



Негосударственное частное образовательное учреждение высшего
образования
«Технический университет УГМК»

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ
ПО ДИСЦИПЛИНЕ**

**АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ
ПРОИЗВОДСТВОМ**

Специальность	<u>21.05.04 Горное дело</u>
Направленность (профиль)	<u>Подземная разработка рудных месторождений</u>
Уровень высшего образования	<u>Специалитет</u> <i>(бакалавриат, специалитет, магистратура)</i>

Автор - разработчик: Худяков П.Ю., канд. физ.-мат. наук

Рассмотрено на заседании кафедры механики и автоматизации технологических процессов и производств

Одобрено Методическим советом университета 30 июня 2021 г., протокол № 4

г. Верхняя Пышма
2021

ИНТЕГРАЦИЯ SCADA TRACE MODE С ПРОЕКТОМ SIMATIC

Цель работы

Изучить настройку сервера OPC для реализации HMI в TRACE MODE. Произвести настройку сервера OPC и реализацию функций управления технологическим процессом через экран оператора.

1. ОБЩИЕ СВЕДЕНИЯ

SCADA системы сейчас, являются неотъемлемой частью современных автоматизированных систем управления процессами или как ее еще называют средой визуализации. Внедрение SCADA систем приводит к существенному удешевлению эксплуатации вторичного оборудования на крупных объектах, путем переноса индикации и накопления технологической информации на пульт диспетчера АРМ.

Не смотря на большое разнообразие SCADA систем на рынке, большинство из них имеет примерно одинаковый набор функциональных возможностей позволяющих выполнять основные требования, предъявляемые к верхнему уровню АСУ ТП. Набор стандартных функций в SCADA системах обусловлен общим кругом задач при разработке систем автоматизации.

Состав основных функций позволяющих выполнить полноценный проект по автоматизации:

- *графический интерфейс* позволяет упростить задачу построения и отображения технологического процесса (ТП). К графической части можно отнести возможность упрощенного или детализированного отображения объектов ТП, средств измерения физических параметров технологических объектов (ТО). Кроме того позволяет отображать кнопки, индикаторы, панели стрелочных или цифровых индикаторов, регуляторов и других вторичных приборов которые раньше располагались на панели шкафа автоматизации. Поддержка библиотек изображений и видео позволяющая выводить графическую информацию сторонних разработчиков на графическую панель SCADA системы, такие как элементы мнемосхем, динамические объекты.

- *архив измерений, событий и аварийных ситуаций*, происходящих на ТО, с отображением изменений информации в окне временного тренда.

- упрощенный язык составления алгоритмов управления ТП, математических вычислений.

- *драйвера устройств и оборудования согласованной работы со SCADA системой*, находящихся на нижнем и среднем уровнях АСУ ТП, такие как датчики, вторичное оборудование контроллеры.

- *поддержка других языков программирования* высокого уровня.

- средства защиты от несанкционированного доступа к файлам и компонентам.

Проекты, разработанные на базе Trace Mode, имеют инсталляции в энергетической, металлургической, атомной, нефтяной, газовой, химической, космической и других отраслях промышленности. Нашли применение при разработке АСДУ ЖКХ и сельском хозяйстве России.

Trace Mode - это первая интегрированная информационная система для управления промышленным производством, объединяющая в едином целом продукты класса SoftLogic-SCADA/HMI-MES-HRM. SCADA система Trace Mode разработана в 1992 году и к настоящему времени имеет более 7000 внедрений на объектах АСУ ТП.

В состав системы входят бесплатные драйверы для более чем 2-х тысяч контроллеров и УСО.

Для программирования алгоритмов управления технологическими процессами в SCADA системе Trace Mode поддержаны все 5 языков международного стандарта IEC 61131-3 такие как - Techno FBD, Techno LD, Techno SFC и процедурные - Techno ST, Techno IL.

Система разработки Trace Mode содержит ряд новых технологий проектирования АСУТП, отличающих ее от других SCADA-систем. Среди них следующие:

обеспечение единых инструментальных средств (единой линии программирования) как для разработки операторских станций, так и для программирования контроллеров;

- разработка распределенной АСУТП как единого проекта;
- технология автопостроения проекта.

Единая линия программирования

Введены функции программирования контроллеров и выпущена специальная исполнительная система для контроллеров – Микро MPB.

Среди встроенных алгоритмов – ПИД, ПДД, нечеткое, позиционное регулирование, динамическая балансировка, алгоритмы массового обслуживания, блоки моделирования объектов, арифметические, алгебраические, тригонометрические, статистические функции, функции расчета технико-экономических показателей и т.д. Добавлен ряд функциональных блоков, ориентированных на контроль и управление типовыми технологическими объектами (клапан, задвижка, привод и т.д.). Также разработаны алгоритмы адаптивных и модальных регуляторов.

1.1. Разработка сетевого комплекса как единого проекта

В Trace Mode распределенная АСУ, включающая ПК и контроллеры, *рассматривается как один проект*. Поэтому каждый узел (ПК или контроллер) имеет информацию об остальных узлах системы и в случае его модификации автоматически обновляет соответствующие базы на других узлах. При этом АСУ можно создавать как в архитектуре клиент-сервер, так и распределенную – технология разработки АСУТП как единого проекта будет одинаково эффективна.

«Автопостроение»

«Автопостроение» – это группа оригинальных технологий, реализованных в Trace Mode. Суть автопостроения заключается в автоматическом генерировании баз каналов операторских станций и контроллеров, входящих в проект АСУТП на основе информации о числе точек ввода /вывода, номенклатуре используемых контроллеров и УСО, наличии и характере связей между ПК и контроллерами. В соответствии с этим реализованы следующие возможности:

автопостроение баз каналов для связи с УСО в РС-контроллерах. Достаточно указать марку и количество РС-контроллеров, используемых в проекте и запустить автопостроение – Trace Mode сформирует базу каналов для каждого контроллера и произведет настройку на УСО автоматически.

- автопостроение баз каналов для связи с обычными контроллерами. Эта процедура

автоматически генерирует базы каналов операторских станций и производит настройку на наиболее распространенные контроллеры.

- автопостроение связей между узлами «ПК-ПК», «ПК-контроллеры», «контроллеры-контроллеры» осуществляет создание, автоматическое поддержание и обновление коммуникаций между узлами распределенной АСУТП.

- автопостроение при импорте баз технологических параметров. Часто для этих целей используются распространенные базы данных и электронные таблицы, например, Excel, Access и др. Trace Mode допускает импорт баз данных с последующим автопостроением.

1.2. Архитектура Trace Mode

Все программы, входящие в Trace Mode, подразделяются на две группы, рис.1: инструментальную систему разработки и исполнительные модули (runtime). Как видно из рисунка, **инструментальная система** разработки содержит три редактора: редактор базы каналов, редактор представления данных, редактор шаблонов.

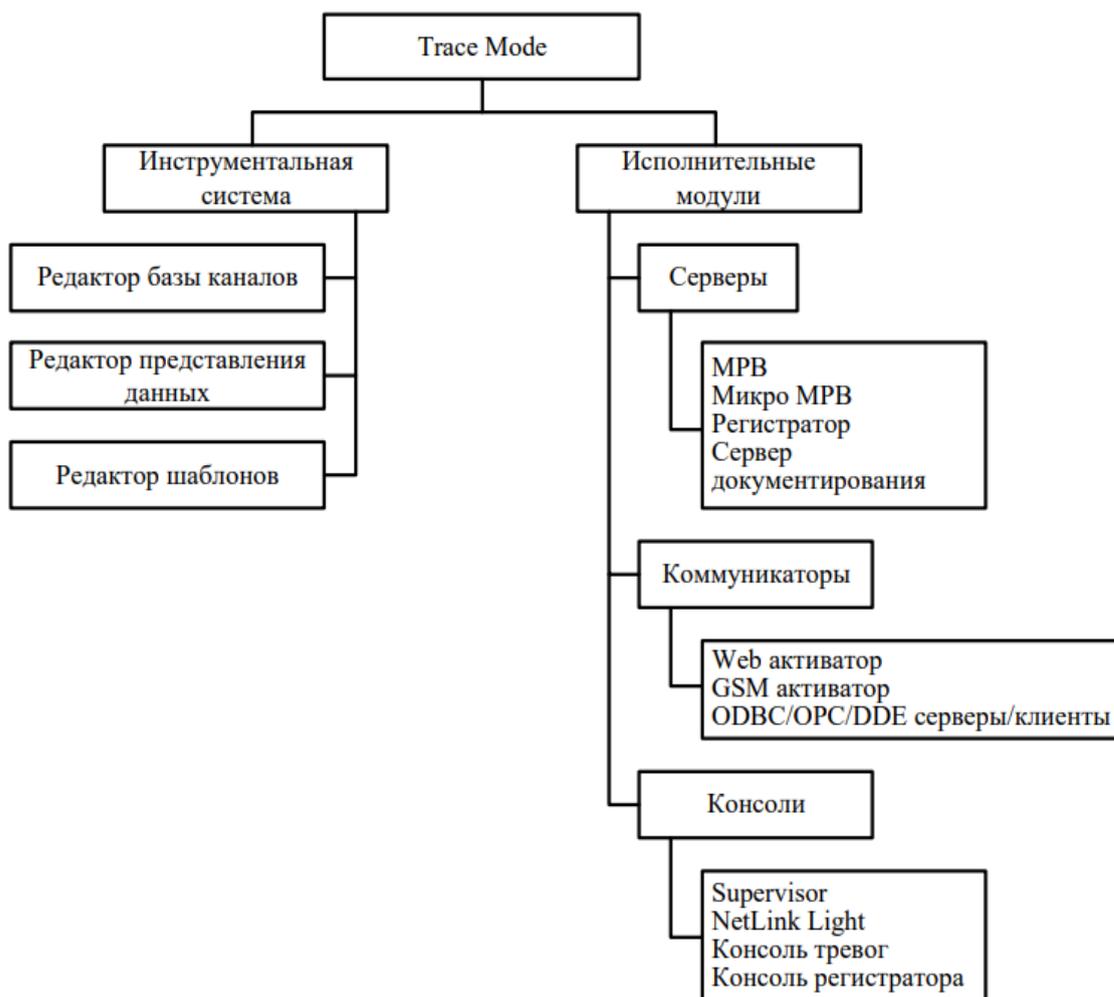


Рис. 1. Архитектура Trace Mode

В редакторе базы каналов создается математическая основа системы управления: описываются конфигурации всех рабочих станций, контроллеров и УСО, а также настраиваются информационные потоки между ними. Здесь же описываются входные и выходные сигналы и их связь с устройствами сбора данных и управления; задаются периоды опроса или формирования сигналов, настраиваются законы первичной обработки и управления, технологические границы, программы обработки данных и управления,

осуществляется архивирование технологических параметров, сетевой обмен, а также решаются некоторые другие задачи.

Результатами работы в этом редакторе являются математическая и информационная структуры проекта АСУТП, которые включают в себя набор баз каналов и файлов конфигурации для всех контроллеров и операторских станций (узлов) проекта, а также файл конфигурации всего проекта с расширением **cmt** (для версии 6 расширение - **prj**). Все остальные файлы проекта хранятся в рабочей директории в каталоге, имя которого совпадает с именем файла конфигурации.

В редакторе представления данных разрабатывается графическая часть проекта системы управления. Сначала создается статичный рисунок технологического объекта, а затем поверх него размещаются динамические формы отображения и управления. Среди этих форм присутствуют такие, как поля вывода числовых значений, графики, гистограммы, кнопки, области ввода значений и перехода к другим графическим фрагментам и т. д.

Кроме стандартных форм отображения, Trace Mode позволяет вставлять в проекты графические формы представления данных или управления, разработанные пользователями. Все формы отображения информации, управления и анимационные эффекты связываются с информационной структурой, разработанной в редакторе базы каналов. Для разработки шаблонов документов в состав инструментальной системы включен **редактор шаблонов**.

Исполнительная система Trace Mode включает в себя **исполнительные модули (мониторы, МРВ)** – программные модули различного назначения, под управлением которых в реальном времени выполняются составные части проекта, размещаемые на отдельных компьютерах или в контроллерах, предназначенные для работы на всех уровнях систем управления, о которых говорилось выше.

Существует ряд программных модулей, назначение которых четко не привязано к функциям одного из перечисленных уровней систем управления. К таким модулям относятся, рис.1:

- глобальный регистратор;
- сервер документирования;
- Web-активатор;
- GSM-активатор.

Они могут использоваться для создания как оперативного, так и административного уровней систем управления.

Глобальный регистратор служит для обеспечения надежного хранения архивов ТП. Он архивирует данные, посылаемые ему посетителями мониторами реального времени (64 000 параметров с дискретностью 0,001 с), обеспечивает автоматическое восстановление данных после сбоя, а также может передавать архивные данные для просмотра мониторами SUPERVISOR. Глобальный регистратор может также выступать как OPC-сервер и DDE-сервер и поддерживает обмен с базами данных через ODBC. Для документирования технологической информации в Trace Mode предусмотрен специальный модуль - **сервер документирования**. Документирование осуществляется по шаблонам, которые создаются в **редакторе шаблонов**. Время или условие генерирования документа, имя файла шаблона, а также направление вывода документа описываются в программах документирования - сценариях. Подготовка отчетов (документов) чаще всего привязывается к астрономическому времени. Например, они могут генерироваться один раз в час, один раз в сутки, один раз в месяц или один раз в десять минут. Кроме того, можно установить режим подготовки документа один раз в смену и затем описать разбивку суток на смены.

Сервер документирования NetLink Light используется для решения задачи документирования технологической информации. Он по команде МРВ, собственному сце-

нарию или по команде оператора интерпретирует созданные заранее шаблоны, запрашивает у МРВ необходимые данные и формирует по ним документы. Эти документы могут быть распечатаны на принтере, отправлены по E-mail или опубликованы на Web- сервере.

Утилита **консоль тревог** позволяет просматривать отчет тревог разных МРВ одного проекта. Для каждого просматриваемого отчета тревог создается отдельное окно. В него можно выводить информацию из файла отчета тревог или сообщения, формируемые МРВ.

Любая рабочая станция системы Trace Mode может выступать в качестве Web-сервера, что позволяет управлять технологическим процессом через интернет. На удаленном компьютере необходимо иметь только доступ к сети интернет и Web-браузер. Для реализации данного режима предназначен модуль **Web-активатор**, который используется в качестве www-шлюза для локальных систем АСУ ТП на базе Trace Mode или для придания функций Web-сервера мониторам реального времени. Использование Web-активатора позволяет быстро превратить существующие АСУТП и АСУП в Internet/Intranet-системы без переделки баз данных реального времени (баз каналов). Доступ к данным реального времени через Web-активатор осуществляется при помощи обыкновенного браузера, работающего под любой операционной системой, позволяющей запуск виртуальной Java-машины. Информация о технологическом процессе представляется пользователю в виде анимированных мнемосхем, трендов и таблиц.

Связь с серверами реального времени Trace Mode может осуществляться практически любыми доступными средствами, например через сотовую сеть стандарта GSM, инфракрасный порт, сеть на основе интерфейса RS-232/485 или модем с использованием высоконадежного протокола TCP/IP. Можно осуществлять подключение и непосредственно через Internet. Для этого достаточно войти в Internet и набрать IP-адрес сервера Trace Mode – подключение произойдет автоматически. Для доступа к данным пользователю достаточно набрать Web-адрес активатора и ввести пароль, тогда весь проект загружается в удаленный компьютер в виде Java-аппрета. Использование стандартного языка Java при написании аппретов позволяет реализовать на удаленных компьютерах не только Windows, но и другие операционные системы, например Unix, Linux, Mac OS и т. д., а также ОС, использующиеся в карманных PC. Проект Trace Mode поступает к пользователю в виде Java-аппрета, объем которого не превышает 300 Кбайт, что дает возможность использовать Web-активатор в сетях с низким качеством связи. Достоинством технологии Java является также повышенная безопасность.

При использовании Web-активатора не требуется установка Web-серверов других производителей (например, MS IE), что выгодно отличает эту программу от решений, примененных в других SCADA.

Для обеспечения мобильных пользователей АСУ оперативной информацией в режиме реального времени на базе Trace Mode разработан программный продукт - **GSM-активатор**. Он предназначен для дистанционного мониторинга и управления технологическими процессами, а также для получения оперативной технико-экономической информации при помощи сверхпортативных компьютеров handheld PC.

В реальном времени GSM-активатор может принимать информацию от 64 000 датчиков, осуществлять супервизорное управление, получать технико-экономическую информацию из баз данных через сервер, использующий стандартные интерфейсы SQL/ODBC.

OPC, DDE и т. д. Вся входящая информация отображается графически в виде анимированных мнемосхем и трендов.

GSM-активатор, относящийся к новому классу систем оперативного управления, отражающих мировую тенденцию к миниатюризации и автономизации компьютерных систем, может быть использован в качестве персональной информационной системы руководителя.

В последней версии Trace Mode все редакторы системы вызываются из одной программы - **Интегрированной среды разработки (ИС)**. ИС – единая программная оболочка, содержащая все необходимые средства для разработки проекта.

Все переменные проекта, к чему бы они ни относились - к контроллеру, к операторской станции, к управлению техобслуживанием или производством **хранятся в единой базе данных проекта**. Единая база проекта устраняет лишнюю работу проектировщика по созданию, поддержке и взаимной увязке во многом одинаковых баз переменных контроллеров и ПК, характерную для систем предыдущего поколения.

Логическая структура проекта полностью отделена от аппаратной части. Благодаря **единому пространству распределенных переменных**, переменные из разных узлов могут связываться между собой также легко, как и в пределах одного узла, любые изменения, вносимые в объект, *автоматически применяются везде*, где он был задействован.

2. РАЗРАБОТКА ИНТЕРФЕЙСА В TRACE MODE

Проектирование систем автоматизации в Trace Mode может быть осуществлено несколькими способами:

- проектирование «от шаблонов»;
- проектирование от технологии;
- проектирование от топологии;
- смешанное проектирование.

Разработка любого проекта автоматизации всегда начинается с запуска интегрированной среды разработки. Для ее запуска необходимо выполнить команду TRACE MODE IDE 6 (base) из группы установки инструментальной системы в меню «Программы» WINDOWS или двойным щелчком ЛК мыши по иконке рабочего стола.

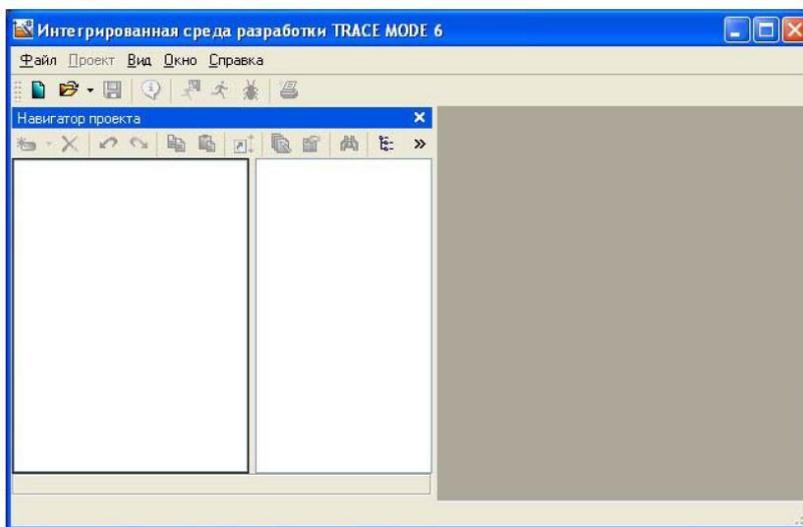


Рис.2. Интегрированная среда разработки

После запуска ИСР в меню «Файл» выбрать команду «**Настройки ИС...**». В появившемся окне выбрать «**Уровень сложности**» и настроить как показано на рис. 3, а затем выбрать «**Отладка**» и настроить как на рис.4.

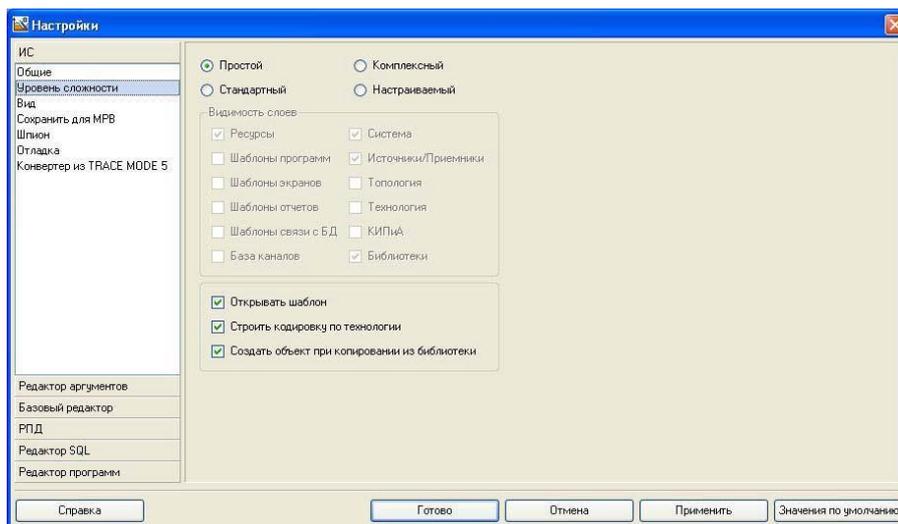


Рис.3. Настройки интегрированной среды разработки

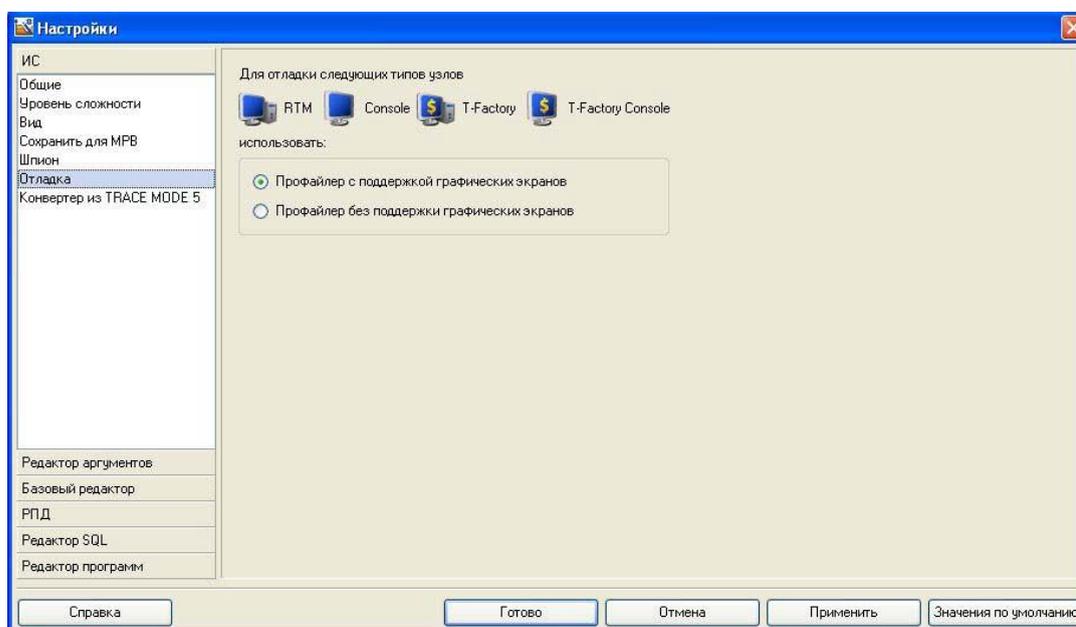


Рис. 4. Настройки интегрированной среды разработки

После проведенных настроек интегрированной среды разработки нажать кнопку «Готово». С помощью инструментальной панели создать новый проект, при этом в открывшемся на экране диалоге рис.5.

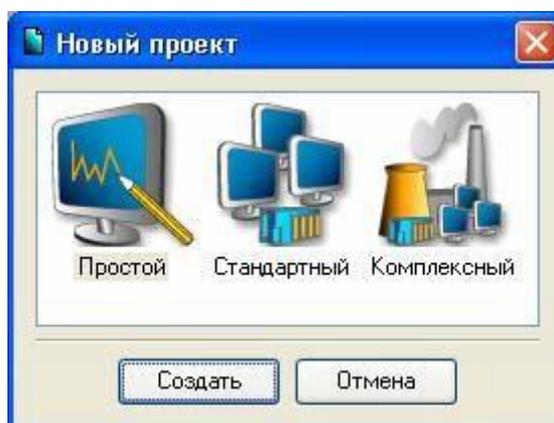


Рис. 5. Создание нового проекта

Уровни сложности определяют количество слоев, доступных при разработке проекта. Ниже перечислены слои каждого из уровней сложности.

Простой:

- ресурсы;
- система;
- источники/приемники;
- библиотека.

Стандартный:

- ресурсы;
- система;
- источники/приемники;
- библиотека;
- шаблоны экранов;
- шаблоны отчетов;
- шаблоны связи с БД.

Комплексный:

- ресурсы;
- система;
- источники/приемники;
- библиотека;
- шаблоны экранов;
- шаблоны отчетов;
- шаблоны связи с БД;
- топология;
- технология;
- КИПиА.

Также доступен «Настраиваемый» уровень сложности, который позволяет самостоятельно выбрать необходимые слои. Создание нового проекта производится в следующей последовательности.

1. Выберем «Простой» стиль разработки. После нажатия левой клавиши мыши (ЛК) на экранной кнопке «Создать», в левом окне «Навигатора проекта» появится дерево проекта с созданным узлом APM RTM_1 рис. 6.

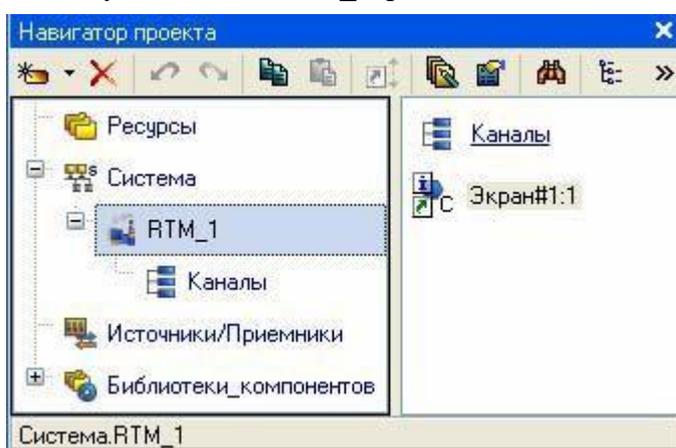


Рис. 6. Навигатор проекта

2. Откроем узел RTM_1 двойным щелчком ЛК, в правом окне «Навигатора проекта» отобразится содержимое узла – пустая группа «Каналы» и один канал класса «Вызов Экран#1», предназначенный для отображения на узле АРМ графического экрана.

В навигаторе структура проекта представлена в виде дерева, рис. 7.

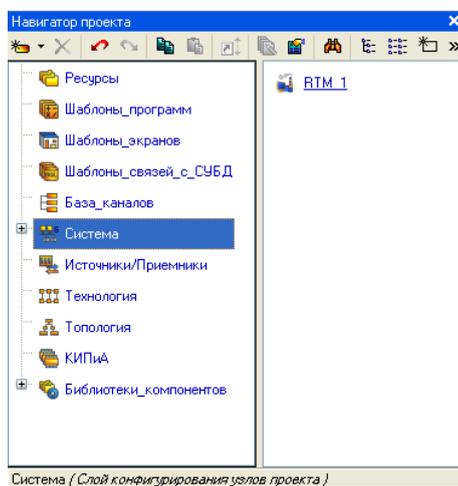


Рис.7. Дерево проекта

Корневые группы этого дерева (*слои*) predeterminedены и создаются автоматически при создании нового проекта (слои отображаются в левом окне навигатора). Элементарные структурные составляющие (листья структурного дерева) называются *компонентами* проекта. Например, компонентами проекта являются: канал; канал, вызывающий шаблон; шаблон; источник данных и т.д.

Группы компонентов, которые могут быть созданы в структуре проекта, предназначены для структурирования проекта.

В ряде случаев группы имеют и другой, вполне определенный, смысл – например, узлы проекта создаются как корневые группы слоя «Система».

В правом окне навигатора отображается содержимое слоя (группы), выделенной в левом окне, – таким образом, компоненты проекта могут быть отображены только в правом окне.

Структура проекта редактируется в навигаторе с помощью команд меню «Проект», контекстного меню и панелей инструментов, а также с помощью метода drag-and-drop.

2.1. Технология разработки проекта в интегрированной среде

Разработка проекта в ИС включает следующие процедуры:

- создание структуры проекта в навигаторе;
- конфигурирование или разработка структурных составляющих – например, разработка шаблонов графических экранов оператора, разработка шаблонов программ, описание источников/приемников и т.д.;
- конфигурирование информационных потоков;
- выбор аппаратных средств АСУ (компьютеров, контроллеров и т.п.);
- создание узлов в слое «Система» и их конфигурирование;
- распределение каналов, созданных в различных слоях структуры, по узлам и конфигурирование интерфейсов взаимодействия компонентов в информационных потоках;

- сохранение проекта в единый файл для последующего редактирования (с помощью команды «**Сохранить**» или «**Сохранить как**»);
- экспорт узлов в наборы файлов для последующего запуска под управлением мониторов Trace Mode (по команде «**Сохранить для МРВ**»).

Перечисленные процедуры (за исключением двух заключительных) и входящие в их состав операции могут выполняться в произвольном порядке. Например, можно начинать разработку проекта с разработки шаблонов графических экранов оператора, с создания узлов и их каналов в слое «**Система**» (если аппаратные средства АСУ известны заранее), можно конфигурировать каналы и информационные потоки после распределения каналов по узлам и т.п.

2.2. Классификация компонентов

По функциональному назначению компоненты проекта относятся к одному из следующих видов:

каналы – компоненты, определяющие алгоритм работы. Каналы могут создаваться в различных слоях, однако их окончательное распределение по узлам в слое «**Система**» обязательно – в противном случае они не будут экспортированы для МРВ;

шаблоны – компоненты, которые при работе в реальном времени могут вызываться каналами с передачей параметров. Передача параметров настраивается при разработке проекта в ИС посредством привязки *аргументов* шаблона к каналам или источникам/приемникам;

источники/приемники – компоненты, тем или иным способом описывающие внутренние переменные различных устройств или приложений, с которыми требуется обмениваться данными. Под устройствами здесь понимаются контроллеры, а также внешние и внутренние модули/платы различного назначения, обмен с которыми поддерживается мониторами Trace Mode (в том числе через драйверы). Системные переменные Trace Mode также создаются в ИС как источники/приемники. Источники/приемники являются шаблонами каналов;

наборы ресурсов – наборы текстов, изображений и видеоклипов, которые могут быть использованы при разработке шаблонов графических экранов;

графические объекты – компоненты, представляющие собой в общем случае несколько графических элементов (из имеющихся в редакторе представления данных), сгруппированных в один. Графические объекты могут быть использованы при разработке шаблонов графических экранов;

последовательные порты – параметры СОМ-портов;

словари сообщений – наборы сообщений, генерируемых при возникновении различных событий;

клеммы – эти компоненты, описывающие электрические контакты (например, монтажных шкафов), являются элементами схемы электрических соединений АСУ.

2.3. Классификация слоев

Предопределенные слои структуры проекта имеют следующее назначение:

«**Ресурсы**» – для создания пользовательских наборов текстов, изображений и видеоклипов, а также графических объектов;

«**Шаблоны программ**» – для создания шаблонов программ;

- «**Шаблоны экранов**» – для создания шаблонов графических экранов;
- «**Шаблоны связей с БД**» – для создания шаблонов связей с базами данных;
- «**Шаблоны документов**» – для создания шаблонов документов (отчетов);

«**База каналов**» – этот слой является хранилищем всех каналов проекта. Выполнять операции с каналами (в том числе создавать их) можно в различных слоях, однако во всех случаях эти операции на самом деле реализуются в слое «**База каналов**». В любом другом слое, где выполняется команда для совершения операции с каналом, ее результат только отображается – поэтому существуют команды удаления и уничтожения каналов. В слое «**База каналов**» можно начинать разработку проекта;

«**Система**» – для конфигурирования узлов и их составляющих (узел создается как корневая группа этого слоя);

«**Источники/приемники**» – для создания описаний источников/приемников в различных устройствах и программных приложениях, обмен с которыми поддерживается мониторами, а также для конфигурирования системных переменных;

«**Технология**» – для разработки проекта от технологии (т.е. с группировкой компонентов по признаку их принадлежности к технологическому объекту). При отладке проекта слой «**Технология**» может играть роль узла – для него определена команда «**Сохранить узел для МРВ**». Кроме того, для этого слоя определены команды взаимодействия с технологической базой данных;

«**Топология**» – для разработки проекта от топологии (т.е. с группировкой компонентов по месту расположения);

«**КИПиА**» – для описания электрических соединений АСУ;

«**Библиотеки компонентов**» – для создания библиотек *объектов* – проектных решений отдельных задач. Этот слой содержит predetermined группы «**Системные**» и «**Пользовательские**». В группе «**Системные**» содержатся библиотеки, подключенные к ИС по умолчанию.

2.4. Классификация узлов

Узлы проекта создаются как корневые группы слоя «**Система**». Предetermined название узла указывает на семейство мониторов, для которых данный узел predetermined. Узел может содержать только те компоненты, которые поддерживаются мониторами соответствующего семейства.

В общем случае, узлы могут выполняться под управлением различных мониторов. Как правило, узел выполняется на отдельном аппаратном средстве. В случае запуска двух и более узлов на одном аппаратном средстве оно должно быть оборудовано соответствующим количеством сетевых карт.

Параметры узлов задаются в соответствующем редакторе.

Узел **RTM** предназначен для запуска на компьютере под управлением исполнительных модулей семейства **RTM** (МРВ) – мониторов с поддержкой отображения графических экранов оператора, поддержкой обмена по последовательному интерфейсу и сети с различным оборудованием и выполняющего пересчет каналов всех классов, кроме каналов T-FACTORY.

Узел **T-FACTORY** предназначен для запуска на компьютере под управлением исполнительных модулей семейства **T-FACTORY** – мониторов для решения задач АСУП.

Узел **MicroRTM** предназначен для запуска на компьютере или в контроллере под управлением исполнительных модулей семейства **Micro RTM**. Основное отличие этих мониторов от МРВ – отсутствие поддержки отображения графических экранов.

Узел **Logger** предназначен для запуска на компьютере под управлением исполнительного модуля **Logger** (регистратор) – монитора, способного вести архивы по каналам всех узлов проекта.

Узел **EmbeddedRTM** предназначен для запуска на компьютере или в контроллере под управлением исполнительных модулей семейства **Embedded RTM** – мониторов с поддержкой мнемосхем, поддержкой обмена с оборудованием по различным протоколам и выполняющего пересчет каналов.

Узел **NanoRTM** предназначен для запуска в контроллере под управлением исполнительного модуля **Nano RTM** – монитора, аналогичного **Micro RTM**, но предназначенного для работы с малым числом каналов.

Узел **Console** предназначен для запуска на компьютере под управлением исполнительных модулей, которые, в отличие от МРВ, не выполняют пересчет каналов, предназначенных для работы с данными. Консоли позволяют получать данные от других узлов проекта по сети, отображать их на графических экранах и управлять технологическим процессом из графики. Консоли не могут взаимодействовать с узлами T-FACTORY.

Узел **TFactory_Console** предназначен для запуска на компьютере под управлением исполнительных модулей, аналогичных консолям, но, кроме того, способных взаимодействовать с узлами T-FACTORY.

Узел **TM OPC_Server** предназначен для запуска на компьютере под управлением OPC-сервера Trace Mode.

2.5. Сохранение проекта для запуска

По команде  «Сохранить для МРВ» меню «Файл» или панели инструментов ИС все узлы экспортируются в наборы файлов для их последующего копирования на аппаратные средства, на которых они должны исполняться под управлением мониторов Trace Mode.

При выполнении команды «Сохранить для МРВ» в директории сервера проекта создается поддиректория <имя файла **prj** без расширения>, в которой для каждого узла создается папка с набором файлов. Папка узла имеет имя <name>_<ordinal + 1>, где **name** – имя, заданное для узла при его конфигурировании в ИС, **ordinal** – порядковый номер узла в слое «Система», начиная с 0 (порядковый номер узла в слое не следует путать с его индивидуальным номером).

Необходимым условием экспорта узла является наличие в нем хотя бы одного канала. По команде «Сохранить узел для МРВ» из меню «Проект» или контекстного меню навигатора выделенный узел (или слой «Технология») экспортируется в произвольную папку, при этом при повторном экспорте резервные копии узла не создаются.

2.6. Создание объектов структуры

Для создания объектов структуры (компонентов и групп компонентов) используются типовые команды меню «Проект», контекстного меню и панели инструментов навигатора

При создании канала класса CALL с предустановленным свойством вызов в соответствующем слое шаблонов создается шаблон, вызываемый каналом. В навигаторе могут быть созданы следующие каналы класса CALL с предустановленным свойством **вызов**:

- «Экран» – канал с вызовом шаблона экрана;
- «Программа» – канал с вызовом шаблона программы;

- «Документ» – канал с вызовом шаблона документа;
- «Связь с БД» – канал с вызовом связи с базой данных.

Меню «Проект», контекстное меню и панель инструментов навигатора содержат команды создания только тех объектов, которые может содержать выделенный слой/группа.

При редактировании сохраненного проекта (в том числе после выполнения ко- манды «Сохранить»/ «Сохранить как») вновь созданные структурные объекты и объ- екты, их содержащие, выделяются в навигаторе синим шрифтом:

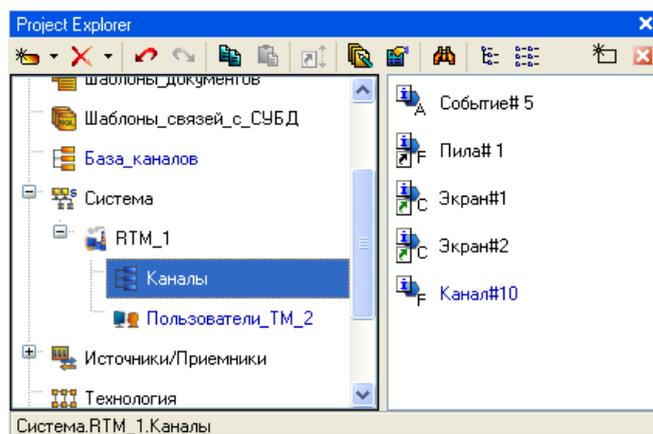


Рис.8. Навигатор проекта

2.7.

Операции с графическими элементами

Размещение графических элементов

Встроенные графические элементы разбиты на группы. Каждой группе соответствует кнопка на панели инструментов «Графические элементы». На кнопку выводится иконка одного из элементов данной группы.

Чтобы выбрать графический элемент для размещения, нужно выполнить следующие действия:

- нажать ЛК на кнопке панели инструментов «Графические элементы». При этом выбирается тот элемент, чья иконка выведена на кнопку (элемент, заданный по умолчанию для соответствующей группы, или элемент, выбранный ранее);
- дважды нажать ЛК на кнопке и затем нажать ЛК на иконке требуемого графического элемента в появившемся меню (меню не открывается, если группа содержит только один графический элемент).

После выбора элемента его иконка выводится на кнопку группы, рис.9.



Рис.9. Меню группы «Прямоугольники»

2.8. Задание типовых свойств графических элементов

В этом разделе описаны типовые свойства графических элементов и инструменты их задания.

Графические элементы имеют следующие настраиваемые свойства:

- атрибуты;
- динамические свойства;
- функции управления.

Эти параметры определяют вид графических элементов и выполняемые ими функции отображения и управления при работе в реальном времени.

Для задания свойств графических элементов (группы графических элементов) используется окно «**Свойства объекта**», содержащее различное число вкладок для разных элементов, рис. 10.



Рис.10. Окно «Свойства объекта»

Чтобы открыть это окно, нужно выполнить команду «**Свойства**» из контекстного меню выделенного графического элемента (выделенной группы графических элементов) или дважды нажать ЛК мыши на графический элемент. При снятии выделения графического элемента, окно его свойств автоматически закрывается. Окно «**Свойства объекта**» недоступно, если графический элемент расположен в слое, редактирование которого запрещено.

Атрибуты – это простейшие свойства графического элемента. Они задаются на вкладке «**Основные свойства**» окна «**Свойства объекта**».

В окне свойств атрибуты могут быть сгруппированы – наименования таких групп выделены подчеркиванием, при двойном нажатии на них ЛК раскрывается список свойств.

Существуют 2 вида атрибутов графических элементов:

- **статические** – атрибуты, которые не изменяются при работе в реальном времени;

- **динамизируемые** – атрибуты, которые могут быть как статическими, так и динамическими (изменяющимися при работе в реальном времени в зависимости от значения привязанного аргумента). Разделы конфигурирования таких атрибутов выделены красным цветом и содержат пункт «**Вид индикации**».

2.9. Динамизация атрибута графического элемента

Динамизацией атрибута называется задание условий его изменения в зависимости от значения привязанного аргумента. При динамизации атрибута графический элемент становится индикатором выполнения заданных условий.

При размещении графического элемента на экране все его динамизируемые атрибуты по умолчанию статические, и разделы их конфигурирования на вкладке «**Основные свойства**» окна свойств содержат инструмент задания соответствующего статического параметра.

Чтобы динамизировать атрибут, нужно дважды нажать на названии ЛК мыши, и в раскрывшемся списке настроить динамические свойства с помощью раздела «**Вид индикации**», рис.11.

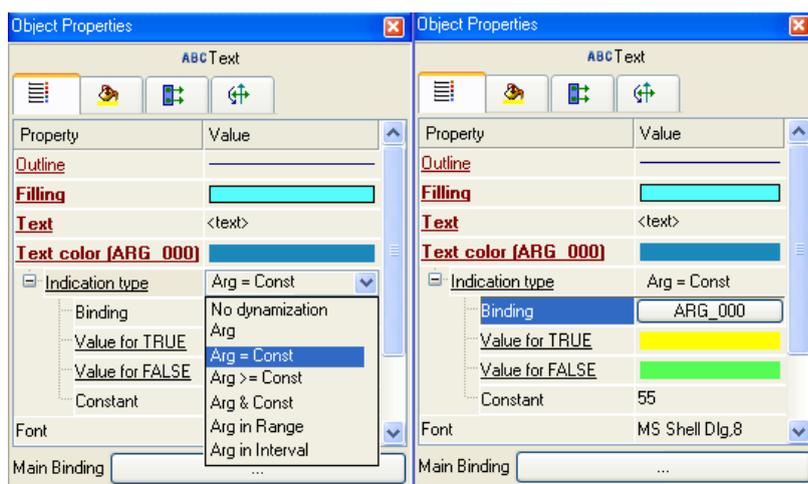


Рис.11. Список настройки динамических свойств

Вид условия (и, соответственно, вид индикатора, создаваемого из графических элементов), выбирается в разделе «**Вид индикации**»:

- **Значение** – индикация значения аргумента;
- **Arg = Констант.** – индикация равенства аргумента заданной константе;
- **Arg >= Констант.** – индикация превышения аргументом заданного порога;
- **Arg & Констант.** – индикация результата побитового умножения аргумента на значение заданной константы;
- **Arg в диапазоне** – индикация нахождения аргумента в заданных диапазонах;

-Arg в интервале – индикация нахождения аргумента в интервалах привязанного канала.

В зависимости от выбранного вида индикации меняются инструменты его конфигурирования.

Индикация значения

Вид индикации «**Значение**» может быть задан при динамизации атрибута «**Текст**». Формат вывода выбирается в списке «**Формат**».

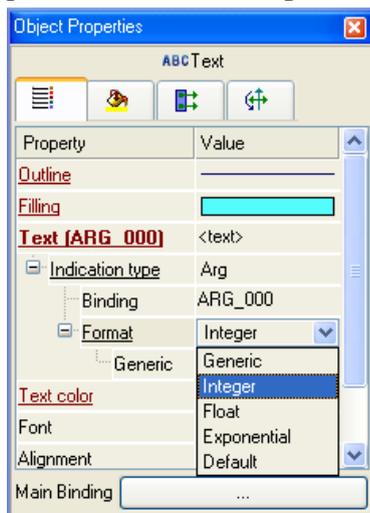


Рис.12. Настройка индикации значения

2.10. Динамические свойства графических элементов

К динамическим свойствам графических элементов относятся динамическая заливка, 3 вида динамической трансформации (перемещение, масштабирование и вращение) и динамический контур.

Динамические свойства графических элементов, как и динамизированные атрибуты, используются для графического отображения значений аргументов экрана при работе в реальном времени.

Динамические свойства настраиваются соответственно на вкладках «Динамическая заливка», «Динамическая трансформация» и «Динамический контур» окна «Свойства объекта».

Динамическая заливка графических элементов

При использовании данного свойства графический элемент отображает значение привязанного аргумента числового формата в виде закрашенной области (слоя). Поддерживаются два вида динамической заливки – одноуровневая (отображает значение одного аргумента) и многоуровневая (отображает значения нескольких аргументов). Оба вида настраиваются на вкладке «Динамическая заливка» окна «Свойства объекта». Для использования динамической заливки нужно на этой вкладке установить флаг «Использовать».

Можно задать несколько слоев заливки. Для добавления/удаления нового слоя используется контекстное меню, вызываемое нажатием ПК мыши на названиях пунктов «Слой/Слой» соответственно. Настройки для всех создаваемых слоев имеют одинаковое назначение.

Динамическое перемещение графических элементов

Это свойство настраивается в разделе «Перемещать» вкладки «Динамическая трансформация» окна «Свойства объекта», рис. 13.

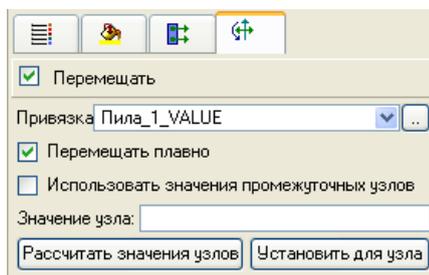


Рис.13. Настройка динамического перемещения

Чтобы использовать данное динамическое свойство, надо установить флаг «Перемещать».

При работе в реальном времени графический элемент перемещается вдоль траектории, которая задается как ломаная линия (количество узлов ломаной не ограничено). Текущее положение графического элемента зависит от значения привязанного аргумента (числовой аргумент для привязки выбирается в списке «Привязка», от значений, заданных для узлов траектории, и флага «Перемещать плавно».

Динамическое масштабирование графических элементов

Это свойство настраивается в разделе «Масштабировать» вкладки «Динамическая трансформация» окна «Свойства объекта», рис.14.

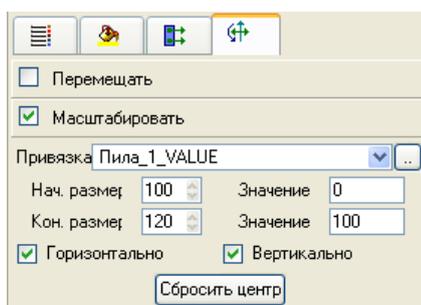


Рис.14. Настройка динамического масштабирования

Для перехода в этот раздел надо нажать ЛК на заголовке раздела. Чтобы данное динамическое свойство было использовано при работе в реальном времени, надо установить флаг «Масштабировать».

Применение свойства «Масштабировать» изменяет координаты точек графического элемента относительно **центра масштабирования**, который по умолчанию располагается в точке привязки графического элемента (в режиме редактирования центр масштабирования отображается на экране в виде красной точки). С помощью метода **drag-and-drop** центр масштабирования может быть перемещен в произвольную точку экрана. При нажатии кнопки «Сбросить центр» центр масштабирования устанавливается в точку пересечения диагоналей прямоугольника, ограничивающего графический элемент, рис. 15.

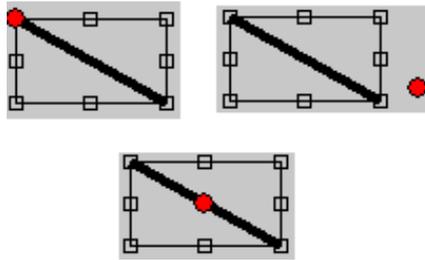


Рис.15. Применение свойства «Масштабировать»

Динамическое вращение графического элемента

Это свойство настраивается в разделе «Вращать» вкладки «Динамическая трансформация» окна «Свойства объекта», рис. 16.

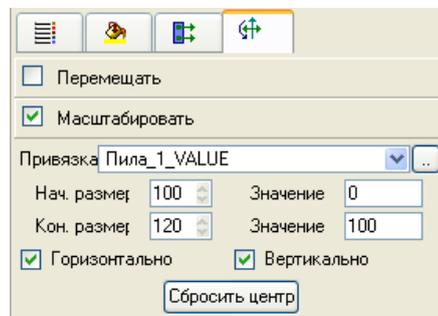


Рис.16. Настройка динамического вращения

Для перехода в этот раздел надо нажать ЛК на заголовке раздела. Чтобы данное динамическое свойство было использовано при работе в реальном времени, надо установить флаг «Вращать».

Точка, относительно которой вращается графический элемент (центр вращения), по умолчанию располагается в точке привязки графического элемента (в режиме редактирования данного динамического свойства центр вращения отображается на экране в виде красной точки). С помощью метода **drag-and-drop** центр вращения может быть перемещен в произвольную точку экрана (в том числе за пределами графического элемента). При нажатии кнопки «Сбросить центр» центр вращения устанавливается в точку пересечения диагоналей прямоугольника, ограничивающего графический элемент, аналогично рис. 15.

Динамический контур графического элемента

Динамический контур представляет собой прокручиваемый по часовой стрелке пунктир (под прокруткой здесь подразумевается дискретное перемещение с шагом, равным длине штриха). Это свойство настраивается на вкладке «Динамический контур» окна «Свойства объекта», рис. 17.

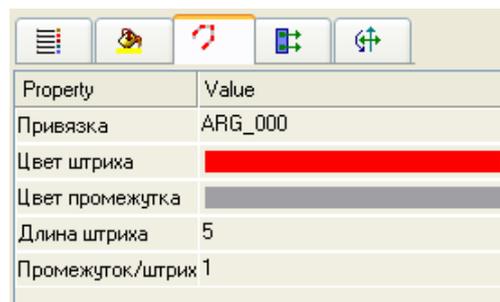


Рис. 17. Настройка динамического контура

4. РАЗРАБОТКА НОВОГО ПРОЕКТА НМИ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА

Порядок создания НМИ проекта после запуска TIA PORTAL включает следующие шаги:

1. Создание проекта АРМ.
2. Определение источников/приемников сигнала – определение таблицы символов
3. Конфигурирование экрана рабочего места.
4. Конфигурирование аргументов экрана

4.1. Создание проекта

Для создания нового проекта в интегрированной системе разработки Trace Mode, необходимо воспользоваться командой «Создать новый проект» в инструментальной панели, после чего в диалоговом окне необходимо выбрать проект «Простого уровня сложности». В навигаторе проекта можно будет увидеть дерево проекта, аналогичное представленному на рис. 18 и включающее в себя такие слои как:

- ресурсы;
- система;
- источники/приемники;
- библиотека компонентов.

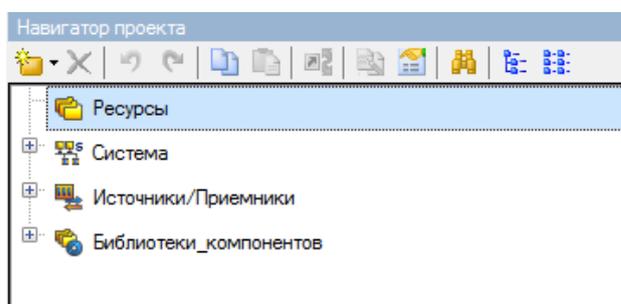


Рис. 18. Основное дерево проекта

Далее необходимо сконфигурировать Источник/Приемник сигналов, а именно добавить OPC сервер как канал обмена с различными устройствами. Данный канал обмена позволит осуществлять управление с моделью производственного процесса. Для этого необходимо в ветке «Источники/Приемники» создать группу OPC устройств, рис.19.

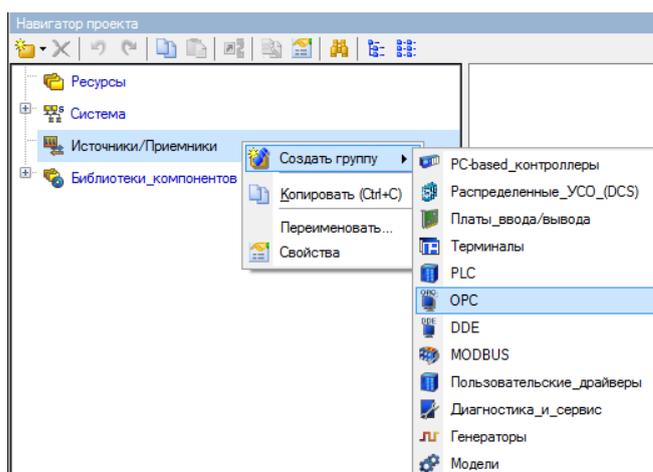


Рис. 19. Создание OPC группы устройств

Группа OPC устройств позволяет добавить в проект *OPC_Сервер*, как это показано на рис.20.

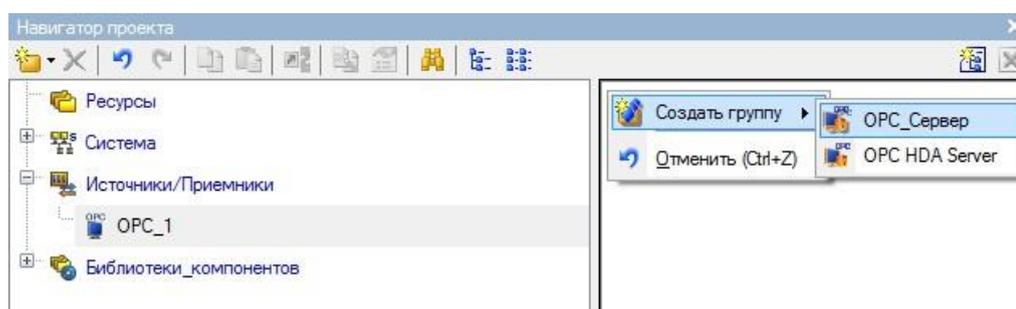


Рис. 20. Добавление OPC сервера к группе устройств

Добавленный канал обмена данными позволяет сконфигурировать компоненты, в свойствах которых имеется возможность установить слово состояния для OPC клиента, через которые будет происходить обмен данными для НМІ проекта.

Процесс работы с компонентами OPC сервера включает в себя:

- добавление OPC компоненты;
- выбор слово состояния;
- настройка свойств элемента.

Окно настройки свойств компоненты OPC сервера представлено на рис.21.

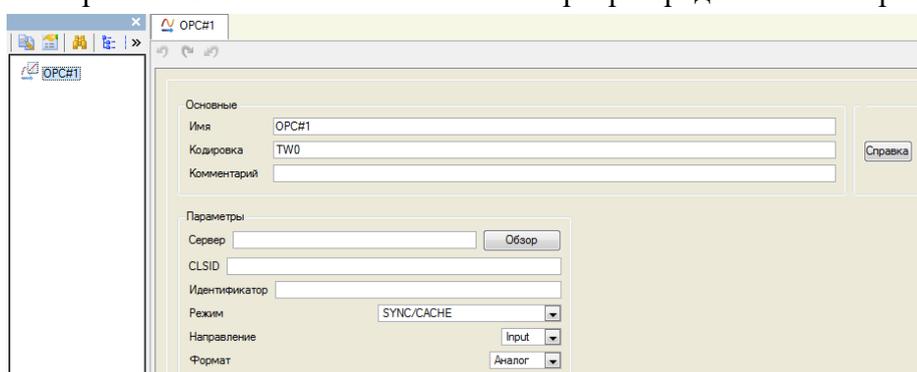


Рис. 21. Окно настройки свойств компоненты OPC сервера

Свойства канала OPC соединения включают в себя заполнение основных свойств наименования и комментарии к нему, а также параметры непосредственного самого *Сервера*, *Направление* и *Формат* сигнала.

В первую очередь для работы необходимо выбрать регистр того сервера, с которым будет происходить непосредственная работа. Для этого необходимо нажать кнопку «Обзор» в свойствах настраиваемого элемента и выбрать необходимое слово состояние, как это показано на рис.22.

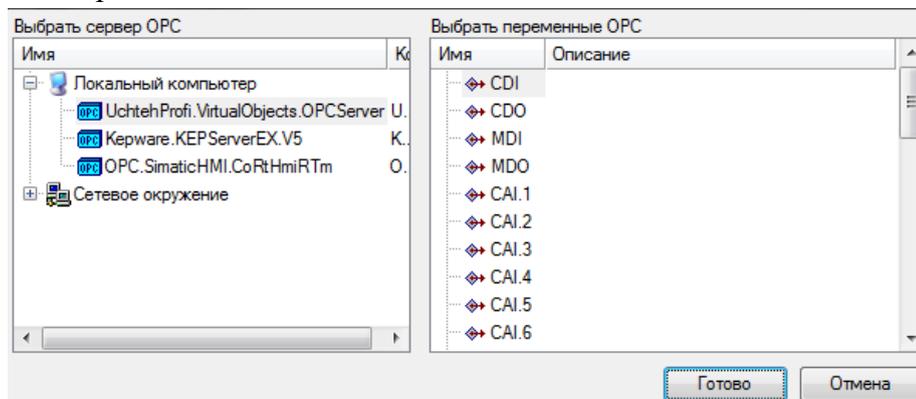


Рис. 22. Окно выбора переменных OPC сервера

Для работы с входными сигналами, необходимо выбрать среди OPC переменных дискретный вход **CDI**, и **CDO** для работы с выходными сигналами соответственно, выставляя при этом в свойствах соответствующее направление **Input/Output**. В итоге окно настроек для связывания с дискретным входом/выходом примет вид, представленный на рис.23.

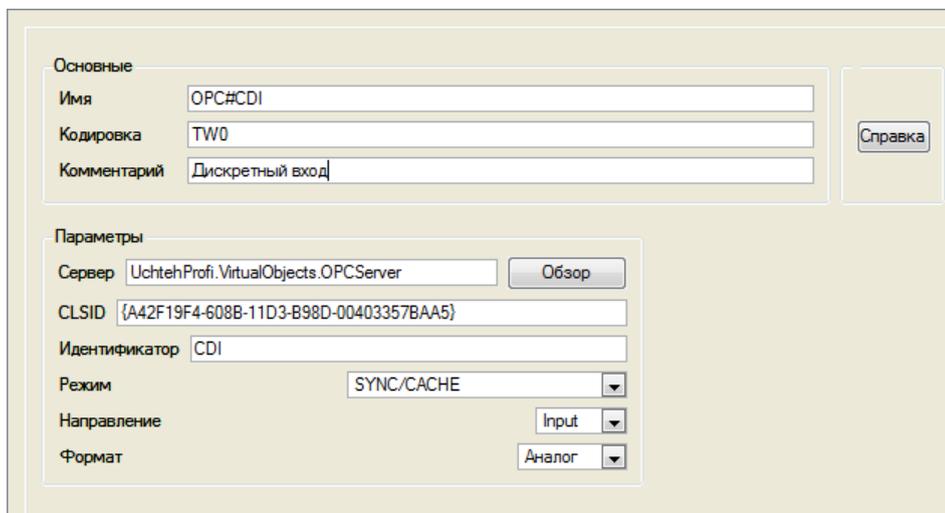


Рис. 23. Окно конфигурирования свойств компоненты «Дискретного входа»

Для создания рабочего места оператора НМІ системы необходимо сконфигурировать экран в проекте. Конфигурирование экранов происходит в ветке «Система»> «RTM»> «Экран» дерева проекта. Открытие конфигурации экрана и запуск редактора производится двойным щелчком мыши по выбранной ветке «Экран». На рис.25 представлен экран конфигурирования рабочего места.

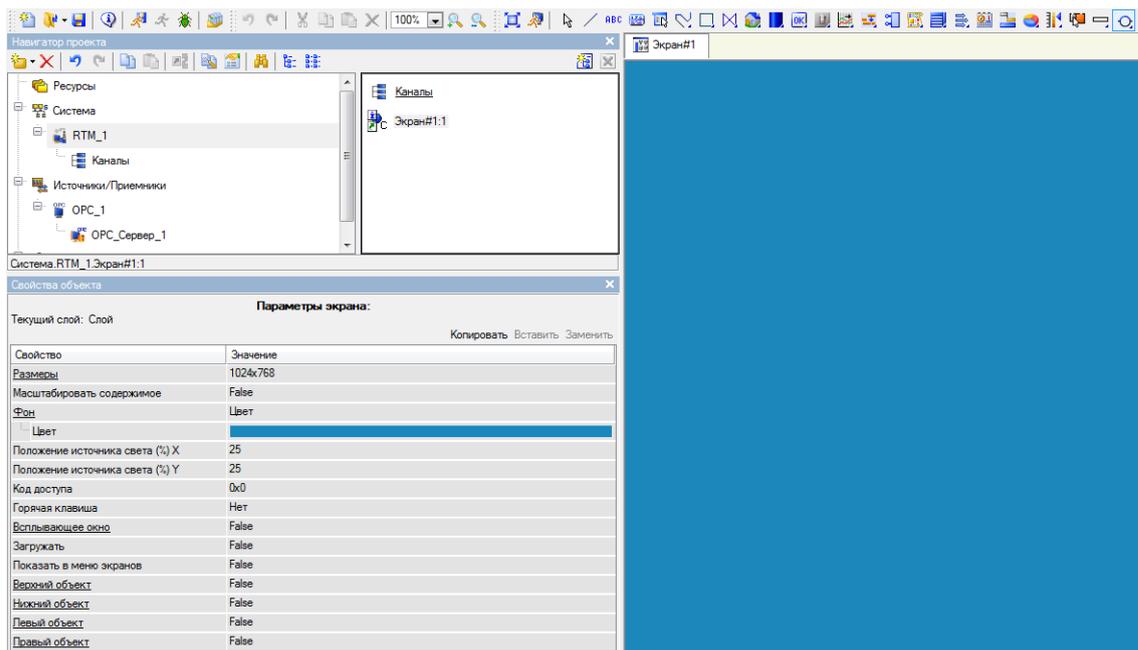


Рис.24. Окно конфигурации экрана HMI

Окно редактирования позволяет конфигурировать интерфейсную часть посредством предустановленных графических элементов, таких как простые геометрические фигуры, стандартные элементы управления и контроля, а также шаблоны готовых объектов. Для этого достаточно поместить вышеописанные элементы на конфигурируемый экран.

С помощью панели инструментов редактирования имеется возможность быстрого изменения цвета, расположения объектов, формирования надписей. Также для каждого из добавляемых в рабочую область объектов доступно быстрое изменение основных свойств.

Для связывания элементов экрана с входными/выходными сигналами OPC сервера, необходимо заполнить панель аргументов экрана. Для этого достаточно перетянуть элемент *OPC_Server* из дерева проекта на панель аргументов. В поле «*Tun*» устанавливается *In* и *Out* для элементов *CDI* и *CDO* соответственно, как показано на рис. 25.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка	Флаги	Группа	Ед.измерения	Комментарий	Кодировка
OPC_CDI_VALUE	IN	REAL		OPC#CDI:Значение (Источники/Приемники.OPC_1.OPC_Сервер_1)					
OPC_CDO_VALUE	OUT	REAL		OPC#CDO:Значение (Источники/Приемники.OPC_1.OPC_Сервер_1)					

Рис.25. Панель аргументов экрана

1.1. Работа HMI системы со сторонними программами

Для исследования работы программ управления и проверки функционирования алгоритма используются модели технологических процессов, позволяющие моделировать движение механизмов, контакты переключателей и концевых выключателей, цифровые табло и т.д. Для выполнения лабораторных работ разработан ряд 2-х мерных моделей, имитирующие такие производственные процессы как:

- роботизированный комплекс транспортировки изделий;
- методическая печь;
- участок сортировки и пакетирования;
- участок транспортировки труб;
- погрузка угля в вагоны.

Для исследования работы системы визуализации удобно использовать модель «Роботизированный комплекс транспортировки изделий». Данная модель позволяет управлять передвижной платформой с механической рукой с щупом для захвата и перемещений заготовок по транспортерам. Вид модели представлен на рис. 26.

Рабочая область комплекса содержит передвижную платформу с механической рукой и захватом. Платформа спроектирована таким образом, чтобы осуществлять перемещение по 2 осям – сама платформа перемещается вертикально вверх и вниз, а расположенная на ней механическая рука с захватом по горизонтали. Также имеются три транспортера, по 2 из которых поступают заготовки, которые необходимо переместить для дальнейшего производственного процесса на самый нижний транспортер посредством захвата деталей механической рукой. Для фиксации движения платформы, механической руки, захвата, детали по транспортерам, комплекс имеет концевые датчики, полный список которых можно посмотреть в справке приложения на ряду с сигналами управления.

После запуска модели «Роботизированный комплекс транспортировки изделий» необходимо подключиться к симулятору, нажатием кнопки «Connect». В меню «Справка» можно посмотреть «Таблицу соответствия» сигналов для указанной модели.

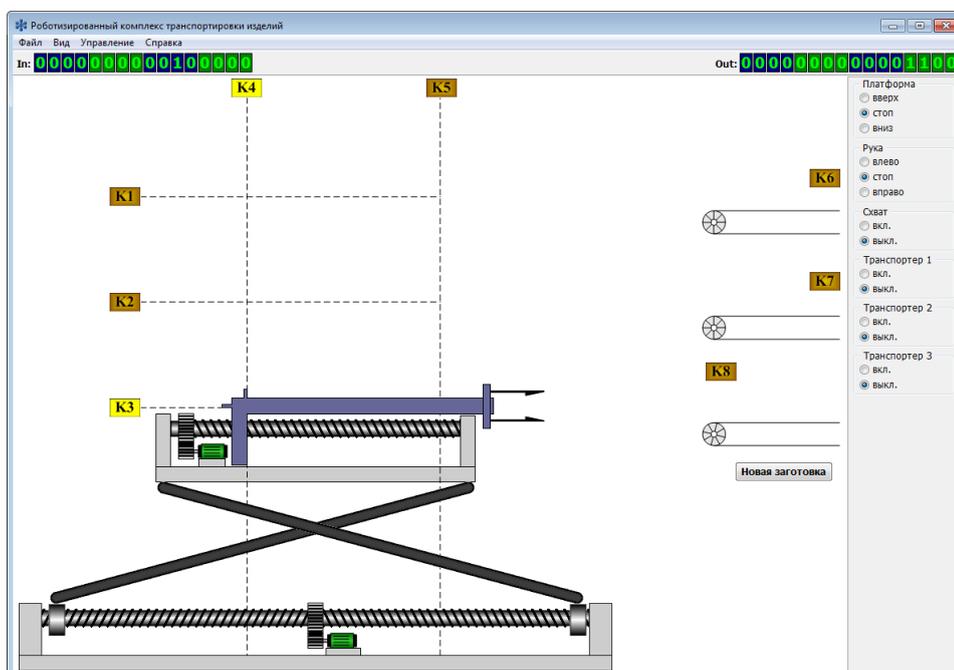
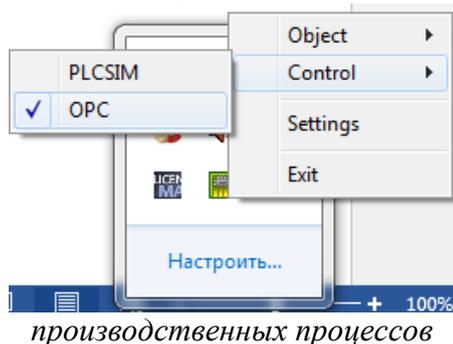


Рис.26. Модель технологического процесса «Роботизированный комплекс транспортировки изделий»

Для осуществления взаимодействия модели производственного процесса с НМІ панелью, необходимо сконфигурировать настройки OPC сервера. Для этого необходимо воспользоваться программой **KEPware** наряду с **VObjectOPC**. Для начала, после запуска модели производственного процесса, необходимо в окне, выводимым после нажатия правой кнопкой мыши в трее по значку программы **VObjectOPC**, выбрать пункт «*Connect to OPC*» для связи объектов со сторонним OPC, как показано на рис. 27.

Рис.27. Окно настройки соединения моделей



При этом также необходимо выбрать вариант «Управления от контроллера» в соответствующей вкладке запущенной программы модели производственного процесса, как показано на рис. 28.

Следующим важным шагом является конфигурирование настроек в программе **KEPware**. Для этого нам потребуется вызвать окно быстрых настроек «*Quick Client*» в панели инструментов, как показано на рис.29. При этом следует учесть, что данная команда будет доступна после старта новой сессии, а именно запуска среды исполнения во вкладке **Runtime**.

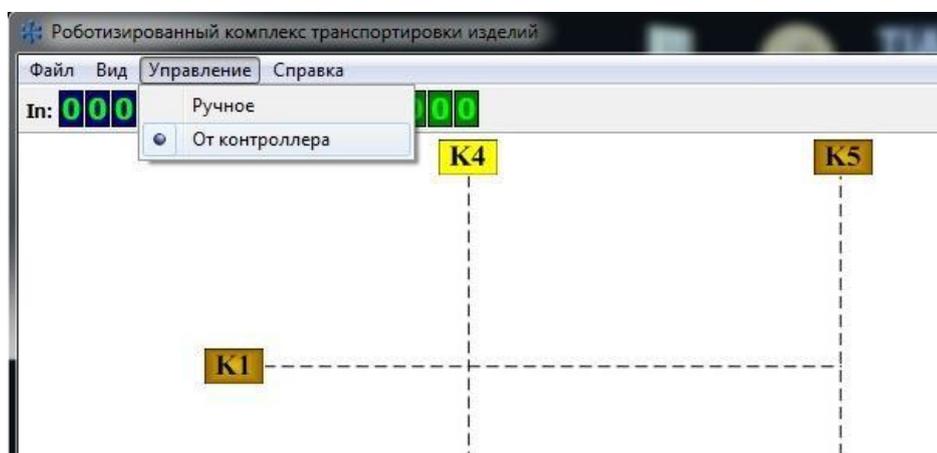


Рис.28. Окно выбора источника управления модели производственного процесса

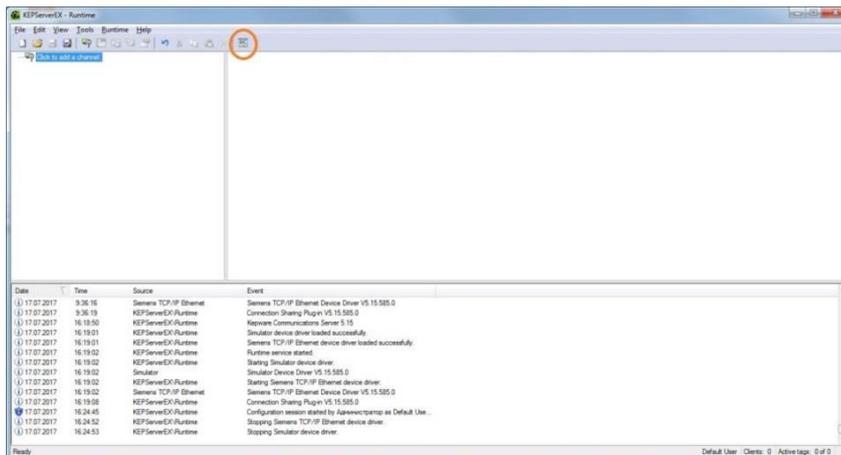


Рис.29. Команда вызова конфигуратора быстрых настроек подключений OPC сервера

В результате выполнения команды, будет вызван диалог настройки соединений, представленный на рис. 30.

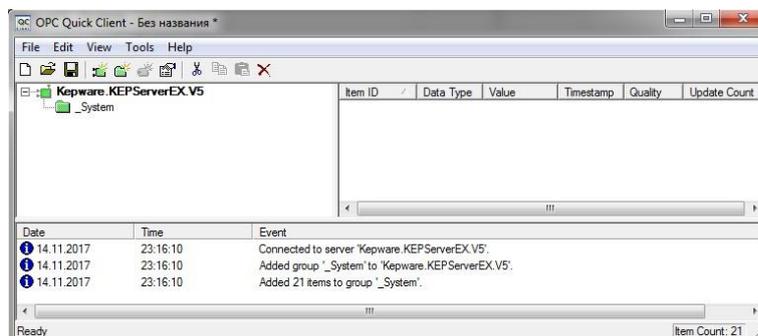


Рис.30. Окно клиента настроек новых соединений

В появившемся окне, в дереве проекта необходимо нажать правой кнопкой мыши на пустом месте и выбрать пункт «**New Server Connection**». Данное действие вызовет окно выбора соединений, представленных на рис. 31, среди которых необходимо выбрать сервер **UchtehProfi.VirtualObjects.OPCServer**.

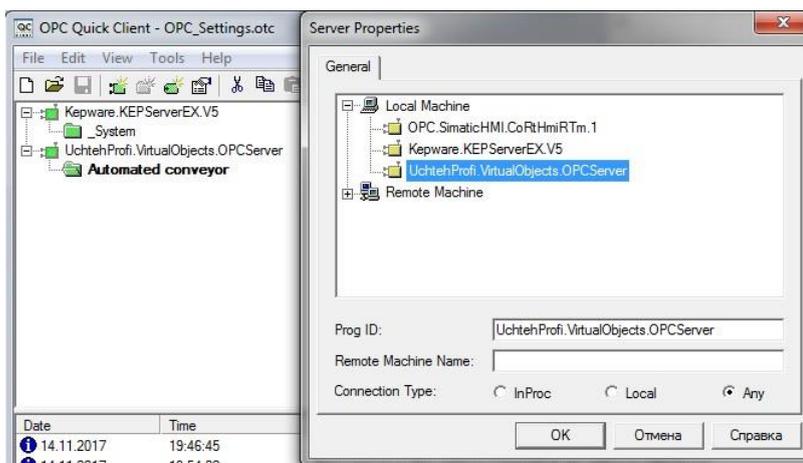


Рис.31. Окно выбора серверов для соединения

Далее необходимо создать ветку «**New Group**» в пункте добавленного OPC сервера, как показано на рис. 32. Это позволит затем добавлять компоненты для работы с дискретными «**Входами/Выходами**».

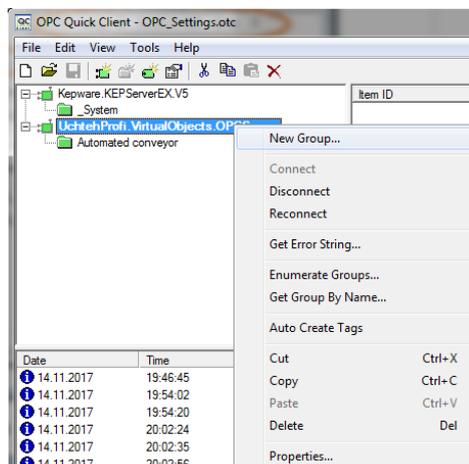


Рис.32. Окно создания новой группы описания соединений для OPC сервера

Далее появится окно настройки добавляемой группы, где необходимо добавить название, как это показано на рис. 33.

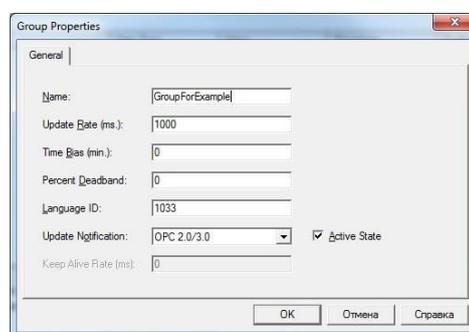


Рис.33. Окно задания основных свойств добавляемой группы

Добавленная группа позволяет создавать элементы каналов дискретных Входов/Выходов по нажатию вкладки **New Item**, как это показано на рис. 34.

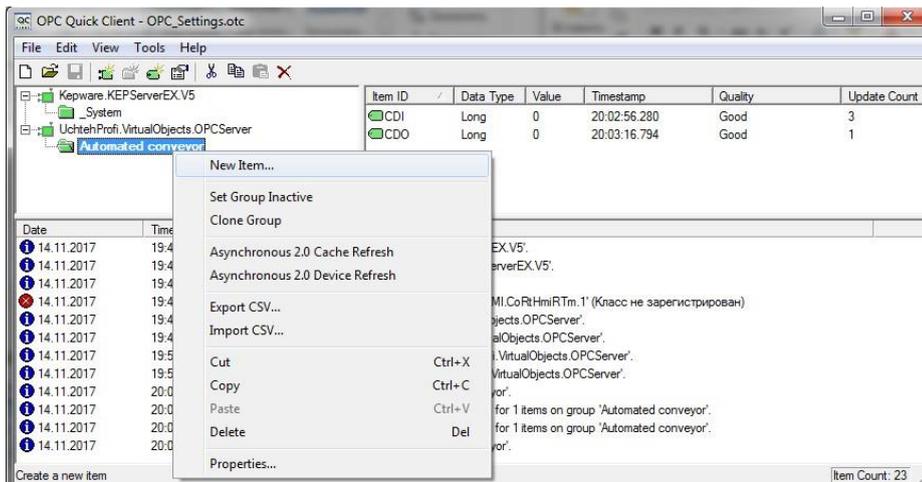


Рис.34. Добавление нового элемента в группу элементов OPC сервера

Добавление нового элемента в группу позволяет сконфигурировать необходимые каналы обмена данными, выбрав нужный элемент из списка, как это показано на рис.35.

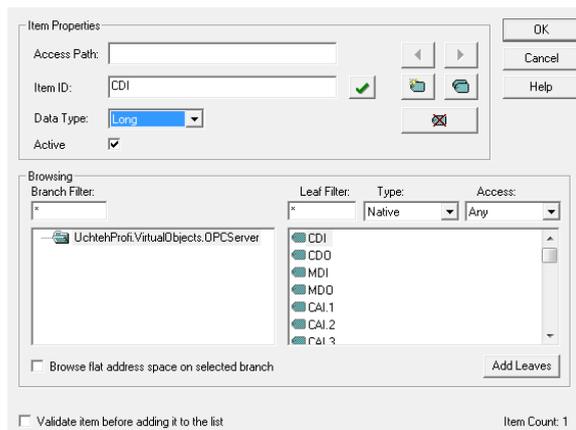


Рис.35. Окно добавления и настройки свойств слово состояний

Итоговый вид настроенных слово состояний группы элементов OPC сервера будет иметь вид, представленный на рис.36.

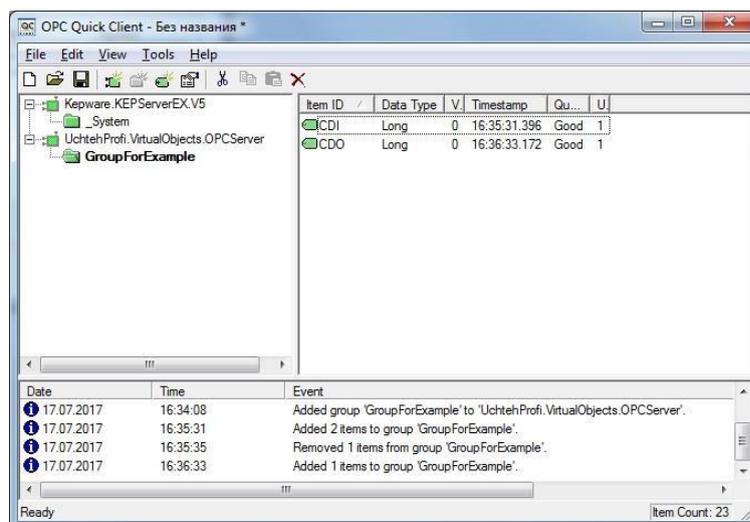


Рис.36. Сконфигурированная группа дискретных Входов/Выходов OPC сервера

Для проверки сконфигурированных настроек необходимо воздействовать на объект, путем изменения слово управления. Для этого необходимо нажать правой кнопкой по строке *CDO* и выбрать пункт *Synchronous Write*, после чего появится окно изменения значений для слова управления, как это показано на рис. 37.

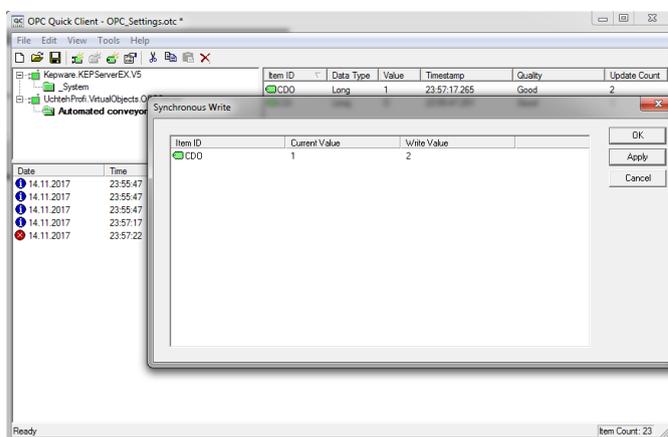


Рис.37. Изменение значения слово управления

Согласно таблице приведенной в каждом варианте виртуального объекта можно сформировать слово управления для необходимого перемещения или действия, а также расшифровать слово состояния.

К примеру, для поднятия платформы нужно ввести в слово управления **CDO** значение «1», а для опускания значение «2». Для того чтобы считать слово состояния необходимо обратить внимание на строку **CDI**. К примеру, если слово состояния имеет значение «12», то платформа находится в крайнем нижнем положении, манипулятор в крайнем правом и отсутствуют сигналы о новых заготовках. Если значение изменилось на «8», то платформа начала движение вверх, но манипулятор все еще в крайнем правом положении и сигналов о новых заготовках все еще не появилось.

2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ «РОБОТИЗИРОВАННЫЙ КОМПЛЕКС ТРАНСПОРТИРОВКИ ИЗДЕЛИЙ»

Порядок создания НМІ проекта после запуска TIA PORTAL включает следующие шаги:

1. Создание проекта АРМ. Включает в себя создание нового проекта Trace Mode простого уровня сложности.
2. Определение источников/приемников сигнала – определение таблицы символов. Добавление к источникам сигналов *OPC сервера* дискретных сигналов **CDI/CDO**.
3. Конфигурирование аргументов экрана – добавление переменных для привязки к объектам экрана рабочего места.
4. Конфигурирование экрана НМІ проекта представляет собой формирование из базовых фигур визуальное представление используемой модели производственного процесса, настройка цветового отображения и настройка анимации с событиями для реагирования на изменения входных/выходных сигналов и действий оператора.

Для обозначения перемещения объекта по конвейеру необходимо визуализировать все участки транспортировочной ленты, а также анимировать ту ее часть, которая работает в данный момент времени. Для этого можно использовать один из базовых графических элементов – прямоугольник, как это показано на рис. 38.

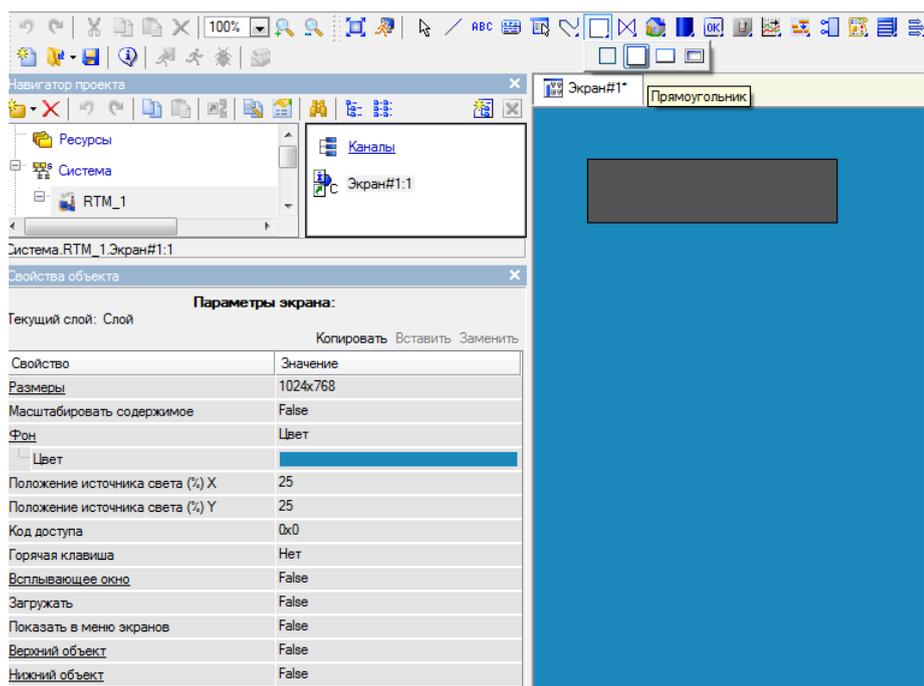


Рис. 38 – Добавление базового графического элемента в режиме конфигурирования экрана

Для большей наглядности у добавленного графического элемента можно задать серый цвет, устанавливая необходимые свойства в панели основных свойств, как это показано на рис. 49.

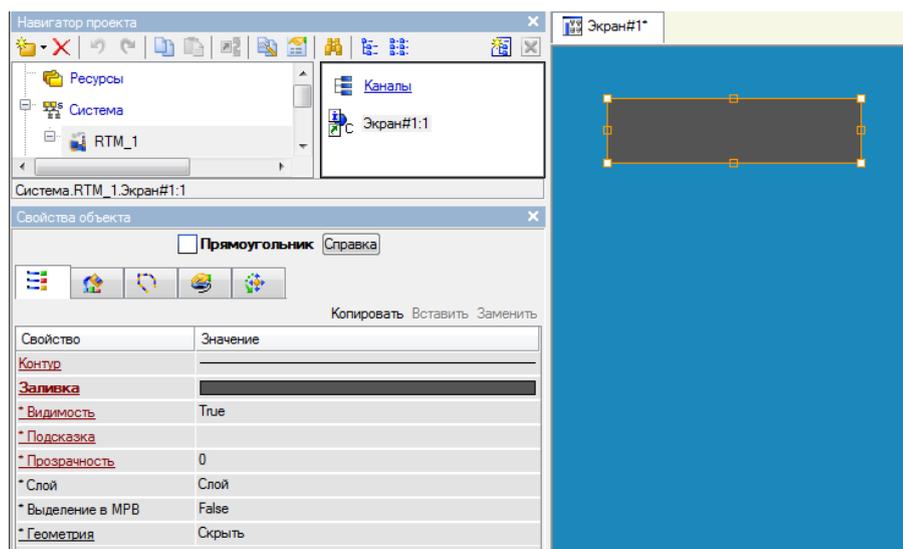


Рис. 49. Редактирование свойств графического элемента

Для отображения режима работы транспортной ленты можно воспользоваться механизмом визуального отображения в случае активации соответствующего выходного сигнала. Следует задать режим отображения данного графического элемента в случае активации соответствующего выходного сигнала. Для этого необходимо в свойствах параметра «Заливка» задать динамизацию объекта вида $Arg=Конст$, задав привязку объекта как слово состоянию **CDO** и установив константу для сравнения равной числу, которое соответствует нужному биту в слово состоянии. Для объектов производственного процесса Транспортера 1, Транспортера 2 и Транспортер 3 - это будут константы равные 64, 128, 256 соответственно. Затем остается задать необходимые

цвета для отображение конфигурируемого объекта при различных результатах сравнения, как это показано на рис.40.

В дополнении к объекту транспортной ленты, необходимо также отобразить направление движения транспортера. Для этого необходимо добавить плоский объект «Стрелка» и расположить ее посередине прямоугольника в рабочей области экрана. При этом режим отображения следует настроить не только для заливки, но и для параметра «Видимости», что позволит отображать направление движения только при включении транспортера. Для этого необходимо добавить динамизацию и настроить аналогично параметру «Заливка».

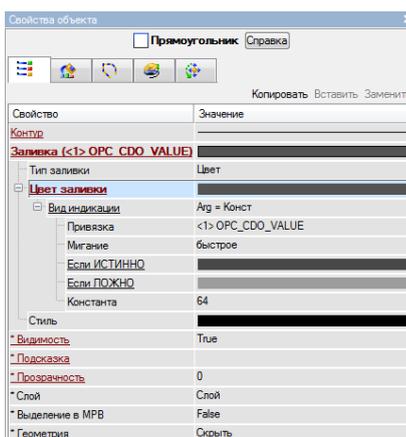


Рис. 40. Конфигурирование индикации графического элемента

В итоге объект «Транспортер», примет вид, представленный на рис. 41.

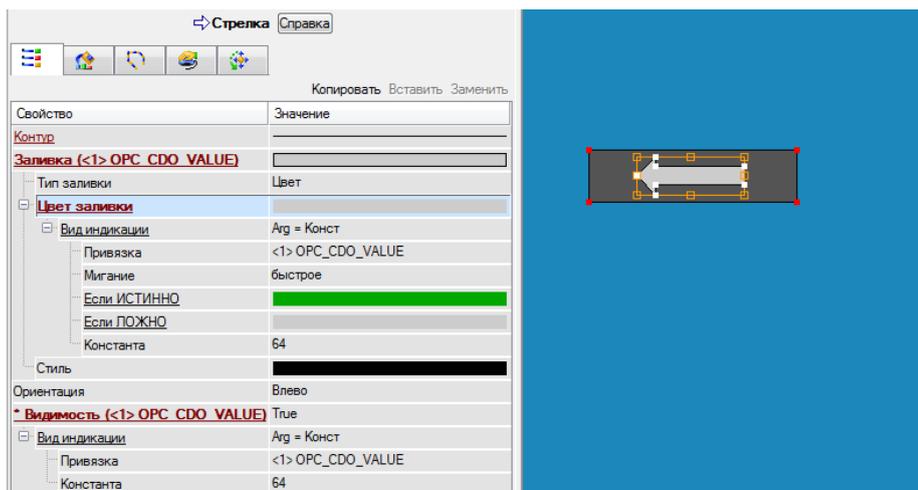


Рис. 41. Конфигурирование динамизации параметров объекта «Стрелка»

В результате добавления и настройки свойств динамизации, графический элемент прямоугольник и стрелка, будут изменять свое свойство видимости и заливки в зависимости от присутствия в контролируемом сигнале соответствующего значения.

Выполняя вышеописанные шаги необходимо сконфигурировать все соответствующие транспортеры. В результате, должен получиться экран, аналогично представленному на рис. 42.

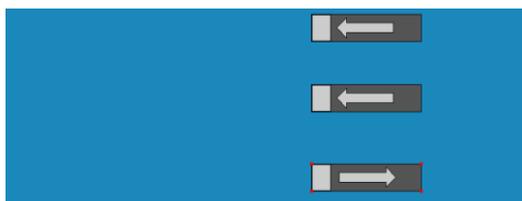


Рис. 42. Участок транспортной линии технологического процесса

Для отображения автоматизированного роботизированного комплекса также следует воспользоваться базовыми элементами панели инструментов. На рис.43 представлен возможный вид данной платформы.

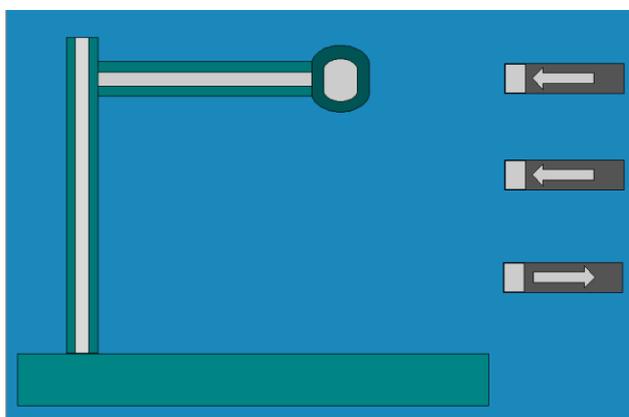


Рис. 43. Вид экрана автоматизированного комплекса транспортировки изделий

Каждая из передвигающихся частей роботизированной платформы имеет объект – индикатор, который изменяет цвет при подаче сигнала смещения платформы по осям, а также включения захвата щупа. На рис.43 такими объектами служат прямоугольные линии серого цвета, а также объект овальной формы в области щупа. Данные компоненты рабочего экрана настроены параметрами динамизации, аналогично транспортерам. Для вертикального участка комплекса использовались пары констант = 1, 2 – как это показано на рис.44.

Для горизонтального участка необходимо использовать пары констант = 4, 8 и для захвата 16, 32 соответственно.

Для отслеживания положения роботизированной платформы в различные моменты времени, а также положения заготовки на каждом из транспортеров – установлены датчики, которые сигнализируют о прохождении объекта определенной зоны.

В качестве датчиков на графическом экране может служить объект «Текст», который меняет цвет заливки при поступлении соответствующего сигнала. Согласно таблице сигналов, представленных в таблице 1, графический экран должен содержать 8 индикаторов положения.

Разместив датчики на графическом экране, как это показано на рис.45, необходимо сконфигурировать их индикацию цвета заливки, настроив на соответствующие биты входного слова состояния, аналогично тому как это было сделано ранее.

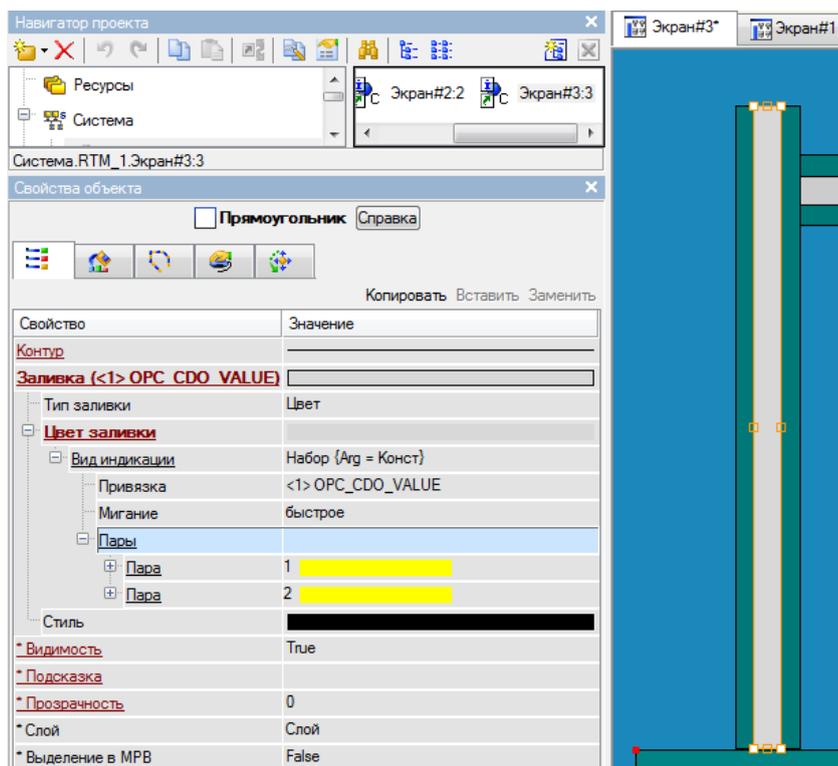


Рис. 44. Конфигурирование заливки вертикального участка роботизированной платформы

Таблица 1.

Входные сигналы с датчиков

Д1	Bool	%I0.0	Датчик верхнего положения платформы
Д2	Bool	%I0.1	Датчик среднего положения платформы
Д3	Bool	%I0.2	Датчик нижнего положения платформы
Д4	Bool	%I0.3	Датчик крайне левого положения манипулятора
Д5	Bool	%I0.4	Датчик крайне правого положения манипулятора
Д6	Bool	%I0.5	Датчик наличия объекта на верхнем транспортёре
Д7	Bool	%I0.6	Датчик наличия объекта на среднем транспортёре
Д8	Bool	%I0.7	Датчик наличия объекта на нижнем транспортёре

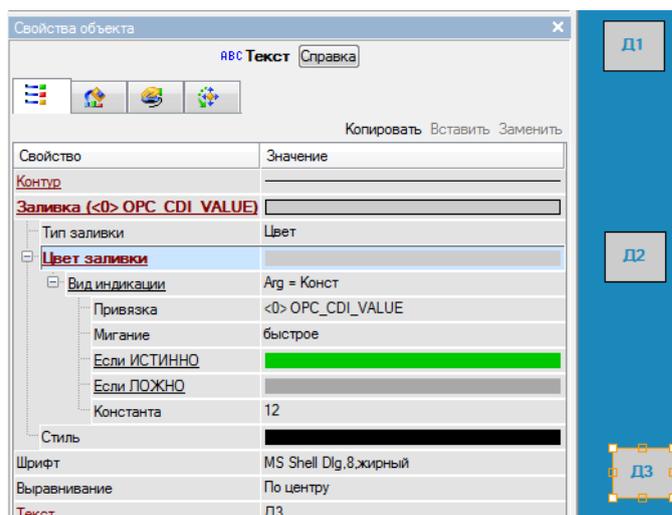


Рис. 45. Конфигурация индикаторов положения

В результате должен получиться экран, аналогично представленному на рис. 46.

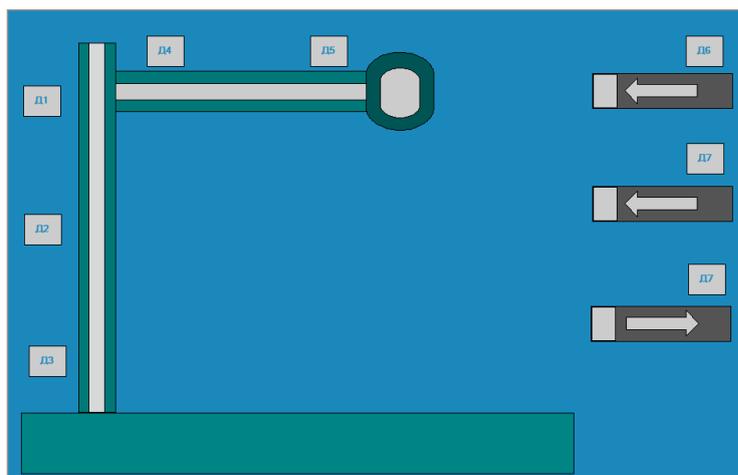


Рис. 46. Индикаторы положения на различных участках рабочего места

Процесс управления моделью технологического процесса осуществляется в ручном режиме. Управление в ручном режиме подразумевает воздействие на управляющие сигналы конвейера через определенные графические элементы, которые реагируют на нажатие мышью. Таким графическим элементом может служить как кнопка, так и базовые объекты с панели графических элементов.

Добавив элемент на экран, необходимо назначить на него событие по нажатию левой кнопки мыши. Назначение событий осуществляется в области конфигурирования и задания параметров «События» > «*MousePress*» > «*Передать значение*». В данной области, в соответствии с рис. 47, необходимо назначить сигнал, на который осуществляется воздействие оператором на один из перечисленных участков конвейера:

- команда на движение платформы вверх;
- команда на движение платформы вниз;
- команда на движение манипулятора влево;
- команда на движение манипулятора вправо;
- команда на сжатие захвата манипулятора;
- команда на разжатие захвата манипулятора;
- команда на движение транспортера первого;
- команда на движение транспортера второго;
- команда на движение транспортера третьего.

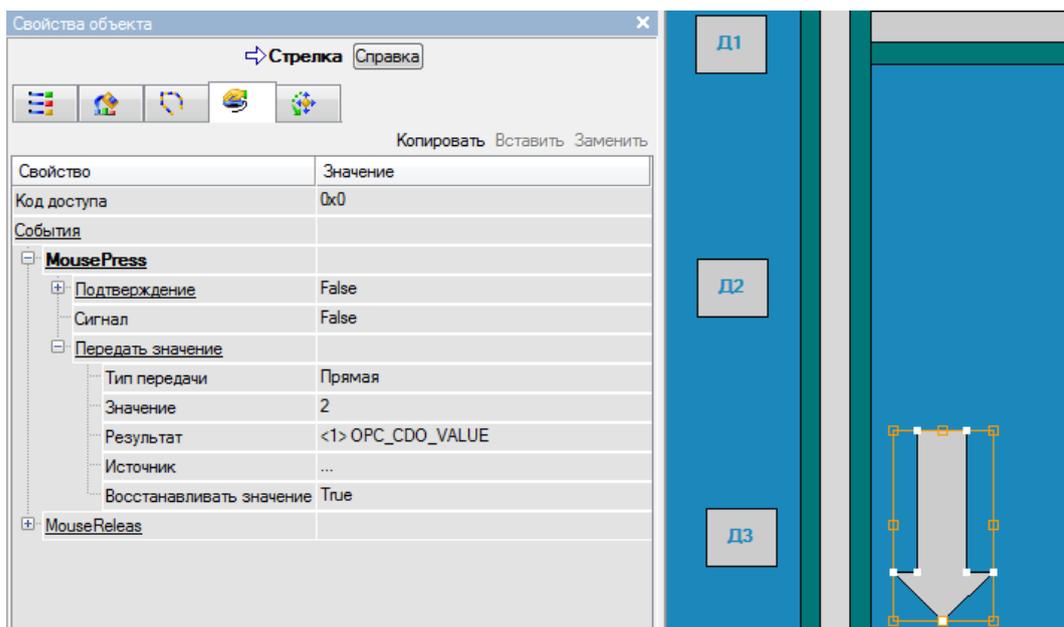


Рис. 47. Назначение событий по нажатию левой кнопки мыши и соответствующей анимации в ручном режиме управления

Для осуществления обратной связи по нажатию кнопок управления конвейером в ручном режиме необходимо вновь воспользоваться динамизации ей стрелки и назначить изменение цвета, аналогично тому, как это делалось ранее.

Для задания значений констант, а именно установки соответствующих битов для команд управления необходимо воспользоваться таблицей 2.

Таблица 2.

Выходные сигналы управления

Вв	Bool	%Q0.0	Команда на движение платформы вверх
Вн	Bool	%Q0.1	Команда на движение платформы вниз
Л	Bool	%Q0.2	Команда на движение манипулятора влево
П	Bool	%Q0.3	Команда на движение манипулятора вправо
Зажим	Bool	%Q0.4	Команда на сжатие захвата манипулятора
Разжим	Bool	%Q0.5	Команда на разжатие захвата манипулятора
Тр1	Bool	%Q0.6	Команда на движение транспортера первого
Тр2	Bool	%Q0.7	Команда на движение транспортера второго
Тр3	Bool	%Q1.0	Команда на движение транспортера третьего

В ходе выполнения работы сконфигурированная панель будет иметь вид, представленный на рис.48.

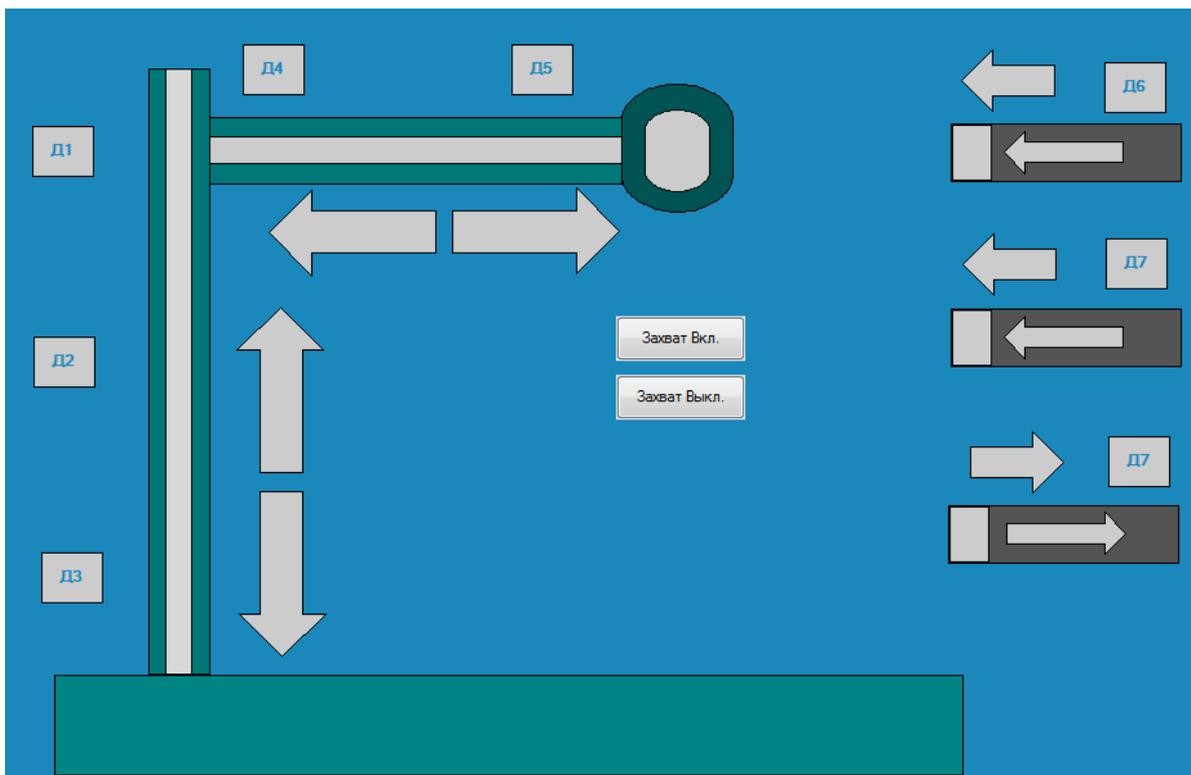


Рис. 48. Сконфигурированный экран технологического процесса

6. ЗАДАНИЯ НА ВЫПОЛНЕНИЕ РАБОТЫ ПО ВАРИАНТАМ

Для выполнения самостоятельных работ по вариантам используются 2-х мерные модели в программе **VObjectsOPC** наряду с **KEPServerEx**. В вариантах необходимо самостоятельно спроектировать и анимировать компоненты интерфейса экрана оператора для функциональных частей технологических объектов и составить программу при необходимости.

Вариант 1.

На примере модели *«Участок транспортировки заготовок в методическую печь»* визуализировать участок методической печи (1) с заготовкой (2), толкателем заготовок (3) и заслонки (4). Под действием толкателя (3) заготовка (2) перемещается в методическую печь (1), после того как заготовка (2) окажется полностью в печи (1), толкатель (3) возвращается в исходное положение, заслонка (4) закрывает печь.

Процесс управления осуществляется в ручном режиме. Воздействие на управляющие сигналы осуществляется по нажатию кнопкой мыши на графические элементы (5), события которых сконфигурированы на активацию соответствующих битов выходных сигналов. Обратная связь осуществляется за счет привязки входных сигналов к датчикам индикации (6). Входные и выходные сигналы представлены в таблице.

Экран проекта (Вариант 1)

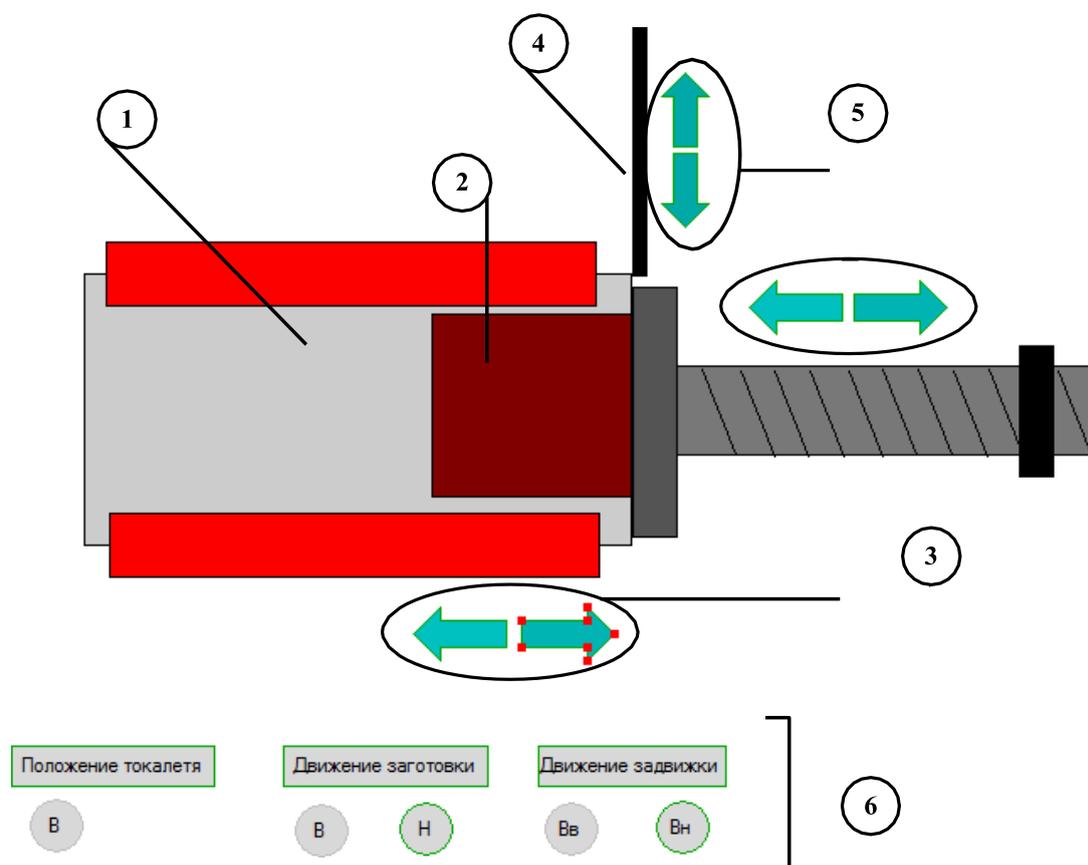


Таблица входных и выходных сигналов управления (Вариант 1)

Имя	Тип	Адрес	Описание
ПТЗ	Bool	X7	Датчик конечного положения толкателя
ПЗО	Bool	X8	Датчик открытия заслонки
ПЗЗ	Bool	X9	Датчик закрытия заслонки
Тв	Bool	Y3	Команда на движение толкателя вперед
Тн	Bool	Y4	Команда на движение толкателя назад
Зо	Bool	Y5	Команда на открытие заслонки печи
Зз	Bool	Y6	Команда на закрытие заслонки печи

Вариант 2.

На примере модели «*Роботизированный комплекс транспортировки изделий*» визуализировать один из участков транспортировки в виде роботизированной руки (1), которая перемещается по горизонтальной оси вместе с захватчиком (2). Сжатие захвата происходит перед началом движения роботизированной руки (1), а разжатие происходит после ее остановки.

Процесс управления осуществляется в ручном режиме. Воздействие на управляющие сигналы осуществляется по нажатию кнопкой мыши на графический элемент (3), события которых сконфигурированы на активирование соответствующих битов выходных сигналов. Обратная связь осуществляется за счет привязки входных сигналов к датчикам индикации (4). Входные и выходные сигналы представлены в таблице.

Экран проекта (Вариант 2)

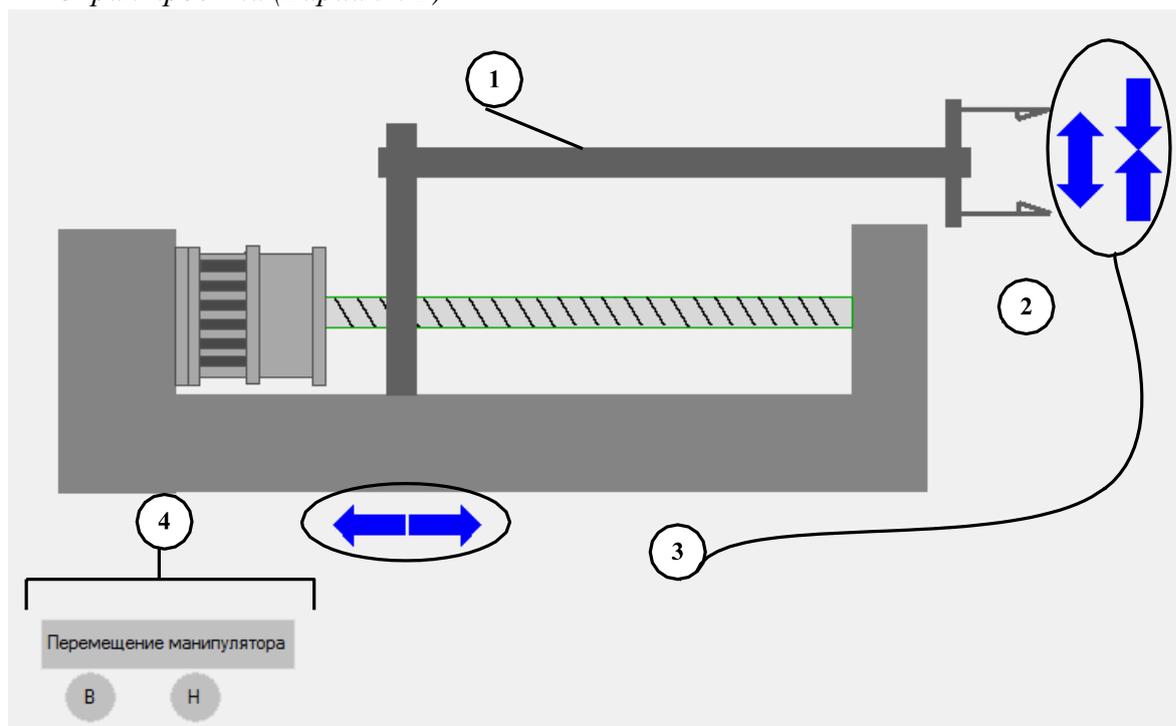


Таблица входных и выходных сигналов управления (Вариант 2)

Имя	Тип	Адрес	Описание
Л	Bool	Y2	Команда на движение манипулятора влево
П	Bool	Y3	Команда на движение манипулятора вправо
Зажим	Bool	Y4	Команда на сжатие захвата манипулятора
Разжим	Bool	Y5	Команда на разжатие захвата манипулятора
К4	Bool	X3	Датчик крайне левого положения манипулятора
К5	Bool	X4	Датчик крайне правого положения манипулятора

Вариант 3.

На примере модели «Участок транспортировки заготовок в методическую печь» реализовать один из его участков в виде основного рольганга (1), которая перемещает заготовку (2), перемещение которой останавливается при достижении сигнальной линии (3). Движение заготовки (2) начинается одновременно с движением рольганга (1).

Процесс управления осуществляется в ручном режиме. Воздействие на управляющие сигналы осуществляется по нажатию кнопкой мыши на графические элементы (4), события которых сконфигурированы на активирование соответствующих битов выходных сигналов. Обратная связь осуществляется за счет привязки входных сигналов к датчикам индикации (5). Входные и выходные сигналы представлены в таблице.

Экран проекта (Вариант 3)

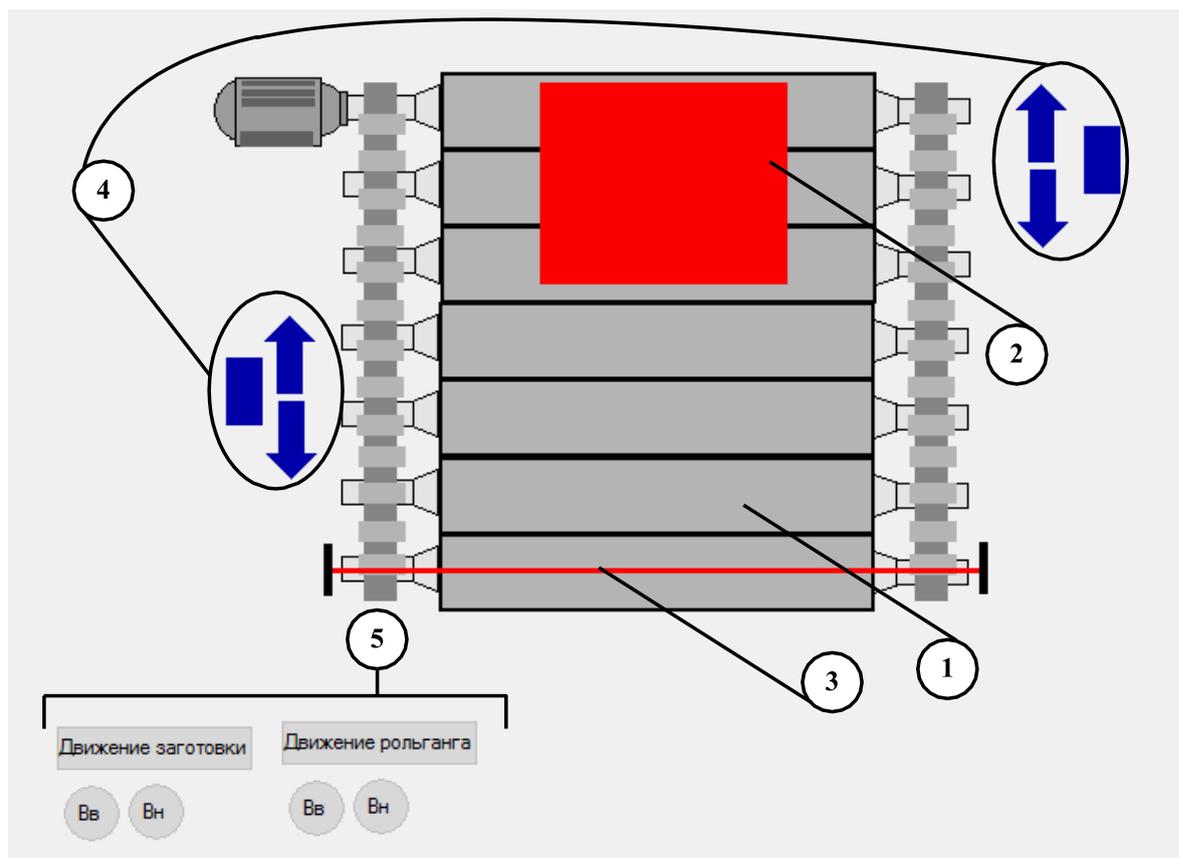


Таблица входных и выходных сигналов управления (Вариант 3)

Имя	Тип	Адрес	Описание
Рв	Bool	Y0	Активация основного рольганга
ПР1	Bool	X1	Датчик для перехода на сниженную скорость основного рольганга
ПР3	Bool	X3	Датчик сигнализации о проезде люка печи

Вариант 4.

На примере модели «Участок сортировки и пакетирования» реализовать один из его участков в виде платформы (1) перемещающейся по основанию (2) в горизонтальном положении и останавливается достигая конца основания. После остановки платформа (1) поднимается, а затем опускается вниз и продолжает свое движение по основанию (2).

Процесс управления осуществляется в ручном режиме. Воздействие на управляющие сигналы осуществляется по нажатию кнопкой мыши на графические элементы (3), события которых сконфигурированы на активирование соответствующих битов выходных сигналов. Обратная связь осуществляется за счет привязки входных сигналов к датчикам индикации (4). Входные и выходные сигналы представлены в таблице.

Экран проекта (Вариант 4)

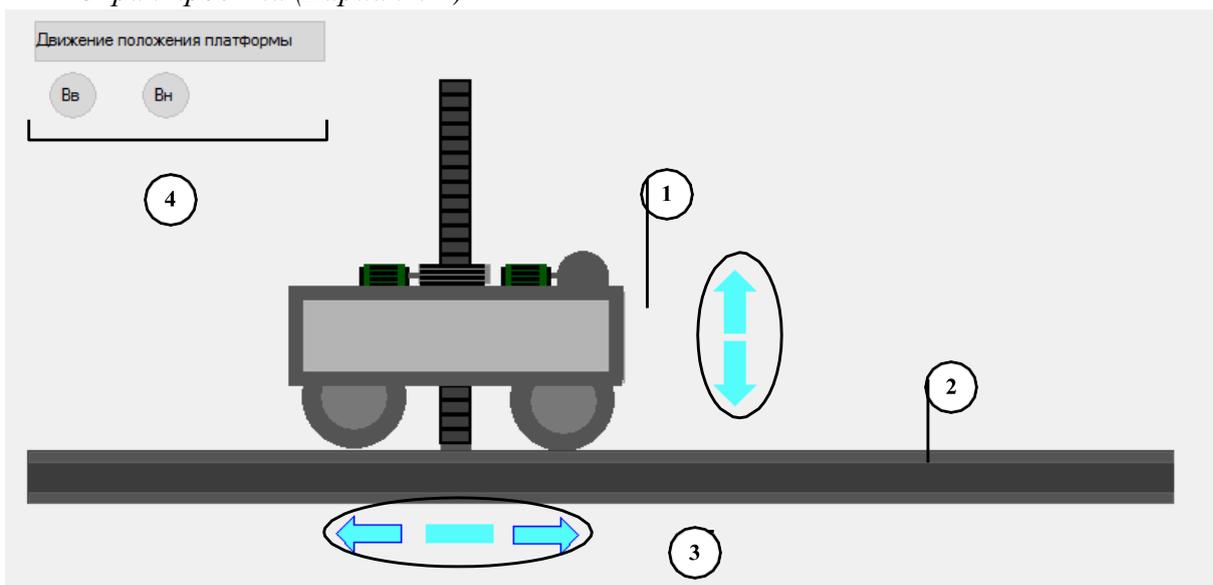


Таблица входных и выходных сигналов управления (Вариант 4)

Имя	Тип	Адрес	Описание
Тл	Bool	Y0	Движение платформы влево
Тп	Bool	Y1	Движение платформы вправо
В	Bool	Y3	Движение платформы вверх
Н	Bool	Y4	Движение платформы вниз
КВ	Bool	X2	Датчик крайнего верхнего положения платформы

Вариант 5.

На примере модели «Погрузка угля в вагон» реализовать один из ее участков в виде выгрузки угля (1) из бака (2) на ленту (3) и его транспортировку.

Процесс управления осуществляется в ручном режиме. Воздействие на управляющие сигналы осуществляется по нажатию кнопкой мыши на графические элементы (4), события которых сконфигурированы на активирование соответствующих битов выходных сигналов. Обратная связь осуществляется за счет привязки входных сигналов к датчикам индикации (5). Входные и выходные сигналы представлены в таблице.

Экран проекта (Вариант 5)

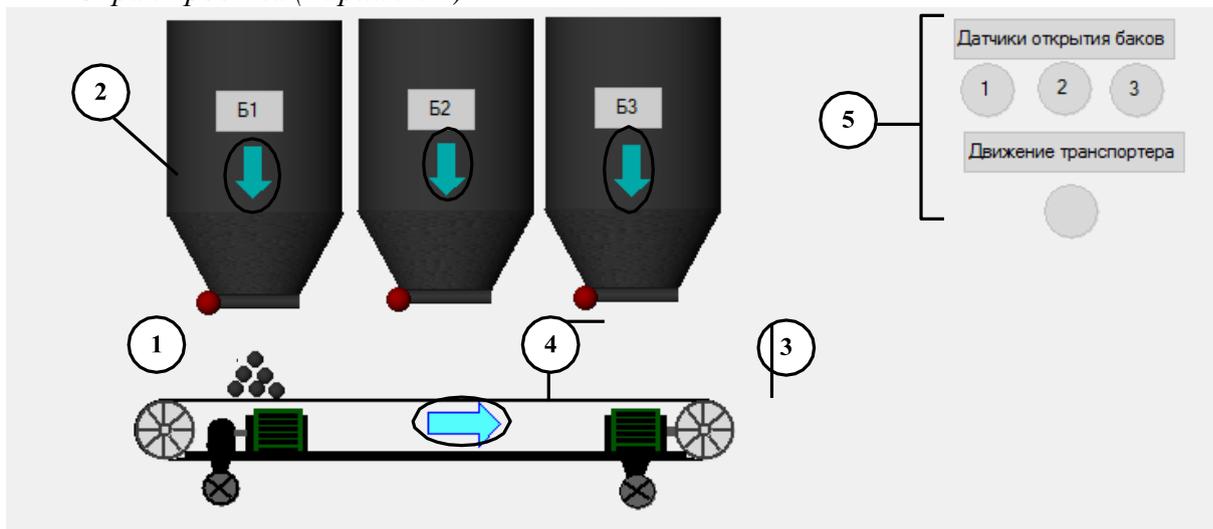


Таблица входных и выходных сигналов управления (Вариант 5)

Имя	Тип	Адрес	Описание
ДТР	Bool	X0	Датчик движения транспортера
Б1	Bool	X1	Датчик открытия бака 1
Б2	Bool	X2	Датчик открытия бака 2
Б3	Bool	X3	Датчик открытия бака 3
ТР	Bool	Y2	Активация транспортера
ЗД1	Bool	Y3	Открытие бака 1
ЗД2	Bool	Y4	Открытие бака 2
ЗД3	Bool	Y5	Открытие бака 3

Вариант 6.

На примере «Погрузка угля в вагоны» реализовать один из ее участков в виде передвижение транспортной ленты (1) по основанию (2) в горизонтальном положении и останавливается достигая конца основания (2).

Процесс управления осуществляется в ручном режиме. Воздействие на управляющие сигналы осуществляется по нажатию кнопкой мыши на графические элементы (3), события которых сконфигурированы на активирование соответствующих битов выходных сигналов. Обратная связь осуществляется за счет привязки входных сигналов к датчикам индикации (4). Входные и выходные сигналы представлены в таблице.

Экран проекта (Вариант 6)

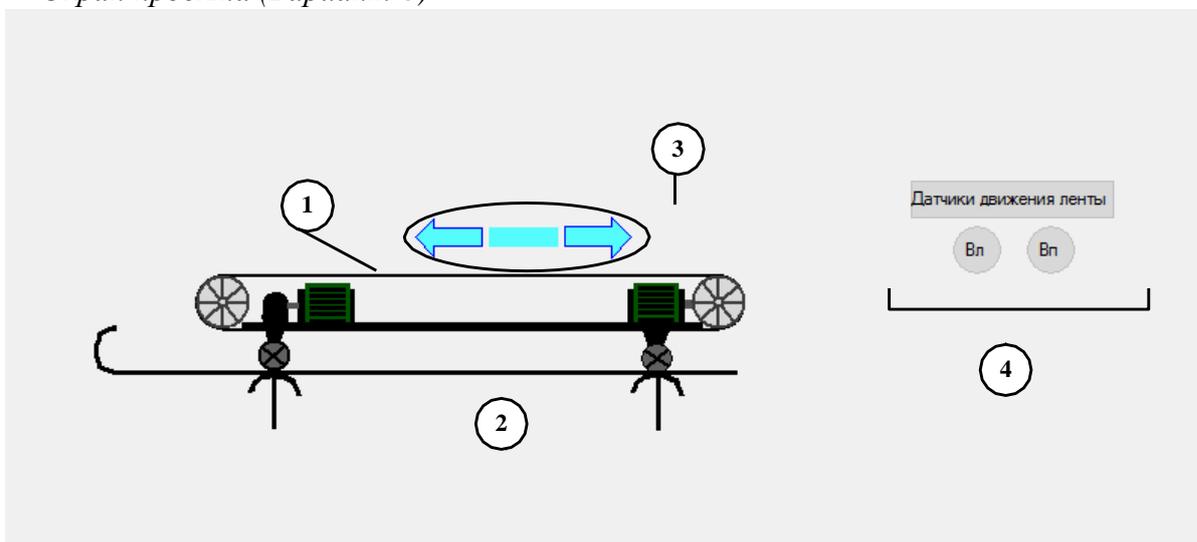


Таблица входных и выходных сигналов управления (Вариант 6)

Имя	Тип	Адрес	Описание
КВ3	Bool	X5	Левый датчик тележки
КВ4	Bool	X6	Правый датчик тележки
ТЛ	Bool	Y0	Движение тележки влево
ТП	Bool	Y1	Движение тележки вправо