЦИЯ ПРОТОКОЛОВ

6.1 МЭК 61850-8-1 MMS

PIC (Protocol Implementation Conformance Statement) для клиентского протокола – см. в электронной справке.

6.2 МЭК 60870-5-104

Формуляры для клиентского и серверного протокола – см. в электронной справке.

6.3 Profibus DP

Объем реализации протокола приведен в таблице:

Наименование	Значение	Примечание
Основные характеристики:		
Версия протокола	PROFIBUS DP-V0	Поддержка обмена входными и выходными данными на уровне базовых функций. Поддержка расширенных диагностических сообщений - Device-based diagnostics – функция DP-V1
Режим работы	Master Class 1	КС – ведущее устройство, инициирует циклический обмен с ведомыми устройствами
Количество подключений к ведомым устройствам	1 Master	
Количество шин (Bus) Количество станций (Stations) в сегменте Количество станций на шине (с использованием повторителей, несколько сегментов)	1 До 32 До 125 (1-125)	
Максимальное количество станций в сегменте	32	
Разовый объем передаваемых данных на одно устройство	До 244 байт	
Скорость обмена	На сторонних компьютерах: 9,6 19,2 Кбит/с. На шлюзе EliGate:	

Наименование	Значение	Примечание
	9,6 19,2 Кбит/с, 0,5 1,5 3 Мбит/с	
Поддержка файлов GSD (General Station Description)	V5.1	
Типы данных	Чтение входов и запись выходов: BOOLEAN, INT8, UINT8, INT16, UINT16, INT32, UINT32, FLOAT32	
Функции:		
Обмен данными	Циклический обмен данных (Cyclic Data)	Обмен данных между КС и периферийными устройствами с интервалами обмена (10-100 мс) по стандарту DP-V0. (КС – ведущее устройство класса 1)
<i>Конфигурирование сети PROFIBUS:</i>		Выполняется в Конфигураторе
Импорт файлов GSD	Да	Импорт файлов с описанием периферийных устройств
Ручная настройка параметров	Да	Изменение параметрической последовательности станции и модулей, настройка адреса сигналов/команд
Автоматический контроль связи в сети Profibus	Определение и устранение ошибок передачи данных	Механизмы контроля битов ошибок, контроля времени ожидания отклика. При потере связи выполняется повторная инициализация станции
Диагностика устройств Profibus	Базовая: Channel-based diagnostics, Identifier-based diagnostics и расширенная (DP-V1): Device-based diagnostics	Получение диагностической информации от ведомых устройств и модулей (modules)

7 СВЕДЕНИЯ О ЛИЦЕНЗИРОВАНИИ

Эликонт-КС может работать с лицензионным ключом (коммерческая лицензия), или без него (в деморежиме). См. Таблицу 7.

При использовании ключа функциональность ПО определяется ограничениями ключа. При работе без ключа максимальное количество сигналов – 100, доступны функции архивирования.

Программа работает в одном из режимов – «Нормальный» и «Ограниченный».

Если работа ведется в рамках установленной функциональности, то режим работы - «Нормальный».

Если ограничения превышены или потеряна связь с ключом, то включается «Ограниченный» режим работы, который лимитирован по времени.

Таблица 7 – Типы лицензий

Типы лицензий	Функциональность	Нормальный режим	Ограниченный режим
Работа с ключом (коммерческая лицензия)	Набор протоколов, наличие функции архивирования: ARCHIVE_POSTGRESQL, ARCHIVE_SQLITE, максимальное допустимое количество сигналов*	Ограничения по функциональности не превышены, есть связь с ключом	При превышении количества сигналов: - 5 часов работы при превышении ограничений, при наличии связи с ключом; - 48 часов работы при потере связи с ключом
Работа без ключа (деморежим)	До 100 сигналов*, все протоколы, наличие функции архивирования	Ограничения по количеству сигналов не превышены	При превышении количества сигналов: - 5 часов работы

* Используемые клиентские сигналы и команды

Примеры отображения информации о работе с ключом и в деморежиме представлены на Рисунках 12 и 13.

Информация о лицензии		
Ключ	Обнаружен	
Идентификатор ключа	#3F7948b4	
Режим	Нормальный	
Ограничения	Клиентские протоколы	GENERATOR CLIENT, IEC104 CLIENT, IEC61850 CLIENT, MODBUS RTU CLIENT, MODBUS TCP CLIENT, MQTT CLIENT, OMRON_FINS CLIENT, OPCUA CLIENT, PROFIBUS_DP CLIENT, S7COMM CLIENT, SNMP CLIENT
	Серверные протоколы	IEC104 SERVER, IEC61850 SERVER, MODBUS RTU SERVER, MODBUS TCP SERVER, MQTT SERVER, OPCUA SERVER
	Функции	ARCHIVE_POSTGRESQL, ARCHIVE_SQLITE
	Максимальное количество сигналов	10000
Конфигурация	Клиентские протоколы	GENERATOR CLIENT, IEC104 CLIENT, PROFIBUS_DP CLIENT
	Серверные протоколы	
	Функции	
	Количество сигналов	106
Оставшееся время	Не ограничено	

Рисунок 12 – Работа с ключом

Информация о лицензии		
Ключ	Отсутствует	
Идентификатор ключа	-	
Режим	Нормальный	
Ограничения	Клиентские протоколы	GENERATOR CLIENT, IEC104 CLIENT, IEC61850 CLIENT, MODBUS RTU CLIENT, MODBUS TCP CLIENT, MQTT CLIENT, OMRON_FINS CLIENT, OPCUA CLIENT, PROFIBUS_DP CLIENT, S7COMM CLIENT, SNMP CLIENT
	Серверные протоколы	IEC104 SERVER, IEC61850 SERVER, MODBUS RTU SERVER, MODBUS TCP SERVER, MQTT SERVER, OPCUA SERVER
	Функции	ARCHIVE_POSTGRESQL, ARCHIVE_SQLITE, RESERV
	Максимальное количество сигналов	100
Конфигурация	Клиентские протоколы	PROFIBUS_DP CLIENT
	Серверные протоколы	
	Функции	
	Количество сигналов	0
Оставшееся время	Не ограничено	

Рисунок 13 – Работа в деморежиме (без ключа)

СОКРАЩЕНИЯ, ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Проект	- описание конфигурации Коммуникационного ядра (или нескольких Коммуникационных ядер). Проект является локальным, т.е. хранится в локальном файле
Команда	- управляющее воздействие, передаваемое от клиента к серверу. В зависимости от типа протокола реализуется процедура передачи управляющего сигнала или уставки регулятору (МЭК 61850, МЭК 60870-5-104) или процедуру записи значения сигнала (OPC UA, Modbus RTU/TCP)
Сигнал аналоговый	- сигнал данных, у которого каждый из представляющих параметров описывается функцией времени и непрерывным множеством возможных значений (ГОСТ 17657-79)
Сигнал цифровой (дискретный)	- сигнал данных, у которого каждый из представляющих параметров описывается функцией дискретного времени и конечным множеством возможных значений (ГОСТ 17657-79)