ЗАДАНИЕ 3
Создание мнемосхемы с имитацией работы оборудования

Цель выполнения задания

Освоение создания элементов пользовательского интерфейса с изображением технологического оборудования, а также приёмов экспорта и импорта компонентов проекта.

Освоение приёмов создания и использования программ проекта, реализующих пользовательские функции.

Содержание задания

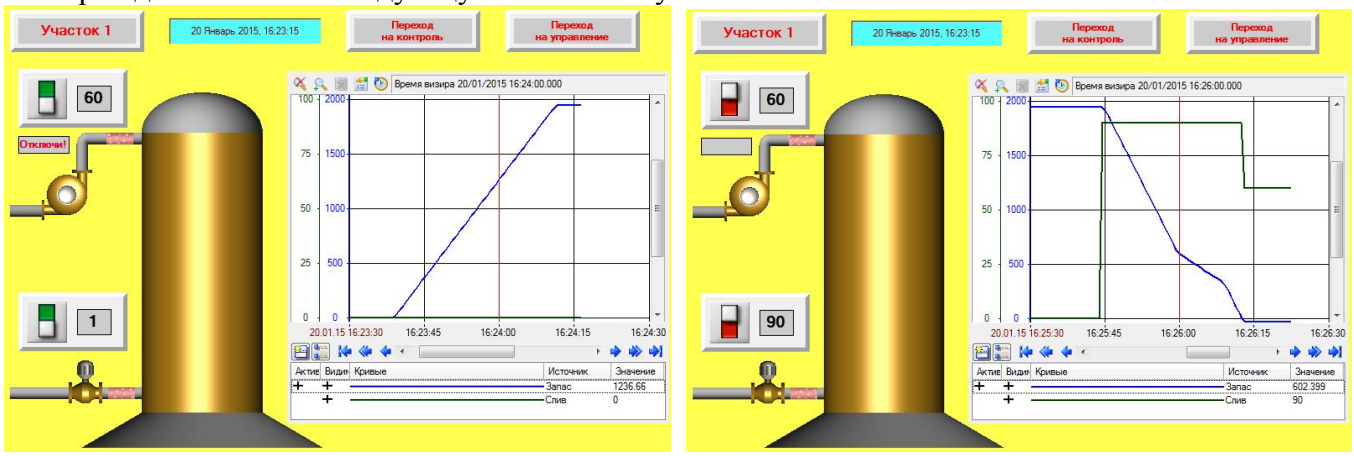
Создать экран с мнемонической схемой управляемого технологического оборудования, а также с элементами управления вызова экранов узла.

Созданный экран должен содержать изображение резервуара для накопления запаса продукта с подводящим и отводящим трубопроводом, с насосом и клапаном и элементами управления их включением и величиной их производительности.

Должна использоваться текстовая сигнализация о необходимости выключении насоса и клапана при заполнении и опустошении резервуара. Движение содержимого трубопроводов должно отображаться видеоклипом, импортированным из ресурсов демонстрационных проектов.

Создание запаса продукта следует имитировать программой, выполненной на языке **Техно FBD**.

Экран должен иметь следующую компоновку своих элементов:



Порядок выполнения задания

Открыть Интегрированную среду разработки TRACE MODE 6.

В окне программы открыть проект, созданный при выполнении предыдущего задания.

Сохранить проект с новым именем, изменив в имени цифру номера предыдущего задания, на номер текущего задания, например: 123_3.

Открыть рабочую папку и проконтролировать наличие в ней файла текущего проекта.

1. Создание графического экрана и элементов управления для перехода по экранам

При создании интерфейса пользователя с множеством экранов возникает необходимость организации в реальном времени возможности перехода по экранам (вызова требуемого экрана). Кроме этого для одного из используемых в узле экранов проекта требуется обеспечить первоочерёдность его отображения (вызова).

Создание нового экрана и элементов управления для вызова других экранов

Вызов в реальном времени других экранов узла можно обеспечить как использованием функциональных клавиш клавиатуры (F2 – F12), так и использование элементов управления с функцией перехода на заданный экран.

Для перехода по экранам обычно создаются экранные элементы, которым назначается соответствующее действие, - функция управления Перейти на экран на нажатие или отжатие ЛКМ с указанием в качестве значения требуемого шаблона экрана.

1.1. В слое Шаблоны экранов структуры проекта создать новый компонент типа **Экран** с именем ЭкранМнемосхемы.

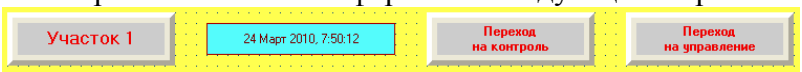
1.2. Открыть созданный шаблон экрана для редактирования. В окне свойств экрана свойству Размеры задать значение 800 x 600 и выбрать желаемый фон.

1.3. Руководствуясь рисунком задания, создать три панели с надписями: панель оформления с надписью Участок 1 и две панели с надписями Переход на контроль и Переход на управление.

Примечание: Панели нового экрана можно создать как копии подобных панелей, предыдущих экранов.

1.4. Используя ГЭ Дата и время создать экранный элемент и выполнить его настройку для отображения текущей даты и времени.

Созданное содержимое экрана может быть оформлено следующим образом:



1.5. Текстовым элементам с надписями о переходе задать действие (функцию) Перейти на экран с назначением для них соответствующих экранов проекта.

1.6. На ранее созданных экранах с функциями контроля и управления создать элементы управления с функцией возврата на экран с мнемосхемой. Для этого:

- открыть на редактирование шаблон ЭкранКонтроля, создать на нём панель с надписью Переход на мнемосхему, текстовому элементу панели задать функцию Перейти на экран и выбрать для неё соответствующий шаблон экрана проекта (можно использовать копию экранного элемента управления с функцией перехода на экран, созданного на шаблоне экрана ЭкранМнемосхемы);
- выполнить предыдущее задание для шаблона ЭкранУправления.

Внимание! *Если шаблон вызываемого в узел экрана не содержит ни одного аргумент, то при запуске узла в профайлере экран не будет отображаться.*

1.7. Из задания следует, что экран должен содержать ряд аргументов для реализации управления работой насоса и клапана, а также для представления на экране налива, слива и изменения запаса содержимого в резервуаре.

Используя окно **Аргументы экрана** создать:

- два аргумента типа OUT с типом данных USINT с именами Вкл_насоса и Вкл_клапана;
- два аргумента типа OUT с типом данных REAL с именами Произв_насоса и Произв_клапана;
- три аргумент экрана типа IN с типом данных REAL с именами Налив, Запас и Слив.

Создание канала для вызова экрана и его настройка

Первоочерёдность вызова экрана обеспечивается соответствующей настройкой канала его вызова. У канала вызова шаблона экрана, который должен отображаться первым, задаётся его отработка при старте. Для этого в редакторе канала, в основных настройках группы ***Системные***, необходимо включить флаг Отработать.

1.8. В узле проекта АРМ, в его группе каналов Каналы_вызова_экранов, создать канал для вызова шаблона экрана ЭкранМнемосхемы или непосредственно - созданием канала класса **CALL** с выполнением настройки его свойства **Вызов** для вызова требуемого шаблона проекта, или процедурой автоматического создания - перетаскиванием шаблона экрана на соответствующую группу каналов.

Созданному каналу назначить имя ВызовЭкранМнемосхемы

1.9. Открыть созданный канал вызова экрана на редактирование.

Для реализации первоочерёдности вызова (отображения) созданного экрана, при запуске файлов узла в реальном времени, выполнить соответствующую настройку канала его вызова - включить флаг Отработать в группе настроек **Системные**.

1.10. Сформировать файлы узла проекта, выделить узел, открыть профайлер, запустить его и проверить в реальном времени действие экранных элементов с функцией перехода на экран.

2. Создание пользовательской библиотеки компонентов

Для тиражирования проектных решений между различными проектами используется пользовательская библиотека компонентов. При этом обеспечивается возможность, как экспорта, так и импорта компонентов проекта.

Пользовательская библиотека компонентов (ПБК) имеет структуру, подобно структуре проекта. Компоненты те-

кущего проекта можно сохранить в соответствующих объектах структуры ПБК, а затем использовать их в других проектах.

Структурно можно создавать набор ПБК. Набор всех ПБК – это файл **`tmdevenv.tmul`**, который размещается в папке ИС.

Создание пользовательской библиотеки компонентов и формирование её содержимого

ПБК создаются в слое Библиотеки компонентов, в его подслое Пользовательская в составе группы Библиотека. Каждая ПБК – это набор групп типа **Объект**, которые имеется предопределённый набор слоёв. Состав слоёв групп типа **Объект** ПБК соответствует составу слоёв структуры проекта.

Для формирования содержимого ПБК надо требуемые компоненты структуры проекта операциями копирования и вставки перенести в соответствующие слои структуры групп ПБК типа **Объект**.

Результаты редактирования содержимого ПБК сохраняются как результаты редактирования текущего проекта.

2.1. В слое Библиотеки компонентов структуры проекта создать пользовательскую библиотеку, а в ней группу типа **Объект** с именем NNN, где NNN - три последние цифры номера зачётной книжки, и сохранить проект с текущими результатами проектирования.

Примечание: Если в ИС имеется пользовательская библиотека, следует создать в ней новую группу типа **Объект** с указанным именем.

2.2. Открыть проект **`Boiler.prj`** из папки **`DEMO`** ИС: **`C:\Program Files\AdAstra Research Group\Trace Mode IDE 6 Base\`**

В группе Анимация структуры ресурсов проекта скопировать компонент Библиотека Видеоклипов.

Открыть в ПБК структуру созданной группы типа **Объект**. В её слое Ресурсы создать группу Анимация и вставить в неё скопированный ранее компонент проекта.

Примечание: Копирование можно выполнить перетаскиванием требуемых компонент.

2.3. Сохранить текущий проект. В данном случае сохраняются результаты редактирования содержимого ПБК.

Создание ресурсов проекта из содержимого пользовательской библиотеки компонентов

Формирование компонентов проекта, в частности его ресурсов, из содержимого ПБК выполняется копированием необходимых компонент из слоёв ПБК и вставкой их в соответствующие слои структуры проекта.

2.4. Используя команду Файл – Последние файлы открыть разрабатываемый проект.

2.5. Открыть в текущем проекте требуемую ПБК.

В ПБК открыть группу Анимация, скопировать в ней компонент Библиотека Видеоклипов и вставить её в группу Анимация ресурсов проекта.

Внимание! *В проекте должно быть две библиотеки видеоклипов.*

Просмотреть содержимое импортированной библиотеки видеоклипов.

3. Создание мнемонической схемы технологического оборудования с элементами управления и индикации

Для создания условных изображений оборудования и трубопроводов, - мнемонической схемы оборудования, используются преимущественно ГЭ из группы **Объёмные фигуры**.

Поверхности фигур (Материал) может иметь базовый цвет с настройкой градиента заливки или цвет типовых металлов с задаваемой текстурой.

Если общий атрибут Толщина стенок не равен нулю, на экране отображается сечение фигуры с заданной толщиной стенок.

У объемных фигур с осевой симметрией имеется атрибут Ориентация, который задает положение их оси.

ГЭ Цилиндр и Труба имеют специфические атрибуты Край1 и Край2, с помощью которых можно задавать разнообразную форму этих частей экранной фигуры.

Для выполнения ортогонального изображения участков трубопроводов, или изображения участков под 45 градусов, следует удерживать нажатой клавишу Ctrl.

ГЭ Сфера, Конус и Тор имеют групповой атрибут Отображаемая часть с двумя параметрами. Этот атрибут может принимать два значения: Часть и Сектор. При значении Часть настроечными параметрами задаётся отображаемая часть экранной фигуры, а при значении Сектор – его начальный угол и угол разворота.

ГЭ Клапан имеет два специфических атрибута: Форма клапана и Форма привода. Значения этих атрибут выбирается из списка. При этом можно назначить отдельный цвет для привода и динамизировать его для сигнализации состояния привода. Если атрибут Цвет фланцев включён (True), то цвет фланцев клапана всегда соответствует статическому значению атрибута Базовый цвет. В противном случае цвет фланцев соответствует цвету материала клапана.

ГЭ Насос имеет атрибут Форма насоса, который определяет форму соответствующего элемента экрана.

Положение элемента экрана, - его положение над или под другими элементами, задаётся соответствующими командами его контекстного меню.

Угловое положение элемента экрана можно изменить только после выбора команды Повернуть его контекстного меню.

Создание изображения резервуара для жидкости

3.1. На шаблоне экрана ЭкранМнемосхемы, используя ГЭ Цилиндр и ГЭ Сфера, из группы **Объёмные фигуры**, создать изображение вертикального резервуара с крышкой из объёмной полусферы и с основанием из объёмного усечённого конуса.

Примечание: Изображение резервуара поместить как можно ближе к нижней кромке экрана

Для поверхности резервуара и крышки использовать окраску под желаемый материал.

3.2. Используя ГЭ Труба, создать изображение подводящего и отводящего трубопровода согласно исходному рисунку задания. Верхний трубопровод следует разместить под крышкой резервуара, а нижний - немного выше дна резервуара.

3.3. На верхнем трубопроводе поместить насос, выбрав для него желаемый материал для корпуса и желаемую его форму.

3.4. На нижнем трубопроводе поместить клапан и выбрать для него форму клапана и форму привода. Повернуть клапан на 180 градусов, используя соответствующий режим редактирования.

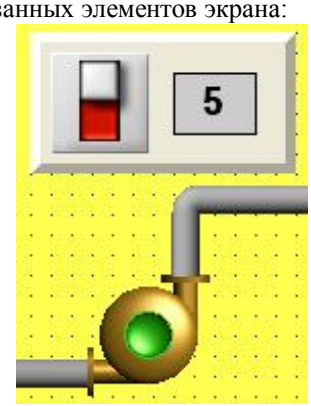
Создание элементов управления насосом и клапаном

Как насос, так и клапан должны имеет элемент управления включением и выключением и элемент для задания величины производительности.

Управление включением и выключением реализуются на основе ГЭ Выключатель и элемент задания производительности - на основе ГЭ Текст.

Указанные элементы должны размещаться на общей панели и иметь всплывающие подсказки с описанием выполняемых ими функций

Рекомендуется следующее размещение указанных элементов экрана:



3.5. На экране, рядом с изображением насоса, создать панель с ГЭ типа **Выключатель** и с ГЭ типа **Текст**.

Настроить созданный ГЭ типа **Выключатель**:

- назначить привязку к аргументу экрана с именем Вкл_насоса;
- для свойства Вид индикации задать условие Arg & Конст.;
- для свойства Константа и Значение задать значение равное 1;
- для свойства Подсказка задать текстовое значение с описанием выполняемой функции.

Настроить созданный ГЭ типа **Текст**:

- для реализации ввода значения производительности назначить на желаемое событие действие Передать значение с типом передачи Ввести и передать с привязкой атрибута Результат к аргументу экрана с именем Произв_насоса;
- динамизировать свойство Текст, назначив ему отображение значения аргумента экрана Произв_насоса;
- для свойства Подсказка задать текстовое значение с описанием выполняемой функции.

3.6. Настроить экранный элемент изображения насоса. Для индикации состояния насоса использовать динамизацию цвета его привода с использованием пороговой индикации на значение порога равное 1 и с привязкой к аргументу экрана Налив.

3.7. Создать копию панели с элементами управления для насоса, поместив её рядом с экранным элементом изображения клапана.

Редактировать настройки созданных копий элементов управления, реализуя функции управления клапаном:

- для копии выключателя назначить привязку к аргументу экрана с именем Вкл_клапана;
- для свойства Подсказка выключателя изменить значение с описанием выполняемой функции;
- для копии текстового элемента редактировать назначенное действие - назначить привязку параметра Результат к аргументу экрана с именем Произв_клапана;
- для копии текстового элемента редактировать динамизацию свойства Текст - назначить привязку к аргументу экрана с именем Произв_клапана;
- для свойства Подсказка копии текстового элемента изменить значение с описанием выполняемой функции.

3.8. Настроить экранный элемент изображения клапана. Для индикации состояния клапана использовать динамизацию цвета его привода с использованием пороговой индикации на значение порога равное 1 и с привязкой к аргументу экрана Слив.

Создание элементов индикации движения содержимого трубопроводов

Для индикации движения содержимого в подводящем (с насосом) и отводящем (с клапаном) трубопроводе использовать видеоклипы движущейся жидкости (fluid_flow) из импортированной библиотеки.

3.9. В группе **Ресурсы** панели **Графические элементы** открыть ресурс Видеоклип и, используя соответствующую библиотеку окна навигатора, создать элементы индикации движения содержимого в трубопроводах: на трубопроводе насоса - элемент индикации налива продукта, а на трубопроводе клапана – элемент индикации слива продукта.

Выполнить настройку размеров созданных элементов экрана и выполнить их привязку к соответствующим аргументам экрана.

Направление движения продукта должно отражать назначение трубопроводов.

Примечание – Для удобства размещения ГЭ на трубопроводе рекомендуется в настройках программы – в настройках РПД, отключить флаг Располагать по сетке, а также использовать масштабирование изображения в редакторе экрана.

3.10. Перейти в режим эмуляции работы экрана и проверить:

- действие выключателей;
- текстовых элементов для задания производительности;
- цветовую индикацию для насоса и клапана их состояния: изменение индикации у насоса при назначении налива и изменение индикацию клапана при назначении слива;
- индикацию наличия налива и слива на основе видеоклипов.

4. Создание программы имитации формирования запаса в резервуаре

Для имитации формирования запаса в резервуаре используется программа, реализующая функцию интегрирования прихода и расхода продукта.

Каждая программа состоит из тела и набора элементов: аргументов, переменных, функций и других элементов, которые образуют структуру программы. Структура программы выводится в отдельном окне при вызове шаблона программы на редактирование. При этом элементы структуры программы отображаются в виде дерева, основой которого является непосредственно сама программа.

Базовыми элементами структуры являются аргументы программы. Через свои аргументы программа обменивается данными с компонентами узлов проекта и прежде всего со значениями каналов узлов проекта. При этом используется два типа аргументов: входные аргументы, которые используются для получения данных для обработки, и выходные аргументы, которые используются для вывода результатов выполнения программы.

Все виды элементов структуры: собственно программа (её содержание), её аргументы и переменные, а также другие элементы структуры, редактируются в соответствующих редакторах. Содержательная часть программы, - её тело, редактируется в редакторе программы, соответствующем используемому языку.

Язык программирования (тип программы) выбирается в окне диалога, которое вызывается при первом нажатии ЛКМ на имени программы в её структуре. После закрытия окна выбора языка открывается соответствующий редактор тела программы.

Создание программы проекта

Программы проекта создаются в слое **Шаблоны программ**. Тип программы (язык программирования) назначается при первом открытии программы на редактирование

4.1. В слое **Шаблоны_программ** окна **Навигатора проекта** создать компонент типа **Программа** и назначить ему имя Резервуар.

4.2. Открыть созданный компонент для редактирования, используя соответствующую команду его контекстного меню или двойное нажатие ЛКМ.

4.3. В окне структуры программы выбрать выделить корень структуры – имя программы. В появившемся окне **Выбор языка программирования** отметить опцию FBD диаграмма и принять сделанный выбор.

Создание содержания FBD-программы

FBD программа представляет собой набор связанных **функциональных блоков**. Функциональный блок – это графическое изображение вызова функции.

Функциональный блок (ФБ) отображается прямоугольником, в верхней части которого выводится обозначение функции. Именованные отрезки слева обозначают информационные входы, а отрезки справа - выходы блока (возвращаемые функцией значения). Вход без имени - это вход управления выполнением (RUN).

Для разработки FBD программы в общем случае нужно выполнить следующие операции:

- разместить необходимые ФБ в рабочем поле FBD-редактора;
- соединить нужные входы и выходы ФБ, реализуя требуемый алгоритм обработки данных;
- создать набор требуемых аргументов и переменных в структуре программы;
- привязать входы/выходы FBD-диаграммы к аргументам и переменным программы или назначить входам требуемые константам.

Выбор ФБ выполняется в окне **FBD блоки**, которое вызывается кнопкой (Показать/скрыть палитру FBD блоков) панели инструментов редактора программы.

ФБ отображаются в составе функциональных групп: ***Арифметические***, ***Логические***, ***Генераторы*** и т.д. Выбор группы выполняется в нижней части окна **FBD блоки**. Для получения справки о вызываемой блоком функции, нужно дважды нажать ЛКМ на его изображении.

В поле FBD-редактора отображается сетка, в ячейках которой размещаются ФБ программы. Требуемый ФБ перетаскивается из окна их размещения в ячейку поля редактора. При этом в одной ячейке можно разместить только один ФБ.

Для передачи данных между ФБ создаются межблочные связи.

Для редактирования FBD программы её элементы (ФБ, их связи и иное) необходимо сначала выделить нажатием ЛКМ. Для выделения группы элементов нужно, удерживая нажатой ЛКМ, указать прямоугольную область выделения с захватом требуемых элементов диаграммы.

Для перемещения ФБ диаграммы её элементы следует выделить и «перетащить» удерживая нажатой ЛКМ.

4.4. На панели инструментов редактора программы нажать кнопку и поместить открывшееся окно **FBD блоки**, в нижней части экрана.

Создание функционала имитации формирования запаса продукта

Для имитации процесса формирования запаса в резервуаре используется интегрирование разности между величиной прихода и величиной расхода продукта.

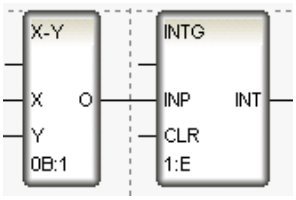
Для реализации требуемых функций следует использовать ФБ Вычитание, из группы ***Арифметические***, и ФБ Определённый интеграл, из группы ***Алгебраические***.

Их последовательное включение реализует интегрирование величины разности между приходом (уменьшаемое) и расходом (вычитаемое). Когда приход больше расхода – запас продукта увеличивается, в противном случае – запас уменьшается.

Для создания межблочных связей надо выделить один из связываемых входов/выходов и, удерживая ЛКМ, указать второй связываемый вход/выход. Возможность создания связи отображается изменением вида курсора: его вид ⃠ - операция невозможна, изменяется на вид - выбор элемента.

4.5. В окне редактора программы создать ФБ для получения разности между приходом и расходом и ФБ интегрирования разности. Для этого:

- в окне **FBD блоки** открыть вкладку (группу) ***Арифметические*** и «перетащить» ФБ Вычитание (X-Y) в поле редактора программы;
- в окне **FBD блоки** открыть вкладку (группу) ***Алгебраические*** и «перетащить» ФБ Определённый интеграл (INTG) в поле редактора программы, поместив его справа от первого ФБ, согласно следующему рисунку:



4.6. Изучить настойки (свойства) используемых ФБ по их описанию в справке ИС. Вызов описания выполняется командой Справка, контекстного меню каждого ФБ программы.

4.7. Создать связь между выходом O ФБ X - Y и входом INP ФБ INTG.

Создание и привязка аргументов программы

Исходные данные программа получает через свои входные аргументы. Для получения исходных данных соответствующие входы ФБ программы привязываются к входным аргументам программы. При этом входы ФБ, связанные с входными аргументами, являются входами программы.

Результаты обработки программой данных, формируемые на выходах соответствующих ФБ, выводятся через привязанные к ним выходные аргументы программы. При этом выходы ФБ, связанные с выходными аргументами, являются выходами программы.

Аргументы программы создаются в редакторе аргументов, который становится доступным при выделении в окне структуры программы ветви Аргументы.

Для создания нового аргумента используется кнопка (Создать аргумент) панели окна редактора.

Для привязки входа/выхода программы к созданным аргументам его надо выделить, Для выбора аргумента следует использовать инструмент (Привязать) или соответствующую команду контекстного меню входа/выхода. При этом возле выделенного входа/выхода выводится поле со списком доступных для привязки аргументов.

При привязке входов и выходов ФБ доступны только аргументы соответствующих типов.

4.8. В окне структуры программы перейти на элемент структуры **Аргументы**.

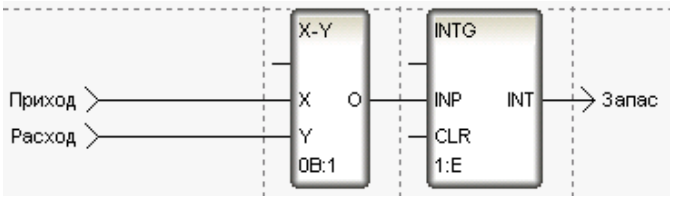
Создать два аргумента тип IN с типом данных REAL:

- с именем Приход для значения величины прихода продукта;
- с именем Расход для значения величины расхода продукта.

4.9. Входы программы, - входы ФБ X - Y, привязать к соответствующим входным аргументам программы: вход X к аргументу Приход и вход Y – к аргументу Расход.

4.10. Создать выходной аргумент с именем Запас, тип OUT, тип данных REAL, и выход программы – выходу ФБ INTG, привязать к созданному аргументу.

Внимание! *Не используйте и не назначайте привязки первым входам ФБ программы – входам без имени.*



5. Компиляция и отладка программы

При редактировании шаблона программы в области панелей инструментов ИС выводится панель **Программа** ( ) с набором инструментов для компиляции и отладки программы, а в меню ИС добавляется пункт Программа с командами для компиляции и отладки.

Перед отладкой программа должна быть скомпилирована – получен её машинный код.

Программы проекта можно использовать **только после их успешной компиляции**.

Компиляция программы

Для компиляции программы нужно выполнить команду Программа - Компилировать из меню пункта, или нажать клавишу F7 или кнопку (Компилировать) панели инструментов. При этом создается код для отладки программы в ИС.

Если при компиляции ошибки не обнаружены, становится доступной кнопки запуска отладки . В противном случае открывается окно **Сообщения** с сообщениями об ошибках компиляции.

Окно с результатами компиляции (и отладки) можно вызвать по кнопке (Сообщения).

5.1. Выполнить компиляцию программы. Если компиляция не выявила ошибок необходимо открыть окно **Сообщения** и ознакомиться с сообщением об успешной компиляции.

Отладка программы

Отладка программы возможна только после ее успешной компиляции.

При отладке программ можно использовать как непрерывный, так и пошаговый режим её выполнения. Для запуска требуемого режима отладки можно использовать команды пункта Программа главного меню или соответствующие кнопки панели инструментов. Непрерывное выполнение программы запускается командой Старт (, F5), а пошаговое выполнение - командами Шаг (, F10) и Трассировка (, F11). Для прекращения выполнения программы используется команда Стоп (, Shift+F10).

Для вывода служебных сообщений отладчика и компилятора используется специальные окна: окно **Переменные** ( ) с набором вкладок, окно **Стек** ( ) и окно **Сообщения** ( ) с вкладками ***Компиляция*** и ***Отладка***. Отображение окон управляется соответствующими командами пункта Вид и кнопками панели инструментов.

5.2. Открыть окно **Переменные**, используя соответствующую команду и пункта Вид меню ИС.

5.3. В поле Десятичные вкладки ***Аргументы*** назначить аргументу Приход желаемое значением в диапазоне от 2 до 5.

Нажатием клавиши F5 запустить программу на выполнение.

Наблюдать изменение значения выходного аргумента как в окне **Переменные**, так и на выходе ФБ INTG в редакторе программы.

5.4. В поле Десятичные вкладки ***Аргументы*** назначить аргументу Расход желаемое значение, большее, чем для аргумента Приход.

Наблюдать изменение значения выходного аргумента как в окне Переменные, так и на выходе ФБ INTG в редакторе программы.

5.5. Открыть окно **Сообщения** и определить время цикла выполнения программы.

5.6. Нажатием клавиши F10 остановить выполнение программы.

Создание функционала имитации работы пустого резервуара

Реализованный функционал формирования запаса обуславливает имитацию получения отрицательной величины запаса, что не может иметь место в реальной действительности. Для исключения указанного в программе имитации работы резервуара необходимо реализовать выполнение следующего условия: если запас становится равным или меньше нуля слив продукта должен быть равен его приходу. Такое состояние работы резервуара возникает, если величина расхода больше величины прихода.

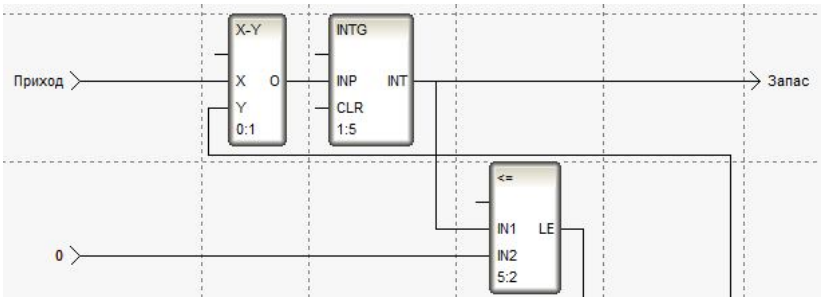
Реализация функционала выполнения требуемого условия надо:

- использовать понятие слив продукта, величина которого при наличии запаса равна расходу, а при отсутствии запаса его приходу
- контролировать обнуление величины запаса;
- контролировать знак разности между приходом и расходом продукта;
- если оба условия выполняются делать равной нулю интегрируемое значение и величину слива сделать равной величине прихода.

5.6. Открыть на редактирование программу Резервуар и создать аргумент тип OUT с типом данных REAL с именем Слив.

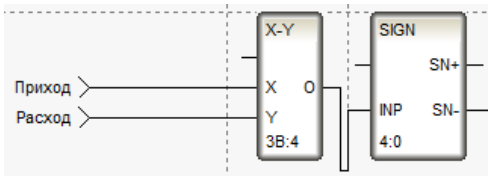
5.7. Редактировать содержание программы.

5.7.1. Для контроля обнуления величины запаса создать ФБ Меньше или равно (<=) из группы ***Сравнение***, назначив его входу IN1 связь с величиной запаса, а входу IN2 задать значение равное нулю.



На его выходе LE сформируется логическая единица (true), когда величина запаса станет отрицательной

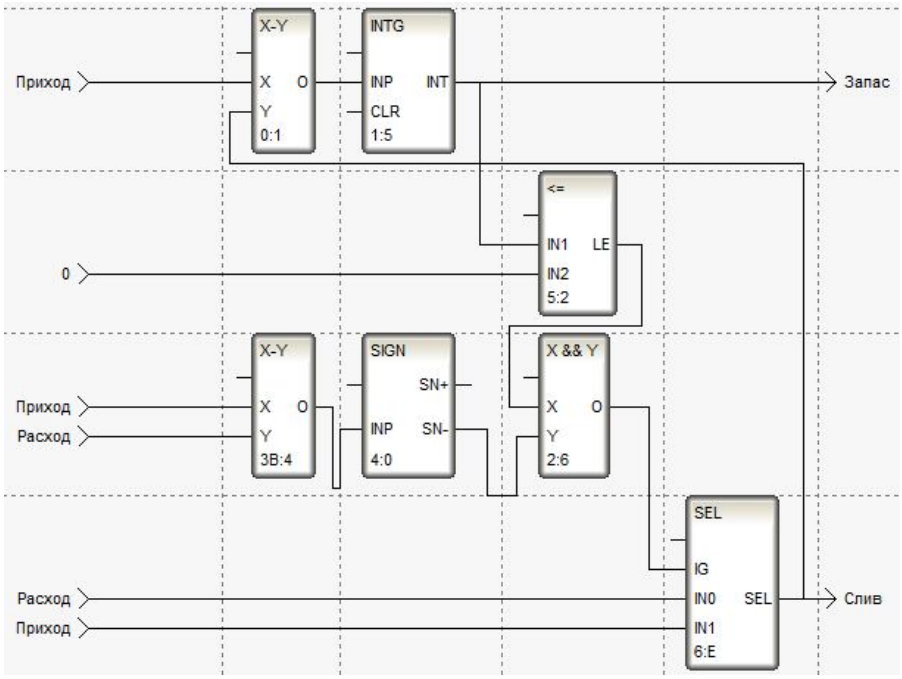
5.7.2. Используя ФБ Вычитание (X-Y) из группы ***Арифметические***, и ФБ Знаковая функция (SIGN) из группы ***Сравнение***, реализовать контроль знака разности между величиной прихода и расхода продукта.



На выходе SN- ФБ SIGN сформируется логическая единица (true), когда величина прихода будет меньше величины расхода – когда разность станет отрицательной.

5.7.2. Используя ФБ Логическое умножение (X&&Y) из группы ***Логические***, и ФБ Выбор из двух (SEL) из группы ***Выбор***, реализовать функционал переключения величины расхода (слива) при выполнении двух условия: обнуления запаса и наличия отрицательной разности между приходом и расходом продукта.

Внутренние связи и привязки к аргументам назначить согласно следующему рисунку:



5.8. Выполнить компиляцию и отладку программы.

Если приход больше расхода, то должен формироваться запас и слив должен быть равен заданному расходу.

Если приход меньше расхода, тот запас должен уменьшаться и когда запас будет меньше нуля слив должен стать равным приходу и величина запаса не должна изменятся.

6. Создание и настройка каналов для имитации работы оборудования

Создаваемая мнемосхема функционально реализует управление процессом накопления и расхода жидкого продукта с использованием резервуара, насоса и клапана. Для имитации их работы требуется использование каналов, реализующих функции соответствующего оборудования.

6.1. В узле АРМ создать группу каналов с именем Оборудование.

Создание и настройка каналов, имитирующего работу насоса и клапана

Работа насоса и клапана функционально подобна. Имитация их работы должна обеспечивать формирование величины прихода для насоса и расхода для клапана, как значений их производительности, при включении оборудования.

Для реализации такой функции следует использовать числовой канал класса **FLOAT** типа Output, которому на вход должны подаваться значения команд на включение и отключение оборудования (1 и 0), а величина производительности оборудования должна задаваться как значение для множителя (KX) его встроенной обработки данных. Тогда выходное значение канала (Q) – его выход, при команде на включение (1), будет принимать значение, равное производительности (Q=1·KX=KX), а при команде на выключение (0) – значение, равное нулю (Q=0·KX=0).

У таких каналов должна быть включена обработка данных.

6.2. Создать и настроить канал, имитирующий работу насоса. Для этого:

- в группе узла Оборудование создать канал класса **FLOAT** с именем Насос;
- открыть канал Насос для редактирования и выполнить следующее:
  - в группе настроек Обработка включить опцию Использовать и назначить для настройки Сглаж. значение из диапазона от 0,2 до 0,4;
  - в группе настроек Системные, в разделе ***Основные***, назначить тип Output;

6.3. Создать и настроить канал, имитирующий работу клапана. Для этого:

- в группе каналов узла Оборудование создать канал класса **FLOAT** с именем Клапан;
- открыть канал Клапан для редактирования и выполнить следующее:
  - в группе настроек Обработка включить опцию Использовать и назначить для настройки Сглаж. значение из диапазона от 0,2 до 0,4;
  - в группе настроек Системные, в разделе ***Основные***, назначить тип Output.

Создание и настройка канала, имитирующего работу резервуара

Работа резервуара имитируется программой Резервуар.

Для использования шаблонов программ в узлах проекта их необходимо вызвать в узел для выполнения. При этом одна программа может вызываться многократно как в один узел проекта, так и в разные узлы проекта.

Шаблон программы может вызываться в узел как каналами, специально созданными для вызова, - каналами класса **CALL** (аналогично каналам для вызова экранов), так и числовыми каналами класса **FLOAT**.

Для реализации в канале функции вызова необходимо выполнить соответствующую настройку канала. Надо в поле его свойства Вызов задать объект вызова - требуемый шаблон проекта (шаблон программы).

Канал с функцией вызова имеет набор аргументов, соответствующий аргументам вызываемого шаблона проекта. Условия использования вызываемого шаблона (вызываемой программы) определяются настройками канала его вызова – назначением постоянных значений или привязок аргументам канала вызова. При этом постоянные значения аргументам задаются как их значения по умолчанию, а в качестве привязок назначаются атрибуты (значения) каналов информационной базы узла.

Каналы для вызова шаблонов проекта (шаблонов программ, экранов и т.п.) создаются в узле как непосредственно, так и процедурой автопостроения каналов.

Для непосредственного создания канала вызова необходимо, в узле проекта или его группе создать компонент - канал класса **CALL**, и в окне его свойств на вкладке ***Информация*** в поле Вызов путём выбора задать вызываемый шаблон программы.

Для создания каналов вызова процедурой автопостроения необходимо в окне навигатора проекта перетащить вызываемый шаблон программы непосредственно на узел или его группу каналов. При этом в соответствующем месте будет создан канал вызова программы.

6.4. В группе каналов Оборудование, узла АРМ, создать непосредственно или процедурой автопостроения канал вызова программы Резервуар.

Канал должен иметь имя Резервуар.

Входные аргументы созданного канал вызова, на которых должны формироваться значения по приходу и расходу продукта, должны получать значения от каналов, имитирующих работу насоса и клапана.

На выходных аргумента имитируется формирование величины запаса и текущего значения слива продукта. Значения выходных аргументов в дальнейшем должна обрабатываться и регистрироваться, поэтому они должна фиксироваться в числовых каналах класса **FLOAT**

6.5. Открыть окно свойств канала Резервуар и выполнить следующее:

- аргументам Приход и Расход назначить привязки к выходным значениям существующих каналов имитации оборудования Насос и Клапан соответственно;

- выделить аргументы Запас и Слив нажатием иконки  (Создать по аргументам каналы с привязкой) на панели инструментов вкладки окна свойств создать одноимённые каналы с привязками.

Имя	Тип	Тип данных	Привязка
Приход	IN	REAL	Насос:Выходное значение (Система.АРМ.Оборудование)
Расход	IN	REAL	Клапан:Выходное значение (Система.АРМ.Оборудование
Запас	OUT	REAL	Запас:Входное значение (Система.АРМ.Оборудование)
Слив	OUT	REAL	Слив:Входное значение (Система.АРМ.Оборудование)

6.6. Каналы, создаваемые по аргументам, размещаются там же, где размещается используемый канал вызова – в группе каналов Оборудование. Созданные каналы содержит значение параметров объекта – величину запаса и слива. Следовательно, они является информационными каналами и должны размещаться в группе каналов информационной базы.

Переместить созданные каналы в группу Каналы информационной базы узла. Для этого используя клавишу Shift и дополнительное окно навигатора проекта перетащить канал Запас из группы Оборудование в целевую группу.

7. Реализация контроля заполнением резервуара

Для управления работой оборудования в реальном времени - отключение насоса при заполнении резервуара, необходимо реализовать контроль заполнения резервуара.

Контроля заполнения резервуара можно осуществить как по показанию величины запаса, так и по сигнализации заполнения и опустошения резервуара.

Реализация контроля величины и динамики изменения запаса продукта в резервуаре

Для реализации контроля величины запаса можно использовать визуализацию текущего значения запаса. Однако для принятия решения по управлению оборудованием важным условием является знание о динамике изменения регулируемого параметра. Наглядной информацией о динамике параметра является представление его изменения во времени. Возможность этого обуславливает элемент экрана на основе ГЭ типа **Тренд**.

Изменение динамики запаса зависит от включения и отключения насоса и клапана. Поэтому информация о связи динамики изменения запаса с состоянием оборудования, обуславливающего его изменение, является важной составляющей для принятия решений по управляющим воздействиям.

7.1. Создать элемент экрана типа **Тренд** со следующим составом кривых:

- кривую с именем Запас с привязкой к соответствующему аргументу экрана и с максимальным значением, равным желаемой максимальной величине запаса, принимая её равным значению в диапазоне от 1000 до 2000 массовых или объёмных единиц (л, кг, галлон, дм куб., м куб.);
- кривую с именами Слив с привязкой к соответствующему аргументу экрана и с максимальным значением, равным 100.

В группе настроек Ось значений основных свойств тренда параметру Показывать задать значение Все оси.

Создание элементов предупредительной индикации

Для управления заполнением резервуара – управления работой насоса, можно использовать индикацию события связанного с завершением наполнения резервуара.

7.2. Создать пороговый индикатор с текстовой мигающей сигнализацией на отключение насоса. Для этого:

- рядом с изображением насоса создать элемент экрана из ГЭ типа **Текст**;
- динамизировать его свойство Текст на условие Arg>=Конст., с привязкой к аргументу экрана Запас, с назначением для атрибута Если ИСТИННО в качестве значения надпись Отключи! и с назначением для атрибута Константа желаемого верхнего предупредительного граничного значения для величины запаса: например 90% от максимального его значения.

7.3. Перейти в режим эмуляции работы экрана и проверить действие элемента предупредительной индикации для насоса.

8. Назначение связей между компонентами проекта

Для реализации управления и визуализации работы имитационного оборудования необходимо назначить информационные связи между компонентами проекта

Для управления и контроля работы насоса и клапана, а также величины запаса необходимо в окне свойств канала вызова экрана Мнемосхема выполнить следующие привязки аргументов:

- аргументу Вкл_насоса назначить привязку к атрибуту Входное значение канала Насос;
- аргументу Вкл_клапана назначить привязку к атрибуту Входное значение канала Клапан;
- аргумента Произв_насоса назначить привязку к атрибуту Множитель канала Насос;
- аргумента Произв_клапана назначить привязку к атрибуту Множитель канала Клапан;
- аргументу Налив назначить привязку к атрибуту Выходное значение канала Насос;
- аргументу Запас назначить привязку к атрибуту Реальное значение канала Запас;
- аргументу Слив назначить привязку к атрибуту Реальное значение канала Слив.

Имя	Тип	Тип данных	Привязка
Вкл_насоса	OUT	USINT	Насос:Входное значение (Система.АРМ.Оборудование)
Вкл_клапана	OUT	USINT	Клапан:Входное значение (Система.АРМ.Оборудование)
Произв_насоса	OUT	REAL	Насос:Множитель (Система.АРМ.Оборудование)
Произв_клапана	OUT	REAL	Клапан:Множитель (Система.АРМ.Оборудование)
Налив	IN	REAL	Насос:Выходное значение (Система.АРМ.Оборудование)
Запас	IN	REAL	Запас:Реальное значение (Система.АРМ.Каналы_информационной_базы)
Слив	IN	REAL	Слив:Реальное значение (Система.АРМ.Каналы_информационной_базы)

9. Проверка работы мнемосхемы в реальном времени

Сохранить результаты проектирования. Обновить файлы узла проекта выполнить команду Сохранить для МРВ, выделить узел и открыть его в профайлере.

Запустить профайлер и выполнить имитацию управления насосом и клапаном, руководствуясь величиной уровня в резервуаре и предупредительной индикацией.

Контрольные вопросы

1. Какие как создаются элементы управления для вызова экранов узла?
2. Как обеспечивается первоочерёдность отображения экрана в реальном времени?
3. Какие встроенные ГЭ используются для изображения оборудования, и какие основные настройки они имеют?

4. Как создаётся и используется пользовательская библиотека компонент?
5. Как организована пользовательская библиотека компонент, что она собой представляет и где размещается?
6. Как можно индицировать движение содержимого трубопроводов?
7. Как создаются программы проекта, и из каких элементов состоит их структура?
8. Что такое FBD программа и как создаётся её содержание?
9. Через какие элементы структуры программа обменивается данными?
10. Как программы проекта используются в узлах проекта?
11. Что является источником и приёмником данных для программ, вызываемых в узел проекта?