ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

«СКАДИ»

(ООО «СКАДИ»)

СОГЛАСОВАНО **УТВЕРЖДАЮ**

СИСТЕМА КОМПЛЕКСНОГО ОБУЧЕНИЯ ЯДЕРНЫМ ТЕХНОЛОГИЯМ

Руководство оператора

34185873.425510.005.34.М

(На - листах)

2025

РАЗРАБОТАЛ

СОДЕРЖАНИЕ

1 НАЗНАЧЕНИЕ ПРОГРАММЫ 5
1.1 Сведения о назначении 5
1.2 Основные функции 5
2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ 7
2.1 Общие требования 7
2.2 Наличие лицензии проекта 7
2.3 Требования разработки моделей 8
3 ВЫПОЛНЕНИЕ ПРОГРАММЫ 9
3.1 Серверное программное обеспечение СКОЯТ 9
3.1.1 WoAccess сервер управления доступом 9
3.1.2 WoData сервер доступа к данным об использовании ресурсов 9
3.1.3 WoMark сервер контроля действий обучаемого 9
3.1.4 WoDbsh Утилита доступа к базе данных 10
3.2 Средства автономной разработки СКАДИ-МОДЕЛЬ 10
3.2.1 DsMM Запуск и останов менеджера моделей 10
3.2.2 DsView Просмотр системных реляционных таблиц 11
3.2.3 DsView Просмотр пользовательских реляционных таблиц 18
3.2.4 DsView Просмотр очередей сообщений 20
3.2.5 DsView Специальные возможности просмотра 21
3.2.6 DsAccess Функции доступа и конфигурирования 21
3.2.7 Проверка прохождения сигнала через систему СКОЯТ 23
3.2.8 DsISD интерактивный отладчик значений 26
3.2.9 DsQSD интерактивный отладчик очередей 28
3.3 Средства разработки видеокадров СВУ 30
3.3.1 Организация файлов видеокадров 30
3.3.2 DsEdit Редактирование датчиков аналоговых сигналов 32
3.3.3 DsEdit Редактирование объектов управления 35
3.3.4 Генерация файлов видеокадров 38
3.4 Средства разработки учебно-методических материалов 39
3.4.1 Организация файлов учебных материалов 39
3.4.2 Редактирование материалов Markup 40
3.4.3 Генерация файлов отображения из материалов Markup 43
3.5 Клиентский доступ 44
3.5.1 Вход с авторизацией 44
3.5.2 Выполнение лабораторных работ 46
3.5.3 Мониторинг и режим инструктора 48
3.5.4 Статистика выполненных работ 49
4 СООБЩЕНИЯ ОПЕРАТОРУ 51
4.1 Сообщения рантайм-менеджера 51
4.2 Сообщения утилит СКАДИ 51
4.3 Лог системных сообщений 51
4.4 Сообщения интерактивных отладчиков 52
ПРИЛОЖЕНИЯ 53
Приложение 1 – Состав загружаемых модулей СКАДИ-МОДЕЛЬ 53
Приложение 2 – Wo Опции командной строки серверного ПО СКОЯТ 55

Приложение 3 – Опции трассировки ПО 57
Приложение 3 – Системные реляционные таблицы 59
Приложение 4 – WoDbsh Опции командной строки утилиты доступа к БД 60
Приложение 5 – DsView Опции командной строки утилиты просмотра общей памяти 61
Приложение 6 – Представление значений структур 62
Приложение 7 – Последовательности переменных и параметров 64
Приложение 8 – DsAccess Опции командной строки утилиты доступа и конфигурирования 67
Приложение 9 – DsISD Опции и команды отладчика значений 68
Приложение 10 – DsQSD Опции и команды отладчика очередей 69
Приложение 11 – Перечень суффиксов расширения оборудования 70
ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ 72
ЛИСТ ИЗМЕНЕНИЙ 73

1 НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1 СВЕДЕНИЯ О НАЗНАЧЕНИИ

Наименование изделия — Система комплексного обучения ядерным технологиям «СКОЯТ 1.0».

Сокращенное наименование: «СКОЯТ».

Программа «СКОЯТ» («SCONT») зарегистрирована в реестре программ для ЭВМ № 2025617289 от 25.03.2025.

Наименование документа 34185873.425510.005.33.М, обозначение по ГОСТ 34.201-89.

«СКОЯТ 1.0» разрабатывается на основании документа 34185873.425510.005.ТЗ.М, Техническое задание, утвержденного 2 декабря 2024г.

Программное обеспечение СКОЯТ предназначено для обучения персонала ядерным технологиям в локальных и глобальных сетях.

Программное обеспечение СКАДИ-МОДЕЛЬ предназначено для разработки моделей технологических процессов. ПО СКАДИ-МОДЕЛЬ входит в состав программного обеспечения СКОЯТ.

1.2 ОСНОВНЫЕ ФУНКЦИИ

Программное обеспечение СКОЯТ реализует следующие функции:

- регистрация и управление доступом и ресурсами СКОЯТ;
- учет используемых ресурсов;
- управление технологическим процессом посредством системы верхнего уровня;
- осуществление связи с моделями технологических процессов;
- архивирование действий обучаемого и контрольных точек данных технологического процесса;
- представления обучающих программ;
- оценивания действий обучаемого.

Работа оператора с функциями СКОЯТ осуществляется посредством:

- клиентского доступа;
- доступа разработчика.

Клиентский доступ означает использование разработанной системы с помощью тонкого клиента-браузера. Может быть предложен всем зарегистрированным в системе СКОЯТ пользователям.

Доступ разработчика означает создание и редактирование содержимого обучающей системы. Такой доступ предполагает использование консольных утилит ПО СКАДИ-МОДЕЛЬ и средств редактирования содержания учебных материалов.

Редактирование и конвертирование видеокадров использует графические средства ПО СКАДИ. Подробное описание представлено в документе 34185873.425510.001.34.01.М «Программная платформа АСУ ТП и диагностики. Человеко-машинный интерфейс. Руководство оператора».

При активировании клиентского доступа пользователь СКОЯТ осуществляет контроль и управление моделями технологических процессов посредством браузера. Пользователь категории «экзаменуемый» осуществляет действия по управлению согласно учебно-методическим материалам. Пользователь инструктор осуществляет действия по контролю за обучаемым. Пользователь наблюдатель осуществляет только контроль за обучаемыми. Экран клиентского режима обучаемого представлен на рисунке 1.

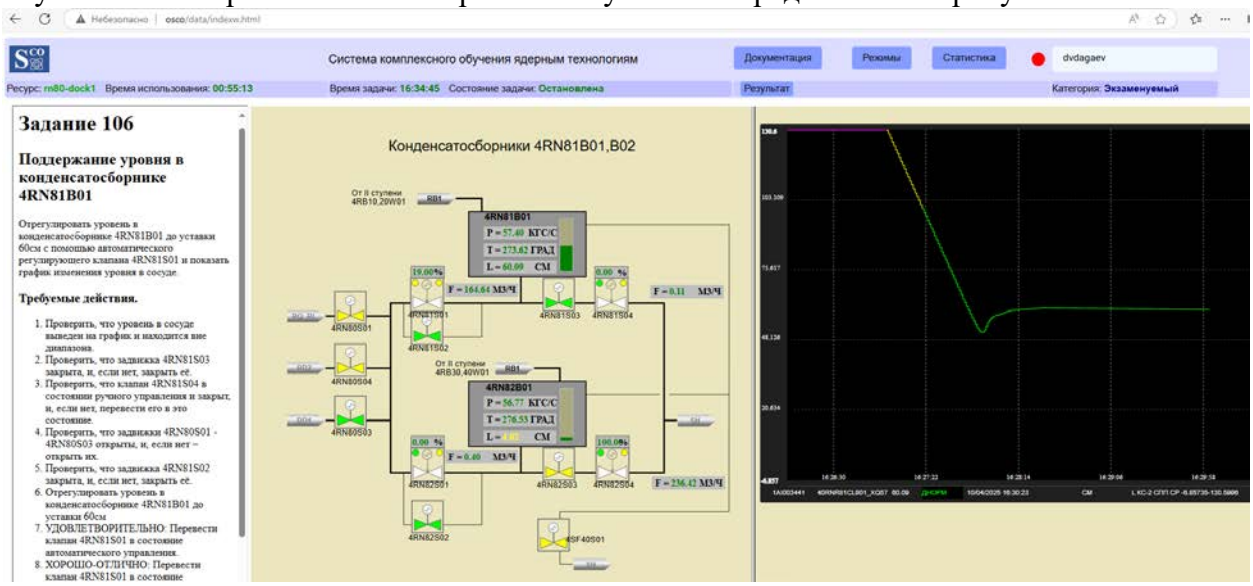


Рисунок 1 – Экран клиентского режима обучаемого

Тонкий клиент функционирует с готовой системой, разработанной и настроенной.

Поэтому требуется разработка моделей и учебных материалов. При наличии доступа разработчика осуществляется подготовка данных, необходимых для работы системы с тонким клиентом. Разработка включает в себя следующие области:

1. Разработка моделей технологического процесса;
2. Разработка видеокадров для системы верхнего уровня;
3. Разработка учебно-методических материалов.

Поэтому для клиентского доступа работает механизм взаимодействия с конечным пользователем. А для доступа разработчика работает механизм взаимодействия с автором конкретной прикладной системы.

2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

2.1 ОБЩИЕ ТРЕБОВАНИЯ

Клиентский доступ требует только наличие браузера. Все требования относятся к доступу разработчика.

Серверная часть программного обеспечения СКОЯТ должна состоять из запущенных задач WoAccess, WoData, WoMark.

```
user@osco:~/osco$ ps x | grep Wo
 741749 pts/1    Sl     0:00 WoAccess -db sqlite
 741750 pts/1    Sl     0:09 WoData -trace_level -1 -jcfg WoDocker.json
 741751 pts/1    Sl     0:12 WoMark -trace_level -1 -jcfg WoDocker.json
```

Программное обеспечение ПО СКОЯТ работает в контексте определенного проекта обучающей системы. Проект предполагает наличие корневого каталога, в котором размещены поддиректории, настроечные и конфигурационные файлы. Для работы ПО СКОЯТ в проектном каталоге должны быть конфигурационные файлы *.json (в примере выше WoDocker.json) и директории labs, reports, web.

Из корневого каталога проекта извлекается имя проекта – в данном случае osco. Используются следующие каталоги:

- labs – каталог учебно-методических материалов;
- reports – каталог протоколов действий обучаемого;
- web – каталог файлов представления данных.

Конфигурационные файлы могут быть переименованы. В таком случае, программа может быть запущена с опцией –jcfg, например WoData –jcfg.

Все приложения запускаются из проектного каталога и используют информацию, относящуюся к проектному каталогу и к объектам, ими порожденными. Такими объектами могут быть контейнеры, созданные из образов.

Для создания контейнеров серверной части необходимы будут исходные образы в версиях, установленных WoDocker.json.

```
user@osco:~/osco$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
model-tk70   v1.0d     fbebd15c096d   18 minutes ago   290MB
model-rn80   v1.5d     1e5ddb63d5dd   5 days ago       290MB
model-oisc   v6d       a028e1a73b1b   2 weeks ago      178MB
model-rb60   v1.1d     6ebcacfd10aa   2 weeks ago      290MB
```

Из приведенных образов будут создаваться контейнеры. Например, контейнер model-tk70-dock1 будет создан из образа model-tk70.

В системе должен быть запущен nginx, настроенный на WoAccess, WoData, WoMark.

2.2 НАЛИЧИЕ ЛИЦЕНЗИИ ПРОЕКТА

Приложения-сервисы WoAccess, WoData, WoMark используют файл лицензии *.lic как необходимое условие запуска для сервисов данного проекта. Лицензия создается разработчиками ПО СКОЯТ и передается как разрешение на использование ПО СКОЯТ в рамках данного проекта <project>. Файл лицензии должен находиться в каталоге <project>. Файл лицензии должен иметь формат Wos_<hostname>.lic, где hostname является именем хоста стандартного сервера. Ниже пример для хоста osco-net-local.

```
user@osco-net-local:~/osco$ ls *.lic
Wos_osco-net-local.lic
```

При отсутствии валидной лицензии приложения-сервисы будут аварийно завершаться с выводом сообщения: “SCONT License error file=Wos_osco.lic”. Работа ПО СКОЯТ возможна только при наличии валидной лицензии.

При просроченной лицензии приложениями-сервисами будет выведено сообщение: “SCONT License file= Wos_osco.lic expired=14/10/2022”. Работа ПО СКОЯТ возможна только при наличии актуальной лицензии.

Лицензия вводит ограничения на максимальное количество одновременно запущенных ресурсов. Каждый ресурс представляет собой пользовательский контейнер. При превышении числа контейнеров над максимальным значением, заданным в лицензии выводится сообщение: “SCONT License resources requested=200 > max=100”. Работа ПО СКОЯТ возможна только при отсутствии выхода количества ресурсов за границы лицензии.

2.3 ТРЕБОВАНИЯ РАЗРАБОТКИ МОДЕЛЕЙ

Программное обеспечение технологических моделей разрабатывается автономно и упаковывается в образы для последующего создания из них контейнеров. Поэтому разработка моделей производится в одной конфигурации, а интеграция – в другой.

Программное обеспечение СКАДИ-МОДЕЛЬ должно состоять из запущенного менеджера моделей DsMM и используемых в данной конфигурации подзадач. Менеджер моделей может использовать как последовательную, так и параллельную обработку. Во втором случае запущенный менеджер стартует подзадачи как процессы.

```
595314 pts/4    S+     0:05 DsMM
595321 pts/4    S+     0:00 /home/ddag/scadi0/sbinlur/DsPersnap
595322  pts/4        S+          0:00  /home/ddag/scadi0/sbinlur/DsModel  -hls
DpgcValves,DpgcCNValves,DpgcPumps,DpgcBlks,DpgcAvrs  -task_name  DsModelGC  -
trace_level 64
595323  pts/4        S+          0:00  /home/ddag/scadi0/sbinlur/DsModel  -hls
DpgcNodes,DpgcPipes -hlm DpmodelMain -cycle_max 30 -task_name DsModelMain -
trace_level 64
```

Последовательная конфигурация будет иметь только процесс DsMM и подзадачи как библиотеки.

В каталоге проекта будут располагаться системные загружаемые модули и пользовательские плагины (таблицы 1-1 и 1-2 приложения 1). Состав подкаталогов динамически загружаемых модулей должен соответствовать таблице 1-3 приложения 1.

3 ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1 СЕРВЕРНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СКОЯТ

3.1.1 WoAccess сервер управления доступом

Сервер управления доступом работает в режиме 24x7 после старта приложения и завершает работу при останове приложения.

Перечень опций командной строки сервера WoAccess приведен в таблице 2-1 приложения 2.

При запуске в режиме трассировки с опцией –trace_level происходит консольный вывод сообщений, относящихся к флагам активизированной трассировки. Перечень опций трассировки приведен в таблице 3-1 приложения 3.

WoAccess является веб-сокет сервером и обменивается данными только через веб-клиент.

При работе WoAccess использует алгоритм авторизации СКОЯТ, имея в составе стандартный для протокола OAuth2.0 фрагмент. В рамках алгоритма осуществляется запрос Яндекса на авторизацию с дальнейшей обработкой результатов.

При успешной авторизации далее осуществляется проверка на наличие авторизированного пользователя в базе данных СКОЯТ. Конкретные параметры соединения с БД устанавливаются опциями –db и –dsn по таблице 2-1. Например, при использовании БД sqlite в качестве имени источника данных устанавливается файл sdb.

```
user@osco:~/osco$ ls -al sdb
-rw-r--r-- 1 user scadi 8192 апр  7 11:35 sdb
```

Предполагается, что источник данных содержит все необходимые таблицы для работы.

По завершении авторизации в передаваемые другим серверам данные добавляется структура ScadiID, имеющая всю информацию об авторизации.

3.1.2 WoData сервер доступа к данным об использовании ресурсов

Сервер доступа к данным об использовании ресурсов работает в режиме 24x7 после старта приложения и завершает работу при останове приложения.

Перечень опций командной строки сервера WoData приведен в таблице 2-2 приложения 2.

При запуске в режиме трассировки с опцией –trace_level происходит консольный вывод сообщений, относящихся к флагам активизированной трассировки. Перечень опций трассировки приведен в таблице 3-1 приложения 3.

WoData является веб-сокет сервером и обменивается данными только через веб-клиент.

При работе WoData выдает информацию о пользователе и данные о пройденных курсах и полученных оценках. Сервис использует аналогичное соединение с СУБД. Конфигурационный файл –jcfg используется для настроек взаимодействия с системой СКОЯТ. В стандартной конфигурации используется WoDocker.json.

3.1.3 WoMark сервер контроля действий обучаемого

Сервер контроля действий обучаемого работает в режиме 24x7 после старта приложения и завершает работу при останове приложения.

Перечень опций командной строки сервера WoMark приведен в таблице 2-3 приложения 2.

При запуске в режиме трассировки с опцией –trace_level происходит консольный вывод сообщений, относящихся к флагам активизированной трассировки. Перечень опций трассировки приведен в таблице 3-1 приложения 3.

WoMark является веб-сокет сервером и обменивается данными только через веб-клиент.

При работе WoMark проводит занятия с технологическими моделями и фиксирует результаты для последующей обработки. Сервис использует аналогичное соединение с СУБД. Конфигурационный файл –jcfg используется для настроек взаимодействия с системой СКОЯТ. В стандартной конфигурации используется WoDocker.json.

3.1.4 WoDbsh Утилита доступа к базе данных

Утилита доступа к БД позволяет осуществлять запросы и модификации БД на уровне SQL из командной строки (приложение 4).

Например, добавление содержимого таблицы пользователей реализуется следующей командой.

```
user@osco:~/osco$ cat mkdb_users.sh
WoDbsh -ef "users.sql" -db_file "sdb"
```

Файл users.sql представляет собой команды добавления таблицы пользователей.

```
user@osco:~/osco$ cat users.sql
drop table if exists sc_users;
create table sc_users(ind int, name text, register_id int, category int,
activity text);
insert into sc_users values(1, 'dvdagaev', 1, 15, "basic");
insert into sc_users values(2, 'ovchinnikov@scadi.tech', 2, 15, "basic");
insert into sc_users values(3, 'gashenkodv', 3, 15, "basic");
```

Выполнение указанного скрипта пересоздает таблицу пользователей sc_users.

Запросы к созданной таблице реализуются следующей командой.

```
WoDbsh -q "select * from sc_users;" -delim "|" -db_file "sdb"
```

В качестве ответа на запрос выдается информация в консольной форме.

```
1|dvdagaev|1|15|basic|
2|ovchinnikov@scadi.tech|2|15|basic|
3|gashenkodv|3|15|basic|
sdb size: 8192
```

Утилита доступа к БД реализует консольный доступ к таблица используемой в СКОЯТ базы данных.

3.2 СРЕДСТВА АВТОНОМНОЙ РАЗРАБОТКИ СКАДИ-МОДЕЛЬ

3.2.1 DsMM Запуск и останов менеджера моделей

Запуск менеджера моделей осуществляется командой DsMM. Опция –cfg_file подставляет заданный конфигурационный файл, опция –trace_level осуществляет трассировку согласно таблице 3-2 приложения 3. Например,

```
DsMM -cfg_file DsmConfigSeq.cfg –trace_level 2
```

Опция запуска –ndb <число> реализует вариант с несколькими клонами-менеджерами моделей для одного проекта. Для этого нужно предварительно клонировать каталог db в db<число>. Например, сделать это следующей командой для номера 7.

```
cp –r db/ db7
```

В такой конфигурации менеджер запускает все задачи независимо от основного набора. При этом обычно реализуется однопроцессная загрузка. Доступ к данным определенного клона осуществляется с явным указанием опции ndb. Например, чтобы воспользоваться командой DsView, нужно будет ее вызывать с опцией DsView –ndb 7 …

При запуске менеджера моделей в многопроцессной загрузке в системе появляются несколько процессов, соответствующих задачам:

- Менеджер моделей DsMM;
- Сервис сохранения данных DsPersnap;
- Процессы оболочки расчетных задач DsModel;
- Система верхнего уровня WsShmCntrl;
- Сервис очередей синхронизации DsDVP.

При запуске менеджера моделей в однопроцессной загрузке в системе появляется только один процесс – DsMM.

При этом процесс DsMM загружает в свое адресное пространство плагины:

- DsPersnapInT - сервис сохранения данных DsPersnap;
- DsModelInT - процессы оболочки расчетных задач DsModel;
- DsDVPInT - сервис очередей синхронизации DsDVP.

Перечень и описание используемых плагинов представлено в таблице 1-2 приложения 1. Подкаталоги, из которых берутся модули для динамической загрузки, представлены в таблице 1-3 приложения 1.

Останов менеджера моделей может осуществляться следующими способами:

- Вызовом DsAccess –end all из командной строки;
- Командой end из отладчика DsISD;
- Убиванием процесса DsMM и всех порожденных им процессов. При наличии запущенных средств отладки DsISD, DsQSD они должны завершить свою работу.

3.2.2 DsView Просмотр системных реляционных таблиц

Таблица 1-3 приложения 1 дает полный перечень утилит сервиса и отладки. DsView предназначена для просмотра содержимого глобалов общей памяти. Перечень опций командной строки DsView приведен в таблице 5-1 приложения 5.

Список заголовков всех глобалов выводится следующей командой.

```
DsView –globtype .
```

Результат, выводимый на консоль, представлен на рисунке 2.

Заголовок каждого глобала содержит поля:

- максимальное число объектов n_total_objs;
- текущее число объектов n_used_objs;
- размер используемой памяти sz_used_mem;
- поле переменного значения usr_data, обычно используемое как индекс или номер типа.

Тип глобала выводится в крайней левой позиции. Поле переменного значения usr_data для глобалов типа q, а определяет индекс текущей записи в кольцевом буфере сообщений. В динамике сигналы приходят и записываются в кольцевые буферы. Если usr_data в глобале q меняется, то данные в кольцевом буфере обновляются. В противном случае обновлений нет. В приведенном примере поле usr_data для глобала Modelq равно 0 и не менятся. Это обусловлено тем, что расчетная задача находится в статусе FREEZE.

Значение поля usr_data для глобала n означает номер типа. Например, для глобала AIs поле usr_data=40. Это означает, что глобал AIs состоит из элементов AI, расположенных в базе данных реального времени под индексом 40.

```
G 0 Globals n_total_objs= 512, n_used_objs= 46, sz_used_mem= 45056, usr_data= 0
E 1 Enums n_total_objs= 400, n_used_objs= 119, sz_used_mem= 171200, usr_data= 5000
T 2 Types n_total_objs= 80, n_used_objs= 56, sz_used_mem= 11520, usr_data= 1
A 3 Attrs n_total_objs= 900, n_used_objs= 747, sz_used_mem= 187200, usr_data= 100
S 4 Syslogs n_total_objs= 30000, n_used_objs= 6, sz_used_mem= 6480000, usr_data= 2991
s 5 Consts n_total_objs= 420, n_used_objs= 210, sz_used_mem= 26880, usr_data= 17
s 6 Characteristics n_total_objs= 32, n_used_objs= 12, sz_used_mem= 5248, usr_data= 18
s 7 Usergroups n_total_objs= 16, n_used_objs= 5, sz_used_mem= 2688, usr_data= 19
s 8 Tasks n_total_objs= 6, n_used_objs= 6, sz_used_mem= 3168, usr_data= 20
s 9 HGroups n_total_objs= 1, n_used_objs= 1, sz_used_mem= 592, usr_data= 21
s 10 Hosts n_total_objs= 2, n_used_objs= 2, sz_used_mem= 608, usr_data= 22
s 11 XGroups n_total_objs= 1, n_used_objs= 1, sz_used_mem= 1064, usr_data= 23
s 12 Xchs n_total_objs= 2, n_used_objs= 2, sz_used_mem= 1008, usr_data= 24
s 13 CSets n_total_objs= 200, n_used_objs= 200, sz_used_mem= 441600, usr_data= 27
s 14 Cells n_total_objs= 6100, n_used_objs= 6100, sz_used_mem= 6661200, usr_data= 26
s 15 Reports n_total_objs= 100, n_used_objs= 100, sz_used_mem= 65600, usr_data= 29
s 16 Repdatas n_total_objs= 30000, n_used_objs= 30000, sz_used_mem= 10080000, usr_data= 28
s 17 WSets n_total_objs= 200, n_used_objs= 200, sz_used_mem= 276800, usr_data= 31
s 18 WDatas n_total_objs= 4100, n_used_objs= 4100, sz_used_mem= 13333200, usr_data= 30
s 19 VFiles n_total_objs= 100, n_used_objs= 100, sz_used_mem= 17200, usr_data= 32
s 20 FVars n_total_objs= 10000, n_used_objs= 10000, sz_used_mem= 3240000, usr_data= 33
s 21 Alarms n_total_objs= 3200, n_used_objs= 3200, sz_used_mem= 332800, usr_data= 34
s 22 TFiles n_total_objs= 50, n_used_objs= 50, sz_used_mem= 8600, usr_data= 38
s 23 TVars n_total_objs= 10000, n_used_objs= 10000, sz_used_mem= 3240000, usr_data= 39
s 24 TAlarms n_total_objs= 53, n_used_objs= 53, sz_used_mem= 13568, usr_data= 35
s 25 Archs n_total_objs= 1, n_used_objs= 1, sz_used_mem= 168, usr_data= 36
s 26 Jobs n_total_objs= 256, n_used_objs= 256, sz_used_mem= 186368, usr_data= 37
s 27 Gstats n_total_objs= 512, n_used_objs= 46, sz_used_mem= 81920, usr_data= 25
n 28 AIs n_total_objs= 15000, n_used_objs= 14916, sz_used_mem= 5520000, usr_data= 40
i 29 AIi n_total_objs= 15000, n_used_objs= 14916, sz_used_mem= 540000, usr_data= 0
n 30 DIs n_total_objs= 150000, n_used_objs= 148707, sz_used_mem= 50400000, usr_data= 41
i 31 DIi n_total_objs= 150000, n_used_objs= 148707, sz_used_mem= 5400000, usr_data= 0
q 32 GCq n_total_objs= 2000, n_used_objs= 0, sz_used_mem= 96000, usr_data= 0
n 33 VLVs n_total_objs= 100, n_used_objs= 17, sz_used_mem= 28000, usr_data= 46
n 34 CNVs n_total_objs= 100, n_used_objs= 2, sz_used_mem= 43200, usr_data= 47
n 35 PMPs n_total_objs= 100, n_used_objs= 6, sz_used_mem= 37600, usr_data= 48
n 36 NODs n_total_objs= 100, n_used_objs= 22, sz_used_mem= 119200, usr_data= 52
n 37 CPLs n_total_objs= 500, n_used_objs= 3, sz_used_mem= 100000, usr_data= 54
n 38 PIPs n_total_objs= 100, n_used_objs= 25, sz_used_mem= 68800, usr_data= 51
q 39 Modelq n_total_objs= 2000, n_used_objs= 0, sz_used_mem= 96000, usr_data= 0
n 40 NPMs n_total_objs= 1, n_used_objs= 1, sz_used_mem= 152, usr_data= 55
n 41 HEAs n_total_objs= 20, n_used_objs= 1, sz_used_mem= 4160, usr_data= 53
n 42 BLKs n_total_objs= 1000, n_used_objs= 12, sz_used_mem= 288000, usr_data= 49
n 43 AVRs n_total_objs= 1000, n_used_objs= 1, sz_used_mem= 344000, usr_data= 50
q 44 DVPq n_total_objs= 1000, n_used_objs= 0, sz_used_mem= 48000, usr_data= 0
q 45 Controlq n_total_objs= 1000, n_used_objs= 0, sz_used_mem= 48000, usr_data= 0
```

Рисунок 2 – Печать заголовков всех глобалов

Печать глобалов по типу globtype, имени globname или номеру global производится в формате, зависящем от типа глобала. Печать глобала G (рисунок 3), содержащего список глобалов, осуществляется одной из команд.

```
DsView –globtype G
DsView –globname Globals
DsView –global 0
```

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -globtype G
G 0 Globals n_total_objs= 512, n_used_objs= 46, sz_used_mem= 45056, usr_data= 0
n_globals= 46
Global 0 -globtype G -globname Globals -globsize 45080 -max_count 512 -olength 0 -gnext 0 -persist_inte
rval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 01/01/70 03:00:00.000 -task_modified 0 -flags {}
Global 1 -globtype E -globname Enums -globsize 171224 -max_count 400 -olength 0 -gnext 0 -persist_inter
val 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.355 -task_modified 0 -flags {}
Global 2 -globtype T -globname Types -globsize 11544 -max_count 80 -olength 0 -gnext 0 -persist_interva
l 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.357 -task_modified 0 -flags {}
Global 3 -globtype A -globname Attrs -globsize 187224 -max_count 900 -olength 0 -gnext 0 -persist_inter
val 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.357 -task_modified 0 -flags {}
Global 4 -globtype S -globname Syslogs -globsize 6480024 -max_count 30000 -olength 0 -gnext 0 -persist_
interval -1 -refl 0 -used $TRUE -open $TRUE -t_modified 23/04/25 15:26:44.076 -task_modified 3 -flags {
}
Global 5 -globtype s -globname Consts -globsize 26904 -max_count 420 -otype 17 -olength 64 -gnext 0 -pe
rsist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:52.355 -task_modified 0 -fl
ags {}
Global 6 -globtype s -globname Characteristics -globsize 5272 -max_count 32 -otype 18 -olength 164 -gne
xt 0 -persist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.358 -task_modifi
ed 0 -flags {}
Global 7 -globtype s -globname Usergroups -globsize 2712 -max_count 16 -otype 19 -olength 168 -gnext 0
-persist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.358 -task_modified 0
-flags {}
Global 8 -globtype s -globname Tasks -globsize 3192 -max_count 6 -otype 20 -olength 528 -gnext 0 -persi
st_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.358 -task_modified 0 -flags
 {}
Global 9 -globtype s -globname HGroups -globsize 616 -max_count 1 -otype 21 -olength 592 -gnext 0 -pers
ist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.358 -task_modified 0 -flag
s {}
Global 10 -globtype s -globname Hosts -globsize 632 -max_count 2 -otype 22 -olength 304 -gnext 0 -persi
st_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.358 -task_modified 0 -flags
 {}
Global 11 -globtype s -globname XGroups -globsize 1088 -max_count 1 -otype 23 -olength 1064 -gnext 0 -p
ersist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.359 -task_modified 0 -f
lags {}
Global 12 -globtype s -globname Xchs -globsize 1032 -max_count 2 -otype 24 -olength 504 -gnext 0 -persi
st_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.359 -task_modified 0 -flags
 {}
Global 13 -globtype s -globname CSets -globsize 441624 -max_count 200 -otype 27 -olength 2208 -gnext 0
-persist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.369 -task_modified 0
-flags {}
Global 14 -globtype s -globname Cells -globsize 6661224 -max_count 6100 -otype 26 -olength 1092 -gnext
0 -persist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.515 -task_modified
0 -flags {}
Global 15 -globtype s -globname Reports -globsize 65624 -max_count 100 -otype 29 -olength 656 -gnext 0
-persist_interval 0 -refl 0 -used $TRUE -open $TRUE -t_modified 21/04/25 08:13:42.517 -task_modified 0
-flags {}
```

Рисунок 3 – Печать списка глобалов

При печати выводятся параметры номера, имени, типа и размера глобала, маскимальное число объектов max_count, тип и длина объекта, интервал сохранения, ссылки на другие глобалы ref1 и флаги при наличии. Переменные t_modified и task_modified указывают на время обновления и задачу, которая данное обновление осуществила.

Печать глобала T (рисунки 4, 5), содержащего типы объектов, осуществляется одной из команд (используемые опции oid, limit).

```
DsView -globtype T
DsView -globname Types
DsView -global 2
```

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -global 2 -limit 16
T 2 Types n_total_objs= 80, n_used_objs= 56, sz_used_mem= 11520, usr_data= 1
Type 0  -descr {} -size 4 -alignment 4 -var 0 -stdotype 0 -crc16 0 -first 1 -gfirst 0 -default_attr 0 -
lckproc_attr 0 -tmp_attr 1
Type 1 I1 -descr {1 byte integer} -size 1 -alignment 1 -var 1 -stdotype 0 -crc16 2953 -first 0 -gfirst
0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 2 I2 -descr {2 bytes integer} -size 2 -alignment 2 -var 1 -stdotype 0 -crc16 23091 -first 0 -gfirs
t 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 3 I4 -descr {4 bytes integer} -size 4 -alignment 4 -var 1 -stdotype 0 -crc16 -250 -first 0 -gfirst
 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 4 I8 -descr {8 bytes integer} -size 8 -alignment 8 -var 1 -stdotype 0 -crc16 -19667 -first 0 -gfir
st 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 5 R4 -descr {4 bytes float} -size 4 -alignment 4 -var 1 -stdotype 0 -crc16 13468 -first 0 -gfirst
0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 6 R8 -descr {8 bytes float} -size 8 -alignment 8 -var 1 -stdotype 0 -crc16 30903 -first 0 -gfirst
0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 7 C8 -descr {8 bytes command structure} -size 8 -alignment 8 -var 1 -stdotype 0 -crc16 23463 -firs
t 0 -gfirst 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 8 E16 -descr {16 bytes event structure} -size 16 -alignment 8 -var 1 -stdotype 0 -crc16 8959 -firs
t 0 -gfirst 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 9 BOOL -descr {1 byte boolean} -size 1 -alignment 1 -var 1 -stdotype 0 -crc16 -1083 -first 0 -gfir
st 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 10 STR -descr {up to 16 bytes string} -size 16 -alignment 1 -var 1 -stdotype 0 -crc16 -14434 -firs
t 0 -gfirst 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 11 VREF -descr {variable reference} -size 16 -alignment 4 -var 1 -stdotype 0 -crc16 2746 -first 0
-gfirst 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 12 CHAR -descr {1 byte char} -size 1 -alignment 1 -var 1 -stdotype 0 -crc16 -25912 -first 0 -gfirs
t 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 13 I2QTS -descr {2 byte integer or bool value, quality, timestamp} -size 16 -alignment 8 -var 1 -s
tdotype 0 -crc16 10612 -first 0 -gfirst 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 14 I4QTS -descr {4 bytes integer value, quality, timestamp} -size 16 -alignment 8 -var 1 -stdotype
 0 -crc16 30376 -first 0 -gfirst 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
Type 15 R4QTS -descr {4 bytes float value, quality, timestamp} -size 16 -alignment 8 -var 1 -stdotype 0
 -crc16 -12229 -first 0 -gfirst 0 -default_attr 0 -lckproc_attr 0 -tmp_attr 0
```

Рисунок 4 – Начало глобала Types

Начало глобала ограничим 16 записями –limit 16. В начале глобала Types идут встроенные типы СКАДИ (I1, I2, I4, I8, R4 R8, C8, E16, BOOL, VREF, CHAR).

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -global 2 -oid 40 -limit 16
T 2 Types n_total_objs= 80, n_used_objs= 56, sz_used_mem= 11520, usr_data= 1
Type 40 AI -descr {Analog Input} -size 368 -alignment 8 -var 23 -stdotype 0 -crc16 30863 -first 456 -gf
irst 28 -default_attr 466 -lckproc_attr 0 -tmp_attr 470
Type 41 DI -descr {Discrete Input} -size 336 -alignment 8 -var 23 -stdotype 0 -crc16 -29494 -first 471
-gfirst 30 -default_attr 472 -lckproc_attr 0 -tmp_attr 476
Type 42 TI -descr {Text Input} -size 816 -alignment 8 -var 23 -stdotype 0 -crc16 -24933 -first 477 -gfi
rst 0 -default_attr 478 -lckproc_attr 0 -tmp_attr 480
Type 43 NI -descr {Numeric Input} -size 336 -alignment 8 -var 23 -stdotype 0 -crc16 25664 -first 481 -g
first 0 -default_attr 482 -lckproc_attr 0 -tmp_attr 487
Type 44 AC -descr {Analog Calculated} -size 920 -alignment 8 -var 23 -stdotype 0 -crc16 -28271 -first 4
88 -gfirst 0 -default_attr 497 -lckproc_attr 0 -tmp_attr 505
Type 45 DC -descr {Discrete Calculated} -size 872 -alignment 8 -var 23 -stdotype 0 -crc16 26068 -first
506 -gfirst 0 -default_attr 506 -lckproc_attr 0 -tmp_attr 513
Type 46 VLV -descr {Valve Component} -size 280 -alignment 8 -var 13 -stdotype 0 -crc16 22837 -first 514
 -gfirst 33 -default_attr 0 -lckproc_attr 0 -tmp_attr 530
Type 47 CNV -descr {Control Valve Component} -size 432 -alignment 8 -var 13 -stdotype 0 -crc16 23084 -f
irst 531 -gfirst 34 -default_attr 0 -lckproc_attr 0 -tmp_attr 565
Type 48 PMP -descr {Pump Component} -size 376 -alignment 8 -var 13 -stdotype 0 -crc16 -19981 -first 566
 -gfirst 35 -default_attr 0 -lckproc_attr 0 -tmp_attr 595
Type 49 BLK -descr {Blocking logic component} -size 288 -alignment 8 -var 13 -stdotype 0 -crc16 15936 -
first 596 -gfirst 42 -default_attr 0 -lckproc_attr 0 -tmp_attr 609
Type 50 AVR -descr {AVR logic component} -size 344 -alignment 8 -var 13 -stdotype 0 -crc16 -11774 -firs
t 610 -gfirst 43 -default_attr 0 -lckproc_attr 0 -tmp_attr 626
Type 51 PIP -descr {Pipe Link} -size 688 -alignment 8 -var 13 -stdotype 0 -crc16 29943 -first 627 -gfir
st 38 -default_attr 0 -lckproc_attr 0 -tmp_attr 667
Type 52 NOD -descr {Node Point} -size 1192 -alignment 8 -var 13 -stdotype 0 -crc16 -30346 -first 668 -g
first 36 -default_attr 0 -lckproc_attr 0 -tmp_attr 727
Type 53 HEA -descr {Heat Exchanger} -size 208 -alignment 8 -var 13 -stdotype 0 -crc16 -5695 -first 728
-gfirst 41 -default_attr 0 -lckproc_attr 0 -tmp_attr 735
Type 54 CPL -descr {Capillary of pipe} -size 200 -alignment 8 -var 13 -stdotype 0 -crc16 -25463 -first
736 -gfirst 37 -default_attr 0 -lckproc_attr 0 -tmp_attr 740
Type 55 NPM -descr {Node-Pipe Model Main} -size 152 -alignment 8 -var 13 -stdotype 0 -crc16 16204 -firs
t 741 -gfirst 40 -default_attr 0 -lckproc_attr 0 -tmp_attr 746
```

Рисунок 5 – Окончание глобала Types

Оканчивается глобал типами, заданными пользователями. Приведенный запрос с параметрами –oid 40 –limit 16 печатает не более шестнадцати записей, начиная с номера 40. В нашем случае выведены записи от 40 до 55 включительно.

Такая схема индексации записей используется для всех типов глобалов утилиты DsView.

Завершается таблица пользовательскими типами NOD, HEA, CPL, NPM.

Для каждого типа выводятся параметры описания, размера, выравнивания, контрольных сумм, атрибута по умолчанию и др.

Для глобала Attrs может быть задействован фильтр otype для вывода атрибутов, относящихся только к данному типу, рисунки 6, 7. В противном случае консольная печать всех атрибутов займет много времени (используемые опции oid, limit, otype).

```
DsView –globtype A –otype AI
DsView –globname Attrs –otype AI
DsView –global 3 –otype AI
```

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -global 3 -otype AI -limit 14
A 3 Attrs n_total_objs= 900, n_used_objs= 747, sz_used_mem= 187200, usr_data= 100
Attr 1 oid -type  -otype 0 -attr_number 0 -next 0 -offset 0 -size 4 -tlength 4 -strlength 13 -alias  -d
escr {Object Identifier} -category 4 -atype 3 -alignment 4 -crc32 1116997197 -link_number 0 -dim 1 -arc
h_num 0 -readonly $TRUE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 0
Attr 2 ncadr -type  -otype 0 -attr_number 1 -next 0 -offset 4 -size 4 -tlength 4 -strlength 13 -alias
-descr {Number of Cadr} -category 4 -atype 3 -alignment 4 -crc32 115437978 -link_number 0 -dim 1 -arch_
num 0 -readonly $TRUE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 0
Attr 3 name -type  -otype 0 -attr_number 2 -next 0 -offset 8 -size 32 -tlength 16 -strlength 32 -alias
 -descr {Object Name String Idl} -category 4 -atype 10 -alignment 1 -crc32 -2137497883 -link_number 0 -
dim 2 -arch_num 0 -readonly $TRUE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 4 proflags -type  -otype 0 -attr_number 3 -next 0 -offset 40 -size 1 -tlength 1 -strlength 7 -alia
s  -descr {Processing state flags} -category 4 -atype 1 -alignment 1 -crc32 484287557 -link_number 0 -d
im 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 5 alloflags -type  -otype 0 -attr_number 4 -next 0 -offset 41 -size 1 -tlength 1 -strlength 7 -ali
as  -descr {Object allocation flags} -category 4 -atype 1 -alignment 1 -crc32 1261498965 -link_number 0
 -dim 1 -arch_num 0 -readonly $TRUE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 6 host_local -type  -otype 0 -attr_number 5 -next 0 -offset 42 -size 1 -tlength 1 -strlength 7 -al
ias  -descr {Host Number for Local data} -category 4 -atype 1 -alignment 1 -crc32 885084818 -link_numbe
r 0 -dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 7 fsm_state -type  -otype 0 -attr_number 6 -next 0 -offset 43 -size 1 -tlength 1 -strlength 7 -ali
as  -descr {State of FSM Automata} -category 4 -atype 1 -alignment 1 -crc32 -2142166205 -link_number 0
-dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 0
Attr 8 dst_mask -type  -otype 0 -attr_number 7 -next 0 -offset 44 -size 4 -tlength 4 -strlength 13 -ali
as  -descr {Destination Mask} -category 4 -atype 3 -alignment 4 -crc32 33919209 -link_number 0 -dim 1 -
arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 9 revnum -type  -otype 0 -attr_number 8 -next 0 -offset 48 -size 4 -tlength 4 -strlength 13 -alias
  -descr {Revision Number} -category 4 -atype 3 -alignment 4 -crc32 -2054086092 -link_number 0 -dim 1 -
arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 10 revres -type  -otype 0 -attr_number 9 -next 0 -offset 52 -size 4 -tlength 4 -strlength 13 -alia
s  -descr {Revision update result} -category 4 -atype 3 -alignment 4 -crc32 543184658 -link_number 0 -d
im 1 -arch_num 0 -readonly $TRUE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 0
Attr 11 revdate -type  -otype 0 -attr_number 10 -next 0 -offset 56 -size 8 -tlength 8 -strlength 21 -al
ias  -descr {Revision Date} -category 4 -atype 4 -alignment 8 -crc32 561601821 -link_number 0 -dim 1 -a
rch_num 0 -readonly $FALSE -resetable $FALSE -def_type 2 -sub_type 0 -sync_type 2
Attr 12 code -type  -otype 0 -attr_number 11 -next 0 -offset 64 -size 32 -tlength 16 -strlength 32 -ali
as  -descr {Object technological} -category 4 -atype 10 -alignment 1 -crc32 -1448210309 -link_number 0
-dim 2 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 13 pin -type  -otype 0 -attr_number 12 -next 0 -offset 96 -size 32 -tlength 16 -strlength 32 -alia
s  -descr {Object contact id} -category 4 -atype 10 -alignment 1 -crc32 -1254421730 -link_number 0 -dim
 2 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
Attr 14 descr -type  -otype 0 -attr_number 13 -next 0 -offset 128 -size 80 -tlength 16 -strlength 80 -a
lias  -descr {Object Description} -category 4 -atype 10 -alignment 1 -crc32 970577726 -link_number 0 -d
im 5 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 2
```

Рисунок 6 – Начало выборки глобала Attrs

В начале глобала Attrs расположены атрибуты, относящиеся ко всем типам из структуры OGeneric в составе oid, ncadr, name, proflags, alloflags, host_local. Далее идут атрибуты из типа OConfigurable в составе fsm_state, dst_mask, revnum, revres, revdate, code, pin. После них идут атрибуты из типа OManagable в составе descr и так далее.

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -global 3 -otype PIP -oid 660 -limit 8
A 3 Attrs n_total_objs= 900, n_used_objs= 747, sz_used_mem= 187200, usr_data= 100
Attr 660 vold -type PIP -otype 51 -attr_number 33 -next 661 -offset 608 -size 8 -tlength 8 -strlength 2
0 -alias  -descr {Velocity from previous time step} -category 1 -atype 6 -alignment 8 -crc32 1907821896
 -link_number 0 -dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_typ
e 1
Attr 661 a -type PIP -otype 51 -attr_number 34 -next 662 -offset 616 -size 8 -tlength 8 -strlength 20 -
alias  -descr {gv=dP*a+v} -category 1 -atype 6 -alignment 8 -crc32 -984961487 -link_number 0 -dim 1 -ar
ch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 1
Attr 662 b -type PIP -otype 51 -attr_number 35 -next 663 -offset 624 -size 8 -tlength 8 -strlength 20 -
alias  -descr {gv=dP*a+v} -category 1 -atype 6 -alignment 8 -crc32 1547960203 -link_number 0 -dim 1 -ar
ch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 1
Attr 663 cpl0 -type PIP -otype 51 -attr_number 36 -next 664 -offset 632 -size 16 -tlength 16 -strlength
 16 -alias  -descr {Capillary 0 upleft ref} -category 1 -atype 11 -alignment 4 -crc32 1829559109 -link_
number 0 -dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 8 -sub_type 0 -sync_type 2
Attr 664 cpl1 -type PIP -otype 51 -attr_number 37 -next 665 -offset 648 -size 16 -tlength 16 -strlength
 16 -alias  -descr {Capillary 1 dnright ref} -category 1 -atype 11 -alignment 4 -crc32 436988883 -link_
number 0 -dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 8 -sub_type 0 -sync_type 2
Attr 665 epos0 -type PIP -otype 51 -attr_number 38 -next 666 -offset 664 -size 8 -tlength 8 -strlength
20 -alias  -descr {Effective position 0 upleft} -category 1 -atype 6 -alignment 8 -crc32 -1538117158 -l
ink_number 0 -dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 1
Attr 666 epos1 -type PIP -otype 51 -attr_number 39 -next 667 -offset 672 -size 8 -tlength 8 -strlength
20 -alias  -descr {Effective position 1 dnright} -category 1 -atype 6 -alignment 8 -crc32 -749403828 -l
ink_number 0 -dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 1
Attr 667 epos -type PIP -otype 51 -attr_number 40 -next 0 -offset 680 -size 8 -tlength 8 -strlength 20
-alias  -descr {Effective position of pipe} -category 1 -atype 6 -alignment 8 -crc32 -1935685776 -link_
number 0 -dim 1 -arch_num 0 -readonly $FALSE -resetable $FALSE -def_type 0 -sub_type 0 -sync_type 1
```

Рисунок 7 – Окончание выборки глобала Attrs

В конце выборки выводятся атрибуты, относящиеся к пользовательским типам. В частности, для типа PIP они состоят из полей vold, a, b, cpl0, cpl1, epos0, epos1, epos и т.д.

Печать глобала S (рисунок 8) с системными сообщениями осуществляется одной из команд (используемые опции fld, oid, limit).

```
DsView –globtype S
DsView –globname Syslogs
DsView –global 4
```

```
23/04/25 08:58:20.799 INFO 2978# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpmodelMain hmo:2 loaded
 for object 'NPM'
23/04/25 15:26:40.973 INFO 2979# DsMM 1@0 DsrmMMServ.Run ==================== RESTART ================
===
23/04/25 15:26:40.974 INFO 2980# DsMM 1@0 DsrmMMServ.RestartUpdater DsMM task is updating periods=0 50
23/04/25 15:26:44.066 INFO 2981# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpgcValves hmo:1 loaded
for object 'VLV'
23/04/25 15:26:44.067 INFO 2982# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpgcCNValves hmo:1 loade
d for object 'CNV'
23/04/25 15:26:44.068 INFO 2983# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpgcPumps hmo:1 loaded f
or object 'PMP'
23/04/25 15:26:44.068 INFO 2984# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpgcBlks hmo:1 loaded fo
r object 'BLK'
23/04/25 15:26:44.069 INFO 2985# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpgcAvrs hmo:1 loaded fo
r object 'AVR'
23/04/25 15:26:44.069 WARNING 2986# DsMM 1@0 DsmodelHandlers.CheckInitObj init config type: VLV, obj:V9
, rc:-1
23/04/25 15:26:44.069 WARNING 2987# DsMM 1@0 DsmodelHandlers.CheckInitObj init config type: VLV, obj:Vl
0, rc:-1
23/04/25 15:26:44.069 WARNING 2988# DsMM 1@0 DsmodelHandlers.CheckInitObj init config type: VLV, obj:Vl
1, rc:-1
23/04/25 15:26:44.070 INFO 2989# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpgcNodes hmo:1 loaded f
or object 'NOD'
23/04/25 15:26:44.071 INFO 2990# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpgcPipes hmo:1 loaded f
or object 'PIP'
23/04/25 15:26:44.076 INFO 2991# DsMM 1@0 DsrmModelInT.AddOneOvh value handler DpmodelMain hmo:2 loaded
 for object 'NPM'
```

Рисунок 8 – Стандартная выборка системных событий

Печать системных событий разделяется на сессии. Каждая сессия начинается с ссобщения === START === или === RESTART ===.

Печать системных сообщений Syslogs выдает все системные события. Их можно отфильтровать по коду важности (0-нет, 1-список, 2-доп, 3-инфо, 4-предупреждение, 5-ошибка, 6-фатально), начиная с 4 – оставив только ошибки и фатальные ошибки.

```
DsView -globname Syslogs -fld 4
```

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -globname Syslogs -fld 4
S 4 Syslogs n_total_objs= 30000, n_used_objs= 6, sz_used_mem= 6480000, usr_data= 2991
21/04/25 08:13:52.354 ERROR 2941# DsMM 1@0 DsrmLUtils.in_groups wrong in_groups item='DVP'
21/04/25 08:13:52.355 ERROR 2943# DsMM 1@0 DsrmLUtils.in_groups wrong in_groups item='DVP'
23/04/25 08:54:55.135 FATAL 2961# DsMM 1@0 DsrmMMServ.CheckLicense SCADI License error file=Dsm_.lic re
s=-1()
23/04/25 08:54:56.882 FATAL 2963# DsMM 1@0 DsrmMMServ.CheckLicense SCADI License error file=Dsm_.lic re
s=-1()
23/04/25 08:57:13.776 FATAL 2965# DsMM 1@0 DsrmMMServ.CheckLicense SCADI License error file=Dsm_.lic re
s=-1()
```

Рисунок 9 – Список системных ошибок

Далее идут глобалы типа «s» - системные реляционные таблицы. Системные глобалы выводят в табличном виде (используемые опции fld, fld_page, oid, limit). Например, глобал Consts содержит список целочисленных констант, рисунок 10.

```
DsView -globname Consts
```

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -globname Consts
s 5 Consts n_total_objs= 420, n_used_objs= 210, sz_used_mem= 26880, usr_data= 17
|#   |ncadr       |name                          |proflag|allofla|host_lo|fsm_sta|dst_mask     |val
       |vdef        |vmin        |vmax        |
|0   |0           |host_id                       |0      |0      |0      |0      |0            |0
       |-1          |-1          |-1          |
|1   |0           |n_db                          |0      |0      |0      |0      |0            |7
       |-1          |-1          |-1          |
|2   |0           |ldr_task_msgs                 |0      |0      |0      |0      |0            |5000
       |-1          |-1          |-1          |
|3   |0           |ldr_cells_in_set              |0      |0      |0      |0      |0            |60
       |-1          |-1          |-1          |
|4   |0           |ldr_cells_small_set           |0      |0      |0      |0      |0            |1
       |-1          |-1          |-1          |
|5   |0           |ldr_sets_per_host             |0      |0      |0      |0      |0            |2
       |-1          |-1          |-1          |
|6   |0           |ldr_repdata_per_report        |0      |0      |0      |0      |0            |300
       |-1          |-1          |-1          |
|7   |0           |ldr_wdatas_in_set             |0      |0      |0      |0      |0            |40
       |-1          |-1          |-1          |
|8   |0           |ldr_wdatas_small_set          |0      |0      |0      |0      |0            |40
       |-1          |-1          |-1          |
|9   |0           |ldr_vars_in_file              |0      |0      |0      |0      |0            |100
       |-1          |-1          |-1          |
|10  |0           |ldr_tvars_in_file             |0      |0      |0      |0      |0            |200
       |-1          |-1          |-1          |
|11  |0           |ldr_n_tfiles                  |0      |0      |0      |0      |0            |50
       |-1          |-1          |-1          |
|12  |0           |ldr_alarmsets_per_host        |0      |0      |0      |0      |0            |3
       |-1          |-1          |-1          |
|13  |0           |ldr_alarms_in_set             |0      |0      |0      |0      |0            |200
       |-1          |-1          |-1          |
```

Рисунок 10 – Фрагмент списка целочисленных констант глобала Consts

Опция limit служит для ограничения числа строк. Ограничение числа столбцов использует явное их задание в опции fld.

```
DsView -globname Consts -limit 5 -fld "'name val vdef'" -fld_page 3
```

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -globname Consts -limit 5 -fld "'name val vdef'" -fld_page 3
s 5 Consts n_total_objs= 420, n_used_objs= 210, sz_used_mem= 26880, usr_data= 17
|#   |name                          |val         |vdef        |
|0   |host_id                       |0           |-1          |
|1   |n_db                          |7           |-1          |
|2   |ldr_task_msgs                 |5000        |-1          |
|#   |name                          |val         |vdef        |
|3   |ldr_cells_in_set              |60          |-1          |
|4   |ldr_cells_small_set           |1           |-1          |
```

Рисунок 11 – Ограничители строк и столбцов

Приведенная команда со списком “name val vdef” выводит только указанные поля. Параметр fld_page задает интервал печати заголовка: через каждые 3 записи будут выводиться имена столбцов.

Другие системные глобалы s выводятся аналогично.

Состояние задач выводится в глобале Tasks, который использует перечисления при отображении поля task_status. При выводе на консоль числовым значениям перечислений соответствуют статусы 0-OFF 1-INIT 2-LOAD 3-ERROR 4-END 5-WAITING 6-STARTING 7-SINGLE 8-PRIMARY 9-BACKUP 10-FAILOVER 11-FREEZE. Примером является глобал Tasks, рисунок 12.

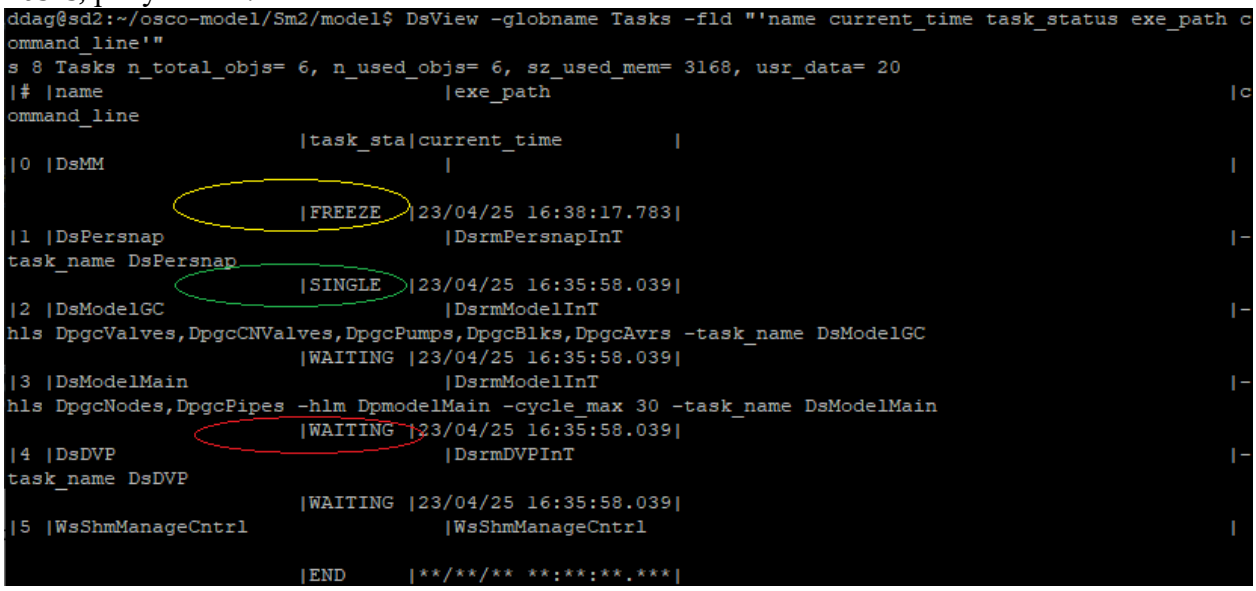


Рисунок 12 – Вывод списка задач Tasks

Состояние задачи в символьном виде отмечено на рисунке. Остановленное приложение DsMM имеет статус FREEZE, отмеченный желтым. Работающая задача DsPersnap имеет зеленый статус SINGLE и обрабатывает поступающие команды run, freeze, snap, reset. Не стартовавшая задача DsModelMain, имеющая статус WAITING, отмечена красным. Глобал показывает состояние задач, временные характеристики, коды ошибок, информация о полученных и переданных данных.

Аналогичную таблицу можно получить по хосту (global 10 Hosts) с выводом статусов и характеристик, рисунок 13.

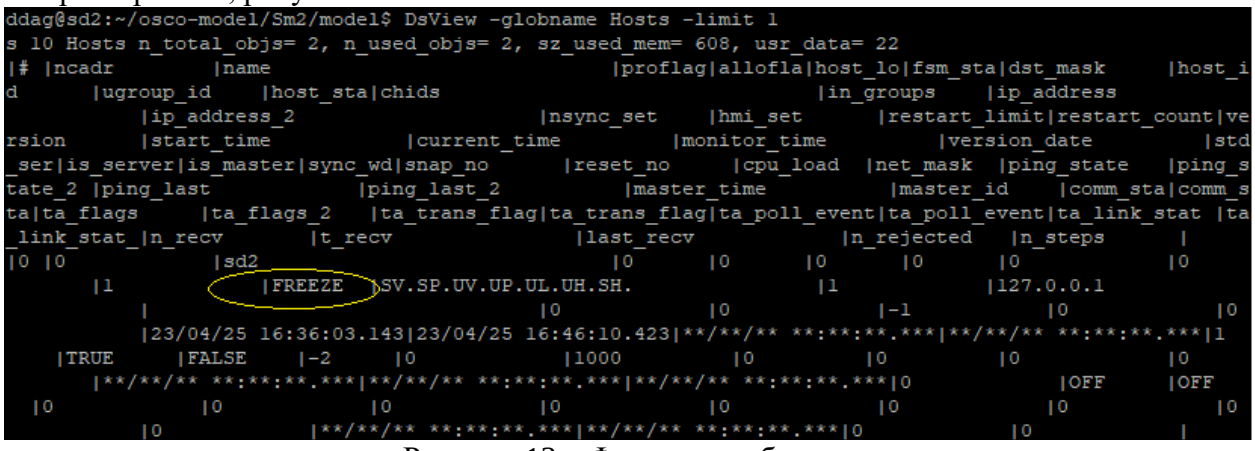


Рисунок 13 – Фрагмент таблицы хостов

3.2.3 DsView Просмотр пользовательских реляционных таблиц

Пользовательские реляционные таблицы образуются как массивы объектов определенных типов. Встроенные типы представляют собой:

- AI – аналоговые входные;
- DI – дискретные входные;

- NI – целочисленные диагностические входные;
- AC – аналоговые расчетные;
- DC – дискретные расчетные.

Объекты из данных типов имеют предопределенный набор атрибутов, который может быть расширен в конфигурации types.tsd. Пример расширения в конфигурации приведен ниже.

```
Type AI -descr {Analog Input}
Attr location -descr {Sensor design position} -atype STR -sync_type parm -dim
3
#
Type DI -descr {Discrete Input}
Attr tref -descr {TPTS reference data} -atype VREF -def_type ptr -sub_type STR
-sync_type parm
#
Type NI -descr {Numeric Input}
#
```

Пользовательские объекты AI, DI расширяются специфическими для проекта атрибутами location и tref соответственно. Объект NI используется в проекте в исходном виде.

Пользовательские объекты появляются под номерами в таблице типов. Например, объект AI будет иметь номер 40 (см. рисунок 5), если в таблице Types содержится запись следующего вида.

```
Type 40 AI -descr {Analog Input} -size 368 -alignment 8 -var 23 -stdotype 0 -
crc16 30863 -first 456 -gfirst 28 -default_attr 466 -lckproc_attr 0 -tmp_attr
470
```

Аналогично исходные и добавленные атрибуты должны быть в соответствующей таблице (см. рисунок 12).

```
…
Attr 457 lf -type AI -otype 40 -attr_number 1 -next 458 -offset 288 -size 4 -
tlength 4 -strlength 16 -alias  -descr {full range low limit} -category 4 -
atype 5 -alignment 4 -crc32 -1838801095 -link_number 0 -dim 1 -arch_num 0 -
readonly $FALSE -resetable $FALSE -def_type 4 -sub_type 1 -sync_type 2
Attr 458 hf -type AI -otype 40 -attr_number 2 -next 459 -offset 292 -size 4 -
tlength 4 -strlength 16 -alias  -descr {full range high limit} -category 4 -
atype 5 -alignment 4 -crc32 -167060931 -link_number 0 -dim 1 -arch_num 0 -
readonly $FALSE -resetable $FALSE -def_type 4 -sub_type 2 -sync_type 2
```

Пользовательские реляционные таблицы при просмотре используют эту мета-информацию о типах и атрибутах, рисунок 14.

```
DsView -globname AIs -oid 1534 -limit 3 -fld "'name code lf hf lw hw'"
```

Типы, определенные пользователем, должны иметь тип и определенный список полей.

```
ddag@sd2:~/osco-model/Sm2/model$ DsView -globname AIs -oid 1534 -limit 3 -fld "'name code lf hf lw hw'"
n 28 AIs n_total_objs= 15000, n_used_objs= 14916, sz_used_mem= 5520000, usr_data= 40
|#     |name                          |code                          |lf              |hf
    |lw              |hw              |
|1534  |1AI001535                       |40YDR20CF001_XQ01             |0.0             |1.6000000238
418 |0.6999999880790 |~               |
|1535  |1AI001536                       |40YDR20CF004_XQ01             |0.0             |1.6000000238
418 |~               |1.0             |
|1536  |1AI001537                       |40YDR20CP001_XQ01             |0.0             |250.0
    |75.0            |175.0           |
```

Рисунок 14 – Просмотр таблицы AIs

Глобал AIs представляет собой массив из пользовательских объектов AI, поле глобала usr_data=40 соответствует номеру объекта AI (отмечено красным). В состав полей объекта глобала входят предопределенные поля lf hf (отмечено желтым).

Аналогично можно просмотреть структуру DI, рисунок 15.

```
root@model-rn80-dock1:/home/osco# DsView -globname DIs -oid 054239 -limit 3 -fld "'name code dtype dcmd d
v'"
n 30 DIs n_total_objs= 150000, n_used_objs= 148707, sz_used_mem= 50400000, usr_data= 41
|#      |name                          |code                          |dv              |dcmd
   |d|
|54239  |1DI054240                     |40RNR81AS801_OU               |0               |c3@38:1,2
   | |
|54240  |1DI054241                     |40RNR81AS804_F0               |0               |0@0:0,0
   | |
|54241  |1DI054242                     |40RNR81AS804_F1               |0               |0@0:0,0
   | |
```

Рисунок 15 – Просмотр таблицы DIs

Глобал DIs представляет собой массив из пользовательских объектов DI, поле глобала usr_data=41 соответствует номеру объекта DI (отмечено красным).

В распечатке глобала DI есть поля dcmd, содержащие значение команды управления (отмечено зеленым). Значение поля dcmd представляет собой структуру и, следовательно, выводится в текстовом формате в виде, принятом для командных структур (таблица 6-1 приложения 6). Значение dcmd=c3@38:1,2 означает, что была подана команда управления пользователем №3 и программой №38. При этом значение команды равно 2 (соответствует «Открыть Задвижку»), а статус команды равен 1=PUT, что означает отсутствие передачи на сервер связи с нижним уровнем. При последующей обработки модель воспринимает команду в переменной DI:~40RNR81AS801_OU.dcmd.

3.2.4 DsView Просмотр очередей сообщений

Очереди сообщений в СКАДИ-МОДЕЛЬ являются кольцевыми буферами из записей n_total_objs, где поле n_total_objs входит в заголовок глобала типа ‘q’. В качестве индекса в очереди используются поля переменного значения заголовка usr_data, n_used_objs. При записи данных кольцевой буфер постоянно переписывается usr_data обозначает позицию новой записи, а n_used_objs становится равной usr_data по окончанию такой записи. Печать состояния очереди исходит из того, что n_used_objs указывает на окончание записи. Для печати используется цикл с началом в n_used_objs и окончанием, когда увеличивающийся параметр вновь достиг n_used_objs. Это возможно, т.к. при достижении предела кольцевого буфера n_total_objs параметр снова сбрасывается в 0.

```
DsView -globname Controlq -limit 10
```

```
Seq 120 ncadr= 7 n_datas= 1 source_id=1@1^02^000 subscr_id=0^000033 flags={UPDATED VAL} srvstatus= 7 cloc
k=25/04/25 14:53:32.927 oldclock=**/**/** **:**:**.*** srvclock=25/04/25 14:53:32.927
Data 121 ref=#41,53884,472,0.13 'DI:1DI053885.dv' val='0' q= 48 ts=25/04/25 14:53:32.927 oldval='1' oldq=
 48 oldts=25/04/25 14:52:21.113
Seq 122 ncadr= 8 n_datas= 1 source_id=1@1^02^000 subscr_id=0^000033 flags={UPDATED VAL} srvstatus= 7 cloc
k=25/04/25 14:53:39.159 oldclock=**/**/** **:**:**.*** srvclock=25/04/25 14:53:39.159
Data 123 ref=#41,53884,472,0.13 'DI:1DI053885.dv' val='2' q= 48 ts=25/04/25 14:53:39.159 oldval='0' oldq=
 48 oldts=25/04/25 14:53:32.927
Seq 124 ncadr= 9 n_datas= 2 source_id=1@1^02^000 subscr_id=0^000034 flags={UPDATED VAL} srvstatus= 7 cloc
k=25/04/25 14:53:50.300 oldclock=**/**/** **:**:**.*** srvclock=25/04/25 14:53:50.300
Data 125 ref=#40,3431,466,0.15 'AI:1AI003432.av' val='5.0' q= 48 ts=25/04/25 14:53:50.300 oldval='0.0' ol
dq= 48 oldts=25/04/25 14:52:21.113
Data 126 ref=#41,54233,472,0.13 'DI:1DI054234.dv' val='2' q= 48 ts=25/04/25 14:53:50.300 oldval='0' oldq=
 48 oldts=25/04/25 14:52:21.113
Seq 127 ncadr= 10 n_datas= 1 source_id=1@1^02^000 subscr_id=0^000034 flags={UPDATED VAL} srvstatus= 7 clo
ck=25/04/25 14:53:51.349 oldclock=**/**/** **:**:**.*** srvclock=25/04/25 14:53:51.349
Data 128 ref=#40,3431,466,0.15 'AI:1AI003432.av' val='10.0' q= 48 ts=25/04/25 14:53:51.349 oldval='5.0' o
ldq= 48 oldts=25/04/25 14:53:50.300
```

Рисунок 16 – Печать очередей сообщений

Последовательности выводятся в виде строки заголовка Seq и строк данных Data. Описание последовательностей приведено в приложении 7.

На рисунке 16 видно, как одна очередь следует за другой. За очередью дискретных сигналов (Seq 120) следует очередь смешанных, включая аналоговых, (Seq 124) и снова дискретных сигналов (Seq 127).

Строки данных содержат поля ссылки на объект и значения. Строковое представление ссылки на объект описано после таблицы 6-3 приложения 6. Например, для ссылки #40,3431,466.15 можно определить тип объекта otype=40 или AI, идентификатор объекта oid=3431, атрибут объекта attrid=466 или av, а 15 соответствует типу значения atype=XT_R4QTS. Далее определяется имя объекта AI:1AI003432.av. Тип atype используется при форматировании значения для вывода в виде val=’10.0’ q=48 ts=25/04/2025 14:53:51.349.

3.2.5 DsView Специальные возможности просмотра

Специальные команды выводят на консоль следующую информацию:

- DsView -global -1 вывод размеров структур;
- DsView -global -2 вывод кодов ошибок;
- DsView -global -3 вывод типов глобалов.

3.2.6 DsAccess Функции доступа и конфигурирования

DsAccess для доступа и конфигурирования глобалов общей памяти.

Запуск приложения следующий.

```
DsAccess ?-option value? ..
```

Перечень опций приведен в таблице 8-1 приложения 8.

Для удобства наборы опций разделяются на команды.

Команда end для завершения всех процессов имеет следующий вид.

```
DsAccess –end all
```

Команда end устанавливает статус end для всех задач, приводя к их завершению.

Команда end для конкретного процесса (на примере DsModel) имеет следующий вид.

```
DsAccess –end DsModel
```

По команде приложение завершает свою работу, переменная статуса Task:DsModel.task_status устанавливается в значение END.

Команда init стартует завершившую работу задачу и имеет следующий вид (на примере DsModel).

```
DsAccess –init DsModel
```

Для рестарта задачи DsModel правильная последовательность команд такая: end, ждем статуса END 1секунду, init.

Команда getvars выводит на консоль значения переменных.

```
DsAccess -getvars "'Task:DsModel.task_status Task:DsMM.task_status
AI:1AI0011.av'"
Task:DsModel.task_status=END q=* ts=*
Task:DsMM.task_status=SINGLE q=* ts=*
AI:1AI0011.av=224.15742492675 q=GNSPEC ts=21/10/22 17:28:13.059
```

Список переменных обрамляется кавычками и разделяется пробелами. В некоторых консольных средах приходится заключать список в двойные и одинарные кавычки.

Команда setvars формирует очередь сообщений из новых значений переменных и вносит эти переменные в базу данных реального времени. Список значений и кодов качества записывается аналогично. При наличии одного значения можно использовать опцию –val.

```
DsAccess -setvars "'AI:1AI000011.av AI:1AI000012.av DI:1DI000002.dv'" -sync val
-vals "'101 102 1'" -quals "'51 51 49' –global 32
```

Команда формирует очередь сообщений для глобала номер 32 (в н) с флагом VAL с установленными значениями и кодами качества. Печать изменившихся значений возможна через getvars.

```
DsAccess -getvars "'AI:1AI000011.av AI:1AI000012.av DI:1DI000002.dv'"
```

Изменившиеся значения доступны с кодами качества и метками времени, рисунок 17.

```
ddag@sd2:~/osco-model/Sm2/model$ DsAccess -getvars "'AI:1AI000011.av AI:1AI000012.av DI:1DI000002.dv'"
AI:1AI000011.av=101.0 q=GLOLO ts=28/04/25 15:27:08.299
AI:1AI000012.av=102.0 q=GLOLO ts=28/04/25 15:27:10.811
DI:1DI000002.dv=1 q=GALM ts=28/04/25 15:27:13.260
```

Рисунок 17 – Значения записанных изменений

При невозможности доступа к значениям параметров на запись выдается сообщение об ошибке.

```
ddag@sd2:~/osco-model/Sm2/model$ DsAccess -setvars "'AI:1AI000011.av AI:1AI000012.av DI:1DI000002.dv'" -sync val -vals "'101 102 1'" -quals "'51 51 49'" -global 32
* unable to write: AI:1AI000011.av
* unable to write: AI:1AI000012.av
* unable to write: DI:1DI000002.dv
```

Для того, чтобы таких сообщений не было, должны быть установки в конфигурации загрузки, разрешающие обновление переменных из утилиты DsAccess.

При этом доступ к данным по чтению сохраняется.

Команда modify запускает пакет обновления параметров базы данных реального времени.

```
DsAccess -modify mod_almpri_ai
```

Введенная команда запускает модификацию по содержимому файла обновлений db/<modfile>.tsd. В нашем случае файл будет db/mod_almpri_ai.tsd.

Пакет обновления параметров представляет собой текстовый файл следующего формата.

```
Mod <name> -option value …
```

Имя name должно однозначно определять экземпляр объекта в таблице. Предлагаемые обновления имеют следующий вид.

```
Mod AI:1AI000001 -almpri 2 -hf 350

Mod AI:1AI000002 -almpri 1 -hf 350

Mod AI:1AI000003 -almpri 4

Mod AI:1AI000004 -almpri 3
```

Пакет обновления формирует очередь обновлений параметров в базе данных реального времени.

При отсутствии или некорректности файла обновлений выдаются сообщения об ошибках.

Команда control формирует последовательность из команд управления для базы данных реального времени. Команды управления выглядят следующим образом.

```
DsAccess -control "'DI:1DI054678.dcmd DI:1DI054688.dcmd'" -vals "'1 2'" -global 43
```

Поле control содержит имена команд конфигурации и управления. Поле vals содержит значения команд конфигурации и управления. Сформированные команды конфигурации и управления аналогично записываются в очередь сообщений, рисунок 18.

```
Seq 7 ncadr= 5 n_datas= 1 source_id=1@1^99^000 subscr_id=0^000000 flags={UPDATED CMD} srvstatus= 11 clock
=28/04/25 16:42:23.236 oldclock=**/**/** **:**:**.*** srvclock=28/04/25 16:42:23.236
Data 8 ref=#41,54677,474.16i99 'DI:1DI054678.dcmd' val='c1@0:1,2' ts=28/04/25 15:50:03.293 oldval='c3@86:
1,2' oldts=28/04/25 15:50:03.293
Seq 9 ncadr= 6 n_datas= 2 source_id=1@1^99^000 subscr_id=0^000000 flags={UPDATED CMD} srvstatus= 11 clock
=28/04/25 16:45:07.542 oldclock=**/**/** **:**:**.*** srvclock=28/04/25 16:45:07.542
Data 10 ref=#41,54677,474.16i99 'DI:1DI054678.dcmd' val='c1@0:1,1' ts=28/04/25 15:50:03.293 oldval='c1@0:
1,2' oldts=28/04/25 15:50:03.293
Data 11 ref=#41,54687,474.16i99 'DI:1DI054688.dcmd' val='c1@0:1,2' ts=**/**/** **:**:**.*** oldval='0@0:0
,0' oldts=**/**/** **:**:**.***
```

Рисунок 18 – Очередь команд конфигурации и управления

Последовательность готова для синхронизации в системе. Флаг CMD указывает на команды конфигурирования и управления в последовательности. Флаг UPDATED указывает на то, что значения уже обновлены в глобалах.

3.2.7 Проверка прохождения сигнала через систему СКОЯТ

Прохождение сигнала через систему СКОЯТ будем отслеживать на примере управления запорной арматурой. Окно управления запорной арматурой представлено на рисунке 19. Мнемосхема отображает состояние арматуры. В закрытом состоянии оба сектора мнемосхемы – зеленые, в открытом состоянии сектора желтые, в промежуточном состоянии один сектор зеленый, а другой желтый. Кнопками Откр и Закр арматура переводится из закрытого состояния в открытое и наоборот.

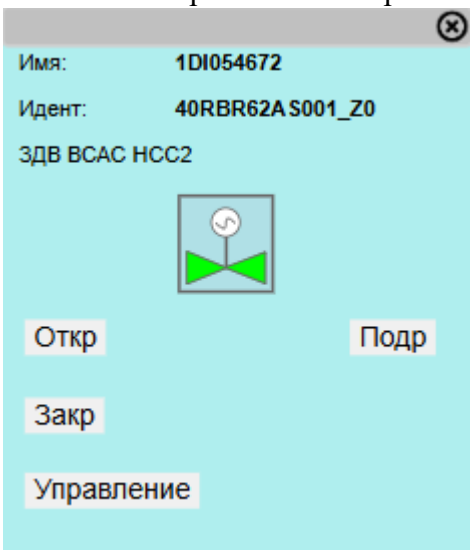


Рисунок 19 – Окно управления запорной арматурой

При некорректной работе требуется проверка прохождения сигнала через систему, состоящая из этапов, описанных ниже.

Этап 1 – проверка наличия сервиса СВУ. Окно управления возникает при наличии загруженной схемы и контейнера с моделью. Для проверки наличия сервиса СВУ надо зайти в контейнер, например model-rb60-dock1 и проверить процесс WsShmCntrl или WsManageCntrl для порта 12020.

```
user@osco:~/osco$ docker exec -it model-rb60-dock1 bash
root@model-rb60-dock1:/# ps x
…
     13 pts/0    Sl+    0:08 /usr/local/scadiweb/sbinur/WsShmCntrl -port 12020
-add_port username -interval 100ms -lic_hostname rb60-image
…
```

Наличие загруженного сервиса говорит о наличии программного обеспечения приема команд пользователя.

Этап 2 – Выдача команды в очередь сообщений. В первую очередь необходимо определить имя очереди сообщений – глобала q, ассоциированного с процессом Ws*Cntrl. Это делается вызовом DsView для всех глобалов. Вызов нужно сделать из проектного каталога (/home/osco в нашем примере).

```
cd /home/osco
root@model-rb60-dock1:/home/osco# DsView -globtype G | grep Cntrl
Global 43 -globtype q -globname Controlq -globsize 48024 -max_count 1000 -olength 0 -gnext 0 -persist_interval 0 -ref1 WsShmCntrl -used $TRUE -open $TRUE -t_modified 29/04/25 14:22:27.142 -task_modified 0 -flags {}
```

Обнаружен глобал Controlq. Далее следует инициировать выдачу команды нажатием кнопки Откр. В глобале Controlq должна появиться последовательность с командой управления.

```
root@model-rb60-dock1:/home/osco# DsView -globname Controlq
q 43 Controlq n_total_objs= 1000, n_used_objs= 6, sz_used_mem= 48000, usr_data= 6
Seq 4 ncadr= 3 n_datas= 1 source_id=3@1^04^000 subscr_id=0^000000 flags={CMD} srvstatus= 7 clock=29/04/25 15:30:50.368 oldclock=**/**/** **:**:**.*** srvclock=29/04/25 15:30:50.368
Data 5 ref=#41,54677,474.16 'DI:1DI054678.dcmd' val='c3@66:1,2' ts=29/04/25 15:30:50.368 oldval='0@0:0,0' oldts=**/**/** **:**:**.***
```

Значение команды управления val='c3@66:1,2' равно двум – команда открыть. Команда выдается на дискретный тэг 1DI054678. Данному тэгу соответствует технологическое имя 40RBR62AS001_OU, что проверяется следующим запросом по базе дискретов DI.

```
root@model-rb60-dock1:/home/osco# DsView -globname DIs | grep 1DI054678
|54677  |0            |1DI054678                        |7      |7      |0
|0        |0                |0                |0                |**/**/**
**:**:**.***|40RBR62AS001_OU                |                              |
!! !!2                                                                |101
|6            |0            |FALSE      |0        |**                  |**
|0/0/0/0:0>0:0>0 |0@0:0,0 |0              |1              |0      |c3@66:1,2
| |FALSE    |
```

Технологическое имя 40RBR62AS001_OU должно соответствовать имени 40RBR62AS001_Z0 в окне управления с точностью до суффикса. Суффикс OU означает команду управления, суффикс Z0 – состояние оборудования.

Если выдача команды не происходит, надо проверить состояние задачи СВУ запросом по глобалу Tasks.

```
root@model-rb60-dock1:/home/osco# DsView -globname Tasks -fld "'name task_status'"
s 8 Tasks n_total_objs= 6, n_used_objs= 6, sz_used_mem= 3168, usr_data= 20
|# |name                          |task_sta|
|0 |DsMM                          |FREEZE  |
|4 |WsShmCntrl                    |FREEZE  |
```

Например, интересующая нас задача WsShmCntrl может находиться в статусе FREEZE. Если статус задачи SINGLE, то должна формироваться команда управления.

Этап 3 – Прием команды в таблицу значений. Обновление команды по значениям в очередях циклических буферов сообщений реализует задача DsDVP. Корректное обновление должно подтверждаться текущими значениями сигналов.

```
root@model-rb60-dock1:/home/osco# DsAccess -getvars "'DI:~40RBR62AS001_OU.dcmd DI:~40RBR62AS001_OU.dv'"
DI:~40RBR62AS001_OU.dcmd=c3@66:1,2 q=* ts=29/04/25 15:30:50.368
DI:~40RBR62AS001_OU.dv=1 q=GNSPEC ts=29/04/25 15:30:44.156
```

Текущее значение в таблице должно быть равно последнему (в смысле метки времени) значению в очереди. Если это не происходит, необходимо проверить статус DsDVP.

```
s 8 Tasks n_total_objs= 6, n_used_objs= 6, sz_used_mem= 3168, usr_data= 20
|# |name                                |task_sta|
|0 |DsMM                                |SINGLE  |
|1 |DsPersnap                           |SINGLE  |
|2 |DsModelGC                           |SINGLE  |
|3 |DsModelMain                         |SINGLE  |
|4 |WsShmCntrl                          |SINGLE  |
|5 |DsDVP                               |SINGLE  |
```

Значения статуса не должны идентифицировать выгруженную задачу.

Этап 4 – Обработка команды в расчетной модели. В расчетной модели должна производиться обработка команды с выдачей значений в очередь сообщений, соответствующей расчетной модели. Для модели DsModelGC очередь сообщений устанавливается аналогично, это GCq.

При подаче команды управления DI:~40RBR62AS001_OU.dcmd в расчетной задаче формируется сигнал DI:~40RBR62AS001_Z0.dv. Причем этот сигнал проходит значения 0-промежуточное и 2-открытое. Очередь сообщений для указанного сигнала представлена ниже.

```
root@model-rb60-dock1:/home/osco# DsView -globname GCq
q 32 GCq n_total_objs= 2000, n_used_objs= 223, sz_used_mem= 96000, usr_data=
223
…
Seq 219 ncadr= 10 n_datas= 1 source_id=1@1^02^000 subscr_id=0^000033
flags={UPDATED VAL} srvstatus= 7 clock=29/04/25 15:30:51.300 oldclock=**/**/**
**:**:**.*** srvclock=29/04/25 15:30:51.300
Data 220 ref=#41,54671,472,0.13 'DI:1DI054672.dv' val='0' q= 48 ts=29/04/25
15:30:51.300 oldval='1' oldq= 48 oldts=29/04/25 15:30:42.279
Seq 221 ncadr= 11 n_datas= 1 source_id=1@1^02^000 subscr_id=0^000033
flags={UPDATED VAL} srvstatus= 7 clock=29/04/25 15:30:57.380 oldclock=**/**/**
**:**:**.*** srvclock=29/04/25 15:30:57.380
Data 222 ref=#41,54671,472,0.13 'DI:1DI054672.dv' val='2' q= 48 ts=29/04/25
15:30:57.380 oldval='0' oldq= 48 oldts=29/04/25 15:30:51.300
```

Аналогично по базе можно определить, что DI:1DI054672.dv соответствует технологическому имени DI:~40RBR62AS001_Z0.dv.

В отличие от команд управления выходные величины в расчетных моделях уже записываются в таблицы значений. Об этом свидетельствует флаг flags={UPDATED VAL}, поле UPDATED которого отмечает обновленные значения в таблице.

При невыдаче значений аналогично проверяется статус расчетной задачи.

Работают или нет расчетные задачи – это вопрос, на который можно ответить изменением размера очередей сообщений. Если параметр usr_data очередей сообщений не меняется, то данные не меняются вовсе. Это можно проверить следующим образом.

```
root@model-rb60-dock1:/home/osco# DsView -globtype . | grep "q "
q 32 GCq n_total_objs= 2000, n_used_objs= 146, sz_used_mem= 96000, usr_data=
146
q 39 Modelq n_total_objs= 2000, n_used_objs= 396, sz_used_mem= 96000, usr_data=
396
q 40 DVPq n_total_objs= 1000, n_used_objs= 0, sz_used_mem= 48000, usr_data= 0
q 43 Controlq n_total_objs= 1000, n_used_objs= 0, sz_used_mem= 48000, usr_data=
0
```

Нулевое значение предполагает, что очередь не менялась. При нормальной работе значения меняются. Застывшие ненулевые значения определяют статус FREEZE. Это нормальное состояние, при котором задачи работают, но расчеты не производятся.

Этап 5 – Выдача сигнала в СВУ. Текущие значения считываются сервисом Ws*Cntrl СВУ и направляются клиентскому приложению для обновления графических элементов. При нормальной работе значения аналоговых и дискретных сигналов меняются.

Если данные не меняются, а в очередях есть изменения, то проверяется статус задачи Ws*Cntrl. Рабочий статус – SINGLE.

3.2.8 DsISD интерактивный отладчик значений

DsISD служит для отслеживания изменения значений переменных в разделяемой памяти и задания этих значений. Для него используется одна опция командной строки – start_page – имя первоначально загружаемой страницы – набора отслеживаемых переменных (таблица 9-1 приложения 9). Если эта опция не задана, то начальный набор переменных является пустым.

Запуск приложения с пустым начальным набором переменных:

```
DsISD
```

В результате выполнения открывается консоль с пустым набором переменных:

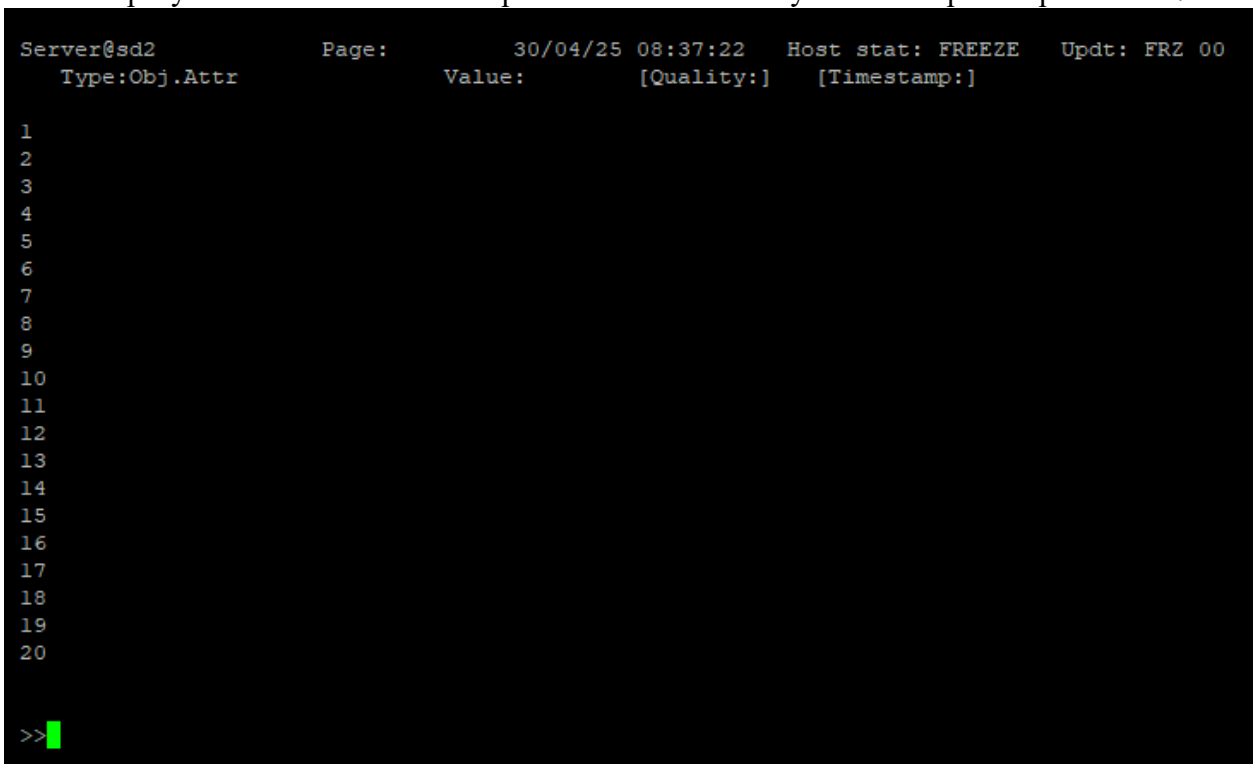


Рисунок 20 – вид консоли DsISD с пустым набором переменных

Приложение работает в символьном режиме в консоли и требует размер окна как минимум 26строк*80столбцов. При размере, меньшем требуемого, выдается сообщение.

```
ddag@sd2:~/osco-model/Sm2/model$ DsISD
Terminal window is too small. rows= 23 columns= 80
Should be: rows>= 26 columns>= 80
```

В заголовочной строке консоли DsISD выводится имя хоста, имя набора переменных, текущее время хоста, состояние хоста Host stat, состояние приложения DsISD Updt (RUN – значения обновляются, FRZ – обновление остановлено), номер начального состояния reset. На одной странице выводится до 20 переменных. Переменные пронумерованы от 1 до 20 по вертикали.

Запуск приложения с заданным начальным набором переменных:

```
DsISD -start_page avr
```

В результате открывается консоль с заданным набором переменных:

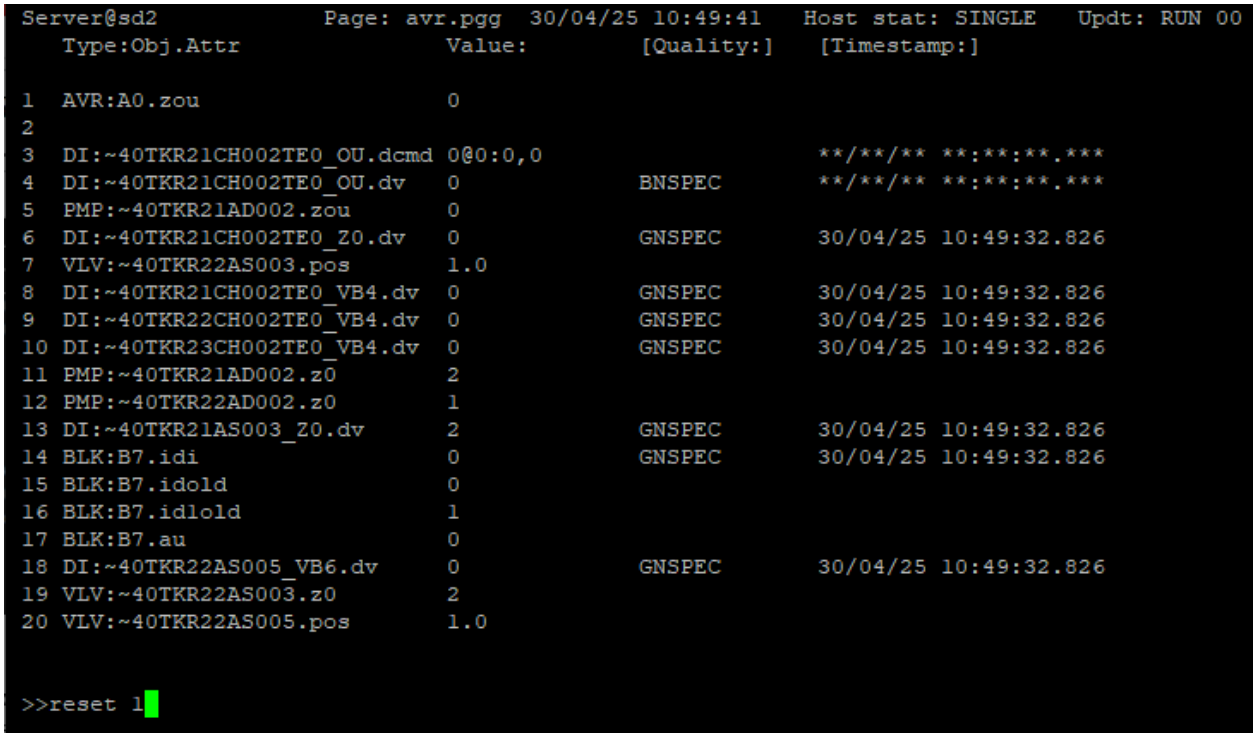


Рисунок 21 – вид консоли DsISD с заданным набором переменных

Для каждой переменной выводится ее имя, значение, признак качества при его наличии и метка времени при ее наличии.

В нижней части консоли DsISD расположена строка для ввода интерактивных команд. Список возможных команд представлен в таблице 9-2 приложения 9. На приведенном рисунке вводится команда reset 1.

Строка ввода вида N=V, где N – целое число, а V – строка, интерпретируется как задание переменной с номером N значения V. Любая строка, отличная от возможной команды и от задания значения переменной, интерпретируется как добавление новой переменной к текущему набору.

Например, строка PMP:~40TKR21AD002.z0 добавляет указанную переменную и помещает ее в свободную ячейку. Строка с символом /<номер> содержит номер ячейки. Строка PMP:~40TKR21AD002.z0/6 помещает переменную в требуемую ячейку.

Типовая последовательность команд при автономной отладки управления насосом будет следующая. Загрузить начальное состояние 1 командой.

```
>reset 1
```

Убедиться, что команда управления и состояние насоса включены в список. В противном случае необходимо их включить.

```
>PMP:~40TKR21AD002.zou/5
>PMP:~40TKR21AD002.z0/11
```

Проконтролировать, что после выполнения reset система переходит в статус FREEZE. Установить переменной PMP:~40TKR21AD002.zou из ячейки 5 значение 1, соответствующей команде управления Откл.

```
>5=1
```

Сделать 1 шаг вычислений командой step или step 1. Проконтролировать, что после выполнения шага значение состояния насоса PMP:~40TKR21AD002.z0 в ячейке 11 перешло с 2 в 1.

```
>run
```

Если все произошло, запускается расчет командой run. Если загруженная страница изменилась, надо ее сохранить командой save <file>. Команда без указания имени сохраняет в тот же файл.

```
>save avr_new
```

Сохранение осуществляется в текстовый файл wrk/avr.pgg.

Завершить работу отладчика командой q. Команда end завершает работу как отладчика, так и рантайм-менеджера.

```
>end
```

3.2.9 DsQSD интерактивный отладчик очередей

DsQSD служит для отслеживания изменения значений переменных в глобальных и локальных очередях сообщений.

Для отслеживания изменений необходимо задать от одной до четырех глобальных очередей сообщений и (или) от одной до четырех локальных очередей сообщений. Также в командной строке можно задать имена от одной до трех отслеживаемых переменных. Для отслеживания большего числа переменных их можно добавить с помощью интерактивных команд. Опции командной строки представлены в таблице 10-1 приложения 10.

Например, для отслеживания в очередях сообщений изменений состояния задвижки 40RBR62AS003 при управлении необходим мониторинг переменных DI:~40RBR62AS003_OU.dcmd DI:~40RBR62AS003_OU.dv, DI:~40RBR62AS003_Z0.dv. Предлагается выполнить следующую команду для трех точек:

```
root@model-rb60-dock1:/home/osco# DsQSD -varname1 DI:~40RBR62AS003_OU.dcmd -
varname2 DI:~40RBR62AS003_OU.dv -varname3 DI:~40RBR62AS003_Z0.dv
```

В результате выполнения открывается окно-консоль:

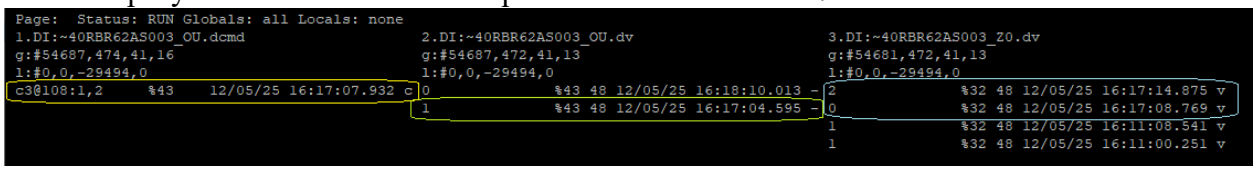


Рисунок 22. Управление оборудованием по консоли DsQSD

В заголовочной строке консоли DsQSD выводится имя текущего набора переменных (если он задан), состояние приложения DsQSD (RUN – значения обновляются, FRZ – обновление остановлено), имена глобальных и локальных очередей сообщений. Изменяющиеся значения выводятся в виде трех столбцов. В заголовке каждого столбца выводится имя переменной и ее идентификаторы. Идентификаторы выводятся в формате строкового представления структуры VariableRef согласно Приложению 6. Для каждого изменения значения выводится значение, признак качества (в виде целого числа, при его наличии), метка времени при ее наличии, флаг типа записи в очереди (приложение 7, таблица 7-3).

При использовании DsQSD для контроля прохождения команд управления задвижкой 40RBR62AS003 наблюдаются несколько изменений с временными метками. Во-первых, при активации окна управления (см. рисунок 19) устанавливается ненулевое значение переменной DI:~40RBR62AS003_OU.dv, связанной с командой управления (отмечено зеленым цветом посередине). Это означает, что сигнал выбора объекта в управления подан. Далее при нажатии кнопки открыть сформировался сигнал открытия в виде структуры C8B с флагом ‘c’ (отмечен желтым цветом для DI:~40RBR62AS003_OU.dcmd слева). По команде логика управления начинает открывать запорную арматуру, переводя сначала ее состояние DI:~40RBR62AS003_Z0.dv в ПРОМЕЖУТОЧНОЕ=0, а затем в ОТКРЫТО=2. Оба значения состояний выделены голубым цветом в правой части.

По завершению управления пользователь убирает окно управления, и сигнал выбора объекта в управление снимается DI:~40RBR62AS003_OU.dv=0. На рисунке это значение не помечено и находится в верхней средней части.

В правом нижнем углу консоли DsQSD расположена строка для ввода интерактивных команд.

Ввод строки, отличной от возможных команд, рассматривается как добавление новой переменной к текущему набору. Аналогичный набор переменных в DsQSD получается вводом вышеупомянутых имен.

```
>DI:~40RBR62AS003_OU.dcmd
>DI:~40RBR62AS003_OU.dv
>DI:~40RBR62AS003_Z0.dv
```

3.3 СРЕДСТВА РАЗРАБОТКИ ВИДЕОКАДРОВ СВУ

3.3.1 Организация файлов видеокадров

Процесс разработки видеокадров предполагает модификацию видеокадров редактором DsEdit с выходом в виде файлов *.bgd и последующую генерацию разработанных файлов в формат svg. Среда разработки должна быть сконфигурирована согласно рисунку 23.

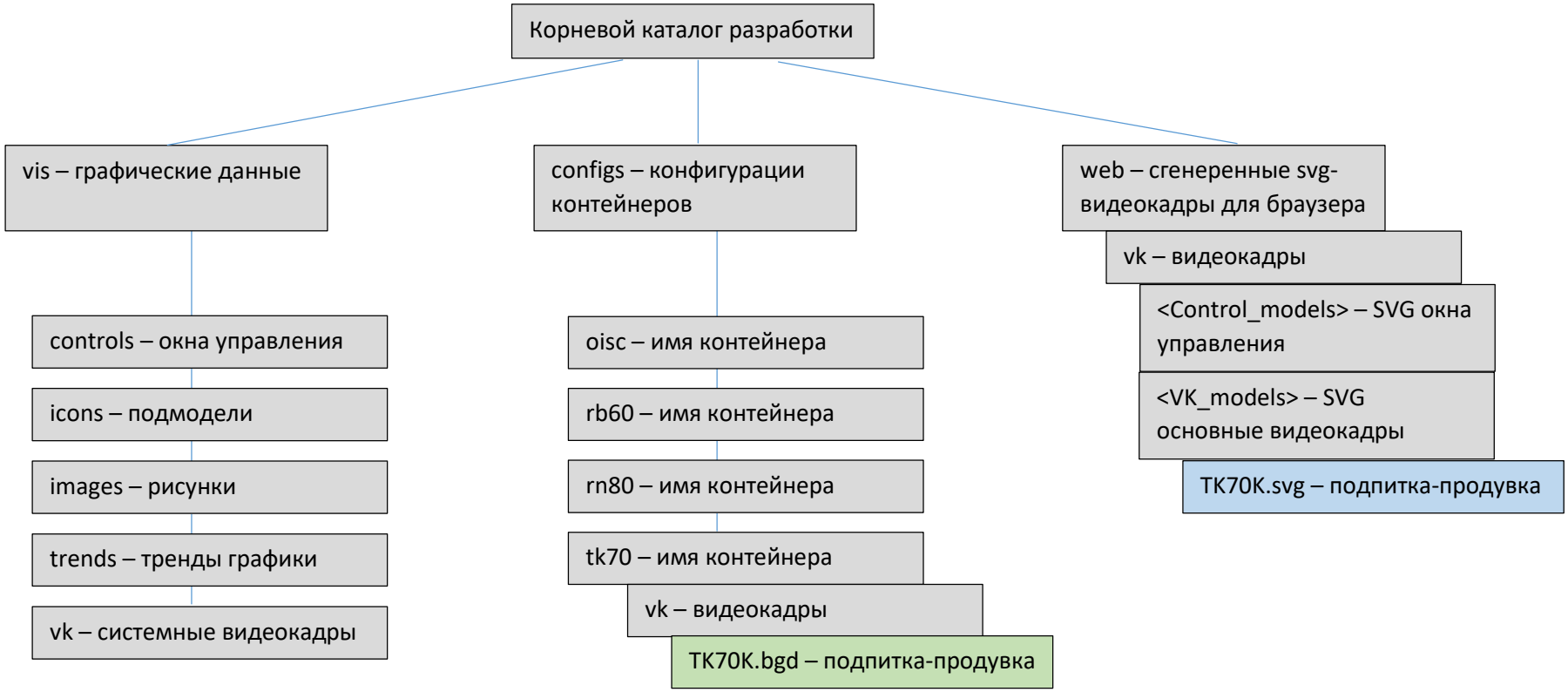


Рисунок 23. Структура каталогов среды разработки

Проект должен иметь корневой каталог разработки с разделом графических данных vis, где установлены используемые в проекте окна управления, подмодели, рисунки, графики и системные видеокадры.

Проект содержит конфигурации для всех используемых в нем контейнеров с моделями процессов. В каждой конфигурации контейнера располагается каталог видеокадров vk, содержащий файлы, используемые для данного контейнера.

На рисунке приведен в качестве примера отмеченный зеленым видеокадр TK70K.bgd для контейнера tk70. В процессе разработки видеокадр TK70K.bgd меняется и сохраняется в папке vk для контейнера tk70.

Для отображения в браузере видеокадр TK70K.bgd конвертируется в файл TK70K.svg и копируется в каталог основных видеокадров для web, полный путь web/vk/<VK_models>/. В качестве VK_models подставляется проектно-зависимый каталог с основными видеокадрами. При создании образа контейнера видеокадры, относящиеся к контейнеру tk70, копируются в образ для tk70. Впоследствии из образа создаются и запускаются контейнеры. В частности, запустится контейнер model-tk70-dockNN, содержащий в себе web/vk/<VK_models>/TK70K.svg.

При генерации в файл svg вставляются подмодели из каталога vis графических данных.

Помимо основных видеокадров в svg формат конвертируются окна управления и графики. Окна управления берутся из графического раздела vis/controls. При запуске окна управления параметризуются KKS-именами конкретного оборудования, которое переводится в состояние управления. Графики получают на вход наборы идентификаторов сигналов для мониторинга. Наборы идентификаторов прописываются в учебно-методических материалах на задание.

3.3.2 DsEdit Редактирование датчиков аналоговых сигналов

DsEdit представляет собой редактор видеокадров ПО СКАДИ. Подробное описание редактора представлено в документе 34185873.425510.001.34.01.М «Программная платформа АСУ ТП и диагностики. Человеко-машинный интерфейс. Руководство оператора». Для запуска наберите DsEdit.

Датчики аналоговых сигналов служат для отображения значений и кодов качества объектов AI. Добавление датчика аналогового сигнала осуществляется через палитру подмоделей.

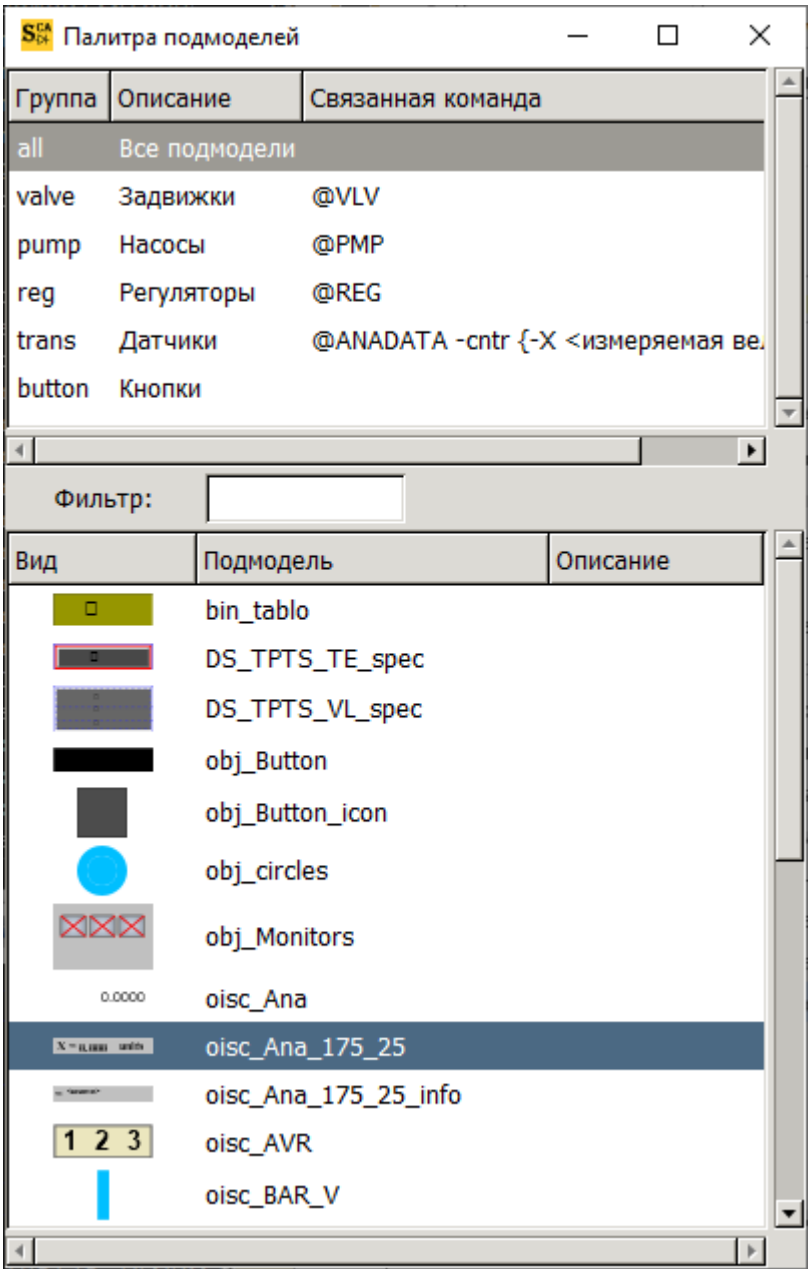


Рисунок 24. Выбор аналогового сигнала в палитре подмоделей

В палитре необходимо выбрать одну из подмоделей аналоговых датчиков, например, oisc_Ana_175_25. Экземпляр датчика на основе выбранной подмодели помещается на текущий видеокадр.

Вставленный на видеокадр датчик остается выбранным. Позиционируйте датчик согласно отображаемому на видеокадре расположению.

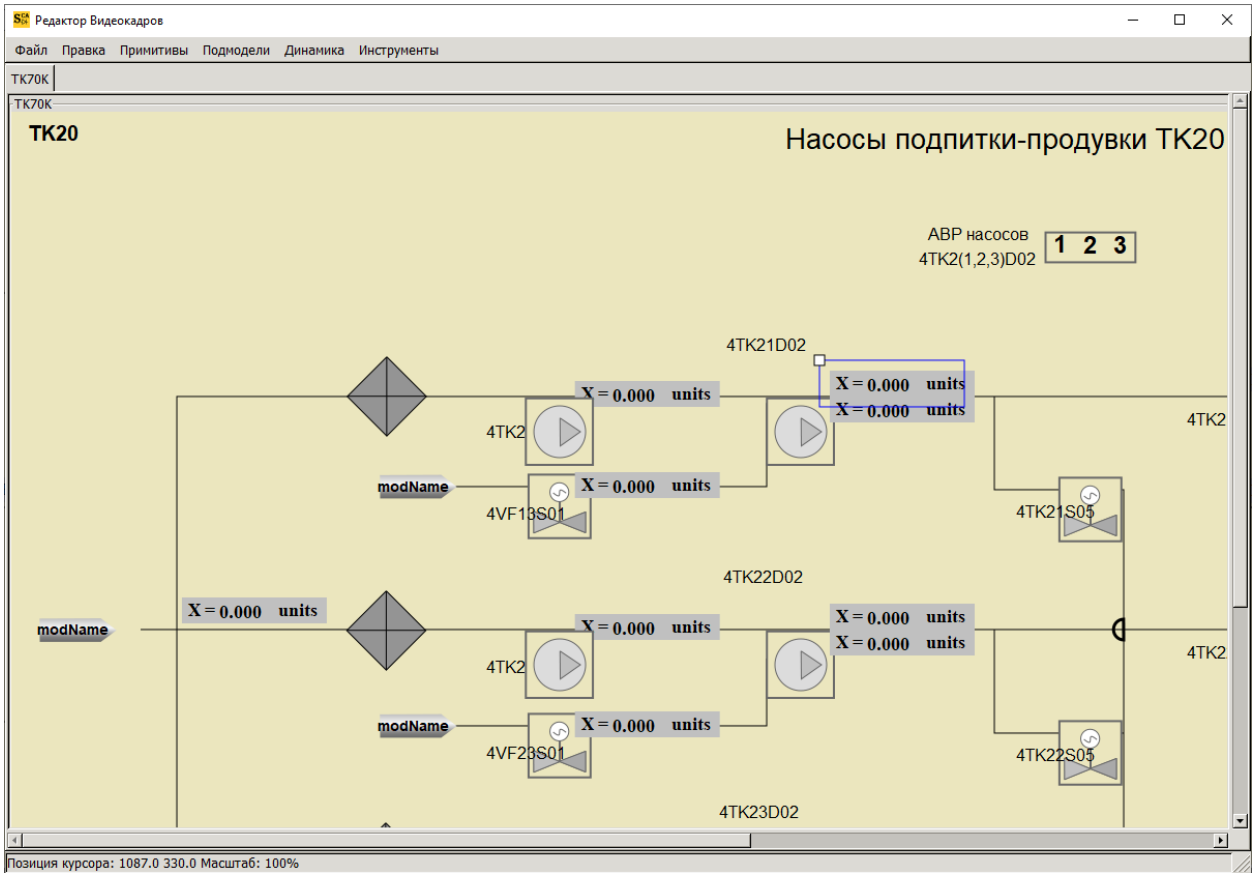


Рисунок 25. Расположение датчика на видеокадре.

Расположенный на видеокадре датчик требует привязки к технологическим параметрам. Если привязка отсутствует, то датчик отображается на видеокадре голубым цветом и нулевыми значениями как непривязанный.

Для привязки необходимо выбранному датчику нажать правой кнопкой мыши для получения окна параметров подмодели, рисунок 26. Часть параметров подмодели задается по умолчанию. Но параметры технологической привязки, отмеченные зеленым, требуют определения.

Параметр id-code требует задания технологического имени – параметра code из глобала AIs базы данных. Необходимо ввести значение, соответствующее коду датчика с суффиксом из глобала аналоговых сигналов AIs. Значение 40TKR21CF902_XQ01 соответствует записи в глобале аналоговых сигналов.

```
Obj 251 -name 1AI000252 -code 40TKR21CF902_XQ01 -descr {F НАП ПН1 0-80} -units
М3/Ч -lf 0 -hf 80 -la 8 -ha 78 -lw ~ -hw ~ -lr ~ -hr ~ -almpri 1029 -dst_mask
1 -xarea_id 101 -proflags 31 -alloflags 31 -is_work TRUE -fun_type 0 -zone 0.0
-ndigits 0
```

Параметр type-name требует задания типа. В нашем случае почти всегда AI, но могут быть другие типы, например, AC – аналоговые расчетные величины.

Параметр type-cntr записывается в виде –имя подстановка. В данном случае, вместо имени X= в редакторе осуществляется подстановка на F= в браузере. Один и тот же датчик может отображать значения расходов (F=), температур (T=), давлений (P=) и т.д.

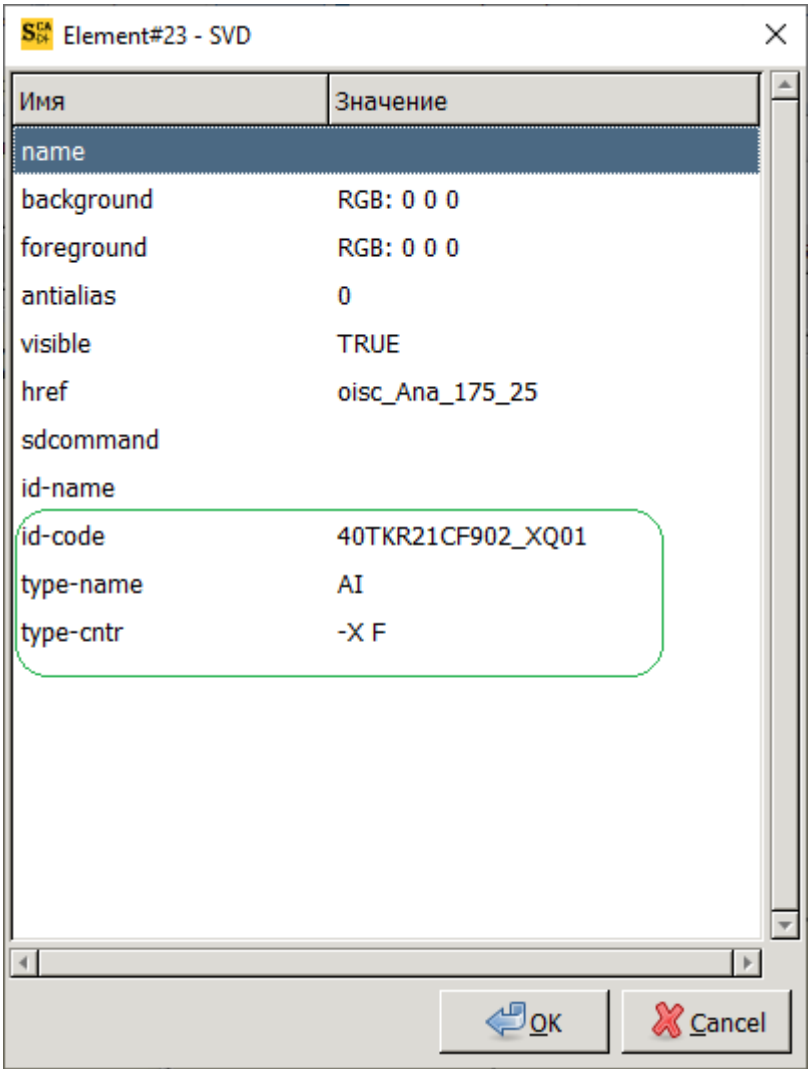


Рисунок 26. Окно параметров подмодели датчика.

После редактирования параметров и сохранения файла датчик становится привязан к аналоговому сигналу с указанным технологическим кодом. При отображении в браузере привязанного датчика появляется параметризация и динамика F = 41.294 М3/Ч.

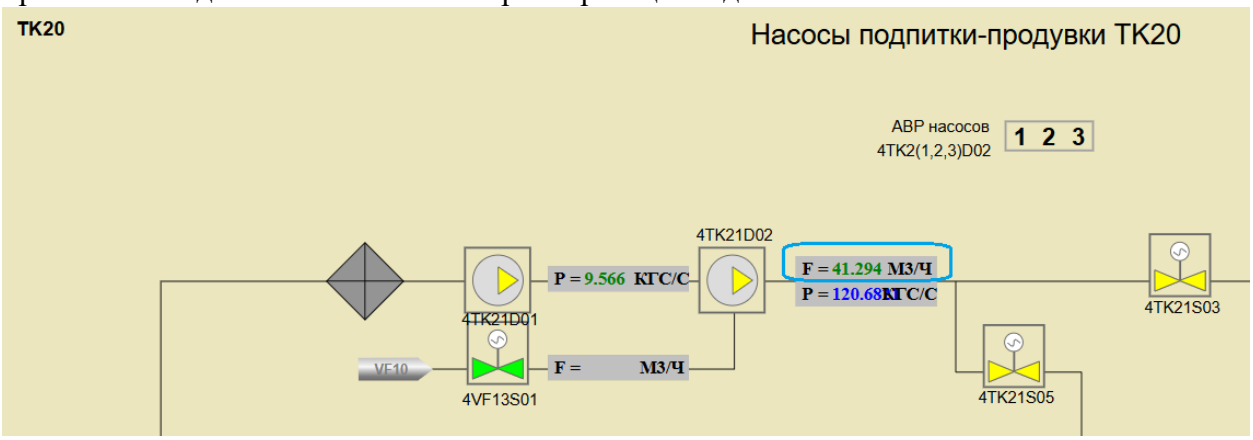


Рисунок 27. Отображение привязанного к технологическому параметру датчика

Цвет отображения зависит от кода качества рассчитываемого параметра, получаемого при сравнении с аварийными и предупредительными границами.

3.3.3 DsEdit Редактирование объектов управления

Объекты управления отображают состояние множества сигналов. Например, задвижки отображают сигналы объектов DI. В свою очередь, регулирующие клапаны отображают типы DI и AI. Добавление нового объекта управления осуществляется через палитру подмоделей.

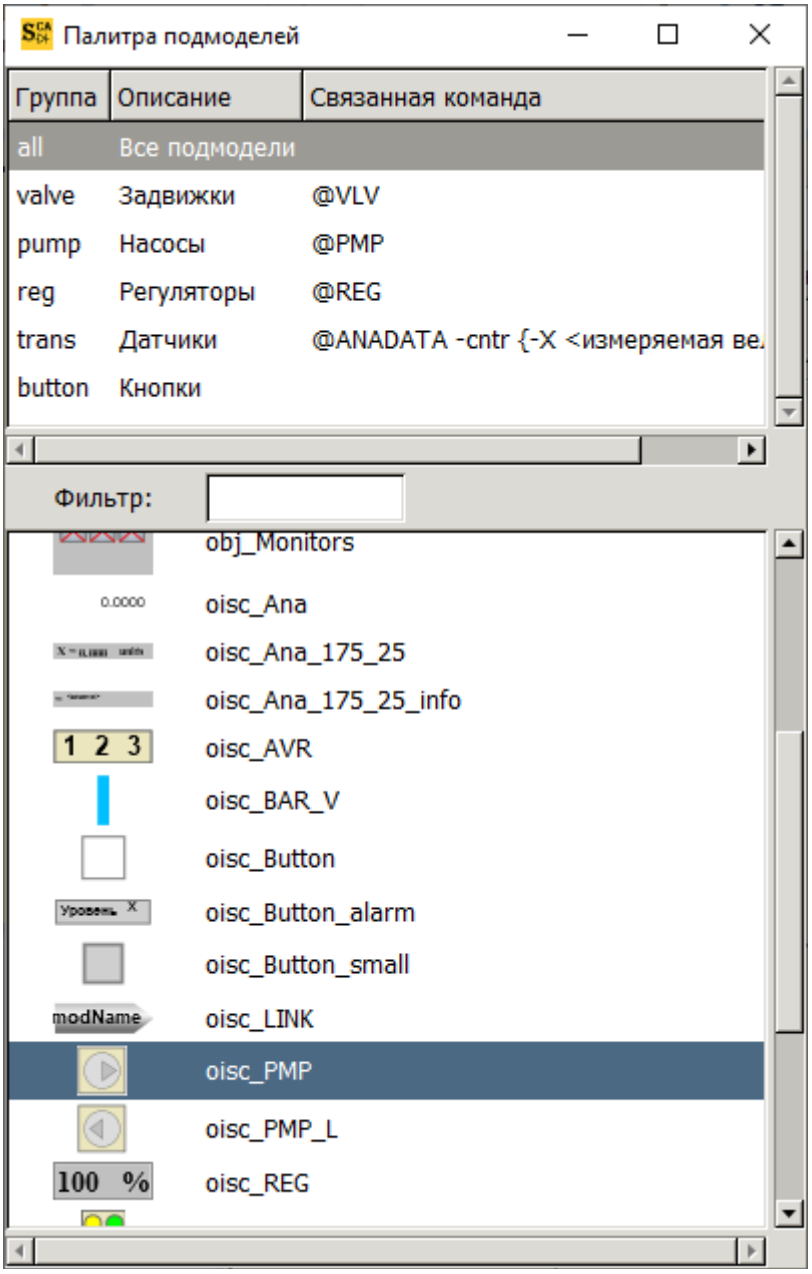


Рисунок 28. Выбор насоса в палитре подмоделей

В палитре необходимо выбрать одну из подмоделей насоса, например, oisc_PMP. Экземпляр насоса на основе выбранной подмодели помещается на текущий видеокадр.

Вставленный на видеокадр объект остается выбранным. Позиционируйте насос согласно отображаемому на видеокадре расположению.

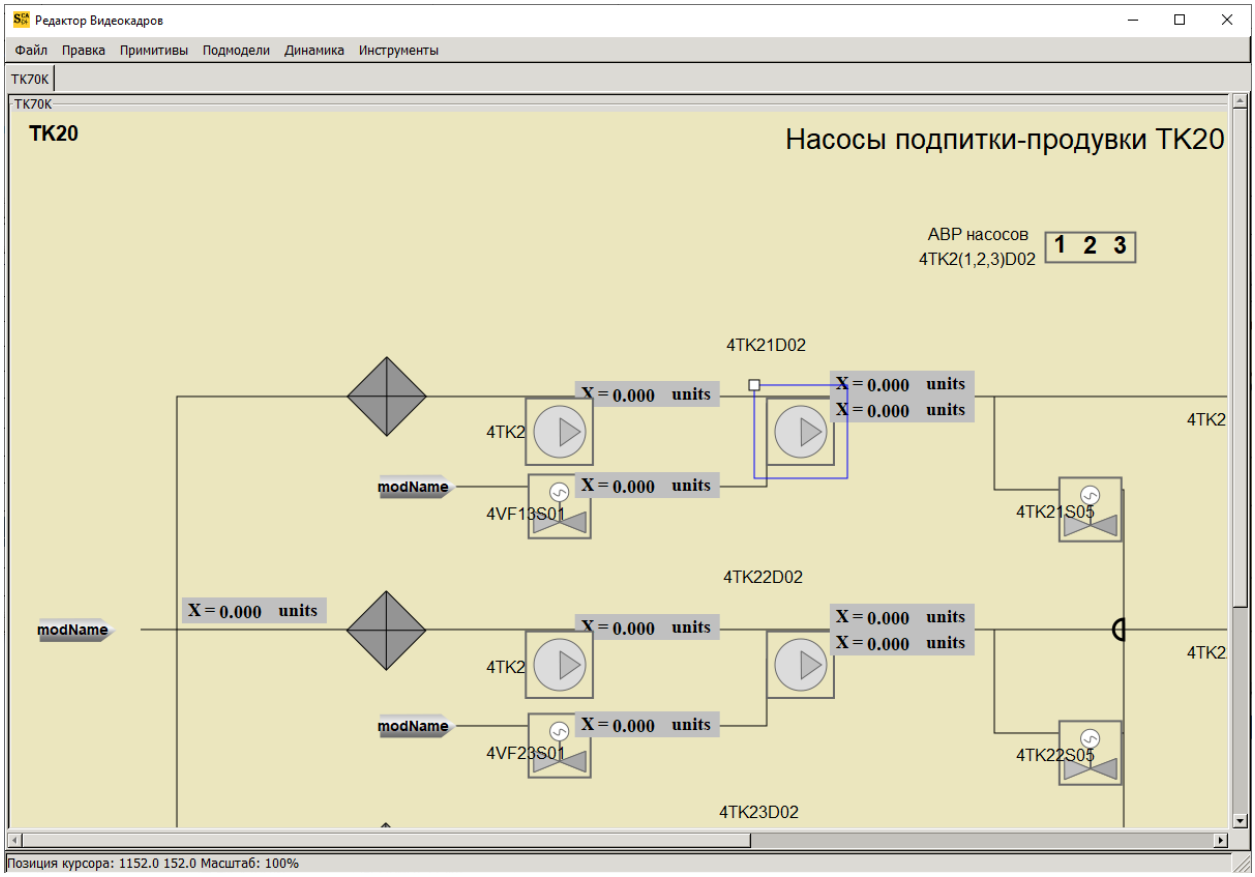


Рисунок 29. Расположение насоса на видеокадре.

Расположенный на видеокадре насос требует привязки к технологическим параметрам. Если привязка отсутствует, то насос отображается на видеокадре голубым цветом как непривязанный.

Для привязки необходимо выбранному насосу нажать правой кнопкой мыши для получения окна параметров подмодели, рисунок 30. Часть параметров подмодели задается по умолчанию. Но параметры технологической привязки требуют определения.

Параметр id-code требует задания технологического имени – KKS. В отличие от датчика, насосам для привязки требуется не один параметр code из глобала дискретных сигналов DIs, а несколько. Поэтому для каждого KKS в базе данных определены несколько параметров, расширенных суффиксами. Перечень суффиксов расширения приведен в приложении 11. Необходимо ввести значение KKS, соответствующее коду оборудования насоса. Значению 40TKR21AD002 соответствуют записи в глобале дискретных сигналов, используемые для отображения и управления.

```
Obj 3023 -name 1DI003024 -code 40TKR21AD002_F0 -descr {ПН1} -almpri 0 -dst_mask
1 -xarea_id 101 -proflags 7 -alloflags 7 -is_work TRUE -fun_type 6 -inver FALSE
Obj 3026 -name 1DI003027 -code 40TKR21AD002_Z0 -descr {ПН1} -tref
TPTS:TP000.name -dtype M -almpri 0 -dst_mask 1 -xarea_id 101 -proflags 7 -
alloflags 7 -is_work TRUE -fun_type 6 -inver FALSE
Obj 3032 -name 1DI003033 -code 40TKR21AD002_OU -descr {ПН1} -almpri 0 -dst_mask
1 -xarea_id 101 -proflags 7 -alloflags 7 -is_work TRUE -fun_type 6 -inver FALSE
```

Параметр type-name требует задания типа. В нашем случае почти всегда DI, но могут быть другие типы, например, DC – дискретные расчетные величины.

Параметр type-cntr не используется.

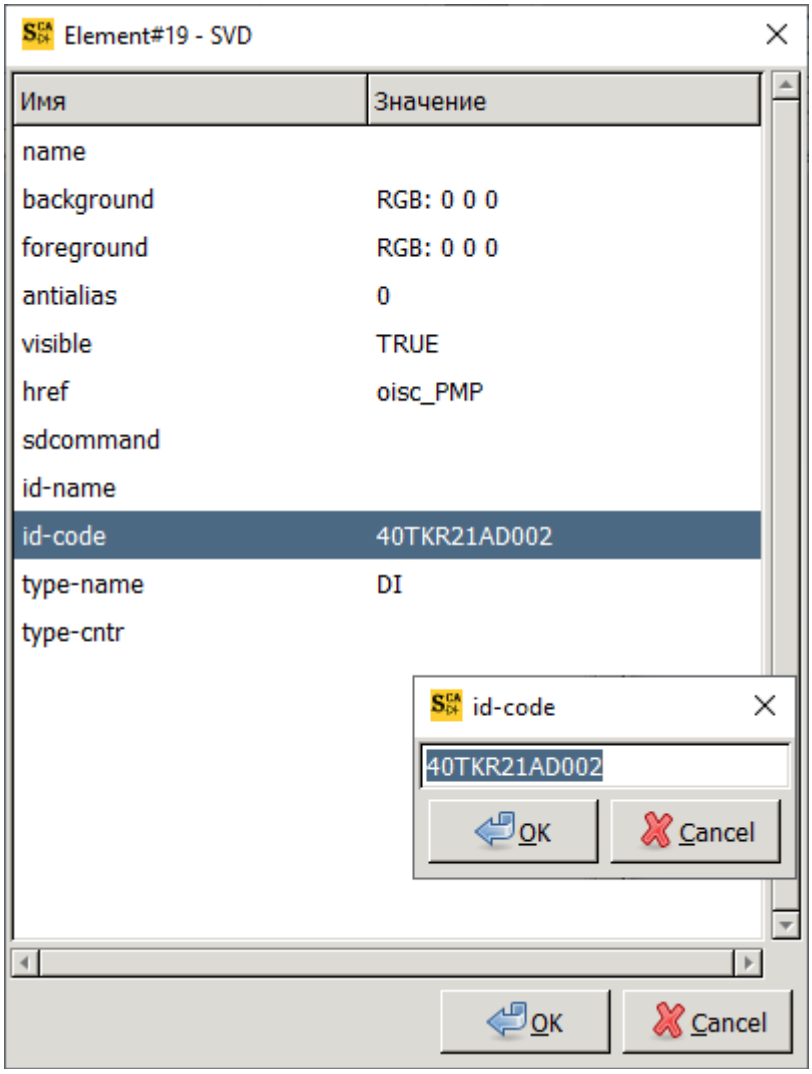


Рисунок 30. Окно параметров подмодели насоса.

После редактирования параметров и сохранения файла объект управления становится привязан к дискретным сигналам с указанными технологическими кодами.

При отображении в браузере привязанного насоса появляется параметризация и динамика включенного состояния.

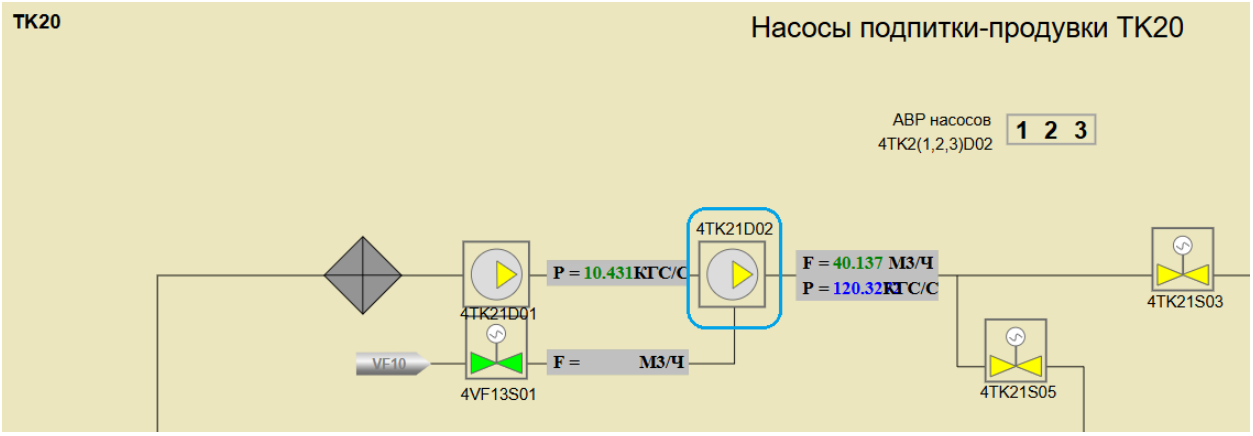


Рисунок 31. Отображение привязанного к технологическим параметрам насоса

Цвет отображения зависит от состояния оборудования.

3.3.4 Генерация файлов видеокадров

Средства генерации файлов видеокадров, помимо основных кадров, охватывает окна управления, тренды, подмодели, рисунки. Процесс генерации видеокадров после их редактирования в графическом редакторе включает в себя два этапа – собственно генерацию видеокадров в формате svg из исходных видеокадров в формате *.bgd и их копирование в определенные проектом рабочие директории.

Для генерации файлов видеокадров наберите py convall.py (в Windows) или ./convall.py (в Linux).

Для копирования сгенерированных файлов в определенные проектом рабочие директории наберите py cpall.py (в Windows) или ./cpall.py (в Linux).

Команды запускаются в корневом каталоге разработки.

3.4 СРЕДСТВА РАЗРАБОТКИ УЧЕБНО-МЕТОДИЧЕСКИХ МАТЕРИАЛОВ

3.4.1 Организация файлов учебных материалов

Учебно-методические материалы разбиты на курсы. В свою очередь, материалы одного курса распределены по контейнерам – по принадлежности.

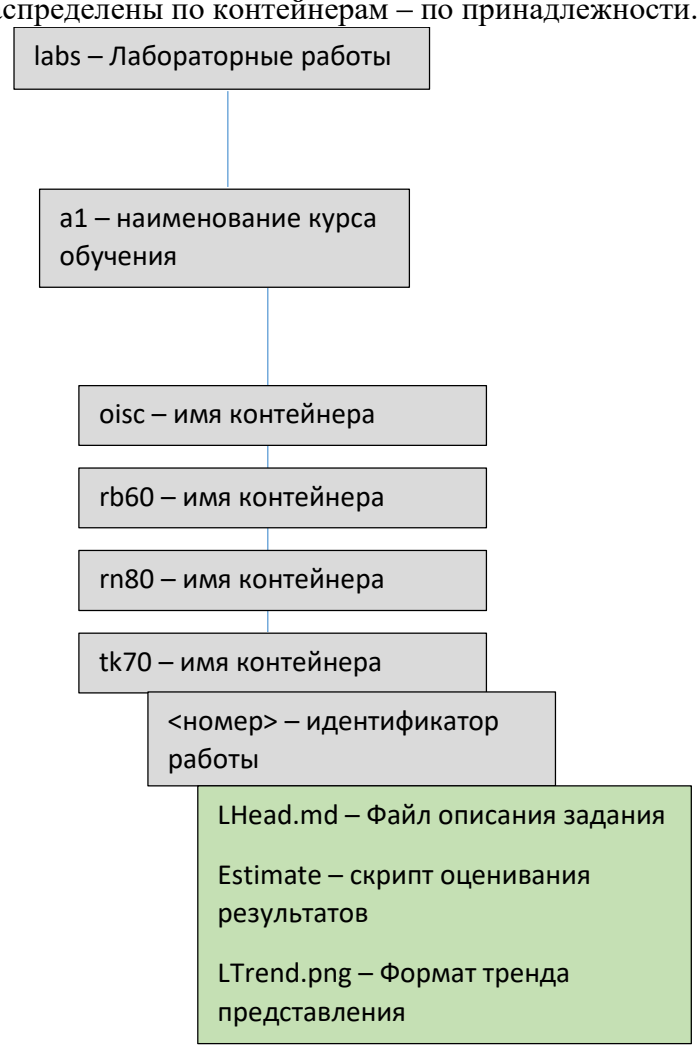


Рисунок 32. Расположение файлов учебных материалов

Материалы располагаются не внутри контейнеров, а в каталоге labs внутри проектного каталога системы СКОЯТ. Курсы могут разрабатываться и корректироваться независимо от изменения моделей и видеокадров. Курсы имеют свою независимую иерархию.

Для сохранения логики распределения данных материалы внутри курса распределяются по контейнерам и далее – по номерам лабораторных работ. Каталог каждой работы должен содержать следующую информацию:

- LHead.md – файлы описания задания на языке Markup;
- LHead.md.html – сгенеренные из языка Markup html-описания задания;
- task.html – описание задачи;
- Estimate – скрипт оценивания результатов.

При необходимости набор базовых файлов может быть расширен. Файл учебных материалов *.md должен редактироваться и быть источником для html.

3.4.2 Редактирование материалов Markup

Для редактирования материалов в формате Markup используется программный продукт Dillinger, доступный из браузера как https://dillinger.io . Пример редактирования LHead.md представлен на рисунке 33.

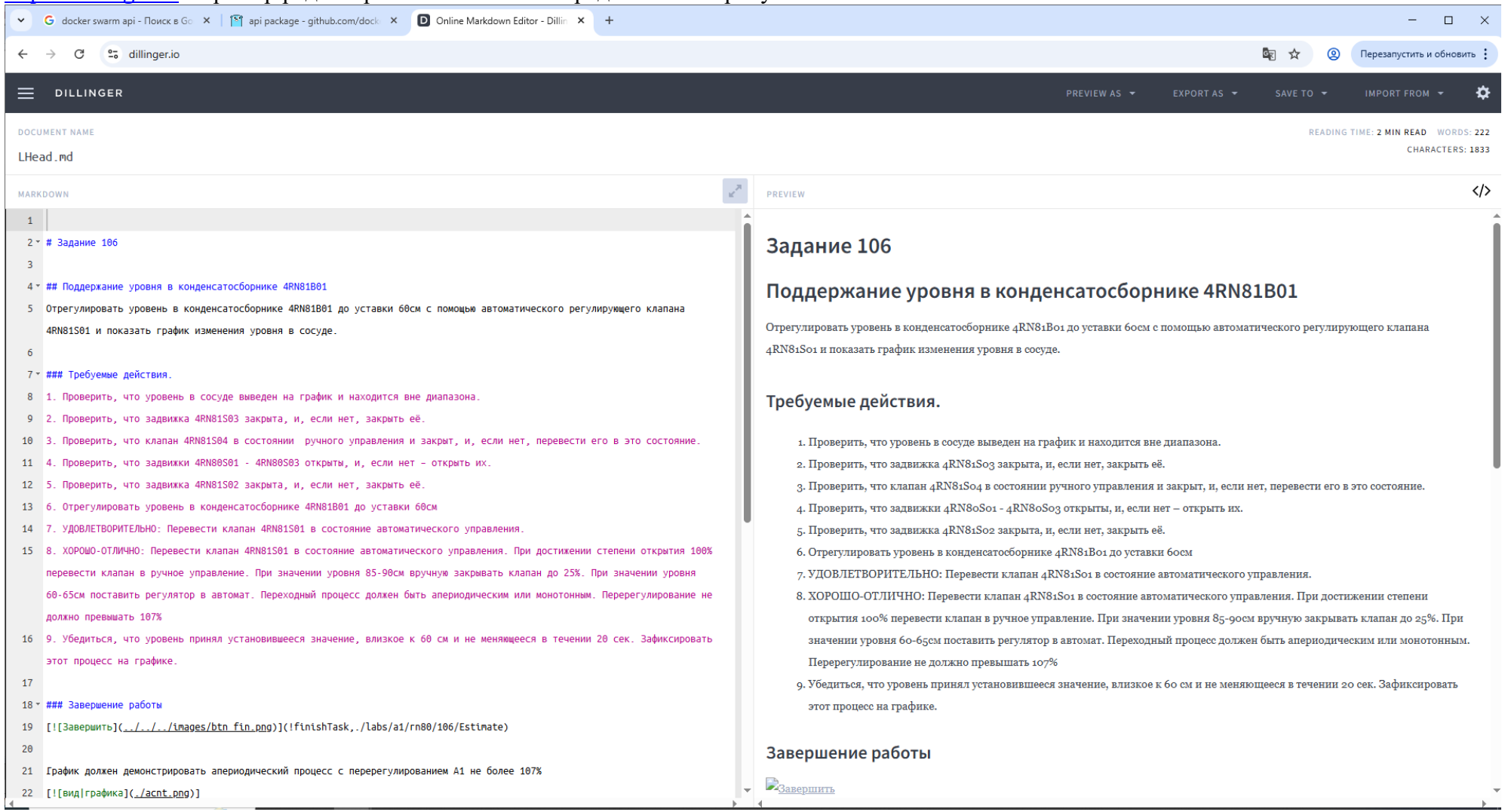


Рисунок 33. Редактирование файла LHead.md

Для загрузки файла редактирования используется меню ‘IMPORT FROM’/’Markdown File’/ с выбором файла из каталога. Левая сторона редактора отображает вводимые строки, правая сторона отображает сгенеренное изображение.

В СКОЯТ используются следующие средства форматирования Markup (Таблица 1):

Форматирование Markup	Тег HTML	Описание
# Текст	Текст	Заголовок 1 (наиболее крупный шрифт для заголовка)
## Текст	Текст	Заголовок 2
### Текст	Текст	Заголовок 3
Текст	Текст	Выделение текста курсивом
__Текст__	<b>Текст<b>	Выделение текста жирным шрифтом
1. Текст-1 2. Текст-2 …	 - Текст-1 - Текст-2 … 	Нумерованный список
- Текст-1 - Текст-2 …	 - Текст-1 - Текст-2 … 	Ненумерованный список
[![Текст](Имя файла)]	<img src='Имя файла' alt='Текст'></img> 	Вставка рисунка из файла
[![Текст](Имя файла)](!Имя команды,Параметр)	 <img src='Имя файла' alt='Текст'></img> 	Команда управления
## Параметр	Параметр	Параметр запуска задания. Примечание. Если параметр не находится в списке поддерживаемых параметров, то он отображается как Заголовок 2

Таблица 1 – используемые средства форматирования Markup.

Поддерживаемые команды управления перечислены в Таблице 2.

Команда	Назначение	Форматирование Markup
[Завершить]	Завершение выполнения задания и вызов скрипта оценивания	[![Завершить](../../../images/btn_fin.png)] (!finishTask,скрипт оценки)
[Выбрать]	Добавление выбранной переменной в данную серверную группу	[![Выбрать](../../../images/btn_sel.png)] (!addCode,имя группы)
[Удалить]	Удаление выбранной переменной из данной серверной группы	[![Удалить](../../../images/btn_del.png)] (!removeCode,имя группы)

Таблица 2 – поддерживаемые команды управления.

Пример команды завершения задания.

```
[![Завершить](../../../images/btn_fin.png)](!finishTask,./labs/a1/rn80/107/Estimate)
```

Поддерживаемые параметры запуска задания перечислены в Таблице 3.

Параметр	Наименование	Дополнительные параметры
Reset	Начальное состояние задания	<число> – номер начального состояния
RstObj	Имя объекта, в котором задается начальное состояние задания	<строка> - имя объекта. Если такой объект не определен в серверной БД, то данный параметр не задается
Elapsed	Время, отведенное на выполнение задания	<секунды>
Interceptor	Имя переменной, значение которой передается на сервер при ее изменении	<строка> - имя (код) переменной
FrmSrc_main	Имя видеокадра, загружаемого в головной фрейм системы верхнего уровня	<строка>
FrmSrc_trend	Имя видеокадра, загружаемого во фрейм отображения графика системы верхнего уровня	<строка>
Trend	Имя переменной, выводимой на график	<строка> - имя (код) переменной
IntervalSec	Интервал отрисовки графика	<секунды>. Рекомендуемое значение – время, отведенное на выполнение задания + 20 секунд

Таблица 3 – поддерживаемые параметры запуска задания.

Пример задания параметров запуска.

```
## Interceptor 40RNR82CL901_XQ57
## Interceptor 40RNR82AS002_Z0
```

```
## Interceptor 40RNR82AS002_OU
## Interceptor 40RNR82AS003_Z0
## Interceptor 40RNR82AS003_OU
## Interceptor 40RNR82AS801_Z0
## Interceptor 40RNR82AS801_OU
## Interceptor 40RNR82AS804_Z0
## Interceptor 40RNR82AS804_OU
## Elapsed 240
## Reset 2
## FrmSrc_main NPP_models/RB3K.svg
## FrmSrc_trend NPP_models/RB3T.svg
## Trend 40RNR82CL901_XQ57
## IntervalSec 260
```

В данном примере задано 9 переменных, значения которых должны передаваться на сервер при их изменении, время выполнения задания – 240 секунд (4 минуты), интервал отрисовки графика – 260 секунд (согласно рекомендуемому значению), загружаемые видеокадры, начальное состояние (состояние №2), переменная, выводимая на график.

3.4.3 Генерация файлов отображения из материалов Markup

Материалы в формате Markup используются для редактирования и хранения данных. Отображаемые данные конвертируются в html-формат средствами генерации. Конверсия материалов, созданных для данного курса, осуществляется путем запуска команды py muconv_all.py (в Windows) или ./muconv_all.py (в Linux). Команда запускается в каталоге labs внутри проектного каталога системы СКОЯТ.

3.5 КЛИЕНТСКИЙ ДОСТУП

3.5.1 Вход с авторизацией

Авторизация в системе СКОЯТ осуществляется через механизм Яндекс OAuth на основе алгоритма OAuth2.0. Для этого пользователю нужно зарегистрироваться в сервисах Яндекс (например, завести почту на Яндексе). Целью регистрации является получение Яндекс ID. Далее наберите адрес, например http://osco .

Наличие Яндекс ID служит необходимым, но не достаточным условием для работы в системе. Необходима регистрация имени, полученном в Яндекс ID, в базе данных пользователей системы. Это может сделать системный администратор СКОЯТ.

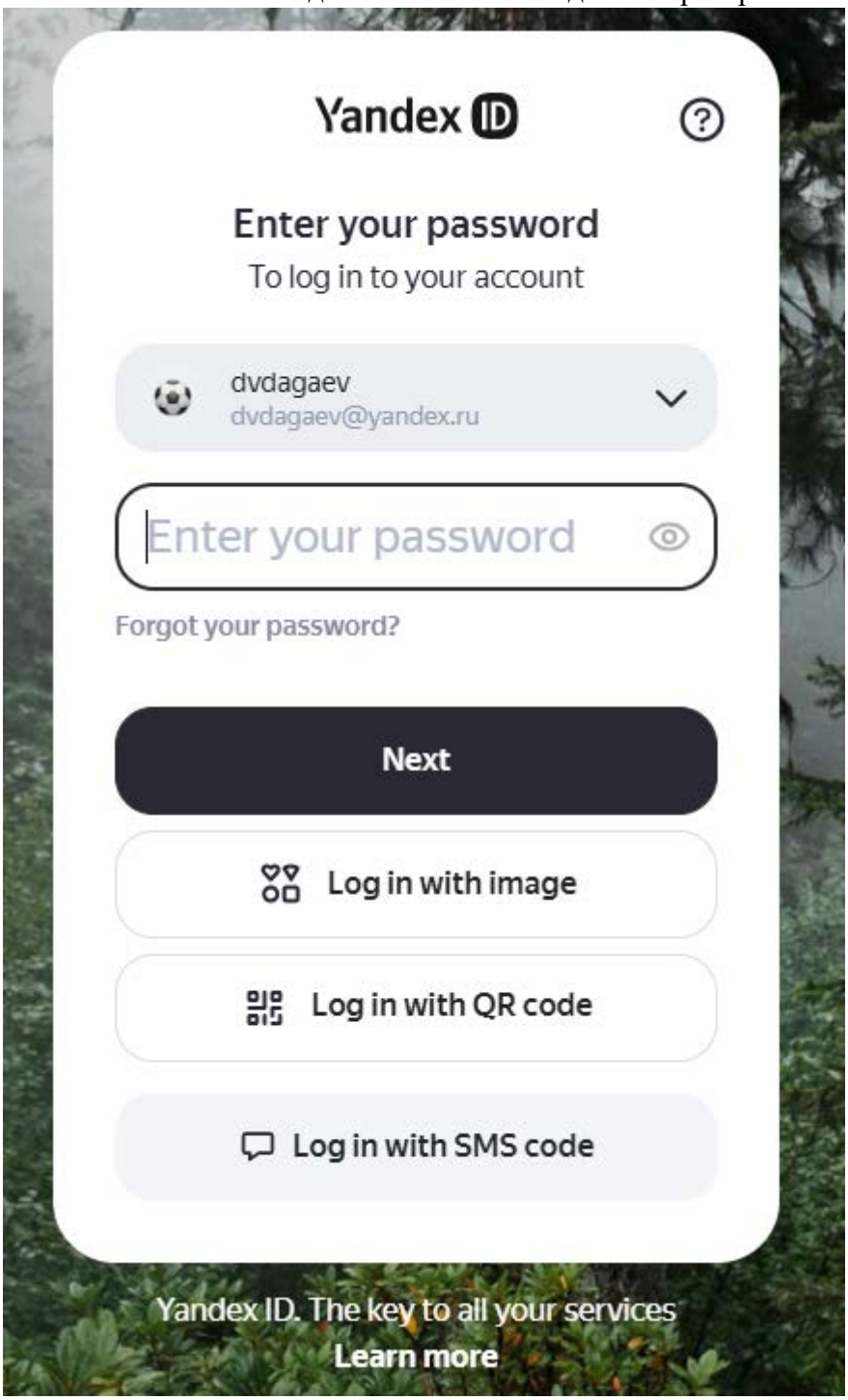


Рисунок 34. Авторизация логином и паролем Яндекс ID

Вход в систему для зарегистрированного пользователя осуществляется вводом логина и пароля, рисунок 34. При необходимости может потребоваться авторизация через номер телефона и подтверждающий SMS, рисунок 35.

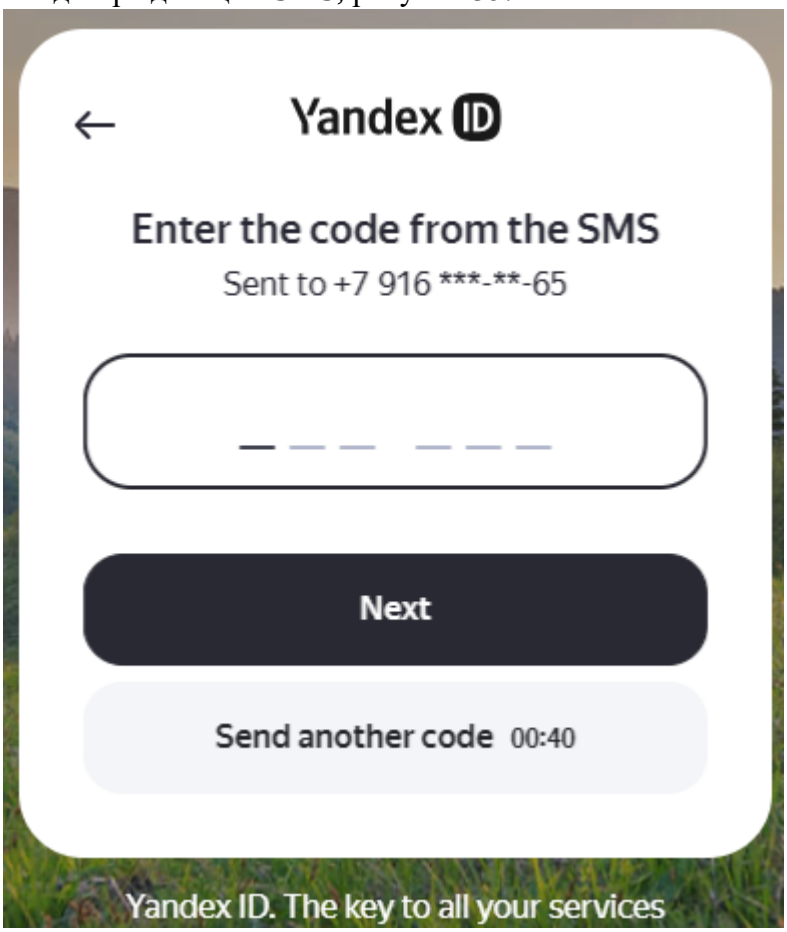


Рисунок 35. Подтверждение с помощью SMS

По завершении авторизации пользователь попадает в фрейм с основным окном – Система комплексного обучения ядерным технологиям, рисунок 36.



Рисунок 36. Фрейм основного окна

Имя авторизированного пользователя появляется в правом углу, отмеченным зеленым овалом. Для начала работы необходимо воспользоваться синими кнопками выбора меню.

Для окончания работы необходимо нажать на прямоугольник пользователя и выполнить нажатие на появившуюся кнопку Выход.

Также выходом из системы считается смена корневого адреса в строке браузера (например, yandex.ru вместо osco).

Введенные данные авторизации сохраняются на некоторое время, определяемое настройками браузера. При таком сохранении последующая авторизация не требуется какое-то время.

Для сброса настроек браузера нужно удалить журналы за необходимый интервал времени. Если в удаленный интервал вводились данные авторизации, то введенные данные будут сброшены.

3.5.2 Выполнение лабораторных работ

Для выполнения лабораторных работ необходимо вначале выбрать режим. Меню – Режим / Лабораторные Работы. При успешном переходе в данные режим пользователь получает категорию: Экзаменуемый, рисунок 37.

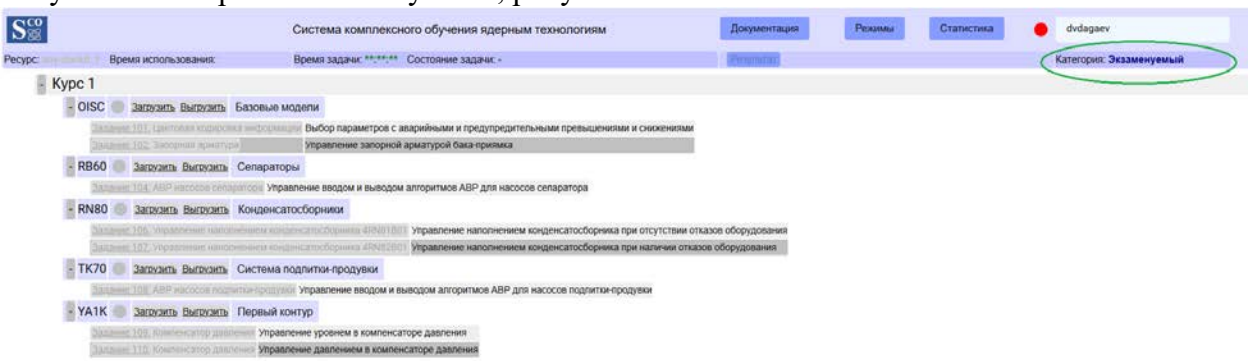


Рисунок 37. Выбор заданий для экзаменуемого

Пользователь категории Экзаменуемый имеет возможность выбора заданий из дерева, которое состоит из разделов <Курс>/<Секция>/<Задание>. Например, Курс1/RB60/104.АВР насосов сепаратора. Каждому заданию соответствует ресурс, доступ к которому может запросить пользователь. Ресурс представляет собой контейнер с моделями, отделенных от контейнера с моделями других пользователей. Ресурс предоставляется в полное распоряжение пользователю. Если в настройках системы сделана возможность разделенного ресурса, то к ресурсу пользователя могут подключаться другие в режиме наблюдателя или в режиме инструктора. Но распоряжается ресурсом исключительно пользователь категории Экзаменуемый.

Для выбора задания необходимо загрузить ресурс, например, Курс1/RB60/104.АВР насосов сепаратора. Это осуществляется нажатием RB60 Загрузить, после этого индикатор загрузки меняет цвет с серого на желтый и на зеленый (оранжевая рамка).

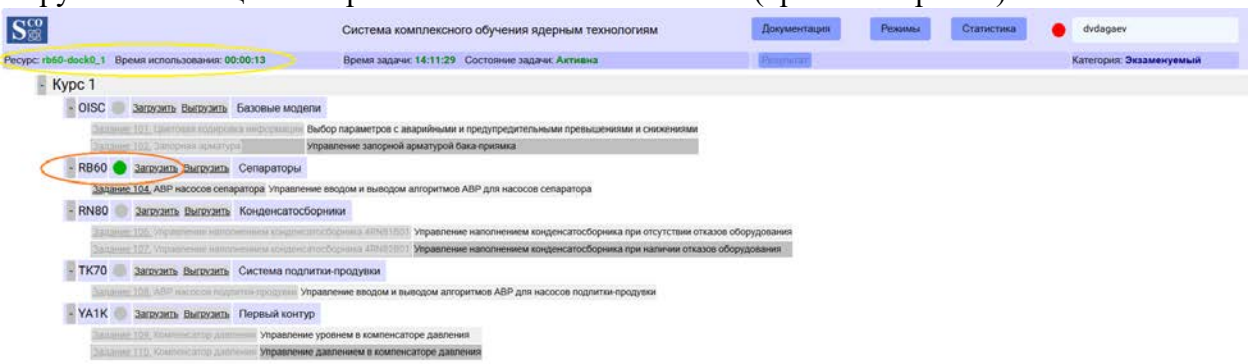


Рисунок 38. Загрузка ресурса - контейнера

При неуспешной загрузке индикатор загорается красным цветом. При успешной загрузке возникает информация по ресурсу и времени использования (желтая рамка). В нашем примере загружается ресурс rb60-dock0_1 и начинается отсчет его времени использования.

После загрузки ресурса активируется ссылка и заданиями, относящимися к данному ресурсу. Например, Задание 104 АВР насосов сепаратора. Переход по указанной ссылке запускает выполнение упомянутого задания.

По завершении работы производится выгрузка ресурса. Если ресурс не выгружен, выгрузка производится автоматически и определяется настройками системы.

Выполнение задания происходит при переходе по ссылке при условии загруженного ресурса. Отсчет времени выполнения фиксируется в верхней части справа (красная рамка). Задание разбивается на несколько фреймов. Наиболее общий вид – на рисунке 39 – имеет фрейм учебных материалов Markup, фрейм мнемосхемы, фрейм графиков. При выполнении задания необходимо осуществлять управление оборудованием. Для этого используются окна управления, выделенные голубым цветом. В окне необходимо нажать кнопку «Управление», чтобы активировать остальные кнопки.

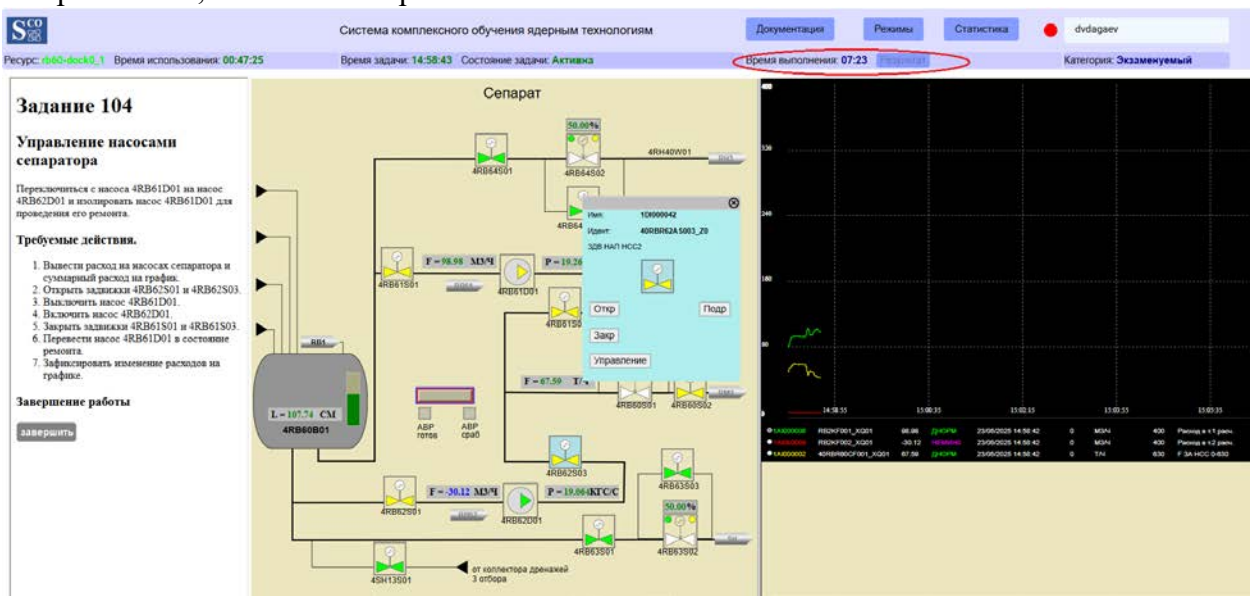


Рисунок 39. Управление задвижкой при выполнении задания

Экзаменуемому необходимо выполнить действия, указанные в левой части. По окончании нажимается кнопка Завершить.

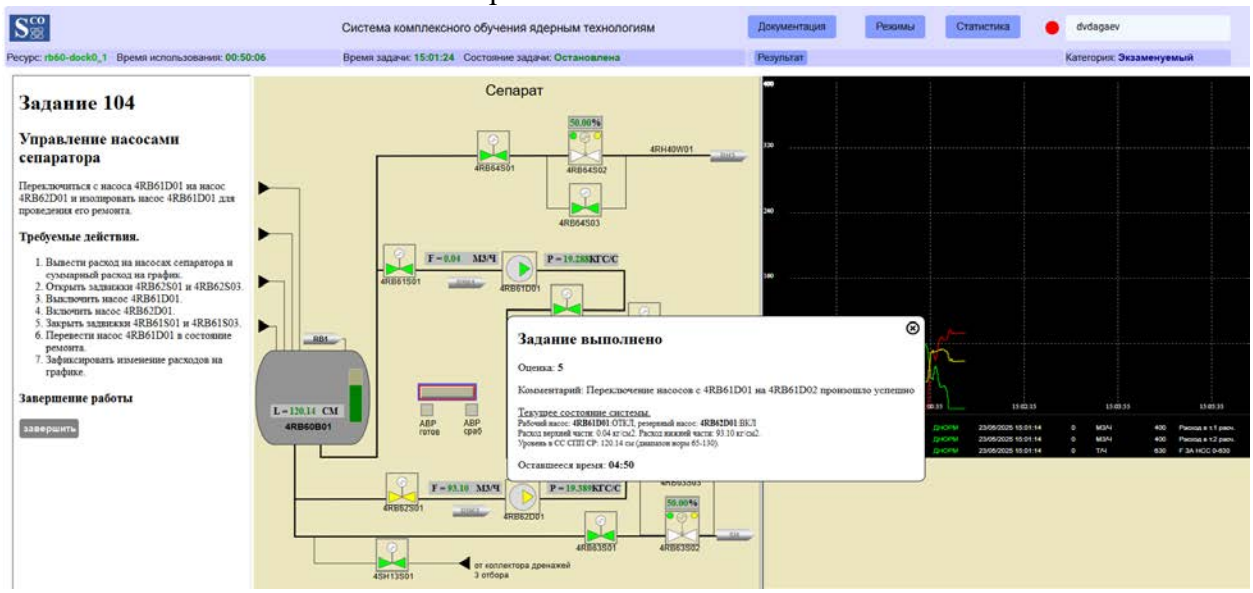


Рисунок 40. Результат выполнения задания

По завершении система производит оценивание результата выполнения работы. Результат выполнения с комментариями и информацией о состоянии системы выводится во всплывающем окне. При этом окно можно закрыть или активировать снова по кнопке Результат в верхней части экрана.

3.5.3 Мониторинг и режим инструктора

Режим мониторинга позволяет наблюдать выполнение лабораторных работ другим экзаменуемым. Меню – Режим / Мониторинг процесса.

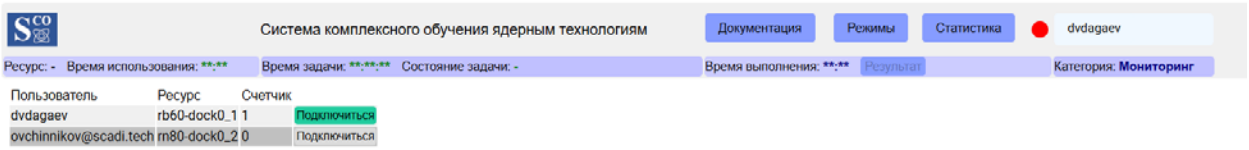


Рисунок 41. Окно выбора ресурса для мониторинга

При выборе режима мониторинга производится переход на видеокадр со списком задач, к которым можно подключиться. Подключение в режиме мониторинга приводит к параллельному отображению задания в составе фрейма учебных материалов, фрейма мнемосхемы и фрейма графика. Управление в режиме мониторинга невозможно.

Режим инструктора наблюдать выполнение лабораторных экзаменуемым и корректировку его действий. Меню – Режим / Инструктор.

Подключение в режиме инструктора позволяет осуществлять расширенное управление экзаменуемым. При таком управлении инструктор имеет возможность ввода обобщенных неисправностей на выбранное оборудование, рисунок 42.

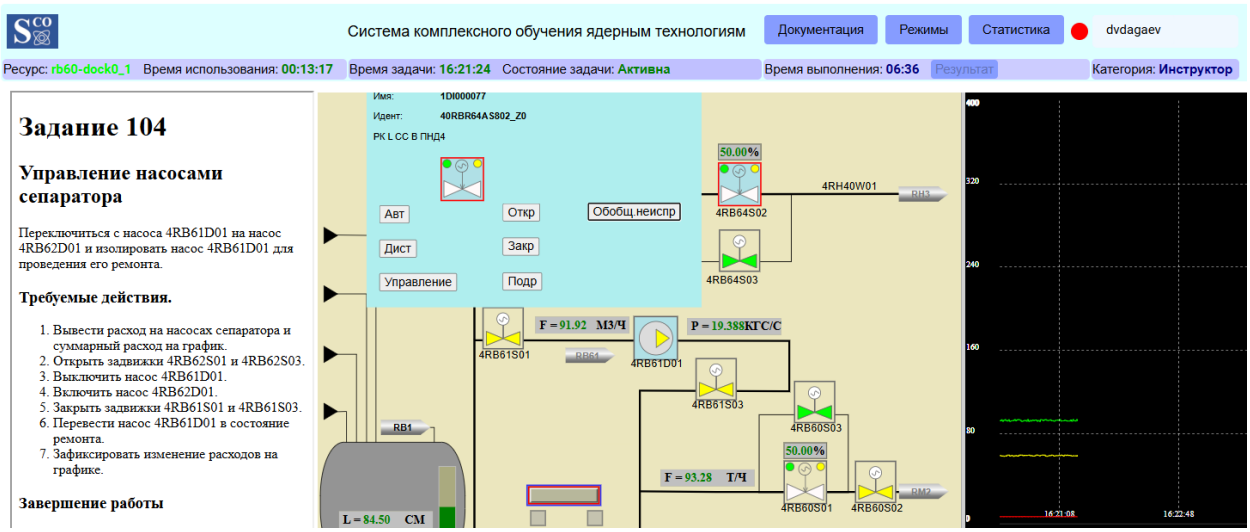


Рисунок 42. Ввод отказа инструктором

На приведенном рисунке инструктор выбирает регулирующий клапан 40RBR64AS802. Выбранный клапан подсвечивается голубым цветом фона. Далее инструктор нажимает на ввод обобщенной неисправности и вокруг клапана появляется красная рамка индикации неисправности. На видеокадре обучаемого также появляется подсветка выбора и рамка неисправности.

В то же самое время экзаменуемый управляет насосом 40RBR61AD001 с помощью окна управления. Экзаменуемый имеет перед собой следующий видеокадр (рисунок 43).

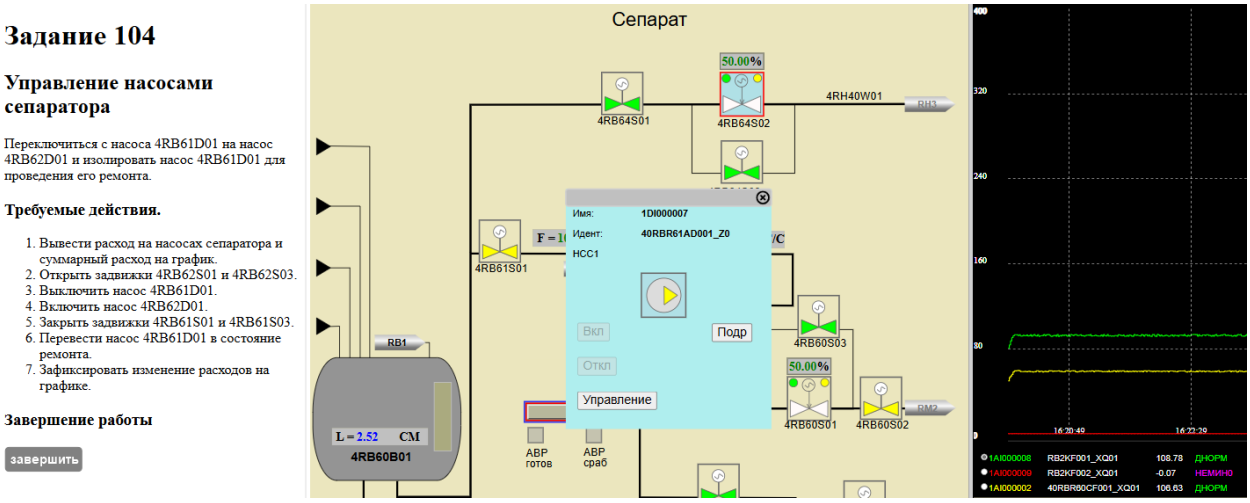


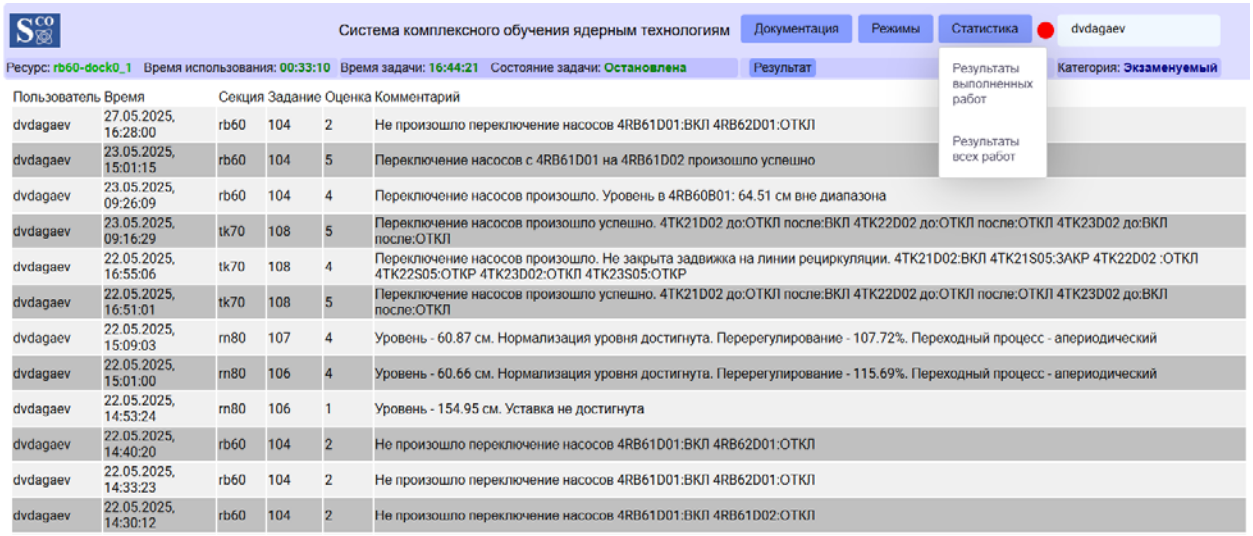
Рисунок 43. Экзаменуемый с инструктором

Экзаменуемый видит выбранный в управление регулирующий клапан 40RBR64AS802 с голубой подсветкой. Экзаменуемый видит появление красной рамка отказа регулирующего клапана. Экзаменуемый имеет возможность корректировки действий в связи с изменившейся ситуацией в объекте управления.

Инструктор может помогать экзаменуемому, т.е. осуществлять за него управление. Инструктор имеет права на все действия по управлению экзаменуемого. При этом кнопка ЗАВЕРШИТЬ остается прерогативой экзаменуемого.

3.5.4 Статистика выполненных работ

Пользователь имеет право выдачи статистики. Меню – Статистика / Результаты выполненных работ представлена на рисунке 44.

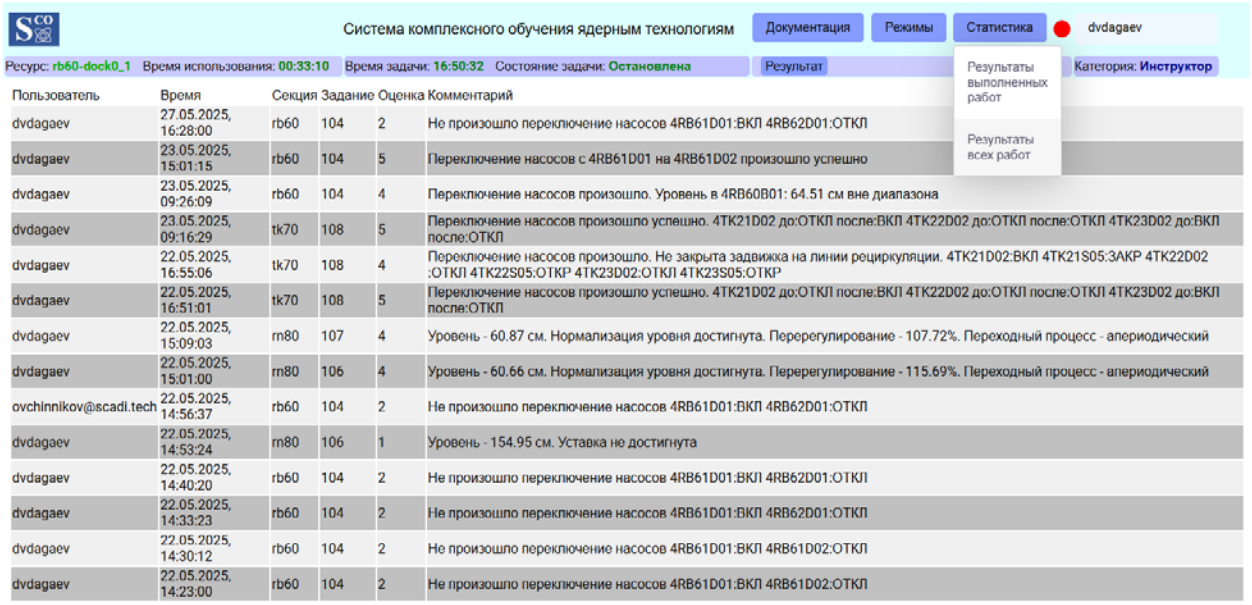


Пользователь	Время	Секция	Задание	Оценка	Комментарий
dvdagaev	27.05.2025, 16:28:00	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB62D01:ОТКЛ
dvdagaev	23.05.2025, 15:01:15	rb60	104	5	Переключение насосов с 4RB61D01 на 4RB61D02 произошло успешно
dvdagaev	23.05.2025, 09:26:09	rb60	104	4	Переключение насосов произошло. Уровень в 4RB60B01: 64.51 см вне диапазона
dvdagaev	23.05.2025, 09:16:29	tk70	108	5	Переключение насосов произошло успешно. 4TK21D02 до:ОТКЛ после:ВКЛ 4TK22D02 до:ОТКЛ после:ОТКЛ 4TK23D02 до:ВКЛ после:ОТКЛ
dvdagaev	22.05.2025, 16:55:06	tk70	108	4	Переключение насосов произошло. Не закрыта задвижка на линии рециркуляции. 4TK21D02:ВКЛ 4TK21S05:ЗАКР 4TK22D02 :ОТКЛ 4TK22S05:ОТКР 4TK23D02:ОТКЛ 4TK23S05:ОТКР
dvdagaev	22.05.2025, 16:51:01	tk70	108	5	Переключение насосов произошло успешно. 4TK21D02 до:ОТКЛ после:ВКЛ 4TK22D02 до:ОТКЛ после:ОТКЛ 4TK23D02 до:ВКЛ после:ОТКЛ
dvdagaev	22.05.2025, 15:09:03	rn80	107	4	Уровень - 60.87 см. Нормализация уровня достигнута. Перерегулирование - 107.72%. Переходный процесс - апериодический
dvdagaev	22.05.2025, 15:01:00	rn80	106	4	Уровень - 60.66 см. Нормализация уровня достигнута. Перерегулирование - 115.69%. Переходный процесс - апериодический
dvdagaev	22.05.2025, 14:53:24	rn80	106	1	Уровень - 154.95 см. Уставка не достигнута
dvdagaev	22.05.2025, 14:40:20	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB62D01:ОТКЛ
dvdagaev	22.05.2025, 14:33:23	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB62D01:ОТКЛ
dvdagaev	22.05.2025, 14:30:12	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB61D02:ОТКЛ

Рисунок 44. Статистика выполненных работ

В приведенной выборке фиксируются работы, закончившиеся командой ЗАВЕРШИТЬ или автоматически прекращенные по окончании отведенного времени. Для режима Экзаменуемый или Мониторинг организуется выдача данных только для указанного пользователя.

Инструктор имеет возможность получения статистика для всех пользователей системы.



Пользователь	Время	Секция	Задание	Оценка	Комментарий
dvdagaev	27.05.2025, 16:28:00	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB62D01:ОТКЛ
dvdagaev	23.05.2025, 15:01:15	rb60	104	5	Переключение насосов с 4RB61D01 на 4RB61D02 произошло успешно
dvdagaev	23.05.2025, 09:26:09	rb60	104	4	Переключение насосов произошло. Уровень в 4RB60B01: 64.51 см вне диапазона
dvdagaev	23.05.2025, 09:16:29	tk70	108	5	Переключение насосов произошло успешно. 4TK21D02 до:ОТКЛ после:ВКЛ 4TK22D02 до:ОТКЛ после:ОТКЛ 4TK23D02 до:ВКЛ после:ОТКЛ
dvdagaev	22.05.2025, 16:55:06	tk70	108	4	Переключение насосов произошло. Не закрыта задвижка на линии рециркуляции. 4TK21D02:ВКЛ 4TK21S05:ЗАКР 4TK22D02 :ОТКЛ 4TK22S05:ОТКР 4TK23D02:ОТКЛ 4TK23S05:ОТКР
dvdagaev	22.05.2025, 16:51:01	tk70	108	5	Переключение насосов произошло успешно. 4TK21D02 до:ОТКЛ после:ВКЛ 4TK22D02 до:ОТКЛ после:ОТКЛ 4TK23D02 до:ВКЛ после:ОТКЛ
dvdagaev	22.05.2025, 15:09:03	rn80	107	4	Уровень - 60.87 см. Нормализация уровня достигнута. Перерегулирование - 107.72%. Переходный процесс - апериодический
dvdagaev	22.05.2025, 15:01:00	rn80	106	4	Уровень - 60.66 см. Нормализация уровня достигнута. Перерегулирование - 115.69%. Переходный процесс - апериодический
ovchinnikov@scadi.tech	22.05.2025, 14:56:37	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB62D01:ОТКЛ
dvdagaev	22.05.2025, 14:53:24	rn80	106	1	Уровень - 154.95 см. Уставка не достигнута
dvdagaev	22.05.2025, 14:40:20	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB62D01:ОТКЛ
dvdagaev	22.05.2025, 14:33:23	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB62D01:ОТКЛ
dvdagaev	22.05.2025, 14:30:12	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB61D02:ОТКЛ
dvdagaev	22.05.2025, 14:23:00	rb60	104	2	Не произошло переключение насосов 4RB61D01:ВКЛ 4RB61D02:ОТКЛ

Рисунок 45. Статистика всех работ всех пользователей

При необходимости организуются запросы по конкретным пользователям или группам пользователей.

3.5.5 Информационный раздел документации

Пользователь имеет доступ к справочной документации. Меню – Документация / Задание 101 цветовая кодировка информации представлена на рисунке 46.

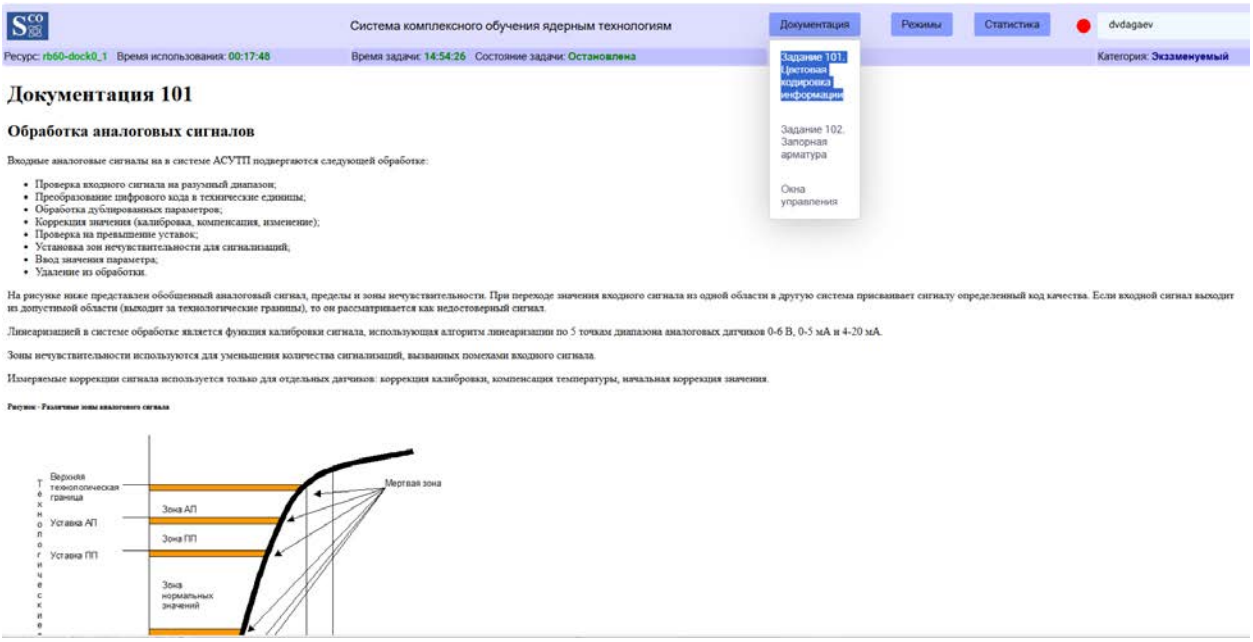


Рисунок 46. Выборка документации из меню

Заполнение раздела документации является полностью проектно-зависимым.

4 СООБЩЕНИЯ ОПЕРАТОРУ

4.1 СООБЩЕНИЯ СЕРВИСОВ И МЕНЕДЖЕРА МОДЕЛЕЙ

Существует ряд фатальных ошибок, приводящих к завершению работы сервисов ПО СКОЯТ. Ошибки переменных окружения приводят к следующим сообщениям.

```
2025/05/28 15:52:50 REDIRECT_URL environment variable required
```

Ошибки лицензии приводят к следующим сообщениям.

```
SCONT License error file= Wos_osco.lic res=-8
```

Работа ПО СКОЯТ прекращена ввиду отсутствия валидной лицензии.

```
SCONT License file= Wos_osco.lic expired=DD/MM/YYYY”
```

Работа ПО СКОЯТ прекращена ввиду того, что лицензия просрочена.

```
“SCADI License License resources requested=1000 > max=100”
```

Работа ПО СКОЯТ прекращена ввиду превышения количества ресурсов-контейнеров, разрешенных в лицензии.

Существует также ряд фатальных ошибок, приводящих к завершению работы менеджера СКАДИ-МОДЕЛЬ. Ошибки лицензии приводят к следующим сообщениям.

```
“SCADI license error file=Dsm_xxxxx.lic”
```

Работа ПО СКАДИ-МОДЕЛЬ прекращена ввиду отсутствия валидной лицензии.

```
“SCADI License file=Dsm_xxxxx.lic expired=DD/MM/YYYY”
```

Работа ПО СКАДИ-МОДЕЛЬ прекращена ввиду того, что лицензия просрочена.

```
“SCADI License amount of tags=140000 exceeds maximum=100000”
```

Работа ПО СКАДИ-МОДЕЛЬ прекращена ввиду превышения количества тэгов, разрешенных в лицензии.

Ошибки конфигурации DsMM приводят к следующим сообщениям.

```
*DsMM cfg_file 'DsmConfig.cfg' not loaded
```

Работа ПО СКАДИ-МОДЕЛЬ прекращена ввиду отсутствия конфигурационного файла.

```
*DsMM CreateGlobals error res=
```

Работа ПО СКАДИ-МОДЕЛЬ прекращена ввиду невозможности создания глобалов. Например, отсутствует каталог db.

4.2 СООБЩЕНИЯ УТИЛИТ СКАДИ-МОДЕЛЬ

Утилиты СКАДИ-МОДЕЛЬ, которые требуют загрузки, при отсутствии последней выдают сообщение следующего вида.

```
DsView -globname AIs -limit 2
* SCADI connect error= -14
```

Проверка загрузки осуществляется командой DsAccess. При отсутствии загрузки выдается сообщение следующего вида.

```
DsAccess
* SCADI connect error= -14
```

4.3 ЛОГ СИСТЕМНЫХ СООБЩЕНИЙ

После инициализации глобалов менеджером моделей появляется доступ к логу системных сообщений.

Просмотр лога системных сообщений осуществляется командой DsView –global 4 (либо DsView –globname Syslogs), вывод на консоль приведен на рисунке 47.

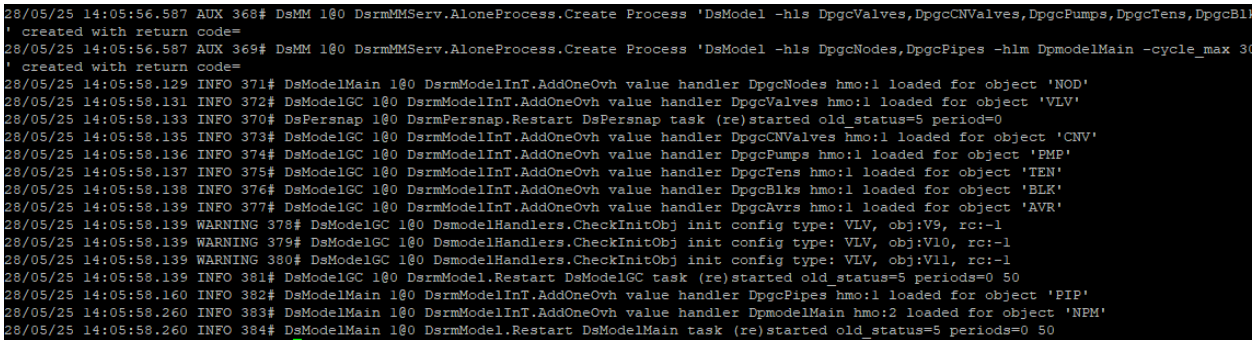


Рисунок 47 – Фрагмент лога системных сообщений глобала Syslogs

Сообщения системного лога представляют собой записи определенного формата Syslog. Например, можно разобрать по полям следующее сообщение.

```
28/05/25 14:05:58.139 WARNING 378# DsModelGC 1@0 DsmodelHandlers.CheckInitObj
init config type: VLV, obj:V9, rc:-1
```

Поле timestamp содержит время, в строковом представлении равное «`28/05/25 14:05:58.139`», поле важности severity – код WARNING, поле номера сообщения msg_id – номер 378, поле имени процесса module – имя программы DsModelGC, строка 1@0 означает user_id = 1, host_ind = 0 (нулевой хост), поле имени действия operation содержит «`DsmodelHandlers.CheckInitObj`», поле сообщения message – шаблон `init config type: VLV, obj:$, rc:$`, а поле params – параметры «V9 -1».

При необходимости можно просматривать сообщения из лога.

4.4 СООБЩЕНИЯ ИНТЕРАКТИВНЫХ ОТЛАДЧИКОВ

Минимальный размер окна терминала для запуска DsISD – 26 строк, 80 столбцов. Если размер окна терминала недостаточен для запуска приложения, то выводится сообщение вида:

```
Terminal window is too small. rows= 24 columns= 103
Should be: rows>= 26 columns>= 80
```

Минимальный размер окна терминала для запуска DsQSD – 32 строки, 138 столбцов. Если размер окна терминала недостаточен для запуска приложения, то выводится сообщение вида:

```
Terminal window is too small. rows= 24 columns= 103
Should be: rows>= 32 columns>= 138
```

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1 – СОСТАВ ЗАГРУЖАЕМЫХ МОДУЛЕЙ СКАДИ-МОДЕЛЬ

В таблице 1-1 приведен состав динамически загружаемых модулей СКАДИ-МОДЕЛЬ, относящиеся непосредственно к системе.

Имя	Описание	Использование	Примечание
Dshm	Базовые модули для работы ПО СКАДИ	все	Только символьные файлы
Dsmodel	Базовые модули для работы ПО СКАДИ-МОДЕЛЬ	DsModel, DsModelInT	
Dsrm	Плагины встраиваемых задач	DsMM в режиме ldr_is_sequential=TRUE	Кодовые и символьные файлы. Головной модуль задачи DsModel называется DsrmModel.

Таблица 1-1 – Состав динамических системных модулей СКАДИ-МОДЕЛЬ

В таблице 1-2 приведен пример состава динамически загружаемых пользовательских плагинов в системе СКАДИ-МОДЕЛЬ.

Имя	Описание	Использование	Назначение
Dpgc	Обобщенные компоненты	DsModel, DsModelInT	Пообъектная обработка задвижек, насосов, регулирующих клапанов, АВР, блокировок
Dpmodel	Теплогидравлическая модель	DsModel, DsModelInT	Расчетная модель теплогидравлики по схеме
Dpxch	Плагин агента обмена	DsXServ	Передача/прием данных для сервера обмена внешнего тренажера

Таблица 1-2 – Пример состав пользовательских плагинов СКАДИ-МОДЕЛЬ

Пользовательские плагины СКАДИ-МОДЕЛЬ загружаются задачами DsModel (с головным модулем DsrmModel).

В таблице 1-3 приведен состав подкаталогов динамически загружаемых модулей. Описанные ниже подкаталоги располагаются внутри каталогов таблиц 1-1 и 1-2. Например, каталог обобщенных компонентов с исходными текстами Dpgc будет иметь при 32-битной компиляции Dpgc/Mod, Dpgc/Code, Dpgc/Sym.

Имя	Описание	Необходимое условие	Назначение
Mod	Исходные тексты программ	Только для прикладных программ	<Имя>.mob – текстовый формат <Имя>.odc – двоичный формат
GMod	Шаблон исходных текстов программ	При использовании генерации из шаблона	<Имя>.mob – текстовый формат <Имя>.odc – двоичный формат
Code Cbur	Кодовые файлы 32-битного ombc	Компиляция ombc	<Имя>.ocf – кодовый файл двоичного формата

Sym Sbur	Символьные файлы 32-битного ombc	Компиляция ombc	<Имя>.osf – символьный файл двоичного формата <Имя>.ouf – файл ограничений и импорта двоичного формата
Clur	Кодовые файлы 64-битного omlc	Компиляция omlc	<Имя>.o – объектный файл двоичного формата <Имя>.ll – LLVM-код текстового формата <Имя>.bc – LLVM-код двоичного формата
Slur	Символьные файлы 64-битного omlc	Компиляция omlc	<Имя>.ouf – файл ограничений и импорта двоичного формата
Clue	Кодовые файлы 32-битного omlc	Компиляция omlc	<Имя>.o – объектный файл двоичного формата <Имя>.ll – LLVM-код текстового формата <Имя>.bc – LLVM-код двоичного формата
Slue	Символьные файлы 32-битного omlc	Компиляция omlc	<Имя>.ouf – файл ограничений и импорта двоичного формата

Таблица 1-3 – Состав подкаталогов динамически загружаемых модулей

ПРИЛОЖЕНИЕ 2 – WO ОПЦИИ КОМАНДНОЙ СТРОКИ СЕРВЕРНОГО ПО СКОЯТ

Опции сервера управления доступом WoAccess приведены в таблице 2-1.

Имя	Тип	Описание	Значения	По умолчанию
cd	string	Путь стартовой директории		.
db	string	Имя драйвера СУБД	none|sqlite	none
dsn	string	Database source name Имя источника данных		sdb
port	int	Имя порта	Порт в ОС	3000
trace_level	int	Уровень трассировки	Таблица 3.1	0
with_tcp6	bool	Использовать tcp6 при необходимости	false|true	false

Таблица 2-1 – Перечень опций управления доступом WoAccess

Опции сервера доступа к данным WoData приведены в таблице 2-2.

Имя	Тип	Описание	Значения	По умолчанию
cd	string	Путь стартовой директории		.
clear_cycle	duration	Интервал цикла обновления	1h|2m|3s	1m
clear_from	duration	Предел времени хранения	1h|2m|3s	24h
jcfg	string	Имя файла конфигурации	*.json	WoDocker.json
port	int	Имя порта	Порт в ОС	3100
trace_level	int	Уровень трассировки	Таблица 3.1	0
with_tcp6	bool	Использовать tcp6	false|true	false

Таблица 2-2 – Перечень опций доступа к данным WoData

Опции сервера контроля действий обучаемого WoMark приведены в таблице 2-3.

Имя	Тип	Описание	Значения	По умолчанию
cd	string	Путь стартовой директории		.
clear_from	duration	Предел времени хранения	1h|2m|3s	24h
download_after	duration	Время задержки выгрузки	1h|2m|3s	5s
interval	duration	Интервал цикла задания	1h|2m|3s	1s
jcfg	string	Имя файла конфигурации	*.json	WoDocker.json
n_resources	int	Максимальное количество ресурсов	[1, максимум-в-лицензии[	50
nstate_interval	duration	Интервал смены состояния	1h|2m|3s	3s
ping_count	int	Максимальное число пингов при старте ресурса	>= 1	10

ping_interval	duration	Интервал пингования ресурса	1h|2m|3s	3s
port	int	Имя порта	Порт в ОС	3200
touch_interval	duration	Интервал признания просроченным	1h|2m|3s	1h
trace_level	int	Уровень трассировки	Таблица 3.1	0
with_tcp6	bool	Использовать tcp6	false|true	false

Таблица 2-3 – Перечень опций контроля действий обучаемого WoMark

Перечень актуальных опций сервиса выводится посредством вызова Name –help (например, WoData --help).

ПРИЛОЖЕНИЕ 3 – ОПЦИИ ТРАССИРОВКИ ПО

Для всех процессов СКОЯТ и СКАДИ-МОДЕЛЬ можно динамически устанавливать и менять уровень трассировки. Трассировка СКОЯТ представлена в таблице 3-1.

Бит	Значение	Наименование	Обозначение
0	1	Цикл обновления	TRAT_RUNTIME
1	2	Транспортный уровень	TRAT_TRANSPORT
2	4	Состояние поллинга	TRAT_POLLING
3	8	Логический уровень сообщений	TRAT_MESSAGE
4	16	Детализация уровня сообщений	TRAT_MESSAGE_DATA
5	32	Ошибки коммуникации	TRAT_ERRORS
-1	FFFF	Все опции	

Таблица 3-1. Список значений трассировки СКОЯТ

Изменение переменной `Task:@T.trace_level` отслеживается динамически задачами СКАДИ-МОДЕЛЬ. Трассировка для задач СКАДИ-МОДЕЛЬ задается следующими способами:

- из командной строки указанием `–trace_level 34`
- в конфигурационном файле `trace_level=1022`
- динамически из отладчика. Добавить переменную `Task:#0.trace_level`, задать значение `1=36`

Значение величины трассировки представляет собой сумму значений бит, согласно таблице 3-2.

Бит	Значение	Коммуникация	Наименование	Обозначение
0	1	нет	Цикл обновления	Tb.TRAL_RUNTIME
1	2	да	Транспортный уровень	Tb.TRAL_TRANSPORT
2	4	да	Состояние поллинга	Tb.TRAL_POLLING
3	8	да	Логический уровень сообщений	Tb.TRAL_MESSAGE
4	16	да	Детализация уровня сообщений	Tb.TRAL_MESSAGE_DATA
5	32	да	Ошибки коммуникации	Tb.TRAL_ERRORS
6	64	нет	Ошибки задачи	Sb.TRAL_ERRORS
7	128	нет	Уровень обработки в задаче	Sb.TRAL_RUNTIME
8	256	нет	Детализация уровня обработки	Sb.TRAL_AUX

9	512	нет	Данные уровня обработки	Sb.TRAL_DATA
10	1024	нет	Времена обработки	Sb.TRAL_TIMES

Таблица 3-2. Список значений трассировки

Для примера, значение 1022 означает установленные биты {1..9} из таблицы. Все проверки на вхождение бита завершатся включением консольной печати или вывода в лог сообщений для Tb.TRAL_TRANSPORT, Tb.TRAL_POLLING, Tb.TRAL_MESSAGE, Tb.TRAL_MESSAGE_DATA, Tb.TRAL_ERRORS, Sb.TRAL_ERRORS, Sb.TRAL_RUNTIME, Sb.TRAL_AUX, Sb.TRAL_DATA.

Консольная печать, требуемая для отладки, включается после установки соответствующих опций трассировки. Следует заметить, что при необходимости трассировки инициализации задачи, нужно установить опцию trace_level в самом начале конфигурационного файла. Например trace_level=1 в начале конфигурационного файла позволяет трассировать разбор параметров, заданных в конфигурационном файле непосредственно за опцией трассировки.

ПРИЛОЖЕНИЕ 3 – СИСТЕМНЫЕ РЕЛЯЦИОННЫЕ ТАБЛИЦЫ

Перечень системных реляционных таблиц приведен в таблице 3-1.

Структуры данных системных глобалов приведены в таблице 3-1.

#	Структура	Глобал	Описание	Примечание
5	Const	Consts	Целочисленные константы	Целочисленные 32 бит
6	Characteristic	Characteristics	Характеристики системы взаимодействия	Только для информации
7	User	Users	Номера пользователей	Только для информации
8	Task	Tasks	Задачи СКАДИ	Текущее состояние процессов на узле
9	HGroup	HGroups	Группы хостов	
10	Host	Hosts	Таблица хостов	Текущее состояние хоста и хостов
11	XGroup	XGroups	Группы агентов обмена	
12	Xch	Xchs	Таблица агентов обмена	Текущее состояние сервера и агентов обмена
13	CSet	CSets	Параметры групп отображения	
14	Cell	Cells	Ячейки групп отображения	
15	Report	Reports	Задачи отчетов	Состояние печати отчетов
16	Repdata	Repdatas	Данные отчетов	Текущие текстовые данные отчетов
17	WSet	WSets	Набор областей управления	
18	WData	WDatas	Данные областей управления	
19	VFile	VFiles	Таблица файлов выборки	
20	FVar	FVars	Переменные файлов выборки	
21	Alarm	Alarms	Глобал сигнализаций	Массивы структур на основе E16B
22	Arch	Archs	Таблица архивов	
23	GStat	GStats	Статистика глобалов	Все, что не вожло в глобал 0

Таблица 3-1 – Данные системных глобалов

Имена колонок обозначают номер глобала, имя типа, имя глобала, описание и примечание.

ПРИЛОЖЕНИЕ 4 – WODBSH ОПЦИИ КОМАНДНОЙ СТРОКИ УТИЛИТЫ ДОСТУПА К БД

Опции командной строки утилиты WoDbsh приведены в таблице 4-1.

Имя	Тип	Описание	Значения	По умолчанию
tmp_dir	bool	Создание и удаление временной директории		false
db_file	string	Имя файла dsn источника данных		sdb
exec_cmd	string	Выполняемая команда модификации	insert delete update	
exec_file	string	Файл выполняемой команда модификации	insert delete update	
query_cmd	string	Выполняемая команда запроса	select	
query_file	string	Файл выполняемой команда запроса	select	
delim	string	Символ – разделитель данных		“ “ пробел

Таблица 4-1 – Перечень опций утилиты доступа к БД WoDbsh

ПРИЛОЖЕНИЕ 5 – DSVIEW ОПЦИИ КОМАНДНОЙ СТРОКИ УТИЛИТЫ ПРОСМОТРА ОБЩЕЙ ПАМЯТИ

Утилита просмотра общей памяти DsView запускается из консоли с опциями согласно таблице 5-1.

Имя	Описание	Значение	Примечание
global	Номер глобала	<Число>	
oid	Фильтр по oid	<Число>	В сочетании с limit устанавливает начальный номер записи
limit	Максимальное число строк	<Число>	0 – (по умолчанию) не лимитировано
globtype	Тип глобала	<Символ>	‘.’ означает печать заголовков всех глобалов
globname	Имя глобала	<Строка>	
locname	Имя локала	<Строка>	Позволяет смотреть разделяемую память, созданную автономными задачами
fld	Поле или список полей для форматирования	<Строка>	fld=’_’ означает табличное форматирование. Для Syslogs fld=0,1,2,3,4,5,6 – фильтр по коду важности
fld_page	Интервал печати заголовка	<Число>	
otype	Фильтр по типу объекта	<Строка>	
raw	Вывод необработанных данных	<Символ>	Любое значение - Да
seqonly	Только заголовки последовательностей	<Символ>	Любое значение - Да
ndb	Номер каталога с БД	<Число> 1..N	db1/ db2/ …

Таблица 5-1 – Перечень опций DsView

ПРИЛОЖЕНИЕ 6 – ПРЕДСТАВЛЕНИЕ ЗНАЧЕНИЙ СТРУКТУР

Командная структура C8B представляет собой запись следующего вида.

```
C8B* = RECORD
      content*, command_status*,
      program_id*, user_id*: Tb.Int8;
      ival*: Tb.Int32;
END;
```

Список полей структуры C8B приведен в таблице 6-1.

Поле	Наименование	Значения	Примечания
content	Содержимое команды	c, C, i, I, m, M	Управление I4, управление R4, имитация I4, имитация R4, менеджмент I4, менеджмент R4
command_status	Статус команды	<Число>	0-INIT 1-PUT 2-GOT 3-XS_ACCEPT 4-XS_REJECT 5-XCH_ACCEPT 6-XCH_REJECT
program_id	Идентификатор программы	<Число>	Task:@T.program_id
user_id	Идентификатор пульта	<Число>	
ival	Целочисленное значение	<Число>	

Таблица 6-1. Список полей структуры C8B

Строковое представление C8B записывается как: `<content>user_id@program_id:command_status,ival`

Пример в программе печати val='c1@0:1,4'.

Преобразование структуры C8B в строку осуществляется функцией SO.C8B2Str, обратное преобразование в структуру – функцией SO.Str2C8B.

Структура сигнализаций E16B представляет собой запись следующего вида.

```
E16B* = RECORD
      content*, dst_mask*, sev_group*, log_stat*: Tb.Int8;
      evcs*, prevcs*, condition*, pre_condition*: Tb.Int8;
      u*: Tb.Int64;
END;
```

Список полей структуры E16B приведен в таблице 6-2.

Поле	Наименование	Значения	Примечания
content	Содержимое структуры	<Число>	1-ALM отклонение 2-MAN ручной ввод
dst_mask	Адресат сигнализаций	<Число>	Номер адреса
sev_group	Класс важности	<Число>	2-Предупредительная 3-Аварийная 4-Неотложная
log_stat	Протоколирование сигнализаций	<Число>	
evcs	Статус сигнализаций	<Число>	
prevcs	Предыдущее значение статуса сигнализаций	<Число>	
condition	Значение качества параметра	<Число>	Код качества
pre_condition	Предыдущее значение качества	<Число>	Код качества

u	Вариантное представление	<Вариант>	Метка времени для content=1,2

Таблица 6-2. Список полей структуры E16B

Строковое представление E16B записывается как: `content/dst_mask/sev_group/log_stat:prevcs>evcs:pre_condition>condition`

Пример в программе печати val='2/1/2/0:27>31:52>52'.

Преобразование структуры E16B в строку осуществляется функцией SO.E16B2Str, обратное преобразование в структуру – функцией SO.Str2E16B.

Ссылка на объект VariableRef представляет собой запись следующего вида.

```
VariableRef* = RECORD
      atype*, otype*: Sb.VarType;
      dim*: Tb.Int16;
      index*: Tb.Int16;
      oid*, attrid*: Tb.Int32;
END;
```

Список полей структуры VariableRef приведен в таблице 6-3.

Поле	Наименование	Значения	Примечания
atype	Тип атрибута	<Тип>	-1 означает последовательность DataSeq
otype	Тип объекта	<Тип>	
dim	Размерность	<Число>	Отрицательное значение означает номер элемента
index	Индекс параметра	<Число>	
oid	Идентификатор объекта	<Число>	
attrid	Номер атрибута	<Атрибут>	

Таблица 6-3. Список полей структуры VariableRef

Строковое представление VariableRef записывается как:

`#otype,oid,attrid[,dim].atype[iindex]`

Параметры dim и index могут быть опущены, перед номером индекса должна быть буква i. Указание отрицательной размерности означает выборку только данного элемента. Пример в программе печати ‘ref=#27,0,253,-4.10’.

Преобразование структуры VariableRef в строку осуществляется функцией SO.Vref2Str, обратное преобразование в структуру – функцией SO.Str2Vref.

При удаленной передаче на агенты обмена идентификация изменяется с номеров типа otype и атрибута atype на числовое представление их имен. Для этого именам типов и атрибутов устанавливаются контрольные суммы, которые передаются через структуру VariableRef. Вместо индекса типа otype значение устанавливается в контрольную сумму Type:#otype.crc16. Вместо индекса атрибута atype значение устанавливается в контрольную сумму Attr:#atype.crc32.

Например, ссылка вида #-29494,10,1191740757.13i10 будет соответствовать объекту DI:#10.dv.

ПРИЛОЖЕНИЕ 7 – ПОСЛЕДОВАТЕЛЬНОСТИ ПЕРЕМЕННЫХ И ПАРАМЕТРОВ

Очередь сообщений представляет собой кольцевую структуру, содержащую записи последовательностей Seq и Data, рисунок 7-1.

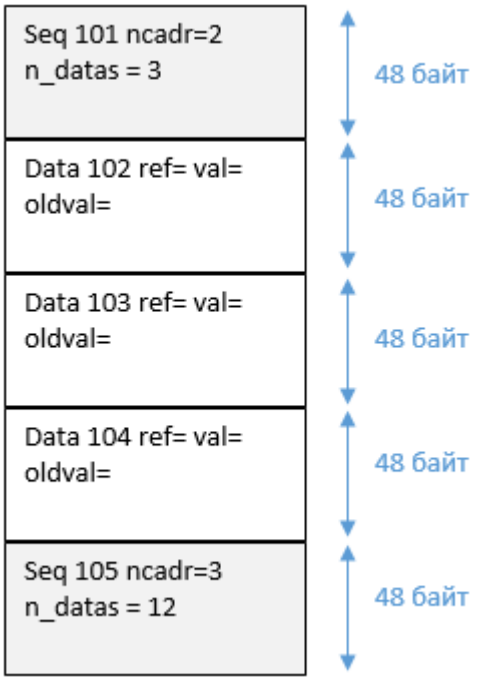


Рисунок 7-1 Очередь сообщений в виде кольцевого буфера

Последовательность имеет головную структуру Seq и следующие за ней n_datas элементов Data.

Список полей структур Data приведен в таблице 7-1.

Поле	Наименование	Значения	Примечание
ref	Ссылка на объект	<Ссылка>	Стандартный тип VREF
val	Текущее значение	<Значение>	Универсальный контейнер значений
oldval	Предыдущее значение	<Значение>	Аналогично

Таблица 7-1. Список полей структуры Data

Список полей структур Seq приведен в таблице 7-2.

Поле	Наименование	Значения	Примечание
n_datas	Число данных последовательности	<Число>	
atype	Тип атрибута	<Тип>	
srvstatus	Статус сервера	<Число>	Enum SRV_STAT
source_id	Идентификатор источника данных	<Число>	
subscr_id	Идентификатор подписки	<Число>	

ncadr	Номер кадра	<Число>	
flags	Флаги передачи	<Число>	Множество
clock	Текущее время	<Время>	
srvclock	Время сервера	<Время>	
oldclock	Предыдущее время	<Время>	

Таблица 7-2. Список полей структуры Seq

Последовательность данных упаковывается в приведенные структуры. Поля структуры Seq относятся ко всей последовательности. Поля структуры Data относятся к каждому элементу данных.

Все структуры располагаются в общей памяти, представляющей собой массив данных фиксированной длины 48 байт. При этом и структура данных Seq, и Data получились одинакового размера. Различение осуществляется по параметру ref.atype, который равен -1 для типа Seq.

Флаги передачи данных являются целочисленным представлением множества, каждый элемент которого есть определенный флаг, согласно таблице 7-3.

№	Имя	Описание
0	SENDALL	Передача полного набора
1	TRIM	Неполные данные
2	HOST_LOCAL	Данные по локальному хосту
3	GAP	Пропуск данных
4	HEAD	Данные заголовков
5	REFRESH	Посылка уже отправлявшихся данных
6	RESEND	Повторная пересылка данных
7	UPDATED	Обновление данных
8	VAL	Значения переменных процесса
9	PARM	Значения параметров процесса
10	LOCAL	Значения локальных величин
11	CMD	Значения команд управления
12	STATE_CMD	Значения команд смены статуса
13	TPACKED	Упакованная по T алгоритму посылка
14	CONTINUED	Продолжение данных в последовательности
15	PARM_UPDT	Обновление параметров
16	PACKED16	Упакованная в 16 бит посылка
17	PACKED32	Упакованная в 32 бита посылка
18	WDATA	Данные по динамической подписке
19	WDU	Управление динамической подпиской
20	GRP	Данные группы
21	XCH	Данные агентов обмена
22	CRC_REF	Наличие контрольных сумм
23	CRC_DONE	Контрольные суммы обработаны
24	AREC	Данные в формате архивной записи
25	ECHO	Эхо последовательность принятой команды
26	NOECHO	Эхо последовательность отвергнутой команды
27	ATINY	Минимальная архивная запись 8 байт
28	ASMALL	Малая архивная запись 16 байт

29	AMIDDLE	Средняя архивная запись 24 байт
30	AFULL	Полная архивная запись 48 байт
30	AQFULL	Полная архивная запись для очередей

Таблица 7-3. Таблица флагов передачи данных

Данные очередей сообщений и передачи по сети группируются в последовательности с источником и временной меткой. Во флаги передачи данных записываются дополнительная информация о характере описываемой последовательности данных. Эта информация содержит флаги, из которых следует, что передаваемая последовательность содержит переменные или параметры или команды управления, и т.д.

ПРИЛОЖЕНИЕ 8 – DSACCESS ОПЦИИ КОМАНДНОЙ СТРОКИ УТИЛИТЫ ДОСТУПА И КОНФИГУРИРОВАНИЯ

Перечень возможных опций утилиты DsAccess приведен в таблице 8-1.

Опция	Название	Значения	Примечание
setvars	Имена переменных	<Строка>	Одно или несколько через пробел в кавычках
getvars	Имена переменных	<Строка>	Аналогично
vals	Значения переменных	<Строка>	Аналогично
val	Значение переменных	<Строка>	
quals	Коды качества	<Числа> или <Строки>	Одно или несколько через пробел в кавычках
end	Завершение задачи	<Строка>	all или имя задачи
init	Старт задачи	<Строка>	Имя задачи
modify	Запуск пакета обновления	<ИмяФайла>	db/<modfile>.tsd
control	Команда конфигурирования	<Строка>	
wdata	Вывод необработанных данных	<Символ>	
global	Имя глобала	<Строка>	
sync	Флаги очереди изменений	val|parm|local	Одно из значений
ndb	Номер каталога с БД	<Число> 1..N	db1/ db2/ …

Таблица 8-1 – Перечень опций DsAccess

ПРИЛОЖЕНИЕ 9 – DSISD ОПЦИИ И КОМАНДЫ ОТЛАДЧИКА ЗНАЧЕНИЙ

Перечень возможных опций утилиты DsISD приведен в таблице 9-1.

Опция	Название	Значения	Примечание
start_page	Имя страницы с переменными	<строка>	Имя файла wrk/<start_page>.pgg
ndb	Номер каталога с БД	<Число> 1..N	db1/ db2/ …

Таблица 9-1 – Перечень опций DsISD

Перечень возможных команд отладчика приведен в таблице 9-2.

Команда	Аргументы	Назначение
setq	Номер переменной – целое число, значение признака качества – целое число	Установить переменной с заданным номером заданное значение признака качества
setts	Номер переменной – целое число	Установить переменной с заданным номером значение метки времени, соответствующее времени в системе
cle, clear	Номер переменной – целое число	Удалить переменную с заданным номером из набора отслеживаемых переменных
load, page	Имя набора переменных - строка	Загрузить набор переменных, ранее сохраненный с заданным именем
save	Имя набора переменных - строка	Сохранить текущий набор переменных с заданным именем
ifrz	Нет	Приостановить обновление значений в отладчике
irun	Нет	Возобновить ранее приостановленное обновление значений в отладчике
frz	Нет	Установить статус в режим FREEZE
run	Нет	Установить статус в режим SINGLE
snap	Номер начального состояния	Записать начальное состояние под номером 1-99
reset	Номер начального состояния	Восстановить начальное состояние под номером 1-99
n, next	Нет	Перейти на следующую страницу набора переменных. Например, если в данный момент отображаются переменные с номерами от 1 до 20, то при переходе на следующую страницу будут отображаться переменные с номерами от 21 до 40 при их наличии в наборе. Иначе команда игнорируется
p, prev	Нет	Перейти на предыдущую страницу в наборе переменных. Если страница первая в наборе (переменные с 1 по 20), то команда игнорируется
end	Нет	Остановить все задачи сервера
failover	Нет	Имитировать неисправность сервера

Таблица 9-2 – перечень команд DsISD

ПРИЛОЖЕНИЕ 10 – DSQSD ОПЦИИ И КОМАНДЫ ОТЛАДЧИКА ОЧЕРЕДЕЙ

Перечень возможных опций утилиты DsQSD приведен в таблице 10-1.

Опция	Название	Значение	Примечание
globname1, globname2, globname3, globname4	Имя глобальной очереди сообщений	<Строка>	По умолчанию – все глобалы q
locname1, locname2, locname3, locname4	Имя локальной очереди сообщений	<Строка>	Не используется
varname1, varname2, varname3	Имя отслеживаемой переменной	<Строка>	
start_page	Имя первоначально загружаемой страницы – набора отслеживаемых переменных	<Строка>	Страницы совместимы с DsISD

Таблица 10-1 – перечень опций DsQSD

Перечень возможных команд приведен в таблице 10-2.

Команда	Аргументы	Назначение
cle, clear	Номер переменной – целое число	Удалить переменную с заданным номером из набора отслеживаемых переменных
load, page	Имя набора переменных - строка	Загрузить набор переменных, ранее сохраненный с заданным именем
save	Имя набора переменных - строка	Сохранить текущий набор переменных с заданным именем
save	Нет	Сохранить текущий набор переменных с текущим именем
ifrz	Нет	Приостановить обновление значений
irun	Нет	Возобновить ранее приостановленное обновление значений
frz	Нет	Установить статус в режим FREEZE
run	Нет	Установить статус в режим SINGLE
n, next	Нет	Перейти на следующую страницу набора переменных. Например, если в данный момент отображаются переменные с номерами 1, 2, 3, то при переходе на следующую страницу будут отображаться переменные с номерами 4, 5, 6 при их наличии в наборе. Иначе команда игнорируется
p, prev	Нет	Перейти на предыдущую страницу в наборе переменных. Если страница первая в наборе (переменные 1, 2, 3), то команда игнорируется
q, quit, x, exit	Нет	Выход из консоли

Таблица 10-2 – перечень команд DsQSD

ПРИЛОЖЕНИЕ 11 – ПЕРЕЧЕНЬ СУФФИКСОВ РАСШИРЕНИЯ ОБОРУДОВАНИЯ

Перечень суффиксов расширения для VLV-запорной арматуры приведен в таблице 11-1.

Суффикс	Назначение сигнала	Значения	Примечание
Z0	Сигналы обратной связи ОТКРЫТО, ЗАКРЫТО от исполнительного механизма	0-ПРОМЕЖУТОЧНОЕ 1-ЗАКРЫТО 2-ОТКРЫТО	Отображение состояния задвижки
F0	Обобщенный сигнал неисправности	Биты неисправностей	Отображения рамки неисправности
B0	Эхо дистанционной команды	1-ЗАКРЫТЬ 2-ОТКРЫТЬ	
OU	Команда управления	1-ЗАКРЫТЬ 2-ОТКРЫТЬ	Привязка к кнопкам ОТКРЫТЬ, ЗАКРЫТЬ окна управления

Таблица 11-1 – перечень суффиксов расширения VLV

Перечень суффиксов расширения для CNV-регулирующих клапанов приведен в таблице 11-2.

Суффикс	Назначение сигнала	Значения	Примечание
Z0	Сигналы обратной связи ОТКРЫТО, ЗАКРЫТО от исполнительного механизма	1-ЗАКРЫТО 2-ОТКРЫТО	Отображение состояния регулирующего клапана
F0	Обобщенный сигнал неисправности	Биты неисправностей	Отображения рамки неисправности
B0	Эхо дистанционной команды	1-ДИСТАНЦИЯ 2-АВТОМАТ 4-ЗАКРЫТЬ 8-ОТКРЫТЬ	
OU	Команда управления	1-ДИСТАНЦИЯ 2-АВТОМАТ 4-ЗАКРЫТЬ 8-ОТКРЫТЬ	Привязка к кнопкам ДИСТАНЦИЯ, АВТОМАТ, ОТКРЫТЬ, ЗАКРЫТЬ окна управления
XQ08	Указатель положения клапана	0-100%	Привязка к индикатору положения

Таблица 11-2 – перечень суффиксов расширения CNV

Перечень суффиксов расширения для PMP-насосов и TEN-тэнов приведен в таблице 11-3.

Суффикс	Назначение сигнала	Значения	Примечание
Z0	Сигналы обратной связи ОТКЛЮЧЕН, ВКЛЮЧЕН от механизма	0-РАЗОБРАН 1-ОТКЛЮЧЕН 2-ВКЛЮЧЕН	Отображение состояния механизмов
F0	Обобщенный сигнал неисправности	Биты неисправностей	Отображения рамки неисправности

B0	Эхо дистанционной команды	1-ОТКЛЮЧИТЬ 2-ВКЛЮЧИТЬ	
OU	Команда управления	1-ОТКЛЮЧИТЬ 2-ВКЛЮЧИТЬ	Привязка к кнопкам ОТКЛЮЧИТЬ, ВКЛЮЧИТЬ окна управления

Таблица 11-3 – перечень суффиксов расширения PMP, TEN

В приложении указаны основные суффиксы отображаемых единиц оборудования. Перечень суффиксов изменяется в сторону расширения.

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

АЭС - Атомная Электростанция

КСА - Комплекс средств автоматизации

ОС - Операционная система

ПО - Программное обеспечение

СВБУ - Система верхнего блочного уровня

СКАДИ - Средства комплексной автоматизации и диагностики

ЧМИ - Человеко-машинный интерфейс

ICMP - Internet Control Message Protocol

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

ЛИСТ ИЗМЕНЕНИЙ

Изм.	Номера листов				Всего листов в изме-ненном документе	Номер документа, утверждаю-щего изменение	Подп.	Дата
	измененных	замененных	новых	аннулированных				