

ПКУ БАРС СУАБ "Комбат". Функциональные характеристики

Документ содержит описание функциональной архитектуры блока архивации, регистрации и связи (БАРС) системы управления аккумуляторной батареей (СУАБ) «Комбат».

Оглавление

Оглавление.....	1
Список сокращений и терминов.....	2
1. Функциональная архитектура.....	3
2. Иллюстрация функциональности ПКУ БАРС СУАБ "Комбат".....	6
3. Метаданные описания БАРС СУАБ.....	16
3.1. Метаданные верхнего уровня.....	18
3.2. Метаданные нижнего уровня. Оборудование.....	19
3.3. Формат Excel-файлов метаданных описания оборудования.....	22
3.4. Метаданные описания оборудования. XML-формат.....	27
3.5. Метаданные описания ЧМИ.....	27
3.6. Метаданные описания ЧМИ. XML-формат.....	33
4. Эмуляторы оборудования.....	34
5. Форматы описания, хранения и алгоритмы съема событий и потоковых данных.....	34
5.1. Периодические операции, события и потоковые данные.....	34
5.2. Алгоритмы съема потоковых данных и прерываний.....	35
5.3. Прерывания (Важные события оборудования).....	36
5.4. Форматы хранения, алгоритмы сжатия и передачи потоковых данных.....	37
6. Язык описания скриптов.....	37
7. Согласование со стандартом МЭК 61850.....	39
8. Согласование со стандартом МЭК 60870.....	41

Список сокращений и терминов

API - Application Programm Interface (программный интерфейс приложения). В данном случае – средство дистанционного доступа к данным сервера посредством Rest API (webAPI на основе протокола http).

JAR-файл – программный файл, исполняемый в Java, содержит все программные компоненты серверных и клиентских приложений БАРС СУАБ. Текущее имя **ESS2Desktop.jar**.

Артефакт (Artifact) – файл, хранящийся на сервере, известный серверу данных. Виды артефактов: аудио-файл, видео, фото, документ (pdf), текст, электронная таблица (xls), исполняемый файл (jar, прошивка), xml-файл.

БД – база данных.

БУС – блок управления стрингом СУАБ, верхний уровень СУАБ, интерфейс ModBus БУС специфицирован как интерфейс оборудования, с которым взаимодействует БАРС СУАБ.

БОМ – блок опроса модуля – элемент второго уровня, групповой контроллер БАК.

БАК – блок аккумуляторного элемента – элемент третьего уровня.

Киоск-приложение – приложение с ЧМИ в виде единственного окна, используется при работе со встроенными мониторами.

ПД – потоковые данные.

ПК-клиент – клиентское приложение, работающее в среде ОС общего применения с использованием стандартного многооконного ЧМИ.

ПКУ БАРС СУАБ "Комбат" - Программный комплекс для управления блоком архивации, регистрации и связи (БАРС) системы управления аккумуляторной батареей (СУАБ) «Комбат».

ПрК – промышленный компьютер.

СУАБ - система управления аккумуляторной батареей.

СУБД – система управления базой данных.

СД - сервер данных, серверное приложение ПКУ БАРС СУАБ "Комбат".

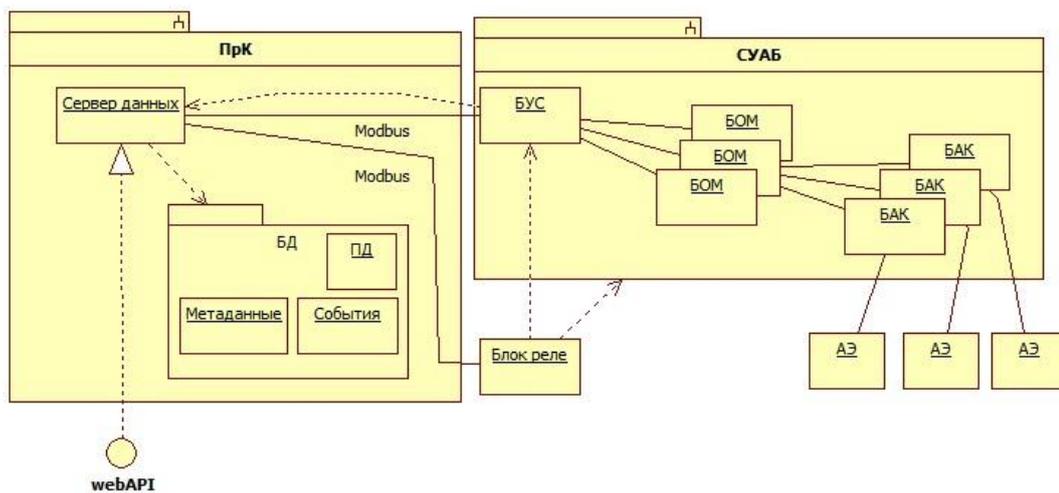
ЧМИ – человеко-машинный интерфейс

1. Функциональная архитектура

Терминология раздела:

- **Конфигурация** – набор спецификаций архитектуры, интерфейсов оборудования и ЧМИ приложений.
- **Развертывание конфигурации** – загрузка и связывание метаданных конфигурации сервером данных и их верификация.
- **Активирование конфигурации** – инициализация соединений с устройствами в развернутой конфигурации в соответствии с мета-данными, идентификация и проверка работоспособности устройств.
- **Сервер данных БАРС СУАБ** – программная компонента, выполняющая основные функции ПУ БАРС СУАБ:
 - реализация webAPI для доступа клиентских приложений к мета-данным описания БАРС СУАБ, к интерфейсам оборудования, авторизации, контроля доступа, администрирования и т.д.;
 - сбор и передача и обработка архивируемых данных;
 - мониторинг состояния оборудования БАРС СУАБ, развертывание и активирование конфигурации БАРС СУАБ.
- **Роль пользователя** – статус, хранящийся в данных авторизации, в соответствии с которым он имеет права на доступ к отдельным компонентам приложений. В БАРС СУАБ определены следующие роли: суперадминистратор, администратор, сервисный инженер, инженер, оператор, посторонний (гость).
- **Уровень доступа** – определяемое в соответствии с ролью пользователя уровень прав на *просмотр* и *изменение* данных при работе с приложением оператора. Для каждой формы приложения в мета-данных устанавливаются свои уровни доступа.
- **ПК-приложение администратора/сервисного инженера/оператора** – клиентское desktop-приложение для администрирования серверных приложений, редактирования мета-данных. Содержит также аналог панели оператора БАРС СУАБ в виде отдельной вкладки.
- **Панель оператора** – киоск-приложение в ПрК, эквивалент панели оператора.
- **Журнал событий** - содержит записи о зафиксированных событиях. Виды журналов:
 - журнал дискретных событий;
 - журнал аварий;
 - журнал изменения конфигураций;
 - журнал системных событий (БАРС СУАБ);
 - журнал изменения уставок.
- **БУС** – блок управления стрингом СУАБ, верхний уровень СУАБ, интерфейс ModBus БУС специфицирован как интерфейс оборудования, с которым взаимодействует БАРС СУАБ.
- **БОМ** – блок опроса модуля – элемент второго уровня, групповой контроллер БАК.
- **БАК** – блок аккумуляторного элемента – элемент третьего уровня.

Программный комплекс (ПК) в рамках которого реализован **ПКУ БАРС СУАБ "Комбат"** представляет собой платформенно-независимую клиент-серверную архитектуру, предназначенную для мониторинга состояния оборудования с протоколом Modbus.



Основные положения его функциональности:

- ПК работает с оборудованием через протокол Modbus;
- оборудование является автономным с точки зрения управляемости со стороны ПК. ПК не реализует алгоритмы управления оборудованием в реальном времени;
- ПК используем **метаданные** описания интерфейсов оборудования для работы с ним в режиме опроса регистров. Описание наряду с общепринятыми видами регистров (данные, уставки, команды, состояния, битовые регистры команд и состояний и пр.) позволяют описывать массивы и коллекции элементов, в т.ч. вложенные;
- Описание интерфейсов оборудования и ЧМИ (разметка и отображение элементов разметки на регистры оборудования) основано на использовании метаданных: создание новой конфигурации оборудования, изменение интерфейсов и разметки ЧМИ не требует пересборки программных компонент;
- Метаданные описания интерфейсов оборудования и ЧМИ могут использовать контейнеры: коллекции и массивы: этим обеспечивается иерархическая структура описаний произвольной глубины;
- Сервер данных реализует периодический опрос регистров оборудования с целью фиксации аварий/предупреждений, дискретных событий, прерываний (важных событий) – изменений разрядов битовых регистров. Структура событий описана в метаданных оборудования;
- Сервер данных реализует периодический опрос регистров для сохранения их содержимого в потоковых данных и мгновенных снимках. Структура потоковых данных описана в метаданных оборудования;
- Собранные потоковые данные используются для их передачи по протоколам МЭК 60870 и МЭК 61850. Описания метаданных БАРС СУАБ автоматически конвертируются в модель описания оборудования для МЭК 61850. Поддержка протоколов реализуется отдельными серверными компонентами;
- Программные эмуляторы оборудования являются серверными компонентами протокола Modbus/TCP, что позволяет использовать совместимые конфигурации и общие метаданные для моделирования в процессе разработки и работы с реальным оборудованием;

ПКУ БАРС СУАБ "Комбат" содержит или использует стандартные следующие клиентские компоненты:

- приложение администратора/разработчика/сервисного инженера (ПК-клиент);

- приложение оператора, запускаемое в режиме киоск-клиента;
- Android-приложение оператора;
- стандартное клиентское приложение протокола МЭК 60870;
- стандартное и оригинальное клиентские приложения протокола МЭК 61850;
- стандартное клиентское приложение для внутреннего протокола оборудования Modbus.

ПКУ БАРС СУАБ "Комбат" осуществляет следующие виды деятельности по работе с оборудованием СУАБ "Комбат":

- мониторинг и управление действующим оборудованием (роли **Оператор, Сервисный инженер**) – деятельность **Мониторинг**;
- конфигурирование программного комплекса в процессе эксплуатации и настройки - настройка параметров интерфейсов оборудования, параметров конфигурации, развертывание конфигурации, подключение к оборудованию, управление протоколами МЭК 60870, МЭК 61850, шлюзом Modbus (роли **Сервисный инженер, Администратор**) – деятельность **Конфигурирование**;
- разработка и редактирование интерфейсов оборудования и ЧМИ приложений в процессе усовершенствования и оборудования и адаптации ЧМИ под новые требования – деятельность **Разработка**.

ПКУ БАРС СУАБ "Комбат" имеет трехуровневую структуру, отраженную в структуре регистров Modbus – БУС-БОМ-БАК (рис.1). БУС содержит регистры уставок и управления конфигурацией и интегрированных данных для всей СУАБ, на каждом уровне (БУС,БОМ,БАК) каждый элемент оборудования имеет свой набор регистров данных (температура, напряжение и т.п.).

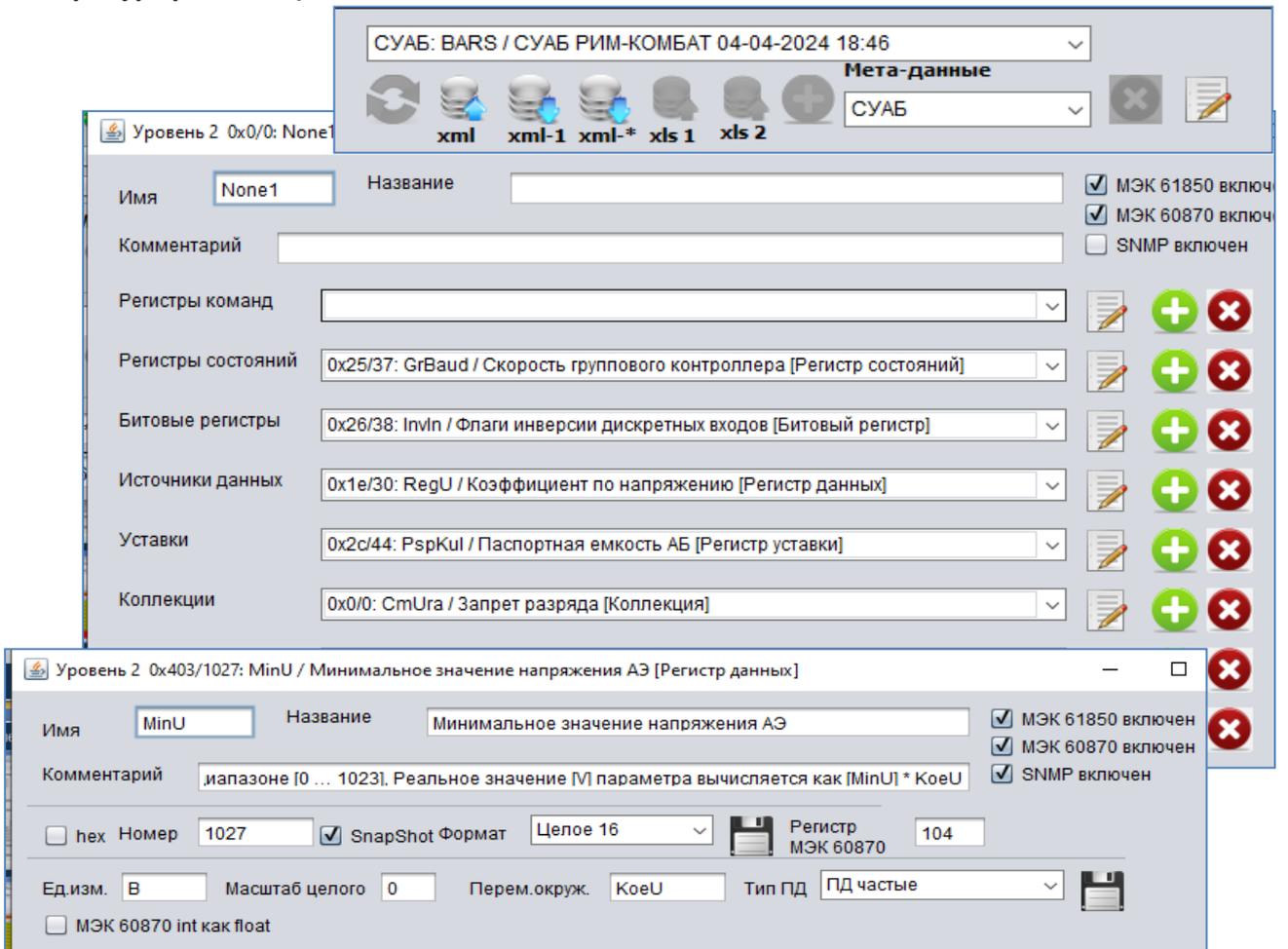
Метаданные верхнего уровня в БД

ESS2Arch	multiUnit	logUnits	metaFile	iec61850L	iec61850U	iec61850T	iec61850L	iec61850U	shortName	title	comment	oid	valid
0	0	0	0	0	0	0	0	0				23	0
2	1	0	62	0	0	0	0	0	BMS	BMS СУАБ 2.0		2	1
2	0	0	28	0	0	0	0	0	RTOS-PLC	ПЛК-RTOS		3	1
2	0	0	60	0	0	0	0	0	...	ACDC		4	1
2	0	0	39	0	0	0	0	0	A3	ДС А3 2.0		5	1
2	0	0	35	0	0	0	0	0	A5	ДС А5 2.0		7	1
2	0	0	33	0	0	0	0	0	A6	ДС А6 2.0		8	1
3	0	0	40	0	0	0	0	0	...	Чукотка-Тура 1.0		9	1
4	0	0	28	0	0	0	0	0	ПП-RTOS	ПП-RTOS		10	0
4	0	0	64	0	0	0	0	0	ArdSoft	Конфигурация тесте		11	1
5	1	0	101	95	96	97	99	ZBAT_1	BARS	BMS BARS		12	1
7	0	0	89	0	0	0	0	0	Thermo	IoT-Device		13	1
8	0	0	93	0	0	0	0	0	RIM-COMBAT	СУАБ РИМ-КОМБАТ		14	0
8	0	0	94	0	0	0	0	0	BMS-4.0-1	СУАБ РИМ-КОМБАТ		15	0
8	0	0	95	0	0	0	0	0	ACDC-4.0-1	ПП ДПК-РМД-РПД		16	1
8	0	0	103	0	0	0	0	0	PLC-4.0-1	МУ 4.0-1		17	1
8	0	0	96	0	0	0	0	0	PLC-4.0-1	Новое		18	0
8	0	0	96	0	0	0	0	0	PLC-4.0-1	Новое МУ 4.0	Интерфейс МУ	19	0
8	1	0	113	0	0	0	0	0	BMS-4.0.1	СУАБ СНЭЭ 10x12	Карта регистров	20	1
8	0	0	115	0	0	0	0	0	DI-4.0-1	Дискретные сигналы	Входные сигнал	21	1
5	0	0	120	0	0	0	0	ZBAT_1	DIB	DIB-Блок реле		22	1

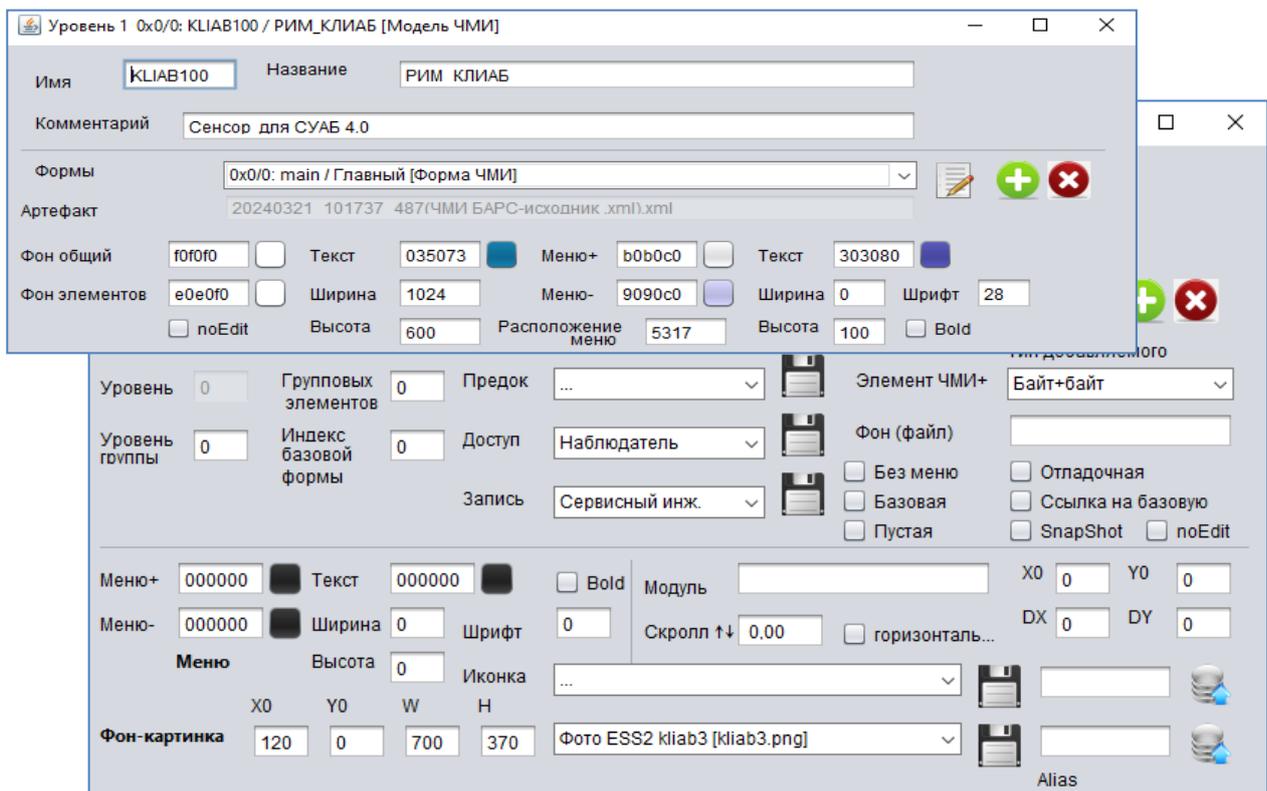
Метаданные. Внутреннее представление в XML-файлах

```
<Equipment shortName="BARS" title="СУАБ РИМ-КОМБАТ" comment="Карта регистров ГК GLS2503 V1" xmlType=
<registers shortName="None1" title="" comment="" out61850Model="false" in60870Model="true" inSNMP=
<list>
<Comment shortName="ShortName" title="" comment="" out61850Model="true" in60870Model="false" i
<SettingRegister shortName="PspKul" title="Паспортная емкость АБ" comment="задается как «Реаль
<SettingRegister shortName="PspI" title="Паспортный ток АБ" comment="задается как «Реальное зн
<SettingRegister shortName="KulVal" title="Остаточная емкость АБ" comment="выдается как число :
<String shortName="SoftVers" title="Версия программы" comment="" out61850Model="true" in60870M
<SettingRegister shortName="ConfReg" title="Система" comment=" мл байт -установка скорости обм
<String shortName="MyName" title="Имя устройства (запись)" comment="По записи (уставка)" out61
<StateRegister shortName="GrBaud" title="Скорость группового контроллера" out61850Model="true"
</states>
<GUIView shortName="KLIAB100" title="РИМ_КЛИАБ" comment="Сенсор для СУАБ 4.0" xmlType=
<forms>
<list>
<GUIForm shortName="main" title="Главный" comment="" level="0" parentName="" modu
<controls shortName="" title="" comment="" type="13" x="0" y="0" dx="0" h="0" f
<list>
<GUIData shortName="BatSI" title="Среднее значение напряжения АЭ" comment="
<regNum equipName="BARS" regNum="1026" ownUnit="true" unitIdx="0"/>
</GUIData>
<GUIData shortName="MinU" title="U АЭ (В) мин./ср./макс." comment="" type="
<regNum equipName="BARS" regNum="1027" ownUnit="true" unitIdx="1"/>
</GUIData>
<GUIData shortName="MaxU" title="Максимальное значение напряжения АЭ" comme
<regNum equipName="BARS" regNum="1028" ownUnit="true" unitIdx="0"/>
</GUIData>
<GUIData shortName="MidG" title="Среднее значение температуры АЭ" comment="
<regNum equipName="BARS" regNum="1029" ownUnit="true" unitIdx="0"/>
</GUIData>
<GUIData shortName="MinG" title="Минимальное значение температуры АЭ" comme
<regNum equipName="BARS" regNum="1030" ownUnit="true" unitIdx="0"/>
</GUIData>
<GUIData shortName="MaxG" title="Максимальное значение температуры АЭ " com
<regNum equipName="BARS" regNum="1031" ownUnit="true" unitIdx="0"/>
</GUIData>
<GUIData shortName="BatSI" title="Среднее значение напряжения АЭ" comment="
<regNum equipName="BARS" regNum="1026" ownUnit="true" unitIdx="1"/>
```

Метаданные описания оборудования. Редактирование (Разработка, Конфигурирование)



Метаданные описания ЧМИ. Редактирование (Разработка, Конфигурирование)



Редактирование ЧМИ «на лету» в процессе рендеринга (Разработка)

ЧМИ
РИМ_КЛИАБ

Эмуляторы
СУАБ БАРС СОМ

Переменные окружения
var=KoeU0 script=KoeU0

Скрипты клиент сервер все

Трасса

Вычисление констант KoeU0 / Переменная окружения

Игнорировать ошибки

Эмуляторы БАРС

Редакт. "на лету" Полный экран

Ориг. пропорции Без данных

МЭК 60870 МЭК 61850 SNMP

Пользователи Сервер Артефакты Настройки сервера Фильтр лога Настройки СНЭЭ Оператор Метаданные

Готовность СУАБ

Высокий ток разряда АКБ

Высокий ток заряда АКБ

Высокое напряжение АЭ

Низкое напряжение АЭ

Высокая температура АЭ

Низкая температура АЭ

Неисправность СУАБ

Запрет заряда

Блок питания 24в

220,0	Напряжение АКБ (В)	220,0
0	Ток АКБ (А)	0
1041,60	Ост. емкость (А*ч)	1041,60
67	SOC (%)	0
0,84 1,19 1,84	U АЭ (В) мин./ср./макс.	0,07 -0,00 -0,00
20,0 33,0 57,0	T АЭ (°C) мин./ср./макс.	10,0 31,0 52,0

08-04-2024 16:12:04

РТЭ

Имя Название

Комментарий

X	Y	W	H	Font	В строке	Элемент
142	210	0	0	0	0	a0a0ff

Оборуд./Unit Регистр hex Прозрач. Уровни: нижний, авария, предупред., верхний

BARS 0 1089 50 0,0 30,0 60,0 100,0

H image W image

130 130 Фото ESS2 bat4 [Bat.png]

Alias

Уровень и индекс группы в форме

Задать здесь уровень 1 Индекс 0

Без перехода

Надпись 000000

Общий noEdit

Общий

Отображать сразу

ПК- и киоск-приложения. Мониторинг состояния оборудования (Мониторинг)

Готовность СУАБ

Высокий ток разряда АКБ

Высокий ток заряда АКБ

Высокое напряжение АЭ

Низкое напряжение АЭ

Высокая температура АЭ

Низкая температура АЭ

Неисправность СУАБ

Запрет заряда

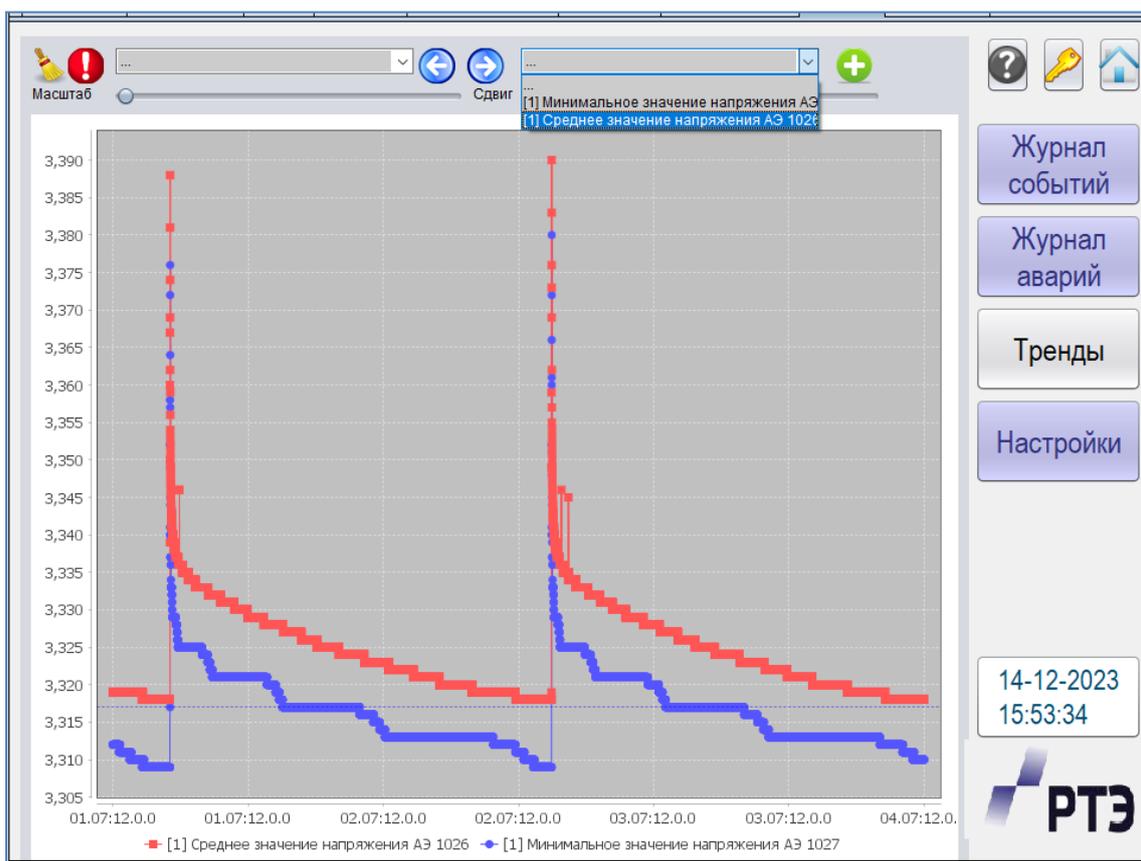
Блок питания 24в

220,0	Напряжение АКБ (В)	220,0
0	Ток АКБ (А)	0
1041,60	Ост. емкость (А*ч)	1041,60
62	SOC (%)	63
0,84 1,09 1,59	U АЭ (В) мин./ср./макс.	0,84 1,10 1,58
20,0 28,0 41,0	T АЭ (°C) мин./ср./макс.	19,0 27,0 40,0

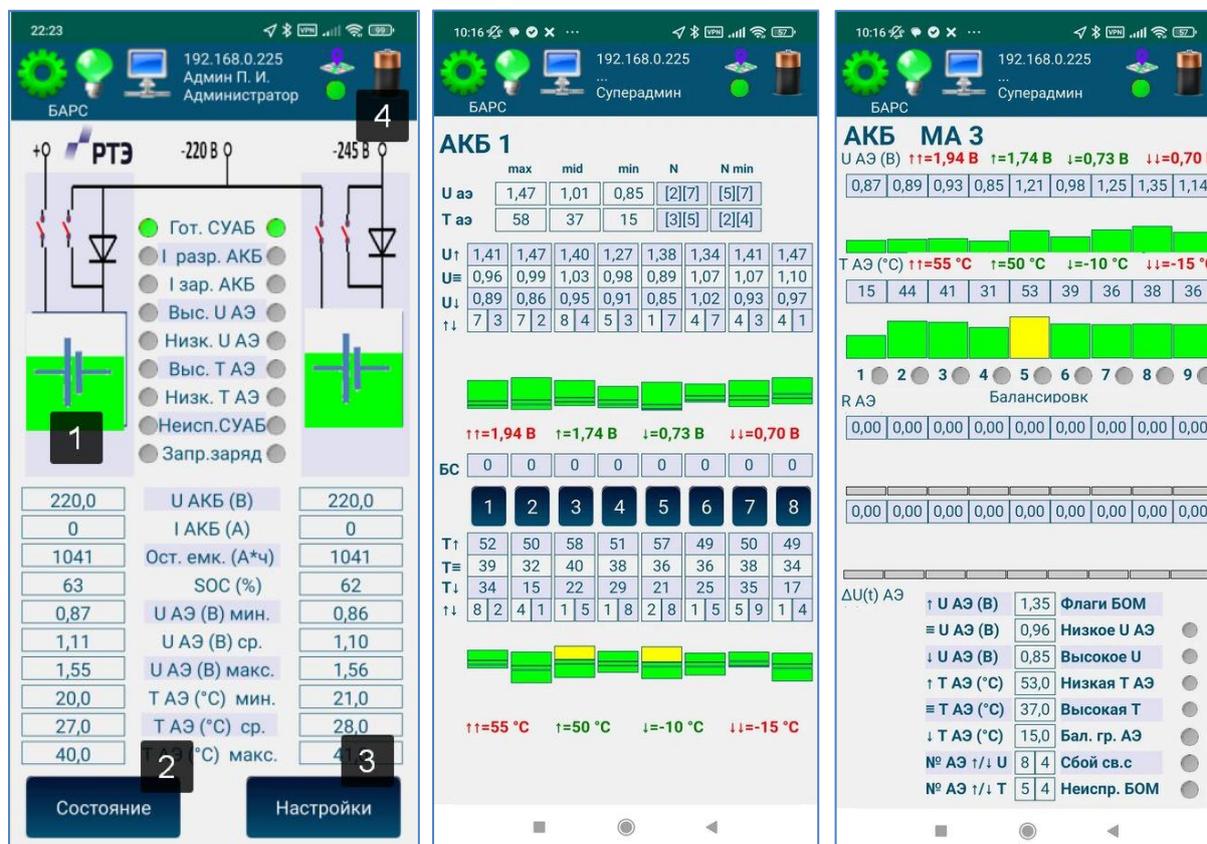
03-07-2024 10:35:31

РТЭ

ПК - и киоск-приложения. Просмотр трендов (Мониторинг)



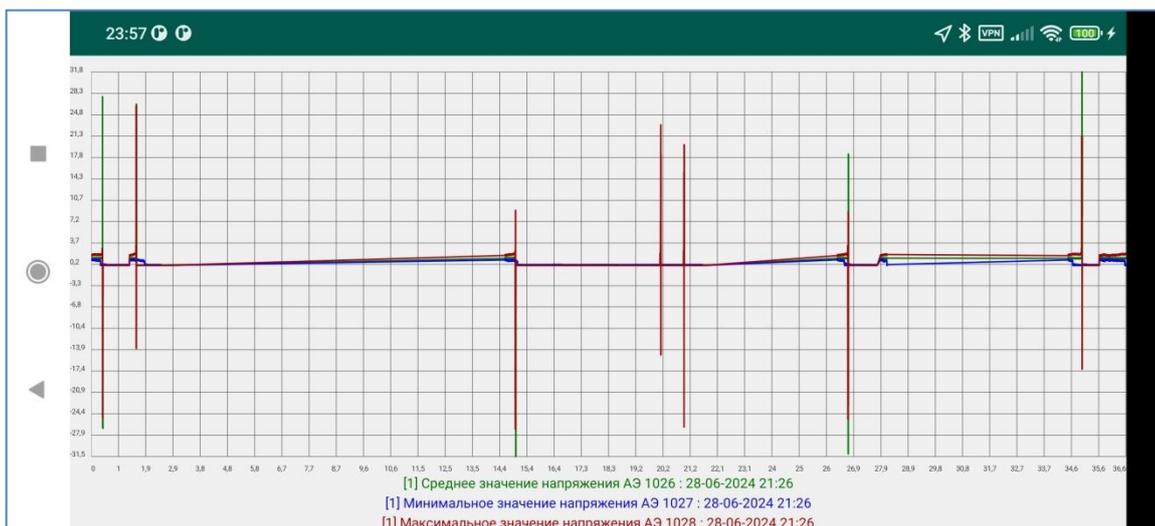
Android -приложение. Мониторинг состояния оборудования (Мониторинг)



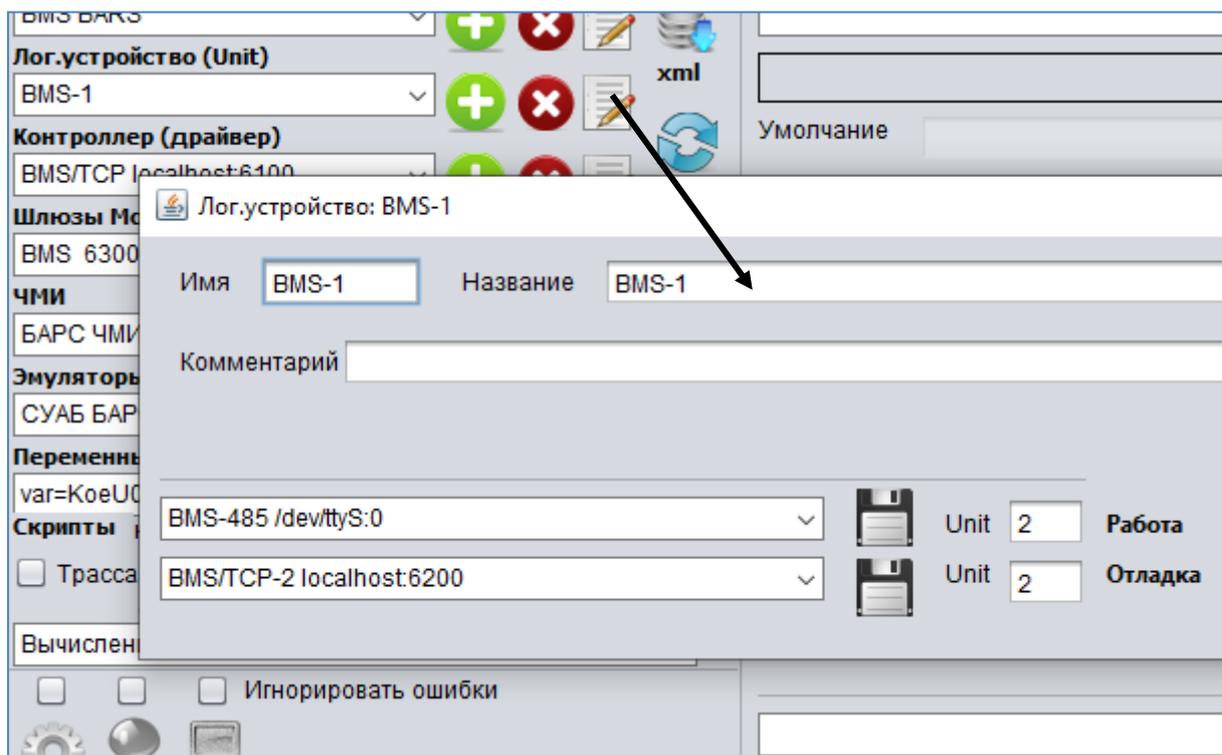
Android -приложение. Аварии и предупреждения (Мониторинг)



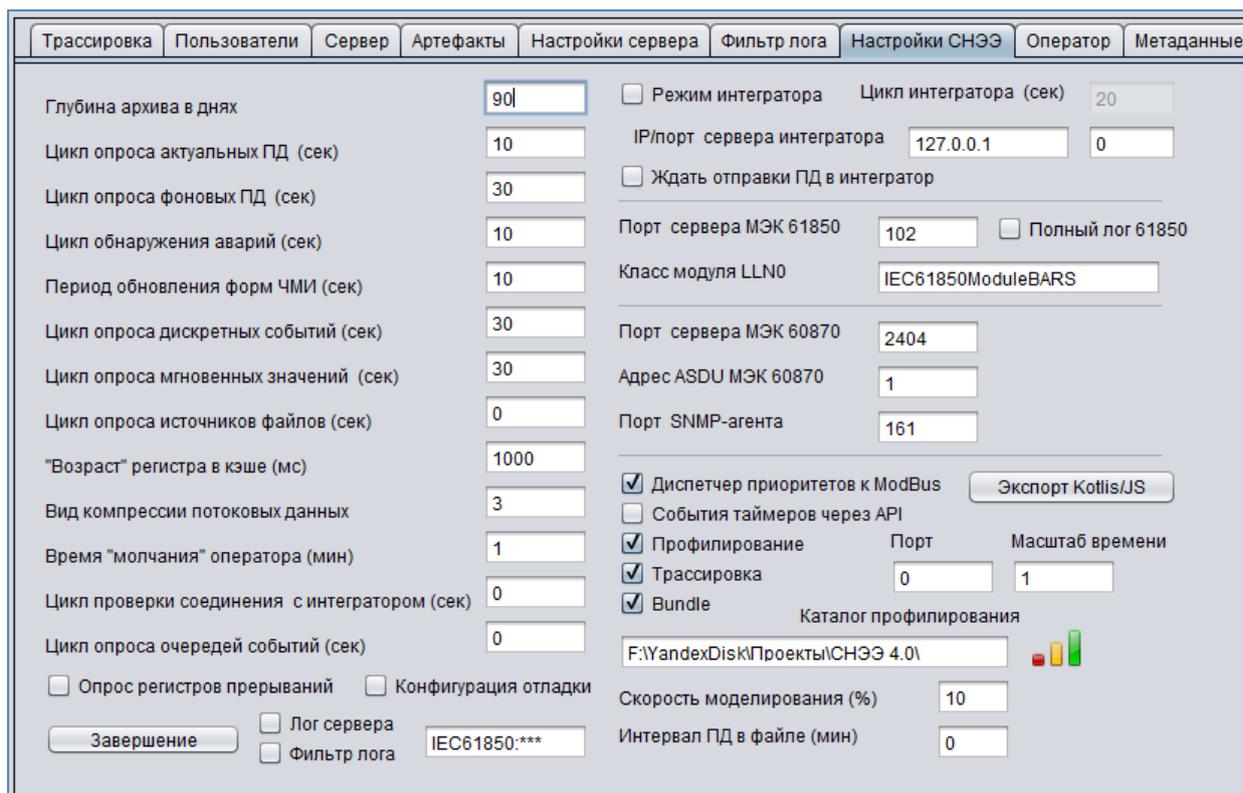
Android-приложение. Просмотр трендов (Мониторинг)



ПК -приложение. Настройка конфигурации (Конфигурирование)



ПК -приложение. Настройка параметров ПКУ (Конфигурирование)



Сторонний клиент МЭК 60870. Мониторинг параметров оборудования (Мониторинг)

The screenshot shows the IEC60870Master application. The main window contains a table with columns: IEC addr, Value, IEC timestamp, quan, N, Value, IEC timestamp, and ASDU type. The table lists data for various IEC addresses (1101-1120). Below the table is a log window showing ASDU received messages with details like CAA, nsI, nri, and ControlFields.I.

IEC addr	Value	IEC timestamp	quan	N	Value	IEC timestamp	ASDU type
1101	0,000	2024.06.30 13:18:09.778	3	3	29,000	2024.06.30 13:18:09.778	36
1102	0,000	2024.06.30 13:18:09.778	3	2	32,000	2024.06.30 13:17:59.707	36
1103	-0,001	2024.06.30 13:18:09.778	3	1	34,000	2024.06.30 13:17:55.906	36
1104	0,000	2024.06.30 13:18:09.778	3				
1105	-0,001	2024.06.30 13:18:09.778	3				
1106	29,000	2024.06.30 13:18:09.778	3				
1107	4,000	2024.06.30 13:18:09.778	3				
1108	54,000	2024.06.30 13:18:09.778	3				
1109	False	-	1				
1110	True	-	1				
1111	True	-	1				
1112	False	-	1				
1113	False	-	1				
1114	True	-	1				
1115	True	-	1				
1116	True	-	1				
1117	False	-	1				
1118	False	-	1				
1119	False	-	1				
1120	False	-	1				

Log window content:

```

13:18:09.992 IEC104M (127.0.0.1): ASDU received (CAA=1, nsI=81, nri=1, ControlFields.I=0x000200A2)
13:18:09.997 IEC104M (127.0.0.1): Information ASDU received (type 36)
13:18:10.001 IEC104M (127.0.0.1): sporadic, number of information objects: 1
13:18:10.005 IEC104M (127.0.0.1): Addr: 2106, Val: 31.00000, Qual: 192, Time: 30.06.2024 13:18:09.843
13:18:10.008 IEC104M (127.0.0.1): ASDU received (CAA=1, nsI=82, nri=1, ControlFields.I=0x000200A4)
13:18:10.012 IEC104M (127.0.0.1): Information ASDU received (type 36)
13:18:10.016 IEC104M (127.0.0.1): sporadic, number of information objects: 1
13:18:10.020 IEC104M (127.0.0.1): Addr: 2107, Val: 0.00000, Qual: 192, Time: 30.06.2024 13:18:09.843
13:18:10.024 IEC104M (127.0.0.1): ASDU received (CAA=1, nsI=83, nri=1, ControlFields.I=0x000200A6)
13:18:10.028 IEC104M (127.0.0.1): Information ASDU received (type 36)
13:18:10.031 IEC104M (127.0.0.1): sporadic, number of information objects: 1
13:18:10.035 IEC104M (127.0.0.1): Addr: 2108, Val: 54.00000, Qual: 192, Time: 30.06.2024 13:18:09.843
    
```

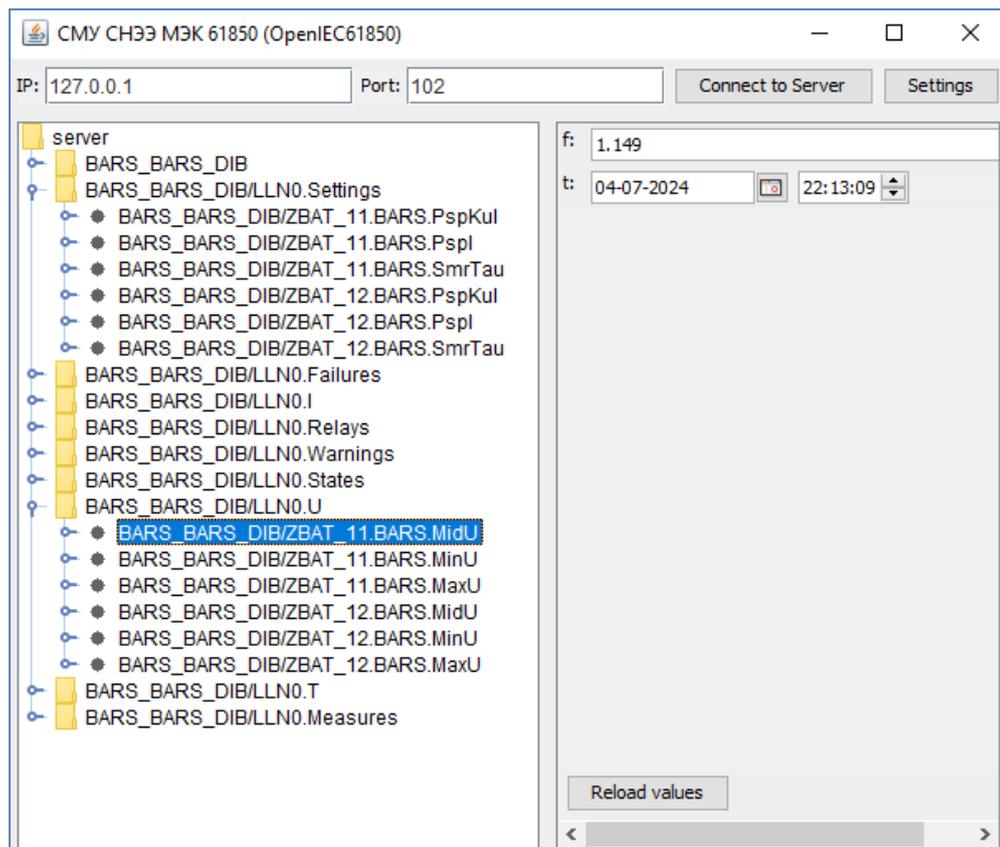
Control panel on the right includes: IEC addresses received: 50, IEC samples received: 76, Start button, Update data checkbox, Full log dropdown, IP: 127.0.0.1, Port: 2404, ASDU: 65535, GI period, sec: 30, Export life, Export full buttons.

Сторонний клиент МЭК 61850. Мониторинг параметров оборудования (Мониторинг)

The screenshot shows the IED Explorer 0.80a interface. The left pane shows a tree view of the IED structure, including BARS_BARS_DIB and various LLNO\$ settings. The right pane shows the IEDDataView table with columns: Name, Type, and Value.

Name	Type	Value
BARS_BARS_DIB/ZBAT_11.BARS.MidU	structure	
----- CHILD NODES -----		
BARS_BARS_DIB/ZBAT_11.BARS.MidU.f	floating_point	1,078
BARS_BARS_DIB/ZBAT_11.BARS.MidU.t	utc_time	04.07.2024 22:1...

Собственный клиент МЭК 61850. Мониторинг параметров оборудования (Мониторинг)



3. Метаданные описания БАРС СУАБ

Терминология раздела:

- **Метаданные описания конфигурации** – адаптация к различным конфигурациям оборудования и приложений реализуется при помощи мета-данных. В состав мета-данных входят *спецификации интерфейса оборудования, спецификации ЧМИ приложений*. Мета-данные редактируются в ПК-приложении администратора/сервисного инженера с использованием соответствующего набора форм (Wizard) и средств предпросмотра ЧМИ (Preview).
- **Спецификации архитектуры БАРС СУАБ** – списки видов и параметров оборудования (устройств) и видов ЧМИ. Хранятся в БД сервера данных ПУ БАРС СУАБ, при загрузке в серверные и клиентские приложения разворачиваются в структуры данных, с которыми работают программные компоненты.
- **Спецификации интерфейса оборудования** – описание структуры регистров оборудования, потоковых данных, аварийных и дискретных событий, а также скрипты (внешние функции) нестандартной обработки отдельных событий интерфейсов. Хранятся в виде xml-файлов, при загрузке в серверные и клиентские приложения разворачиваются в структуры данных, с которыми работают программные компоненты.
- **Спецификации ЧМИ приложения** – описание структуры экранных форм и составляющих их элементов, а также их отображения на элементы интерфейсов оборудования (регистры, разряды, команды и пр.). Хранятся в виде xml-файлов, при загрузке в серверные и клиентские приложения разворачиваются в структуры данных, с которыми работают программные компоненты.

Диаграмма классов метаданных БАРС СУАБ приведена на рис.1. Метаданные описания БАРС СУАБ имеют два уровня:

- метаданные верхнего уровня хранятся в БД сервера данных и содержат списки архитектурных компонент БАРС СУАБ и параметры настройки;
- метаданные нижнего уровня хранятся в виде xml-файлов и содержат описание интерфейсов оборудования - протокола Modbus/TCP и описание ЧМИ.

Ввиду того, что архитектурные классы являются основным элементом представления, описание метаданных будет проводиться *непосредственно в виде заголовков этих классов*. Элементы данных классов представляют собой архитектурные типы данных (целые, вещественные, строки) и ссылки (массивы ссылок) на вложенные (связанные) элементы метаданных. Форматы Excel-файлов будут описаны ввиду необходимости редактирования метаданных в этом представлении, а xml-формат будет цитироваться опционально.

Наследование архитектурных классов соответствует включению элементов данных базовой компоненты в любую производную.

Имена классов метаданных даются в оригинальном виде, используемом в программном коде. В таблицах 1,2,3 приводится соответствие этих имен сущностям предметной области (архитектура БАРС СУАБ, оборудование, ЧМИ).

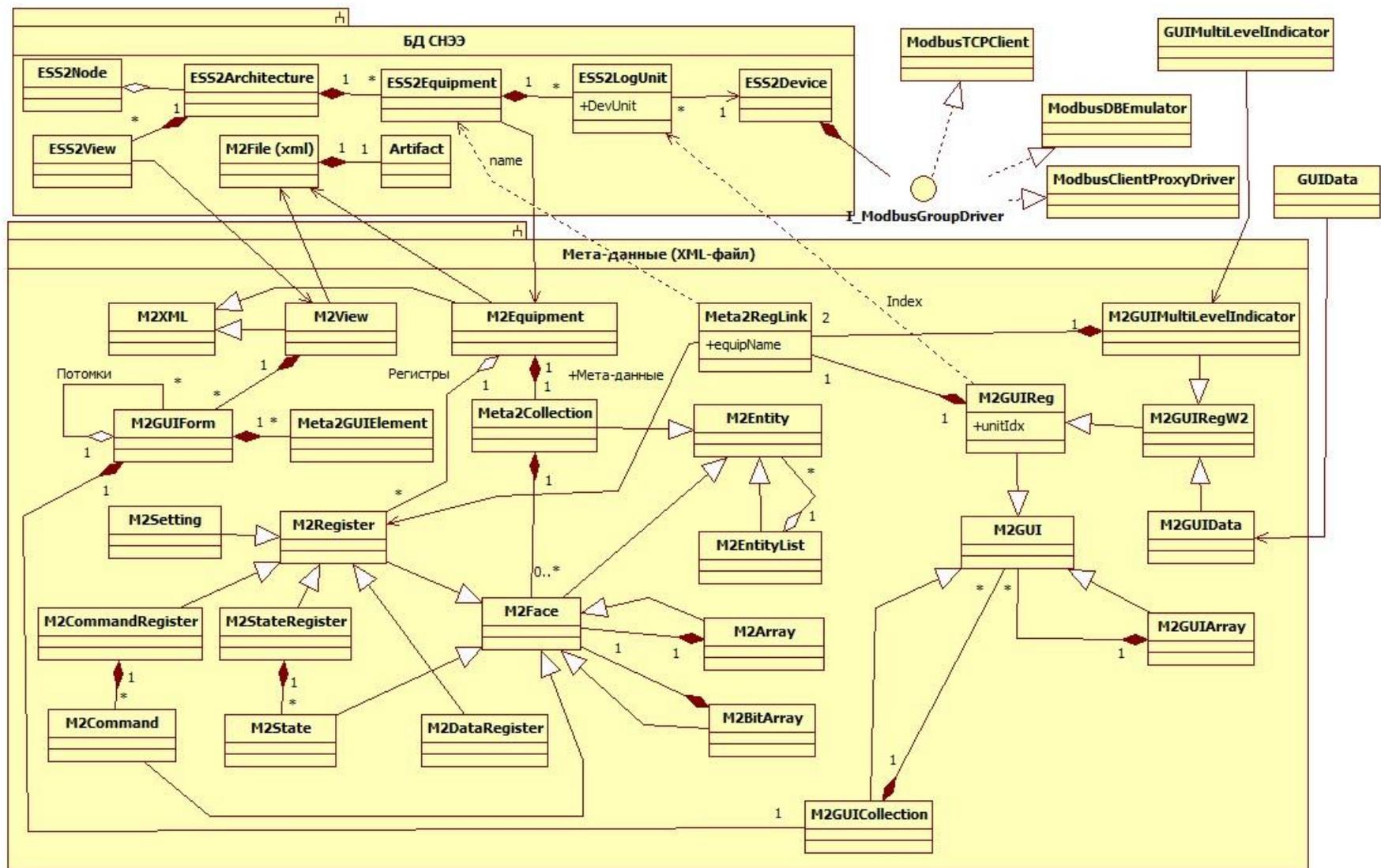


Рис.1. Метаданные ПКУ БАРС СУАБ

3.1. Метаданные верхнего уровня

Терминология раздела:

- **Оборудование (Equipment)** – логическая компонента, существующая как отдельная функциональная единица и имеющая специфицированный интерфейс управления. В данной версии – ModbusTCP.
- **Логический Unit оборудования (логическое устройство)** – каждой единице оборудования соответствует свой логический Unit в описании. Он содержит ссылку на устройство и назначенный ему Unit в этом устройстве.
- **Физическое устройство (Device)** – контроллер Modbus TCP, обеспечивающий подключение нескольких единиц оборудования. В конфигурации БАРС СУАБ может быть несколько однотипных или разнотипных единиц оборудования. Устройство идентифицируется уникальным внутренним IP-адресом/портом, единица оборудования - числовым идентификатором - **Unit единицы оборудования в устройстве (Device Unit)**;
- **Эмулятор оборудования** – программная компонента, воспроизводящее поведение оборудования с протоколом Modbus. Является серверной компонентой Modbus/TCP с соответствующим драйвером, поэтому может свободно конфигурироваться вместо реального оборудования.
- **Скрипт** — исходный текст программы, написанной на языке описания скриптов (ЯОС). Предназначены для описания последовательности действий с интерфейсами оборудования и ЧМИ. Скрипты используются для:
 - вычисления значений переменных окружения;
 - выполнения пусконаладочных работ;
 - обработки событий в интерфейсах оборудования;
 - обработки событий в ЧМИ.
- **Шлюз Modbus** – перенаправление внешнего доступа по протоколу Modbus/TCP на назначенное физическое устройство (контроллер).

Метаданные конфигурации верхнего уровня хранятся в БД и включают в себя:

- **узел** - данные идентификации и доступа в сети БАРС СУАБ;
- **архитектура** – набор конфигурационных данных описания оборудования и ЧМИ для заданной конфигурации оборудования;
- **оборудование** – описание вида оборудования (подсистемы БАРС СУАБ);
- **логическая единица оборудования (модуль)** – логическое устройство, отображаемое на физическое устройство (контроллер);
- **физическое устройство** – контроллер Modbus, имеющий оригинальные параметры подключения (IP/Port для TCP и имя/номер устройства для RS-485);
- **ЧМИ** - описание варианта ЧМИ для текущей архитектуры (файл-артефакт);
- **эмулятор** – описание подключенного эмулятора;
- **скрипт** – описание скрипта на ЯОП (файл-артефакт);
- **переменная окружения** – переменная, которая может использоваться в качестве масштабирующего коэффициента при получении данных от оборудования;
- **шлюз Modbus** – параметры шлюза для внешнего доступа к Modbus оборудования.

Таблица 1. Метаданные верхнего уровня

	Мета-сущность	Класс метаданных	Описание
1	Архитектура БАРС СУАБ	ESS2Architecture	Оборудование, ЧМИ, скрипты, эмуляторы БАРС СУАБ
2	Оборудование	ESS2Equipment	Интерфейс оборудования – связанные метаданные в Meta2Equipment
3	Логическая единица оборудования	ESS2LogUnit	Соответствие логических единиц оборудования контроллерам/Unit
4	Физическое устройство	ESS2Device	Контроллер ModBus
5	ЧМИ (внешний вид)	ESS2View	ЧМИ – связанные метаданные в Meta2View
6	Скрипт	ESS2ScriptFile	Описатель файла-артефакта, содержащего текст скрипта на ЯОП
7	Эмулятор оборудования	ESS2EquipEmulator	Описатель Java-класса – эмулятора оборудования и параметры адресации
8	Файл метаданных	ESS2MetaFile	Описатель xml-файла (артефакта), содержащего метаданные.
9	Файл (артефакт)	Artifact	Описание хранимого на СД файла
10	Переменная окружения	ESS2EnvValue	Масштабируемые коэффициенты для данных от оборудования
11	Шлюз Modbus	ESS2ModBusGate	Описатель внешнего подключения по Modbus к контроллеру

Метаданные верхнего уровня редактируются непосредственно в клиентском ПК-приложении в режиме администратора/сервисного инженера.

3.2. Метаданные нижнего уровня. Оборудование

Терминология раздела:

- **Регистр битовый состояний (битовый регистр)** – регистр, каждый разряд (бит) которого соответствует событию (предупреждение, авария, дискретное событие) или логическому состоянию оборудования.
- **Регистр битовый** – регистр, запись единичного бита в каждый разряд воспринимается оборудованием как отдельная команда.
- **Регистр флажков** – битовый регистр, разряды доступны по чтению/записи. Изменение отдельного разряда при записи должно сопровождаться записью прочитанных значений остальных разрядов.
- **Регистр важных событий оборудования (регистр прерываний)** – битовый регистр, разряды которых являются индикаторами событий (дублируются разрядами в других регистрах оборудования). Реализуются как *защелки*: сброс разряда (события) производится записью единичного значения в соответствующий разряд.
- **Регистр – источник данных** содержит текущее значение параметра оборудования. Может быть отмечен как источник потоковых данных.
- **Регистр команд** – регистр, в который записывается код команды оборудования. Коды команд регистра специфицированы в мета-данных.
- **Регистр состояния** – регистр, из которого читается код состояния оборудования. Коды состояний регистра специфицированы в мета-данных.

- **Регистр оборудования** – 16-разрядный регистр, 32-разрядные вещественные и целые передаются в двух подряд идущих регистрах. Интерфейс оборудования Modbus представляет собой поле регистров, в котором каждый регистр имеет свой номер.
- **Регистр уставки** – содержит значение параметра, устанавливаемого для оборудования. Доступен чтению и записи.

Метаданные конфигурации нижнего уровня хранятся в виде xml-файлов и представляют собой иерархические (древовидные) описания элементов интерфейсов или ЧМИ. Метаданные верхнего уровня (*оборудование, ЧМИ*) содержат ссылки на конфигурационные xml-файлы.

Метаданные в xml-файлах могут быть получены следующими способами:

- импорт Excel-файлов метаданных описания оборудования;
- графическое редактирование древовидной структуры метаданных в клиентском ПК-приложении в режиме администратора/сервисного инженера;
- загрузка сторонних xml-файлов с описанием метаданных.

Таблица 2. Метаданные элементов оборудования

	Мета-сущность	Класс метаданных	Описание
1	Базовый класс метаданных	Meta2Entity, Meta2Face	Базовый класс метаданных нижнего уровня
2	Комментарий	Meta2Comment	Комментарий в Excel-файле
3	Регистр Modbus	Meta2Register	Базовый класс регистра Modbus
4	Регистр данных	Meta2DataRegister	Регистр – источник данных
5	Уставка	Meta2SettingRegister	
6	Регистр команд	Meta2CommandRegister	
7	Команда	Meta2Command	Команда регистра команд
8	Регистр состояния	Meta2StateRegister	
9	Состояние	Meta2State	Состояние регистра состояний
10	Битовый регистр	Meta2BitRegister	
11	Разряд битового регистра	Meta2Bit	
12	Массив мета-элементов	Meta2Array	
13	Коллекция мета-элементов	Meta2Collection	
14	Массив битов	Meta2BitArray	Массив битовых регистров
15	Метаданные	Meta2XML	Базовый класс для объединения метаданных оборудования и ЧМИ
16	Оборудование	Meta2Equipment	
17	Шаблон	Meta2Template	Часть листа Excel-файла, используемая как шаблон
18	Экземпляр	Meta2Implement	Генерация метаданных по шаблону со

	шаблона		списком параметров
19	Строка	Meta2String	Массив регистров (по 2 символа)
20	Дата-время	Meta2DateTime	Группа регистров даты-времени (3 регистра)
21	Код сообщения	Meta2MessageId	Идентификатор сообщения. Сообщения в регистре передаются в потоке

Групповые регистры. Классы – контейнеры

В метаданных возможно определение групповых и вложенных компонент. Мета-класс коллекции Meta2Collection позволяет объединить в одну группу элементы метаданных любых типов (в т.ч. массивов и коллекций).

Класс метаданных Meta2Array – массив, определяет множество размерности size однотипных элементов метаданных (elem), с шагом адресации step в поле регистров. При использовании вложенных структур следует учитывать, что шаг step на каждом уровне учитывает размерности метаданных всех вложенных уровней. Возможен режим генерации описаний групповых элементов в МЭК 61850 не как элемента массива, а как группы независимых описаний элементов с именами xxx0,xxx1,xxx2 и т.д..

С помощью коллекций и массивов можно делать компактные описания иерархических структур регистров Modbus. Элементы ЧМИ имеют аналогичные средства (мета-классы). При рендеринге с помощью таких метаданных создаются групповые элементы ЧМИ и регулярным образом связываются с соответствующими групповыми элементами оборудования.

В качестве примера приведены структуры групповых регистров версии БАРС СУАБ-1, включающие 3 уровня оборудования (СУАБ – ГРК – АЭ, в терминологии этой версии).

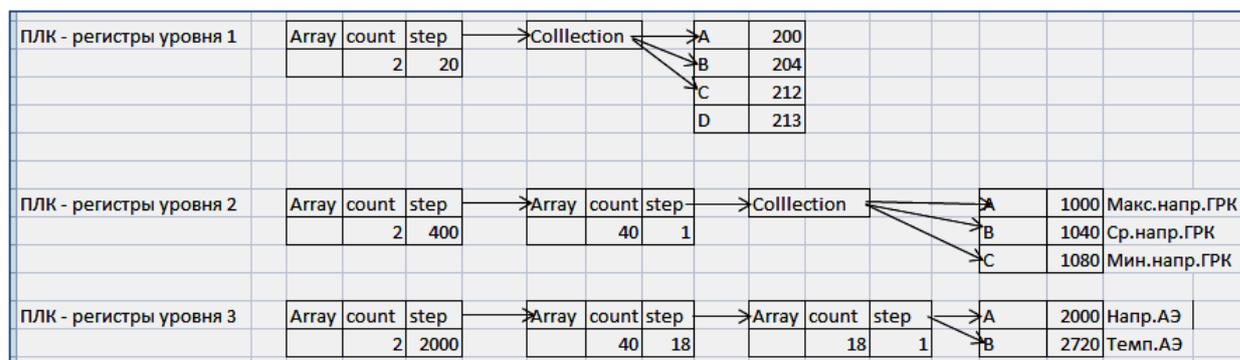


Рис.2. Структуры групповых регистров ПЛК для БАРС СУАБ-1

На рис.3 изображены варианты структуры регистров групповых элементов СУАБ, использованных в работах по БАРС СУАБ-2.

В первой группе регистров имеется 60 ГРК с шагом 80 (0x50) для коллекции регистров ScanFlgs, GrkSize, GrkName и 4 групп регистров по 24 (по одному на каждый АЭ) с шагом 1: CellVolt – 0x500-0x517, CellGrad – 0x518-0x52F, CellPwm – 0x530-0x547. Всего имеется 60 групп таких регистров, номера регистров указаны для первого ГРК, для каждого последующего они смещаются на 0x50 за счет массива верхнего уровня.

Вторая группа более простая. Для 60 ГРК с шагом группы 96 (0x60) имеется 4-ка регистров на каждый из 24 АЭ.

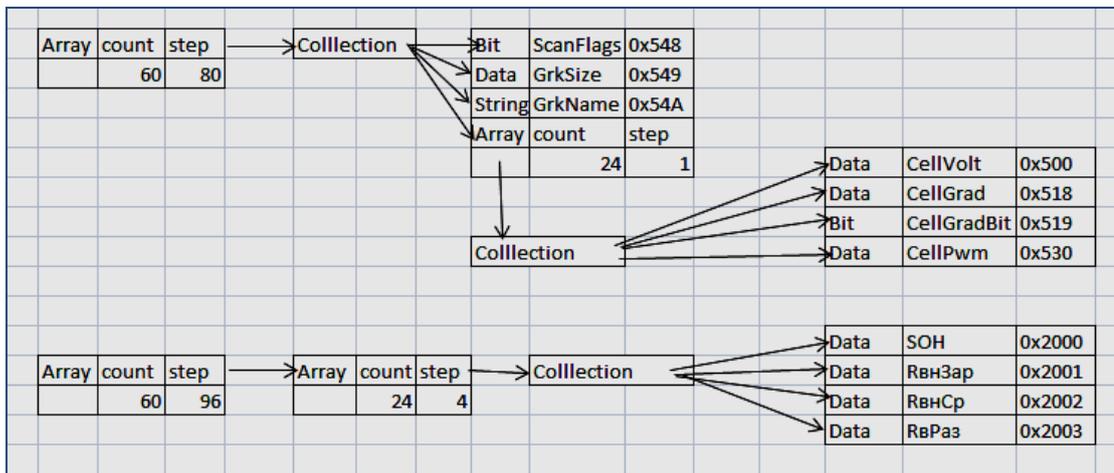


Рис.3. Структуры групповых регистров СУАБ для BARC СУАБ-2

На рис.4 изображена структура регистров групповых элементов СУАБ BARC. Поле регистров БУС включает массив регистров для 8 БОМ с шагом 112(10). Поле регистров БОМ содержит 12 уникальных регистров данных и массивы регистров по одному на каждый БАК. Все массивы описаны в виде одной коллекции.

Замечание: описание метаданных в виде непосредственно массива в массиве недопустимо, поэтому приходится прибегать к вспомогательной коллекции, которая формально объединяет по одному элементу из каждого массива с общим для всех индексом.

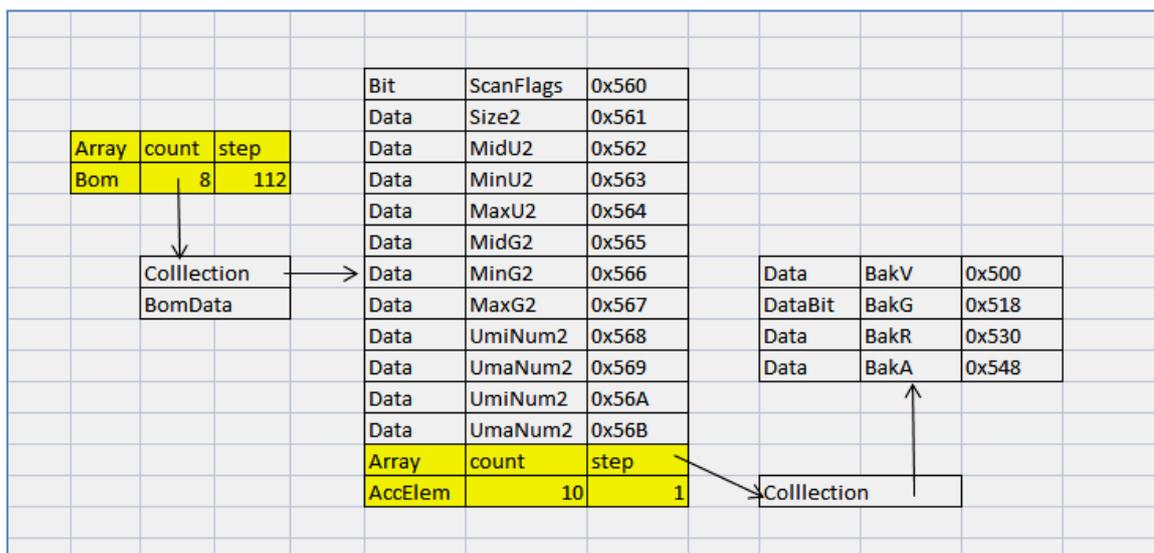


Рис.4. Структура групповых регистров СУАБ BARC для BARC СУАБ-4

3.3. Формат Excel-файлов метаданных описания оборудования

Формат Excel-файла метаданных оборудования является наиболее удобным для документирования интерфейсов с последующим редактированием и импортом (конвертацией) в xml-формат. Соглашения формата:

- с каждого листа производится импорт отдельного вида оборудования;
- первая строка листа содержит имена столбцов;
- фиксированными являются столбцы Title и Comment для всех элементов метаданных;

- поля Type и следующее за ним ShortName фиксируют соответствующие параметры - имя типа (**имя константы в группе MetaElem в классе констант**) и короткое имя;
- во второй строке столбец Type содержит имя из группы констант MetaType (например, MEEquipBMS) – тип оборудования. Следующий за ним столбец ShortName – имя оборудования;
- далее к этим же столбцам привязаны описания корневой коллекции мета-элементов, привязка столбцов – та же;
- оригинальные данные мета-элемента следуют в столбцах, расположенных справа от пары Type/ShortName. Порядок следования для каждого типа метаданных закреплен в коде импорта элементов и не использует имена столбцов;
- для вложенных элементов метаданных расположение полей Type/ShortName и всех последующих смещается на один вправо. К вложенным элементам относятся:
 - разряды в битовых регистрах;
 - команды в регистрах команд;
 - состояния в регистрах состояний;
 - элементы шаблонов;
 - элементы коллекций (кроме корневой);
 - элемент массива.
- элементы, имеющие переменное количество вложенных элементов, могут иметь закрывающийся тэг, совпадающий с именем типа мета-элемента, предваренный символом «/» (все перечисленные выше, кроме массивов);
- закрывающийся тэг может быть в той же строке, что и описание последнего вложенного элемента;
- в одной строке может быть не более одного закрывающегося тэга;
- свободные ячейки столбцов мета-элементов комментариев могут использоваться для именованых столбцов следующих за ними мета-элементов. Аналогичным образом можно использовать свободные ячейки справа от списка оригинальных данных мета-элемента.

Пример описания уставок приведен на рис.5.

MEComment	ShortName	RegNum	Unit	Power	Remote	FailType	DataReg	OverLim	DefValue	MinVal	MaxVal	format	Snapshot	Koeff
/MESetting	Shul	0x0181	A						500			1		
/MESetting	Hrel	0x0182	A						0			1		Koeff
/MESetting	Null	0x0184	A*4						4			1		Koeff
/MESetting	Ball	0x0185	A						80	1	100	1		Koeff
/MESetting	BakSet	0x0186							16	2	380	1		
/MESetting	BomSet	0x0187							2	1	63	1		
MEComment	ShortName	RegNum	Unit	Power	Remote	FailType	DataReg	OverLim	DefValue	MinVal	MaxVal	format	Snapshot	Koeff

Рис.5. Метаданные уставок в Excel-файле

На рис. 6 приведены примеры описания групповых регистров, структура которых изображена на рис.3.

MEArray	Grk	60	96		
	MEArray	AccElem	24	4	
		MECollection	GrkAccElemData		
			MEDataRegister	SOH	0x2000
			MEDataRegister	RвнЗар	0x2001
			MEDataRegister	RвнСр	0x2002
		/MECollection	MEDataRegister	RвнПаз	0x2003

В таблице 3 перечислены имена типов элементов оборудования, используемых в Excel-файлах.

Таблица 3. Имена типов мета-элементов

Код	Тип элемента	Имя константы
1	Регистр команд	MECommandRegister
2	Команда	MECommand
3	Регистр состояний	MEStateRegister
4	Состояние	MEState
5	Битовый регистр	MEBitRegister
6	Разряд регистра	MEBit
7	Регистр данных	MEDataRegister
8	Регистр уставки	MESetting
9	Оборудование	MEEquipment
10	Форма ЧМИ	MEGUIForm
11	Элемент формы	MEGUIElement
12	Коллекция	MICollection
13	Массив	MEArray
14	Битовый массив	MEBitArray
15	Модель ЧМИ (XML-файл)	MEGUIModel
16	Регистр-строка	MEString
17	Комментарий	MEComment
18	Шаблон	METemplate
19	Имплементация	MEImplement
20	XML-файл	MEXMLFile
21	Регистр даты-времени	MEDateTime

3.3.1. Метаданные – шаблоны импорта

При использовании повторяющихся групп мета-элементов с параметрами в Excel-формате можно использовать шаблоны. Сам шаблон представляет собой мета-элемент типа METemplate и имеет формат коллекции (рис.8.)

METemplate	Comparator		Unit	Power	Remote	FailType	DataReg	OverLim	DefValue	MinVal	MaxVal	Format	Snapshot	Koeff
	MESetting	LoUst	*0+0	*7					*2			1		*6
	MESetting	HiUst	*0+1	*7					*3			1		*6
	MESetting	LoTau	*0+2	сек	-1				*4			1		
	MESetting	HiTau	*0+3	сек	-1				*5			1		
	MEBitRegister	Flags	*0+4	5										
		EnFl	0											
		ZonFl	1											
		AbsFl	2											
		Bum1Fl	5											
		Bum2Fl	6											
		Bum3Fl	7											
		Bum4Fl	8											
		Bum5Fl	9											
		Bum6Fl	10											
		TrigFl	15											
	/MEBitRegister													
/METemplate														

Рис.8. Шаблон импорта

В ячейках шаблона можно использовать обозначения целочисленных параметров, подставляемых при создании экземпляра (имплементации шаблона). Параметр шаблона имеет вид:

- ***<индекс параметра>** - подстановка в ячейку значения параметра с заданным индексом;

- *<индекс параметра>+<целое> - подстановка в ячейку значения параметра с заданным индексом с добавлением значения целой константы.

Экземпляр шаблона создается и записывается в метаданные при импорте из Excel-файла мета-элементом MEImplement, имеющим следующую структуру:

- имя типа MEImplement;
- имя шаблона, используемого при создании экземпляра;
- имя экземпляра – коллекции;
- префикс имени, добавляемого к коротким именам основных регистров мета-элементов шаблона. Короткое имя регистра в экземпляре имеет формат <префикс>.<ShortName>. Это необходимо для исключения конфликта имеет в нескольких экземплярах шаблона;
- список параметров экземпляра. Параметру представляют собой целые значения, индексы нумеруются, начиная с 0 (рис.9)

MEImplement	MEComment		regNumBase	Mask	LoUst	HiUst	LoTau	HiTau	Koeff	Unit
MEImplement	Comparator	CmUra	0x00E8	0x087E7	687	725	50	50	KoeU	B
MEImplement	Comparator	CmUza	0x00EE	0x087E7	687	725	50	50	KoeU	B
MEImplement	Comparator	CmCld	0x00F4	0x087E7	30	35	100	300		град.С
MEImplement	Comparator	CmTen	0x00FA	0x087E7	5	10	100	300		град.С
MEImplement	Comparator	CmErr	0x0100	0x087E7	0	1	5	50		
MEImplement	Comparator	CmUb1	0x0106	0x07	50	100	150	150	KoeU	B
MEImplement	Comparator	CmUb2	0x010C	0x07	100	200	150	150	KoeU	B
MEImplement	Comparator	Cm11za	0x0112	0x07	1536	1843	10	10	KoeI	A
MEImplement	Comparator	Cm12za	0x0118	0x087E7	2150	2458	10	10	KoeI	A
MEImplement	Comparator	Cm11ra	0x011E	0x07	-2765	10	10	10	KoeI	A
MEImplement	Comparator	Cm12ra	0x0124	0x087E7	-3072	-2765	10	10	KoeI	A
MEImplement	Comparator	CmMix	0x012A	0x087E7	5	10	100	300		град.С
MEImplement	Comparator	CeU1mi	0x0130	0x07	725	737	50	50	KoeU	B
MEImplement	Comparator	CeU2mi	0x0136	0x087E7	700	736	50	50	KoeU	B
MEImplement	Comparator	CeU1ma	0x013C	0x07	936	950	50	50	KoeU	B
MEImplement	Comparator	CeU2ma	0x0142	0x087E7	936	950	50	50	KoeU	B
MEImplement	Comparator	CeG1mi	0x0148	0x07	-10	-5	150	150		град.С
MEImplement	Comparator	CeG2mi	0x014E	0x087E7	-15	-10	150	150		град.С
MEImplement	Comparator	CeG1ma	0x0154	0x07	50	55	150	150		град.С
MEImplement	Comparator	CeG2ma	0x015A	0x087E7	55	60	150	150		град.С
MEImplement	Comparator	CeUb0	0x0160	0x07	862	865	50	50	KoeU	B
MEImplement	Comparator	CmFun	0x0166	0x07	30	35	150	150		град.С
MEImplement	Comparator	CmSle	0x016C	0x07	10	15	12	50	KoeI	A

Рис.9. Создание экземпляров шаблона импорта

После импорта экземпляры шаблона присутствуют в метаданных как независимые коллекции.

3.4. Метаданные описания оборудования. XML-формат

Файлы метаданных описания оборудования и ЧМИ представляют собой текстовые файлы с сериализацией в XML-формате древовидной структуры мета-объектов. Они хранятся на сервере данных как все прочие файлы-артефакты, при редактировании средствами ПК-приложения перезаписываются. Возможна их выгрузка/загрузка и стороннее ручное редактирование.

```
<Equipment shortName="BARS" title="СУАБ БАРС" comment="Карта регистров ГК GLS2503 V1" xmlTy
  <registers shortName="None1" title="" comment="" DOType="" out61850Model="false">
    <list>
      <Comment shortName="ShortName" title="" comment="" DOType="" out61850Model="false"/>
      <SettingRegister shortName="PspKul" title="Паспортная емкость АБ" comment="задается к
        <envVarValue/>
      </SettingRegister>
      <SettingRegister shortName="PspI" title="Паспортный ток АБ" comment="задается как «Ре
        <envVarValue/>
      </SettingRegister>
      <SettingRegister shortName="KulVal" title="Остаточная емкость АБ" comment="выдается к
        <envVarValue/>
      </SettingRegister>
      <String shortName="SoftVers" title="Версия программы" comment="" DOType="" out61850Mo
        <envVarValue/>
      </String>
      <SettingRegister shortName="ConfReg" title="Система" comment="мл байт -установка ско
        <envVarValue/>
      </SettingRegister>
      <String shortName="MyName" title="Имя устройства (запись)" comment="По записи (установк
        <envVarValue/>
      </String>
      <SettingRegister shortName="GrBaud" title="Скорость группового контроллера" comment=""
        <envVarValue/>
      </SettingRegister>
      <BitRegister shortName="InvIn" title="Флаги инверсии дискретных входов" comment="Уста
        <envVarValue/>
        <bits>
          <list>
            <Bit shortName="InvDI1" title="[0/1] - прямой/инверсный дискретный вход DI1 (п
            <Bit shortName="InvDI2" title="[0/1] - прямой/инверсный дискретный вход DI2 (п
          </list>
        </bits>
      </BitRegister>
```

3.5. Метаданные описания ЧМИ

Терминология раздела:

- **Разметка** – описание ЧМИ (форм и элементов управления) в метаданных, термины, относящиеся к разметке:
 - **рендеринг** – прорисовка и переходы между формами в соответствии с разметкой;
 - **мониторинг** – отображение текущего состояния компонент оборудования в элементах ЧМИ;
 - **графический runtime-редактор разметки** – совмещенное с рендерингом редактирование мета-данных ЧМИ.
- **Модули нестандартных форм ЧМИ** – программная компонента со специфицированным интерфейсом, выполняющий действия в форме, которые не могут быть описаны средствами мета-уровня ЧМИ (тренды, аварийные события). Добавление нового модуля в клиентское приложение требует его пересборки и обновления.

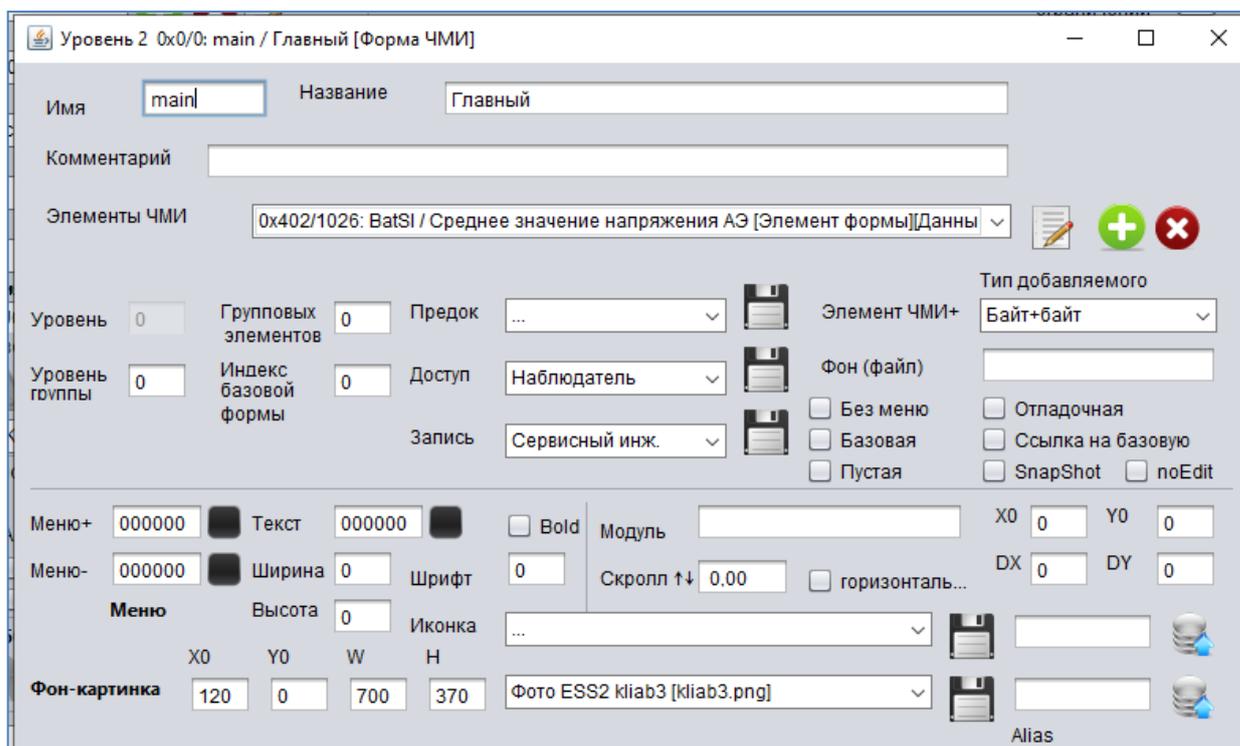
Элементы метаданных ЧМИ аналогично организованы в древовидную структуру. Класс Meta2GUIForm содержит описание экранной формы. Формы организованы в 4-уровневое дерево, соответствующее основному меню.

Формы имеют 4-уровневую древовидную структуру, в соответствии с которой изображается до 4 рядов кнопок меню на панели оператора. level – номер уровня формы, parent – имя (title) формы – предка. Главная форма имеет уровень 0. При выводе формы уровня N выводятся кнопки меню уровней N+1, т.е. от 1 до следующего за текущим.

Для панели оператора установлен отдельный регламент по уровням доступа. Имеются следующие уровни:

- 0 – уровень доступа разработчика, доступны любые операции, игнорируется запрет удаленных;
- 1 - уровень доступа сервисного инженера;
- 2 - уровень доступа инженера;
- 3 - уровень доступа оператора;
- 4 - уровень доступа наблюдателя.

Установлено соответствие ролей авторизации и уровней доступа. Роли разработчика, сервисного инженера, инженера и оператора имеют соответствующие уровни доступа. *Суперадминистратор* имеет уровень доступа разработчика. *Администратор и остальные роли (Гость)* имеют уровень доступа наблюдателя.



В каждой форме задан допустимый уровень по входу (просмотру) – accessLevel и по изменению (выполнение команд и запись уставок) – writeLevel. Права имеют пользователи с установленным уровнем доступа и выше.

Если в форме установлен признак empty, то форма не имеет собственных элементов ЧМИ. При ее открытии происходит автоматический переход к первому непустому потомку. Соответствующий пункт меню является *группирующим*.

Имеется возможность подсветки кнопки меню, соответствующей форме, в зависимости от значения разряда битового регистра (кнопка Сервисный режим). Для

этого используются поля regNum, regBit, color. При отсутствии заполнения опция не работает.

Признак noMenu устанавливается для форм, содержащих групповые элементы, в которых позиционирование осуществляется с помощью элементов управления Meta2GUIFormSelector. При его установке кнопки меню этой формы и ее потомков не выводятся.

С каждой формой может быть связан программный модуль, выполняющий нестандартные или уникальные действия над формой. Его имя в поле Module является именем Java-класса проекта, который обладает специализированным интерфейсом. Соответствие этому интерфейсу проверяется *при открытии формы* (но не при импорте метаданных). Оригинальными модулями реализованы формы визуализации трендов и журналов аварий и событий.

Поле backgroundImage содержит имя файла с фоновым рисунком формы. Файл должен быть размещен в папке resources проекта сервера данных (т.е. предварительно включен в проект перед его компиляцией и сборкой).

Формы, в которых выводится содержимое групповых регистров в регулярном виде (СУАБ-ГРК-АЭ), имеют свои управляющие элементы для переходов между уровнями и выбора начального номера визуализируемого элемента (Meta2GUIFormSelector). Они должны иметь дополнительные параметры в метаданных:

- formLevel – номер уровня групповых данных формы (1-3), вычисляется автоматически по дереву форм;
- elementsCount – количество элементов групповых данных формы – дублирует (количество отображаемых элементов задается в описании конкретного элемента управления).

Типы элементов управления и соответствующие им классы перечислены в таблице 5.

Таблица 5. Элементы управления ЧМИ

Тип	Название	Внутреннее имя (класс)
0	Надпись	Meta2GUILabel
1	Состояние по группе разрядов	Meta2GUIMultiBitState
2	Регистр - источник данных	Meta2GUIData
3	Уставка	Meta2GUISetting
4	Пустой	
5	Команда	Meta2GUIButton
6	Индикатор: Разряд + кнопка изменения	Meta2GUI2StateBox
7	Индикатор: 2 разряда, 3 состояния	Meta2GUI3StateBox
8	Индикатор - 2 разряда, 3 состояния (малый)	Meta2GUI3StateBoxSmall
9	Регистр состояний	Meta2GUIStateSet
10	Индикатор – время	Meta2GUITimeSetting
11	Кнопка: Прямой переход на форму	Meta2GUIFormButton
12	Выбор формы для групповых элементов	Meta2GUIFormSelector
13	Контейнер – коллекция	Meta2GUICollection
14	Контейнер - массив	Meta2GUIArray
15	Индикатор уровня	Meta2GUILevelIndicator
16	Мультииндикатор уровня	Meta2GUILevelMultiIndicator
17	Индикатор – разряд, 2 состояния (малый)	Meta2GUI2StateBoxSmall
18	Скрипт	Meta2GUIScript

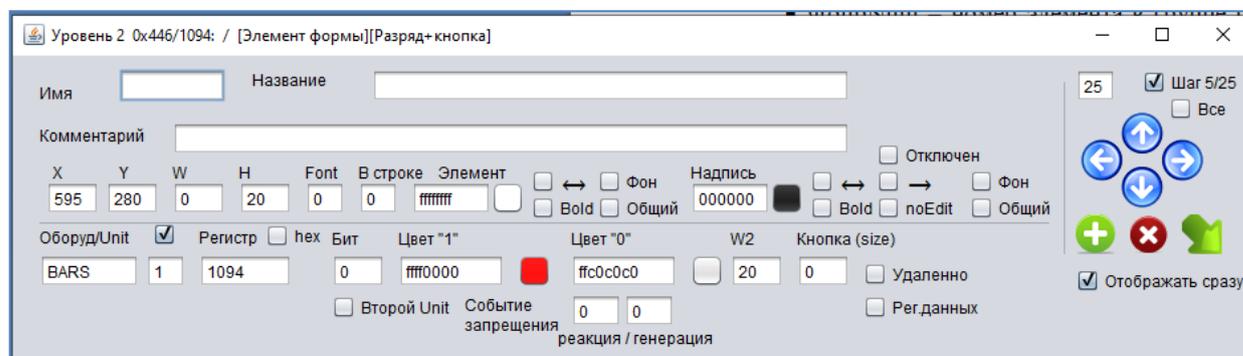
19	Надпись с форматированным значением	Meta2GUIDataLabel
20	Параметр настройки сервера – целое	Meta2GUIESSSettingInt
21	Параметр настройки сервера – строка	Meta2GUIESSSettingString
22	Параметр настройки сервера – логическое	Meta2GUIESSSettingBoolean
23	Переменная окружения	Meta2GUIEnvVar
24	Разряд битового регистра + кнопка команд	Meta2GUIBitStateCmd
25	Надпись-скрипт	Meta2GUIScriptLabel
26	Картинка	Meta2GUIImage
27	Картинка – индикатор: разряд + 2 картинки	Meta2GUIImageBit
28	Картинка-уровень (скрипт)	Meta2GUIImageScriptLevel
29	Регистр даты-времени	Meta2GUIDateTime
30	Текст (регистр-строка)	Meta2GUIString
31	Разряд-команда битового регистра	Meta2GUIBitStateCmd
32	Индикатор: Разряд + кнопка изменения	Meta2GUIBit2Commands (6)
33	Регистр данных – вывод побайтно	Meta2GUITwoBytes
34	Индикатор уровня ч/б	Meta2GUILevelWB
35	Разряд- флажок (checkbox)	Meta2GUIBitCheckBox
36	Картинка-уровень (регистр данных) - переход к форме	Meta2GUIImageDataLevel
37	Надпись – индекс группового элемента	Meta2GUIIndexLabel
38	Текущее время	Meta2GUICurentDateTime
39	Уставка – список «название-значение»	Meta2GUIStateSelector
40	Индикатор – результат скрипта	Meta2GUIBitScript
41	Битовый регистр – список установленных разрядов	Meta2GUIBitsList
42	Картинка-бит-2 команды	Meta2 GUIImageBitCmd
43	Разряд+2 команды+маска	Meta2GUIBitStateCmdEnb
44	Таймер-интервал	Meta2GUITimeInterval

Многие элементы управления имеют одинаковые параметры. Наиболее часто используемые приведены в таблице 6.

Таблица 6. Поля описания свойств элементов управления

1	regNum	Целое	Номер регистра
2	bitNum	Целое	Номер бита
3	shortName	Строка	Короткое имя
4	type	Целое	Тип элемента управления
5	title	Строка	Надпись
6	comment	Строка	Комментарий
6	X	Целое	Координата надписи
7	Y	Целое	Координата надписи
8	W	Целое	Размер надписи (ширина)
9	W2	Целое	Ширина самого элемента (по умолчанию – различное для типов)
10	H	Целое	Общая высота элемента и надписи
11	stringSize	Целое	Количество символов в строке многострочной надписи
12	power	Целое	Масштаб форматированного вывода целого
13	color	Цвет	Явно установленный цвет фона элемента
14	commonColor	Выбор	Элемент -выбран общий цвет элементов для ЧМИ
15	backColor	Выбор	Элемент - выбран цвет фона для ЧМИ
16	onCenter	Выбор	Элемент – выравнивание по центру

17	bold	Выбор	Элемент – жирный шрифт
18	labelColor	Цвет	Надпись – установленный явный цвет
19	labelCommonColor	Выбор	Надпись - выбран общий цвет надписей для ЧМИ
20	labelBold	Выбор	Надпись – жирный шрифт
21	labelOnCenter	Выбор	Надпись – выравнивание по центру
22	labelOnRight	Выбор	Надпись – выравнивание по правому краю
23	labelBackColor	Выбор	Надпись - выбран цвет фона для ЧМИ
24	noEditThere	Выбор	Запрет изменения в текущей форме (уставка)
25	colorYes	Цвет	Цвет логического индикатора для true
26	colorNo	Цвет	Цвет логического индикатора для false
27	font	Целое	Размер шрифта
28	count	Целое	Счетчик групповых элементов первого уровня
29	step	Целое	шаг для групповых элементов (>0 – по вертикали, <0 – по горизонтали)
30	level	Целое	номер уровня групповых элементов для открываемой формы
31	click	Строка	Имя формы для перехода по клику (11- GUIFormButton), по переходу на след. уровень для групповых элементов (12 - GUIFormSelector)
32	groupNum	Целое	Номер группы для нерегулярных групповых элементов
33	regNum2	Целое	База дополнительной группы регистров для мультииндикатора (тип 16), номер 2-го битового регистра для состояния по группе разрядов (тип 1)
34	regNum3	Целое	База дополнительной группы регистров для мультииндикатора (тип 16 - GUILevelMultiIndicator), номер 3-го битового регистра для состояния по группе разрядов (тип 1)
35	bitMask	Маска	маска 3-го битового регистра для состояния по группе разрядов (тип 1)



Особенности некоторых параметров.

- power - масштаб вывода (дублирует данные регистра источника и уставки);
- color - цвет надписи в формате FFRRGGBB, R,G,B – 16-цифры компоненты цвета 0..F;
- colorYes - цвет флажка для true;
- colorNo - цвет флажка для false;
- size - количество символов в строке при многострочной надписи (0 - однострочная), для GUI2StateBox (тип 2) – размер индикатора;
- count:int - счетчик групповых элементов первого уровня;
- step:int=0 - шаг для групповых элементов (>0 – по вертикали, <0 – по горизонтали);

- level – номер уровня групповых элементов для открываемой формы для Meta2GUIFormSelector;
- click – имя открываемой формы для Meta2GUIFormButton, Meta2GUIFormSelector;
- groupNum – номер элемента в группе первого уровня (АКБ) для разрядов битовых регистров, если разные разряды одного регистра используются для событий разных элементов (например, 370[15]-авария АКБ1, 371[2]-авария АКБ2, или 370[5],370[7] (вариации разрядов в зависимости от номера элемента первого уровня).
- приоритеты цветов фона элементов и их надписей:
 - общий цвет фона для ЧМИ;
 - общий цвет для элементов ЧМИ;
 - явно заданный цвет, код цвета не равен 0.

Замечание: Цвет текста для большинства элементов одинаковый и задается в целом для варианта разметки ЧМИ. Для элементов, состоящих только из надписи, параметры цвета элемента (явный, общий, фон) относятся к **тексту**, а не к фону. Общий цвет кнопок соответствует цвету кнопок меню в неактивном (не нажатом) состоянии.

Большинство элементов ЧМИ имеют строку надписи, при W=0 она отсутствует.

Для описания ссылок на регистры оборудования используется класс Meta2RegLink, содержащий имя оборудования, номер регистра и индикатор формирования номера логического устройства (Unit) для регистра - ownUnit. Если он взведен, то номер непосредственно задается полем unitIdx. При сброшенном ownUnit номер выбирается из стека индексов для уровня, соответствующего уровню формы (т.е. определяется выбором, произведенным в родительской форме ЧМИ).

Большинство элементов управления, связанных с регистрами, наследуются от классов метаданных Meta2GUIReg и Meta2GUIRegW2

Признак intValue устанавливает форматирование значения при выводе как целого числа. Признак byteSize используется в особых случаях, когда размерность данных регистра составляет один байт.

В таблице 7 перечислены дополнительные параметры элементов управления (базовый набор: желтый - Meta2GUIReg, зеленый - Meta2GUIRegW2)

Таблица 7. Элементы управления ЧМИ

	Название	Параметры
1	Надпись	
2	Состояние по группе разрядов	bitNum,colorYes,colorNo
3	Регистр - источник данных	
4	Уставка	Meta2GUISetting
5	Команда	cmdCode
6	Цветовой индикатор - 2 состояния	bitNum,colorYes,colorNo
7	Цветовой индикатор - 3 состояния	bitNum,colorYes,colorNo
8	Цветовой индикатор - 3 состояния (малый)	bitNum,colorYes,colorNo
9	Регистр состояний	
10	Индикатор – время	
11	Прямой переход на форму	formName
12	Выбор формы для групповых элементов	formName,formLevel
13	Индикатор уровня	maxFail,maxWarn,minFail,minWarn
14	Мультииндикатор уровня	maxFail,maxWarn,minFail,minWarn,regNum2,regNum3

15	Цветовой индикатор - 2 состояния (малый)	bitNum,colorYes,colorNo
16	Надпись с форматированным значением	colorYes,colorNo
17	Параметр настройки сервера – целое	fieldname,editDisable,colorYes,colorNo
18	Параметр настройки сервера – строка	fieldname,editDisable,colorYes,colorNo
19	Параметр настройки сервера – логическое	fieldname,editDisable,colorYes,colorNo

Индикаторы и мультииндикаторы уровня имеют дополнительные регистровые ссылки Meta2RegLink на уставки – уровни и дополнительные источники данных.

Для отображения групповых элементов и группировки функционально однородных используются классы описания элементов ЧМИ - коллекция и массив. Сами они не порождают элементы управления ЧМИ, но определяют процессы генерации регулярных элементов вложенных описаний.

Класс Meta2GUIArray описывает порождение в цикле элемента управления elem в количестве size и со смещением dxу относительно описания элемента elem. Значение dxу<0 соответствует смещению по оси x с приращением -dxу, значение dxу>0 - 0 соответствует смещению по оси y с приращением dxу.

Класс Meta2GUICollection описывает порождение элементов управления коллекции. Используется для группировки элементов в массивах.

3.6. Метаданные описания ЧМИ. XML-формат

Файлы метаданных описания оборудования и ЧМИ представляют собой текстовые файлы с сериализацией в XML-формате древовидной структуры мета-объектов. Они хранятся на сервере данных как все прочие файлы-артефакты, при редактировании средствами ПК-приложения перезаписываются. Возможна их выгрузка/загрузка и стороннее ручное редактирование.

```
<GUIView shortName="BARS" title="СУАБ 3.0 ЧМИ" comment="Desktop для СУАБ 3.0" xmlType="2" menuButtonOf
<forms>
  <list>
    <GUIForm shortName="main" title="Главный" comment="" level="0" parentName="" moduleName="" empty
      <controls shortName="" title="" comment="" type="13" x="0" y="0" dx="0" h="0" color="0" common
        <list>
          <GUILabel shortName="" title="АКБ 1" comment="" type="0" x="325" y="125" dx="75" h="0" col
          <GUILabel shortName="" title="Версия программы" comment="" type="0" x="65" y="150" dx="265
          <GUIString shortName="" title="Версия программы" comment="" type="2" x="330" y="155" dx="0
            <regNum equipName="BARS" regNum="2" ownUnit="true" unitIdx="0"/>
          </GUIString>
          <GUIData shortName="BatU" title="Напряжение АКБ" comment="" type="2" x="60" y="480" dx="27
            <regNum equipName="BARS" regNum="1036" ownUnit="true" unitIdx="0"/>
          </GUIData>
          <GUIData shortName="BatRI" title="Среднеквадратичное значение тока АБ" comment="" type="2"
            <regNum equipName="BARS" regNum="1024" ownUnit="true" unitIdx="0"/>
          </GUIData>
          <GUIData shortName="BatSI" title="Средневыпрямленное значение тока АБ" comment="" type="2"
            <regNum equipName="BARS" regNum="1025" ownUnit="true" unitIdx="0"/>
          </GUIData>
          <GUIData shortName="BatSI" title="Среднее значение напряжения АЭ" comment="" type="2" x="6
            <regNum equipName="BARS" regNum="1026" ownUnit="true" unitIdx="0"/>
          </GUIData>
          <GUIData shortName="MinU" title="Минимальное значение напряжения АЭ" comment="" type="2" x
            <regNum equipName="BARS" regNum="1027" ownUnit="true" unitIdx="0"/>
          </GUIData>
          <GUIData shortName="MaxU" title="Максимальное значение напряжения АЭ" comment="" type="2"
            <regNum equipName="BARS" regNum="1028" ownUnit="true" unitIdx="0"/>
          </GUIData>
          <GUIData shortName="MidG" title="Среднее значение температуры АЭ" comment="" type="2" x="6
            <regNum equipName="BARS" regNum="1029" ownUnit="true" unitIdx="0"/>
          </GUIData>
          <GUIData shortName="MinG" title="Минимальное значение температуры АЭ" comment="" type="2"
            <regNum equipName="BARS" regNum="1030" ownUnit="true" unitIdx="0"/>
```

4. Эмуляторы оборудования

Эмулятор СУАБ БАРС (ESS2BARSEmulator) разработан под текущую раскладку регистров.

В эмулятор «защиты» параметры уставок и определены объекты с функциями компараторов.

Определен массив для поля регистров Modbus. Все результаты процесса эмуляции записываются туда по соответствующим адресам. Моделируются следующие процессы:

- цикл увеличения/уменьшения значений напряжения и температуры каждой БАК со случайной составляющей;
- определение максимальных/минимальных/средних значений регистров напряжения и температуры для каждого БОМ и БУС в целом;
- определение в линейке номеров БАК с экстремальным значением напряжения/температуры;
- установка разрядов аварий/предупреждений в битовых регистрах Modbus по результатам срабатывания компараторов;
- периодическая установка случайных разрядов регистра блинкеров (0x446) – эмуляция прерываний (для тестирования);
- установка регистра текущей даты/времени и изменение содержимого регистров строки – версии программы (для тестирования);

5. Форматы описания, хранения и алгоритмы съема событий и потоковых данных

Терминология раздела:

- **Периодические операции** – действия, выполняемые сервером данных БАРС СУАБ для отслеживания событий, съема, сохранения и передачи потоковых данных, проверки состояния коммуникаций и т.д. Каждая операция имеет свой задаваемый в настройках период.
- **Событие** – изменение состояния оборудования, состояния сервера данных, выполнение операций обслуживающим персоналом с администраторами сервера данных. Фиксируются в журналах аварий/событий и/или в виде уведомлений.
- **Прерывания** – установка единичных значений в разрядах-защелках битовых регистров прерываний, для которых сервер данных реализует приоритетный опрос, обнаружении, сброс и формирование событий.
- **Потоковые данные (ПД)** - содержимое регистров – источников данных, список регистров содержится в мета-данных. Периоды задаются в настройках СМУ СНЭ. Потоковые данных сохраняются в БД в виде архивированных записей. Виды ПД:
 - **актуальные** - сохраняются с периодичностью порядка 10 сек;
 - **фоновые** - сохраняются с периодичностью порядка 1 мин;
 - **суточные** - сохраняются с периодичностью 1 сутки.
- **Мгновенный снимок (snapshot)** – подмножество потоковых данных, которое сохраняется и обновляется БАРС СУАБ, передается в интегратор и используется для мониторинга состояния оборудования сторонними клиентами.

5.1. Периодические операции, события и потоковые данные

Первая часть операций связана с периодическим опросом регистров оборудования для съема ПД и мгновенного снимка, получения событий об авариях/предупреждениях, важных событиях (прерываниях). В метаданных описания оборудования имеются поля и

признаки, определяющие регистры и разряды, используемые в этих целях. При развертывании конфигурации производится просмотр дерева метаданных и составление списков регистров для периодической проверки. Виды операции и метаданные для составления списков:

- съем и сохранение актуальных ПД – в метаданных регистров-источников устанавливается соответствующий тип ПД;
- съем и сохранение фоновых ПД – в метаданных регистров-источников устанавливается соответствующий тип ПД;
- съем и сохранение суточных ПД – в метаданных регистров-источников устанавливается соответствующий тип ПД;
- съем данных мгновенного снимка – в метаданных регистра имеется признак сохранения в мгновенном снимке;
- фиксация аварий/предупреждений и дискретных событий – в метаданных битовых регистров и их разрядов устанавливаются соответствующие типы;
- фиксация аварий по превышению уставок – определяется типом аварии, установленном для регистра уставки. Событие аварии определяется как превышение считанного значения связанного с уставкой регистра данных;
- фиксация важных событий (прерываний) – устанавливается соответствующий тип для битового регистра, все определенные в нем разряды являются источниками прерываний.

Вторая часть операций связана с проверкой состояния коммуникаций и передачей данных.

- опрос каталога – источника артефактов;
- передача накопленных записей ПД в интегратор;
- проверка состояния соединения с интегратором и keepAlive;
- периодические операции для сервера данных интегратора.

5.2. Алгоритмы съема потоковых данных и прерываний

БАРС СУАБ обеспечивает опрос оборудования для получения данных трех видов:

- Важных событий оборудования (прерываний);
- Актуальных потоковых данных;
- Фоновых потоковых данных.

При анализе изменения состояния регистров актуальных потоковых данных фиксируются дискретные события и аварии (изменение разрядов битовых регистров и выход значений регистров данных за уровни уставок), на основании которых добавляются записи в соответствующие журналы.

В БАРС СУАБ реализован приоритетный опрос состояния и съем данных оборудования с относительными приоритетами. Запросы на чтение данных размещаются в очередях, соответствующих четырем приоритетам.

БАРС СУАБ производит опрос регистров – источников данных и битовых регистров оборудования для следующих типов данных:

- прерывания (важные события) – разряды регистров прерываний (*приоритет 0*);
- актуальные потоковые данные (*приоритет 1*);
- мгновенные снимки (*приоритет 1*);
- фоновые потоковые данные (*приоритет 3*);
- суточные потоковые данные (*приоритет 3*);

- данные регистров, отображаемых в ЧМИ (при выполнении функций API клиентов, *приоритет 2*);
- запись регистров оборудования (при выполнении функций API, *приоритет 3*);
- дискретные события (разряды битовых регистров, *приоритет 3*);
- аварии (изменение разрядов битовых регистров и выход значений регистров данных за уровни уставок, (*приоритет 3*);

Используется приоритетный опрос с относительными приоритетами – запросы на чтение потоковых данных размещаются в трех очередях, соответствующих приоритетам:

- Запросы разных приоритетов не прерывают друг друга, обслуживание следующего начинается после завершения обслуживания текущего;
- Запрос на получение важных событий от устройства выполняется одним циклом чтения регистров ModBus;
- Запросы на чтение актуальных и фоновых потоковых данных разбиваются на запросы чтения групп регистров ограниченного размера (80 регистров, 256 байт) – реализовано в БАРС СУАБ 2.0;
- Цикл чтения актуальных или фоновых потоковых данных задерживается до окончания чтения данных предыдущего цикла.

Таким образом, установленный интервал чтения актуальных и потоковых данных является рекомендуемым, но не обязательным.

Замечание: БАРС СУАБ не производит сбор данных в режиме реального времени. Состояния регистров в потоковых данных и мгновенных снимках определяются временем опроса, которое может варьироваться по всему интервалу цикла опроса.

Замечание: скорость передачи, установленная для RS-485 – 115200 бит/с, соответствует предельной скорости передачи потока данных 10Кб/с, а с учетом накладных расходов протокола Modbus может быть существенно ниже. Определяющим факторами являются:

- Количество и способ группирования регистров, с которых снимаются фоновые потоковые данные;
- Количество единиц оборудования, сконфигурированных в БАРС СУАБ;
- Загрузка процессора потоком передачи/приема (или исполняющего его ядра), не оказывающая влияния на производительность других программных компонент. Фактор загрузки оказывается существенным ввиду того, что прием/передача потока байтов в RS-485 реализуется драйвером в режиме программного прерывания (в отличие от блочной передачи в IP-протоколе или USB).

БАРС СУАБ хранит потоковые данные в БД в течение установленного в настройках периода. Также может быть установлена опция обязательной передачи их в интегратор, при ее установке период хранения продлевается до момента их отправки.

5.3. Прерывания (Важные события оборудования)

В оборудовании могут быть реализованы битовые регистры **важных событий оборудования (регистры прерываний)**, разряды которых являются индикаторами событий (дублируются разрядами в других регистрах оборудования). Свойства:

- являются защелками: взводятся при появлении события, сбрасываются *при записи 1* в соответствующий разряд регистра от Master. *Замечание:* тем самым исключается потеря события при потере ответного кадра Modbus. Разряд сбрасывается, когда его обнаружит ПО на прикладном уровне. Однако это

может привести к потере однотипных событий при их последовательном появлении с малым интервалом;

- дублируют разряды аварий и дискретных событий в обычных битовых регистрах (в метаданных имеются ссылки на них);
- важные события в зависимости от их типа создают записи в журналах дискретных событий или аварий;

5.4. Форматы хранения, алгоритмы сжатия и передачи потоковых данных

Потоковые данные (ПД) снимаются, упаковываются и хранятся следующим образом:

- одномоментно снятые ПД хранятся в виде одной записи в формате текстовой строки (ArchStreamDataSet);
- каждый регистр занимает 1 или 2 16-разрядных слова в (2-4 байта) в зависимости от *финального* формата с учетом масштабирования значения (умножения на коэффициент или показатель степени $10 - power$).
- в конфигурации оборудования на основе сканирования метаданных определяются параметры смещения для всех сохраняемых регистров: для каждого логического устройства каждого вида оборудования создается описатель StreamRegisterGroup, он содержит описатель каждого сохраняемого регистра StreamRegisterData (ссылка на описатель регистра Meta2Register, смещение (для групповых) и смещение в записи ArchStreamDataSet. Один и тот же регистр в метаданных описания устройства порождает уникальные описатели для каждого логического устройства и для каждого регистра в группе (рис.13);

При съеме ПД опрос и сохранение данных от регистров производится в том же порядке. Полученная двоичная запись сжимается одним из способов (см. ниже) и преобразуется в текстовую строку в кодировке Base64.

Для построения тренда из БД извлекается необходимый набор записей, из каждой извлекается значение регистра по общему для всех смещению, соответствующему этому регистру в записях.

6. Язык описания скриптов

Язык описания скриптов (ЯОС) — простой интерпретируемый язык Си-подобным синтаксисом и Бейсик-подобной семантикой. Имеет встроенные функции для работы с интерфейсами оборудования и метаданными БАРС СУАБ.

Архитектура языка

Язык реализуется интерпретатором с компиляцией исходного текста скрипта в промежуточный код с оригинальной системой команд. Система команд работает со стеком операндов (аналогично JVM). При запуске скрипта производится его компиляция с выводом сообщений об ошибках. При отсутствии ошибок запускается интерпретатор полученного промежуточного кода. Любая ошибка исполнения (RunTime) приводит к остановке скрипта.

Скрипты могут быть оформлены в виде функций. В этом случае они могут использоваться в качестве обработчиков событий интерфейсов оборудования БАРС СУАБ и ЧМИ (**TODO**). Автономные скрипты, не имеющие заголовка функций, запускаются вручную.

Типы данных, переменные и константы

В ЯОС используются следующие типы данных:

- пустой тип **void**;
- логический **boolean** и константы **true,false**;
- строковый **string** и константы вида **"y="**. Для записи кавычек внутри константы используется их дублирование, например **"Это строка ""в кавычках""!!!!"**;
- целые типы **short, int, long**, десятичные константы вида **1232**, шестнадцатеричные константы вида **AA56!**;
- вещественные типы **float,double** и константы вида **125.77**.

Определение переменных представляет собой список имен с возможными инициализаторами, перечисленными через запятую и ограниченными символом «;», например **double x=2.5,y=x/2;**

string ss="a-a-a-a";

Использование неопределенных переменных недопустимо.

Операции, выражения, приведение типов

Группы операций (в порядке убывания приоритета):

- вызов функции, скобки приоритета ();
- поразрядные «не (1), и (2),или (3),исключающее или (3)» - **!,&, | , ^** (приоритеты в группе по убыванию 1-2-3);
- арифметические - ****** (возведение в степень,1),*****, **/** (2),**+, -** (3), (приоритеты в группе по убыванию 1-2-3);
- сравнение - **<, <=, >, >=, <>** (не равно), **=** (равно);
- логические «не(1),и(2),или(3),исключающее или(3)»- **not,and,or,xor** (приоритеты в группе по убыванию 1-2-3);

Правила преобразования типов и выполнения операций:

- логические операции применяются только к логическим типам. Логический тип имеют результаты операций сравнения, объявленный результат встроеной функции и логические константы;
- арифметические операции применяются только к числовым типам (целым и вещественным). При наличии в операциях целых и вещественных операндов производится приведение к вещественному типу;
- поразрядные операции применимы только к целочисленным типам;
- к операндам типа **string** применима операция «+» - конкатенация строк;
- при выполнении арифметических операций, в которых один из операндов имеет тип **string**, а другой — числовой, производится преобразование числового операнда во внешнюю форму (строку со значением операнда) с последующей конкатенацией с операндом **string**;

Операторы

Любое выражение, ограниченное символом «;», становится простым оператором. Отсутствие символа «;» в операторе присваивания и выражении является синтаксической ошибкой (стандартный ограничитель выражения).

Виды операторов:

- простой оператор — выражение, ограниченное символом «;»;
- оператор присваивания вида **<переменная>=<выражение>;** . При выполнении присваивания производится приведение типа результата

выражения к типу переменной. Если выражение является строкой, а переменная имеет числовой тип, то выполняется операция преобразования содержимого строки (число во внешней форме) к значению типа переменной. Например, допустимо такое присваивание:

```
int d="1256"+"7";
```

- условный оператор вида **if(<логическое выражение> <оператор>else<оператор>**. Ветка **else** не является обязательной;
- оператор цикла **while(<логическое выражение>)<оператор**
- составной оператор или блок - **{<последовательность операторов>}**
- пустой оператор «;»;

Встроенные функции

Имеется две группы встроенных функций: универсальные и функции предметной области (области применения). Функции имеют типизированные параметры и результат. При вызове функции проверяется возможность приведения типа фактического параметру к типу формального, приведение при вызове.

Таблица 8. Функции предметной области.

	Прототип	Действие
1	short readReg(string devName, long regNum)	Чтение регистра с номером regNum с устройства devName (<имя оборудования>_<индекс контроллера>)
2	void writeReg(string devName, long regNum, long regValue)	Чтение в регистр с номером regNum значения regValue

Таблица 9. Универсальные функции

	Прототип	Действие
1	double sin(double x)	стандартная функция sin
2	double cos(double x)	стандартная функция cos
3	void alert(string title, double v)	вывод в лог значения v с подсказкой title

7. *Согласование со стандартом МЭК 61850*

МЭК 61850 определяет стандарт описания интерфейсов оборудования в энергетике. Язык описания конфигурации оборудования (ГОСТ МЭК 61850-6-2009 «Сети и системы связи на подстанциях. Часть 6. Язык описания конфигурации для связи между интеллектуальными электронными устройствами на электрических подстанциях») представляет собой мета-уровень описания структуры XML-файлов. Для различных классов оборудования разрабатывается рекомендуемая конфигурация (**профиль оборудования**), включающая иерархию компонент, перечни параметров (атрибутов), зарезервированные имена, типы и характеристики форматов данных и пр. Аналогичные профили создаются производителями оборудования.

Работа с оборудованием, поддерживающим стандарт МЭК 61850, обеспечивается следующим образом:

- В соответствии со стандартом создается файл описания устройства (CID-файл);

- Программная компонента оборудования выступает в качестве сервера. Сервер реализуется с помощью общедоступной библиотеки, включенной в состав сервера данных БАРС СУАБ;
- При включении поддержки МЭК 81850 сервер данных БАРС СУАБ разворачивает серверную компоненту МЭК 81850 и передает ей CID-файл (или его содержимое в виде текстового потока, объекта-строки). По переданному описанию сервер МЭК 61850 создает *древовидную систему объектов*. В стандарте имеются соглашения по соответствию узлов модели и *составных имен параметров*;
- Сервер данных БАРС СУАБ периодически загружает текущие значения параметров оборудования в объекты сервера МЭК 61850;
- Клиентские приложения при подключении к серверу получают от него CID-файл конфигурации оборудования;
- Взаимодействие клиента с сервером производится посредством MMS-протокола обмена данными промышленного оборудования с использованием соглашений о *составных именах параметров*.

Для работы с оборудованием возможны два варианта создания клиентских приложений:

- *Специализированное*: создается под конкретный *профиль оборудования*. При работе с сервером клиент получает CID-файл о сервера, верифицирует его структуру. Внешний вид клиента соответствует структуре интерфейса оборудования. Приложение предназначено для конечных пользователей (техников, операторов, сервисных инженеров);
- *Универсальное*: представляет собой средство просмотра структуры и содержимого произвольного дерева объектов, соответствующего CID-файлу, полученному от сервера. Предназначено для разработчиков, наладчиков оборудования. Имеются открытые библиотеки для создания таких приложений и общедоступные приложения.

Ввиду отсутствия *профиля оборудования* для БАРС СУАБ и его компонентов (СУАБ, преобразователя) в текущей версии сервера данных БАРС СУАБ реализована следующая схема:

- При развертывании сервера МЭК 61850 выбирается тип оборудования (в командной строке – индекс из списка, в конфигураторе – выбранный в списке);
- Для дерева метаданных описания оборудования выполняется рекурсивная процедура, которая генерирует соответствующее ему описание оборудования в формате CID-файла. Таким образом, логическая структура описания оборудования совпадает со структурой описания, принятой в метаданных БАРС СУАБ;
- При конфигурировании можно отказаться от включения любой части описания метаданных в CID-файл;
- Полученное содержимое CID-файла передается серверу МЭК 61850, на его основе сервер строит собственную модель структуры и параметров в соответствии с требованиями стандарта МЭК 61850;
- Обновление данных в модели сервера МЭК 61850 производится только для следующего списка опрашиваемых сервером данных значений регистров следующих групп:
 - Битовые регистры прерываний – логические переменные МЭК 61850, соответствующие именованным разрядам регистров;
 - Битовые регистры аварий/предупреждений – логические переменные МЭК 61850, соответствующие именованным разрядам регистров;

- Потокковые данные – регистры данных с высокой частотой опроса («частые» потокковые данные);
- Потокковые данные – регистры данных с низкой частотой опроса («редкие» потокковые данные);
- При построении списков опрашиваемых регистров генерируются соответствующие им *составные имена параметров* для модели МЭК 61850;
- В процессе периодического опроса регистров (потокковых данных, прерываний, аварий/предупреждений) для полученных из регистров данных производится парсинг *составных имен параметров* в модели МЭК 61850 и поиск необходимого узла, в который записываются полученные данные;
- В дальнейших процессах предоставления данных клиентским приложениям сервер данных БАРС СУАБ не участвует.

8. Согласование со стандартом МЭК 60870

Процедуры передачи данных по протоколу МЭК 60870 связаны с остальными процессами опроса и сохранения потокковых данных, фиксации аварий и предупреждений и других процессов, которые используют обращения к оборудованию по протоколу Modbus:

Для накопления данных протокола используется кэш значений регистров, который заполняется процессами опроса потокковых данных. Записываются данные от источников следующих типов:

- потокковые данные с малым периодом опроса (порядка 10 сек)
- разряды битовых регистров, соответствующих авариям/предупреждениям;
- дискретные события – изменения значения регистра состояния или разряда битового регистра, обозначенного как дискретное событие;
- значение регистра – уставки;
- значения регистров данных, составляющих «мгновенный снимок», используемых для выдачи системам верхнего уровня;
- разряды битовых регистров, опрашиваемые с максимально возможной частотой (регистры «прерываний»);

При наличии в кэш значения данного типа/регистра/разряда оно обновляется новым значением. Значение получает метку времени на момент помещения/обновления в кэше.

В самом кэше значения регистров хранятся не в формате значений, полученных по Modbus. Формат передаваемых данных зависит от типа регистра и правил преобразования значений регистров, принятых в оборудовании:

- разряды битовых регистров хранятся в кэш и передаются как логически переменные;
- значения Modbus-регистров в формате float, а также значения в формате int с масштабным коэффициентом (10^N или заданным в виде параметра, например - KoeI, KoeU) – передаются в формате float в соответствии с требуемыми преобразованиями;
- остальные значения передаются в формате int.

Для синхронного получения данных клиент использует команду опроса (General Interrogation), на которую сервер БАРС СУАБ отвечает всеми значениями, хранящимися на данный момент в кэш. Типы ответных сообщений:

- Одноэлементная информация M_SP_NA_1 для boolean (разряды битовых регистров, события);
- Измеряемое значение, масштабированное значение M_MENB_1 для int;
- Измеряемое значение, короткий формат с плавающей запятой M_METF_1, M_MENC_1 для float.

При обновлении значения в кэше в протокол посылается асинхронное сообщение (*Спорадическая передача*). Передаваемые типы сообщений:

- Одноэлементная информация M_SP_TB_1 для boolean (разряды битовых регистров, события);
- Измеряемое значение, масштабированное значение M_METE_1 для int
- Измеряемое значение, короткий формат с плавающей запятой M_METF_1 для float.

В параметре **Адрес объекта** информации в асинхронных сообщениях и сообщениях, поступающих на команду опроса, передается адресные данные оборудования/регистра/разряда в виде *десятичного числа* в формате EURRRRBV, где

- RRRR – цифры номера регистра в диапазоне 0..9999,
- BV – цифра номера разряда, для int/float BV=0, для разрядов битового регистра номер разряда+1;
- EU – две цифры в диапазоне 0...15 – **<индекс оборудования> * 4 + (<индекс логического Unit>+1)**.

Возможна также явная альтернативная нумерация регистров для МЭК 60870 путем записи их уникальных номеров в метаданных.