



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электромеханика и промышленная автоматика»

А.А. ШКРОМАДО
Р.В. ШЕСТОВ
А.Н. БИРЮКОВ

SCADA-СИСТЕМЫ

Лабораторный практикум

Самара
Самарский государственный технический университет
2017

Печатается по решению редакционно-издательского совета СамГТУ

УДК 004.4

S SCADA-системы: Лабораторный практикум /Сост. *А.А. Шкромадо, Р.В. Шестов, А.Н. Бирюков.* – Самара: Самар. гос. техн. ун-т, 2017. – 64 с.: ил.

Рассмотрены основные аспекты применения SCADA-систем для разработки прикладного программного обеспечения систем автоматизации и организации управления технологическими процессами, приведены методические рекомендации к выполнению лабораторных работ по дисциплине «SCADA-системы».

Для студентов высших технических учебных заведений, обучающихся по направлению 15.03.04 «Автоматизация технологических процессов и производств».

УДК 004.4

Рецензенты: доцент кафедры конструкции и проектирования летательных аппаратов Самарского научно – исследовательского университета к.т.н. *Кременецкая М. Е.*

профессор кафедры «Электропривод и промышленная автоматика» ФГБОУ ВО «Самарский государственный технический университет» д.т.н. *Кузнецов П. К.*

© А.А. Шкромадо, Р.В. Шестов,
А.Н. Бирюков, 2017

© Самарский государственный
технический университет, 2017

ВВЕДЕНИЕ

Современные автоматизированные системы управления технологическими процессами (АСУТП) предполагают использование в качестве компонентов программного обеспечения систем сбора данных и диспетчерского управления (SCADA), которые в том числе решают задачи автоматического регулирования технологических параметров процесса.

В данном лабораторном практикуме по дисциплине «SCADA-системы» комплексно рассматриваются теоретические и практические аспекты разработки человека-машинного интерфейса (HMI), а также разработки прикладных программ автоматического регулирования с применением позиционных регуляторов, используя для этой цели инструментальную систему отечественной SCADA Trace Mode 6. Разработка программ выполняется с применением актуальных в промышленной автоматизации языков программирования стандарта МЭК 61131 – Structured Text и Function Block Diagram.

Обучающимся поэтапно предлагается ознакомиться с основными теоретическими сведениями по рассматриваемому материалу, после чего выполнить практическую часть лабораторной работы с подробным описанием каждого шага и его значения. Для отработки и закрепления полученных навыков обучающимся требуется выполнить самостоятельное задание. Для контроля усвоения теоретических положений в конце каждой работы предусмотрены контрольные вопросы.

В данном пособии представлены три лабораторные работы. Первая работа посвящена получению базовых навыков работы в инструментальной системе и изучению принципов создания проекта, знакомству с разработкой интерфейса оператора и языком программирования Structured Text (ST). Вторая и третья работа знакомят с двух- и трехпозиционными регуляторами соответственно, дают навыки разработки соответствующих программ на языке Function Block Diagram (FBD).

Тема 1. ОСНОВЫ РАЗРАБОТКИ ПРОЕКТА В TRACE MODE

ЛАБОРАТОРНАЯ РАБОТА №1

ОСНОВЫ РАЗРАБОТКИ ПРОЕКТА АСУТП В TRACE MODE

Цель работы: освоить порядок и принципы разработки проекта, включающего человеко-машинный интерфейс (HMI) и элементарную программу-эмулятор; освоить автопостроение каналов.

Задача работы: разработать технологический экран (мнемосхему) участка емкости Е-101, экран тренда по уровню в емкости и программу-эмулятор уровня в емкости на языке Structured Text.

Программное обеспечение: инструментальная система Trace Mode 6.

1.1. Порядок выполнения работы:

2. Изучите теоретические сведения, необходимые для выполнения лабораторной работы.

3. Выполните практические задания согласно методическим рекомендациям, описывающим содержание и порядок работы.

4. Составьте отчет по проделанной работе, включив в него результаты в соответствии с требованиями к содержанию отчета.

5. Подготовьте ответы на контрольные вопросы.

1.2. Теоретические сведения

Trace Mode 6 – это программный комплекс, предназначенный для разработки и запуска в реальном времени распределенных автоматизированных систем управления технологическими процессами (АСУТП).

Trace Mode относится к классу *SCADA* (Supervisory Control And Data Acquisition) – систем сбора данных и диспетчерского управления. В составе комплекса можно выделить два основных компонента:

1. *Интегрированная среда разработки проекта (инструментальная система* – ИС), содержащая средства для разработки проекта. Инструментальная система устанавливается на рабочем месте разработчика АСУ.

2. *Исполнительные модули (мониторы реального времени* – МРВ) – программные модули, под управлением которых под управлением которых в реальном времени выполняются составные части проекта, размещаемые на отдельных компьютерах или в контроллерах.

Данная составная часть проекта (группа файлов), размещенная на отдельном компьютере или в контроллере и выполняемая под управлением одного или нескольких МРВ, называется в терминологии Trace Mode **узлом проекта**. В общем случае, узлы могут выполняться под управлением различных мониторов. В случае запуска двух и более МРВ на одном аппаратном средстве, оно должно быть оборудовано соответствующим набором сетевых карт.

Также помимо вышеназванных компонентов комплекса следует выделять **драйвера обмена**, используемые МРВ для взаимодействия с устройствами, для которых отсутствуют встроенные в МРВ протоколы обмена, и дополнительные средства – модули, предназначенные для выполнения специализированных задач за исключением выполнения узлов.

Алгоритм работы любого МРВ заключается в анализе и исполнении структур переменных, именуемых каналами. **Каналы** представляют собой предопределенные алгоритмы обработки данных, часть из которых может настраиваться в процессе разработки проекта. По результатам анализа канала МРВ выполняет определенную операцию, что называется **отработкой канала**.

Для канала настраиваются два свойства – связь и вызов. Свойство «**связь**» позволяет настраивать информационные потоки АСУ. Свойство «**вызов**» позволяет вызвать **шаблон компонента проекта** (программы, экрана, документа и т.д.) с передачей параметров. Это свойство используется при реализации интерфейса оператора, передаче значения в базу данных и т.д. Совокупность каналов узла называется **базой каналов узла**.

Класс канала определяет его назначение. Переменные канала называются **атрибутами** и различаются **типом** (определяет направление передачи данных: Input – прием данных от источников, Output – посылка значения приемникам) и **типом данных** (диапазоном хранимых значений). Атрибуты, которые используются для передачи значений при вызове шаблона, называются **аргументами**.

Исполнение алгоритмов канала и анализ его атрибутов монитором называется **пересчетом канала**.

Под **шаблонами** подразумеваются компоненты проекта, которые при работе в реальном времени могут вызываться каналами с передачей параметров через аргументы. Передача параметров настраивается посредством **привязки** аргументов шаблона к каналам или источникам/приемникам.

Источники/приемники – это шаблоны специализированных каналов для обмена данными с различными устройствами и программными приложениями. Под устройствами понимаются контроллеры, а также модули и платы ввода-

вывода, обмен с которыми поддерживается мониторами Trace Mode, а также встроенные генераторы сигналов, модели объектов, системные переменные и т.д.

Trace Mode имеет встроенные средства для разработки математического обеспечения проекта – шаблонов программ контроля и управления.

Для разработки программ в ИС встроены языки *Техно ST (Structured Text)*, *Техно SFC (Sequential Function Chart)*, *Техно IL (Instruction List)*, *Техно LD (Ladder Diagram)* и *Техно FBD (Function Block Diagram)*, являющиеся модификациями соответствующих языком стандарта МЭК61131-3. Математическая обработка также выполняется внутренними алгоритмами числовых каналов.

Разработка графического интерфейса оператора также может выполняться непосредственно в ИС. Trace Mode обеспечивает графическое представление хода технологического процесса, а также управление процессом с помощью графических средств.

Графический интерфейс реализуется в виде графических экранов, шаблоны которых разрабатываются в *редакторе представления данных* (РПД) для узлов, обладающих значительными графическими возможностями и производительностью (например, АРМ оператора) либо в виде графических панелей, разрабатываемых в еРПД для узлов, обладающих ограниченными графическими возможностями и производительностью (контроллеры с ОС Windows CE).

Интерфейс включает в себя *графические элементы и объекты*, с помощью которых изображается мнемосхема процесса, текст, изображения, видеоклипы (анимацию), тренды, интерактивные элементы и др.

Помимо шаблонов прикладных программ и шаблонов экранов (интерфейса оператора) проект может содержать в себе такие компоненты как шаблоны документов (отчеты) в формате HTML, шаблоны связей с базами данных, описания источников/приемников (аппаратных тегов), словари сообщений и т.д. По результатам работы МРВ могут генерироваться архивы технологических параметров, отчеты тревог и др. Данные объекты рассматриваются далее постепенно в ходе выполнения лабораторных работ.

Набор возможных компонентов проекта определяется используемыми МРВ, поскольку узел может содержать только те компоненты, которые поддерживаются соответствующим монитором реального времени (семейством МРВ). Предопределенное название узла указывает на семейство мониторов, для которых данный узел предназначен.

Например, узел типа *RTM* предназначен для запуска на компьютере под управлением МРВ семейства RTM, поддерживающих графические экраны.

MPB данного типа имеют максимальные возможности, поскольку в них возможно создание шаблонов программ, документов, связей с БД, обмен данными по сетям с различным оборудованием и выполняющим пересчет каналов всех классов (кроме T-factory), и устанавливаются на АРМ оператора или управляющие станции под управление Windows или Linux (LinMPB).

1.3. Содержание и выполнение работы

Разработаем проект системы управления участком емкости Е-101. Оператор управляет вручную работой (пуском/остановом) насоса Н1, отвечающего за наполнение емкости (увеличение уровня), и электромагнитным запорным (открыт/закрыт) клапаном К1, который в открытом состоянии пропускает воду, что способствует уменьшению уровня. Управление насосом и клапаном осуществляется кнопками на экране оператора. Состояние исполнительных устройств отображается цветом (зеленый – включен/открыт, серый – выключен/закрыт).

На емкости производится отображение уровня по принципу смотрового окна с нанесенной шкалой, а также располагается цифровой указатель уровня, показывающий уровень в процентах. Кроме того, динамика изменения уровня отображается на графике (тренде), представляющем собой всплывающее окно.

Проект включает в себя два шаблона экрана и одну программу-эмулятор уровня воды в емкости в зависимости от состояния исполнительных устройств (насоса и клапана) на языке Structured Text.

Создание и сохранение проекта. Откройте инструментальную систему Trace Mode 6, используя ярлык TRACE MODE IDE 6 (base при использовании базовой версии) на рабочем столе либо через пункт меню *Trace Mode 6* → *TRACE MODE IDE 6*.

Если программа выдала запрос на выбор стиля проекта, выберите «Стандартный». В противном случае создайте проект вручную, используя пункт меню «Файл → Новый» либо нажмите кнопку «Создать новый проект»  на панели инструментов «Главная».

Сохраните проект, нажав кнопку «Сохранить текущий проект»  на панели инструментов «Главная» или используя пункт меню «Файл → Сохранить». Выберите местоположение проекта и задайте его имя. В имени файла недопустимо использование пробелов, а также нельзя начинать имя с цифры. Использование кириллицы допускается. В дальнейшем периодически сохраняйте проект вручную.

Учтите, что при каждом сохранении проекта, он сам сохраняется в файл «*имя_файла.prj*», а предыдущая версия сохраняется в виде резервной копии под именем «*имя_файла.~prj*». Ее можно использовать просто удалив (переименовав) «~» из расширения.

Создание узла проекта. Далее нажмите правой кнопкой мыши (ПКМ) на слое «*Система*» в навигаторе проекта и в контекстном меню выберите «*Создать узел→RTM*».

Создание шаблона экрана оператора. Аналогично нажмите ПКМ на созданном узле RTM_1 и выберите в контекстном меню «*Создать компонент → Экран*».

При этом в узле создается канал класса «*Вызов*» (об этом говорит буква *C* в обозначении – от английского CALL) типа *Input* (входной, об этом говорит буква *I* в обозначении)  *Экран#1:1*, вызывающий соответствующий компонент – шаблон экрана, на котором с помощью графических элементов (ГЭ) разработаем человеко-машинный интерфейс – технологический экран (мнемосхему) участка емкости Е-101. О том, что у канала настроено свойство «*связь*» с соответствующим шаблоном экрана, говорит зеленая стрелка на значке канала.

Один и тот же шаблон можно многократно вызывать в разных узлах, а изменения, вносимые в шаблон, отразятся во всех «ссылках» на него. Экраны проекта находятся (группируются) в слое «*Шаблоны_экранов*».

Нажмите ЛКМ на слое «*Шаблоны_экранов*» в навигаторе проекта и затем нажмите ПКМ в его правой части на шаблоне «*Экран#1*» и в контекстном меню выберите пункт «*Переименовать*». Введите название «*Мнемосхема*» и нажмите *Enter*. Вернитесь обратно в слой «*Система*» в узел *RTM_1*.

Обратите внимание, что канал вызова шаблона экрана также сменил свое название на «*Мнемосхема*». Можно было переименовать аналогично только сам канал вызова. При этом шаблон сохранил бы прежнее название.

Дважды нажмите левой кнопкой мыши (ЛКМ) на канале вызова  *Мнемосхема:1* в правой части навигатора проекта. Откроется *редактор представления данных* (РПД), в котором произведем размещение и настройку свойств ГЭ экрана оператора таким образом, чтобы он выглядел, как показано на рисунке 1.1.

Создание ГЭ «Емкость». Первоначально разместим на экране емкость (элемент 1). Для этого нажмите ЛКМ в правом нижнем углу (на треугольнике) значка группы «*Объемные фигуры*»  и выберите из раскрывающегося списка ГЭ «*Емкость*» . Значок группы на панели изменит вид на выбранный ГЭ.

Нажатие ЛКМ на экране определяет положение верхнего левого угла прямоугольника, обозначающего положение и размер ГЭ. Протяните указатель мыши (он в этом режиме имеет форму +) до получения требуемого размера емкости и нажмите любую клавишу мыши для завершения создания ГЭ. Для прекращения создания ГЭ нажмите на клавиатуре кнопку ESC.

Для изменения размеров, положения и иных свойств уже созданных ГЭ достаточно войти в *режим редактирования*, нажатием кнопки  на панели и выделением одинарным или двойным нажатием ЛКМ требуемого ГЭ (убедитесь, что редактируете нужный ГЭ – имя редактируемого элемента отображается вверху бланка свойств).

После этого наведение указателя на любой из маркеров прямоугольника, ограничивающего размер ГЭ, приводит к изменению формы указателя на двунаправленную стрелку, а протягивание ЛКМ

мыши за маркер меняет размер объекта. При помещении указателя над объектом, он принимает форму + , и с зажатой ЛКМ можно осуществлять перемещение объекта.

В свойствах ГЭ «Емкость», используя пункты списков «Верхний край» и «Нижний край» задайте подходящую форму емкости (см. рис. 1.1). Также можете изменить цвет емкости (свойство «Базовый цвет»).

Если свойства объекта не открылись автоматически, войдите в *режим редактирования* нажатием кнопки  на панели и выделите одинарным или двойным нажатием ЛКМ созданный ГЭ.

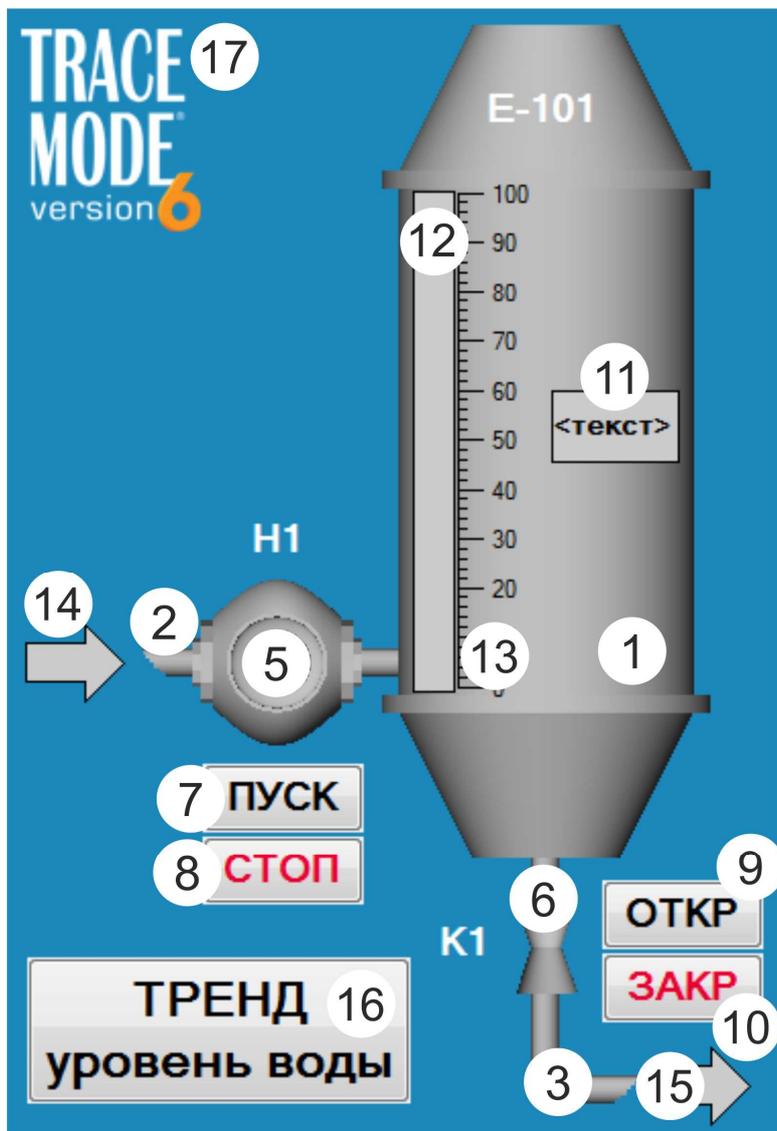


Рис. 1.1. Технологический экран оператора участка емкости E-101 «Мнемосхема»

Здесь и далее предполагается, если не указано иное, что все действия производятся с полями (ввод значений) или раскрывающимися списками (выбор вариантов), расположенными справа от пунктов свойств, чьи названия упоминаются в ходе выполнения работы.

Создание ГЭ «Труба». Далее покажем схематично наливную (элемент 2) и сливную (элемент 3) трубы. Из группы «Объемные фигуры» аналогично ГЭ «Емкость» выберем ГЭ «Труба» .

Создание трубы производится путем размещения ее узлов. Узлами являются начало, конец и каждый изгиб трубы. Установка всех узлов, кроме узла, соответствующего окончанию трубы, осуществляется нажатием ЛКМ; конечный узел ставится нажатием ПКМ. Для отрисовки прямолинейного участка с углом наклона, кратным 45° , нужно удерживать CTRL в процессе установки следующего узла.

Нарисуйте трубы как показано на рис. 1.1. В свойствах ГЭ «Труба» установите толщину 15 (свойство «Толщина») у обеих труб. В свойствах «Край 1» или «Край 2» для наливной и сливной труб соответственно выберите пункт «Диагональ 1», чтобы показать, что трубы выходят за пределы рассматриваемого участка.

Расположение уже созданных узлов трубы можно отредактировать, используя пункт «Редактировать узловые точки» контекстного меню выделенного ГЭ «Труба». При этом станут видны узлы (оранжевые маркеры), перемещая которые можно изменить конфигурацию трубы.

Создание ГЭ «Насос». Разместим на наливной трубе (элемент 2) насос (элемент 5). Для этого из группы «Объемные фигуры» выберем ГЭ «Насос» . Размещение и установка размеров производятся аналогично ГЭ «Емкость». Используя пункт свойств «Форма насоса» данного ГЭ, задайте форму, как показано на рисунке 1.1.

Для установки видимости взаимного расположения ГЭ (насос должен располагаться поверх трубы) можете воспользоваться пунктами «Переместить вверх»  или «Переместить вниз»  контекстного меню выделенного ГЭ. Аналогичные пункты меню доступны в меню «Сервис».

Создание ГЭ «Клапан». Разместим на сливной трубе (элемент 3) запорный (электромагнитный) клапан (элемент 6). Для этого из группы «Объемные фигуры» выберем ГЭ «Клапан» . Размещение и установка размеров производятся аналогично ГЭ «Емкость».

Если требуется повернуть объект (данный или любой другой), то достаточно войти в *режим редактирования*, нажатием кнопки  на панели и выделением требуемого ГЭ. Затем нажать ПКМ на объекте и выбрать в контекстном меню пункт «Повернуть». Наведение указателя мыши на любой из угловых маркеров рамки, ограничивающей ГЭ, приводит к изменению его формы на круговую стрелку. Нажав и удерживая ЛКМ в таком режиме можно осуществить поворот объекта. Разворот на 90° (горизонтально или вертикально) зачастую можно осуществить выбором соответствующего пункта свойства «Ориентация» в свойствах ГЭ.

Настройка отображения состояния насоса. Выберите ЛКМ ГЭ «Насос» на экране. Раскройте пункт свойств «Базовый цвет → Вид индикации» на вкладке «Общие свойства»  (подчеркнутые пункты свойств коричневого цвета имеют подпункты и раскрываются двойным нажатием ЛКМ) и выберите из списка вместо «Нет динамизации» пункт «Arg=const». Появятся дополнительные пункты свойств.

В свойстве «Привязка» нажмите справа на «...». Откроется *табличный редактор аргументов*. Нажатием клавиши *Insert* или кнопки «Создать новый аргумент»  выполните создание нового аргумента. Помните, что аргументы можно создавать не только в процессе разработки компонентов проекта, но и заранее.

Переименуйте аргумент в «Насос» (допускается использование кириллицы в учебных проектах, однако, как и в случае с именем проекта, недопустимо начинать имя аргумента с цифры и использовать в именах пробелы). Тип аргумента в данном случае можно оставить «IN».

Поскольку состояние насоса может принимать только два значения (пуск – логическая «1» либо стоп – логический «0»), то логично использовать тип данных *BOOL* – однобайтную переменную, которая может принимать значения *TRUE* (1) или *FALSE* (0), но в реальных проектах (и далее в данном пособии) возможно использование большего количества состояний (например, запрет пуска, неисправность и т.д.), поэтому будем использовать переменную типа *USINT* (целочисленное без знака) – однобайтную переменную, которая может принимать значения 0...255. Расход памяти при этом не изменяется – и в том, и в другом случае будет использован один байт.

Выберите из списка «Тип данных» пункт *USINT*.

После завершения задания свойств созданного аргумента «Насос», нажмите кнопку «Готово» в нижней части табличного редактора аргументов для со-

здания привязки цвета ГЭ «Насос» к значению одноименной переменной (аргумента).

Изменим цвет, соответствующий остановленному насосу. Для этого здесь же в пункте свойств «Если ЛОЖНО» вместо красного цвета выберем серый, нажав на раскрывающийся список справа от названия свойства. В поле свойства «Константа» введем «1», т.к. именно активное состояние насоса интересует нас в первую очередь – в дальнейшем (при написании программы-эмулятора) именно при присвоении аргументу значения «1» будет осуществляться «включение» насоса и, соответственно, наполнение емкости; в других случаях (любые другие значения аргумента кроме единицы) программа не будет выполнять никаких действий.

Таким образом, если аргумент «Насос» примет единичное значение (значение аргумента насос равно значению константы – единице), ГЭ «Насос» станет зеленого цвета, что соответствует состоянию «Пуск/работа». При любом другом значении одноименного аргумента, ГЭ «Насос» будет серого цвета, что соответствует состоянию «Стоп».

Настройка отображения состояния клапана. Выберите ЛКМ ГЭ «Клапан» на экране. Раскройте пункт свойств «Базовый цвет → Вид индикации» на вкладке «Общие свойства»  и выберите из списка вместо «Нет динамизации» пункт «Arg=const». Появятся дополнительные пункты свойств.

В свойстве «Привязка» нажмите справа на «...». Откроется *табличный редактор аргументов*. Нажатием клавиши *Insert* или кнопки «Создать новый аргумент»  выполните создание нового аргумента. Переименуйте аргумент в «Клапан». Тип аргумента в данном случае можно оставить «IN». Выберите из списка «Тип данных» пункт *USINT*.

После завершения задания свойств аргумента «Клапан», нажмите кнопку «Готово» в нижней части *табличного редактора аргументов*.

Изменим цвет, соответствующий закрытому клапану. Для этого здесь же в пункте свойств «Если ЛОЖНО» вместо красного цвета выберем серый, нажав на раскрывающийся список справа от названия свойства. В поле свойства «Константа» введем «1».

Проверка правильности задания свойств ГЭ (эмуляция). На любом этапе разработки шаблона экрана можно проверить правильность настройки свойств ГЭ и их работоспособность. Для этого используется режим «Эмуляция» . В данном режиме можно проверить только работу экрана, т.е. остальные компоненты проекта при этом не исполняются (программы не выполняются, каналы не пересчитываются и т.д.).

Нажмите кнопку «Эмуляция»  на панели ГЭ. В открывшемся окне значений аргументов введите «1» в поля значений аргументов «Насос» и «Клапан». При правильном задании свойств, данные ГЭ станут зеленого цвета. Теперь введите «0» (т.е. выключите насос, закройте клапан) – клапан с насос станут обратно серого цвета.

Отключите эмуляцию также, как производили ее включение – нажмите кнопку «Эмуляция»  на панели ГЭ.

В режиме эмуляции нельзя редактировать шаблон экрана.

Создание кнопок управления исполнительными устройствами. ГЭ «Кнопка». Создадим кнопки управления насосом.

Выберите на панели ГЭ инструмент «Кнопка» . Разместите кнопку под насосом (элемент 7), как показано на рисунке 1.1. Размещение и задание размера кнопки производится также, как и других ГЭ (см. создание ГЭ «Емкость»).

В пустом поле свойства «Текст» на вкладке «Общие свойства»  введите надпись «ПУСК» и нажмите *Enter* (разворачивать свойство не требуется). Нажатием ЛКМ на параметрах шрифта в свойстве «Шрифт» установите жирное начертание, 14 кегль, выбрав их в списке.

Переключитесь на вкладку «События» . Настроим действия, производимые при нажатии кнопки «ПУСК», а именно передачу значения «1» в аргумент «Насос». Для этого нажмите ПКМ на свойстве *MousePress* (свойство *MousePress* описывает действие, происходящее при нажатии оператором ЛКМ на кнопку). В контекстном меню выберите пункт «Передать значение». Появятся дополнительные пункты свойств.

Оставьте в свойстве «Тип передачи» значение «Прямая». В пустое поле свойства «Значение» введите «1» (логическая «1» – команда включения электродвигателя насоса). В свойстве «Результат» нажмите на «...» и выберите в редакторе аргументов аргумент «Насос», подтвердив выбор нажатием кнопки «Готово» в редакторе. Остальные свойства оставьте без изменений.

Создадим кнопку выключения насоса (элемент 8). Для этого скопируем кнопку с одновременным выравниванием ее относительно кнопки «ПУСК». Выделите кнопку «ПУСК» нажатием ЛКМ в режиме редактирования и дайте команду меню «Сервис → Тиражировать» .

В открывшемся окне в свойство «Число столбцов» введите «1», в «Число строк» введите «2» и нажмите кнопку «Готово». Снизу расположится точно такая же кнопка, уже выровненная относительно оригинала.

Измените значение, передаваемое в свойство «Значение» на вкладке «События»  с «1» на «0», а также измените значение в свойстве «Текст» на

вкладке «Общие свойства»  на «СТОП». Также измените цвет текста данной кнопки с черного на красный. Для этого выберите красный цвет в списке свойства «Цвет текста».

Создадим кнопки управления запорным электромагнитным клапаном. Скопируйте кнопки управления насосом. Для этого выделите их нажатиями ЛКМ с нажатой кнопкой *CTRL* на клавиатуре либо протягиванием (рамкой). После этого нажмите *CTRL+C* (или «Правка → Копировать» в меню), затем *CTRL+V* (или «Правка → Вставить» в меню). При активном режиме редактирования  переместите их, расположив рядом с клапаном на мнемосхеме как показано на рисунке 1.1 (элементы 9 и 10).

Переименуйте кнопку «ПУСК» в «ОТКР». В поле свойства «Текст» на вкладке «Общие свойства»  введите надпись «ОТКР» и нажмите *Enter*.

Поменяйте аргумент привязки данной кнопки. Для этого переключитесь на вкладку «События» , нажмите ЛКМ в поле с именем текущего привязанного аргумента свойства «Результат» и выберите в открывшемся *табличном редакторе аргументов* вместо аргумента «Насос» аргумент «Клапан». Подтвердите выбор нажатием кнопки «Готово» в редакторе.

Переименуйте кнопку «СТОП» в «ЗАКР» и поменяйте аргумент привязки данной кнопки аналогично рассмотренному выше примеру.

Нажмите кнопку «Эмуляция»  на панели ГЭ. Проверьте работоспособность кнопок – соответствующие исполнительные устройства должны менять цвет при их включении и отключении. Отключите эмуляцию.

Создание статических надписей. ГЭ «Текст». Статический текст используется в качестве поясняющих надписей. Подпишем основные элементы мнемосхемы: емкость Е-101, насос Н1 и клапан К1.

Выберите на панели ГЭ инструмент «Текст» . Разместите текст над насосом. Размещение и задание размера текстового блока производится также, как и других ГЭ (см. создание ГЭ «Емкость»).

В свойствах ГЭ «Текст» в пустом поле свойства «Текст» на вкладке «Общие свойства»  введите надпись «Н1» и нажмите *Enter* (разворачивать свойство не требуется). Если необходимо ввести текст в две строки, то переход на новую строку выполняется сочетанием клавиш *CTRL+Enter* или *Enter* на цифровой клавиатуре.

Нажатием ЛКМ на параметрах шрифта в свойстве «Шрифт» установите жирное начертание, 14 кегль, выбрав их в списке. Изменим цвет текста – выберем белый, нажав на раскрывающийся список свойства «Цвет текста» (раскрывать свойство не требуется).

Уберем рамку текста. Для этого в подпункте «*Стиль*» свойства «*Контур*» на вкладке «*Общие свойства*»  нажмите ЛКМ на изображении линии и выберите в списке пустой пункт (без линии).

Уберем заливку текста. Для этого, используя подпункт «*Стиль*» свойства «*Заливка*» (свойство нужно развернуть), нажмите ЛКМ справа на цветовой заливке и выберите в списке пункт «*Без заливки*».

Скопируйте данный текстовый элемент любым, из описанных выше способов, и разместите текстовые блоки так, как показано на рисунке 1.1. Измените надписи, отредактировав свойство «*Текст*» на вкладке «*Общие свойства*» .

Создание динамических надписей. ГЭ «Текст». Динамический текст используется для отображения значений численных величин либо может заменять одну надпись на другую в зависимости от значения аргумента, с которым он связан.

Разместим цифровой индикатор, отображающий уровень воды в емкости в процентах. Для этого используем тот же ГЭ.

Выберите на панели ГЭ инструмент «*Текст*» . Разместите текст на емкости, как показано на рисунке 1.1 (элемент 11). Нажатием ЛКМ на параметрах шрифта в свойстве «*Шрифт*» установите жирное начертание, 10 кегль, выбрав их в списке. Измените цвет текста – выберите черный, нажав на раскрывающийся список свойства «*Цвет текста*» (раскрывать свойство не требуется).

Настроим отображение численного значения уровня данным ГЭ. Раскройте пункт свойств «*Текст* → *Вид индикации*» на вкладке «*Общие свойства*»  и выберите из списка вместо «*Нет динамизации*» пункт «*Значение*». Появятся дополнительные пункты свойств.

В свойстве «*Привязка*» нажмите справа на «...». Откроется *табличный редактор аргументов*. Нажатием клавиши *Insert* или кнопки «*Создать новый аргумент*»  выполните создание нового аргумента. Переименуйте аргумент в «*Уровень*». Тип аргумента в данном случае можно оставить «*IN*». Оставьте в «*Тип данных*» пункт «*Real*».

Real – это четырехбайтный тип данных, который далее будет использоваться для всех численных переменных, хранящих вещественные числа. Максимальное значение составляет $3,402823466 \cdot e^{38}$.

После завершения задания свойств созданного аргумента «*Уровень*», нажмите кнопку «*Готово*» в нижней части табличного редактора аргументов.

Изменим формат отображения численного значения. Здесь же в пункте «*Формат*» свойства «*Текст*» выберите из раскрывающегося списка «*Float*» (с

плавающей точкой) вместо «*Generic*» (общий, текстовый). Уменьшите количество знаков дробной части с трех до одного, исправив «%.3f» на «%.1f».

Нажмите кнопку «Эмуляция»  на панели ГЭ. В открывшемся окне значений аргументов введите любое число в поле значения аргумента «Уровень». При правильном задании свойств данного ГЭ вы увидите введенное значение в цифровом индикаторе уровня. Выключите режим эмуляции.

Создание смотрового окна. ГЭ «Прямоугольник». Разместим на емкости «смотровое окно». Для этого нажмите ЛКМ в правом нижнем углу (на треугольнике) значка группы «Прямоугольники»  и выберите из раскрывающегося списка ГЭ «Прямоугольник» . Значок группы на панели изменит вид на выбранный ГЭ. Разместите ГЭ «Прямоугольник» так, как показано на рисунке 1.1 (элемент 12). Размещение и задание размера прямоугольника производится также, как и других ГЭ (см. создание ГЭ «Емкость»).

Реализуем динамическую заливку «смотрового окна», соответствующую уровню воды в емкости. Переключитесь на вкладку «Динамическая заливка» . Раскройте пункт свойств «Слои → Слой» и в подпункте «Привязка», нажав на «...», выберите в *табличном редакторе аргументов* аргумент «Уровень». Нажмите кнопку «Готово» в нижней части *табличного редактора аргументов*.

Измените цвет заливки, нажав на раскрывающийся список справа от названия свойства «Цвет заливки».

Свойства «Макс» и «Мин» на вкладке «Динамическая заливка»  позволяют задать значения, соответствующие полностью заполненному и пустому прямоугольнику – оставьте их значения без изменений.

Создание шкалы. ГЭ «Ползунок». Разместим на емкости шкалу рядом со «смотровым окном». Для этого нажмите ЛКМ на панели ГЭ на значке группы «Приборы» . По умолчанию будет выбран ГЭ «Ползунок» . Разместите ГЭ «Ползунок» рядом с ГЭ «Прямоугольник», как показано на рисунке 1.1 (элемент 13). Размещение и задание размера ползунка производится также, как и других ГЭ (см. создание ГЭ «Емкость»). Размеры ползунка изменятся при редактировании, поэтому положение подберите условно – по завершении редактирования зададите окончательное положение и размер ГЭ.

В режиме редактирования внесите следующие изменения в свойства ГЭ «Ползунок»: в качестве значения свойства «Ползунок» выберите «False», свойства «Полоса» – «False», «Положение ползунка» – «Слева».

Раскройте свойство «Рамка» и в пункте «Тип» выберите значение «Стандартная». В раскрывающемся списке пункта «Стиль» данного свойства выбо-

рите пустую строку. Раскройте свойство «Фон» и в раскрывающемся списке пункта «Стиль» данного свойства выберите «Без заливки».

Раскройте свойство «Шкала». Измените цвет текста – выберите черный, нажав на раскрывающийся список пункта «Цвет текста» данного свойства.

Раскройте пункт «Уровень 2» свойства «Шкала» и выберите значение «False» в подпункте «Использовать». Разверните пункт «Уровень 1» и введите значение «10» в подпункт «Число делений», нажав после *Enter*. В подпункт «Десятичные знаки» введите значение «0», также нажав после *Enter*.

Нажмите кнопку «Эмуляция»  на панели ГЭ. В открывшемся окне значений аргументов введите любое число от 0 до 100 в поле значения аргумента «Уровень». При правильном задании свойств прямоугольника вы увидите заполнение «смотрового окна» до требуемой отметки на шкале. Выключите режим эмуляции.

Создание плоских фигур. ГЭ «Стрелка». Наряду с объемными фигурами на мнемосхемах могут использоваться и плоские фигуры. Изобразим стрелки, поясняющие направления потоков жидкости на мнемосхеме управления.

Для этого нажмите ЛКМ в правом нижнем углу (на треугольнике) значка группы «Плоские фигуры»  и выберите из раскрывающегося списка ГЭ «Стрелка» . Значок группы на панели изменит вид на выбранный ГЭ. Размещение и задание размера стрелок производится также, как и других ГЭ (см. создание ГЭ «Емкость»). Нарисуйте стрелку для входящего (элемент 14) и исходящего потоков (элемент 15), как показано на рисунке 1.1.

Плоские фигуры также, как и почти любой ГЭ, могут выполнять функции динамизации. Настройте изменение цвета при активации соответствующего исполнительного устройства также, как это было сделано для насоса и клапана (см. Настройка отображения состояния насоса). Процедура аналогична с тем лишь отличием, что работа происходит с пунктом свойств «Заливка → Цвет заливки». Привязка стрелки входного потока выполняется к созданному ранее аргументу «Насос», выходного – «Клапан» соответственно.

Создание всплывающего экрана. Разместим тренд (график) изменения уровня на отдельном экране, оформленном в виде всплывающего окна.

Создайте еще один экран так, как описано в «Создание шаблона экрана оператора». Переименуйте шаблон экрана в «Тренд». Откройте его на редактирование в РПД двойным нажатием ЛКМ.

Измените размер окна. Для этого нажмите дважды ЛКМ на любом свободном пространстве экрана, чтобы вызвать окно свойств самого экрана.

В раскрывающемся списке свойства «Размеры» выберите размер «640x480». Также выберите у свойства «Всплывающее окно» значение «True». Раскройте данный пункт и обратите внимание, что можно установить координаты первого появления окна и необходимость его открытия сразу же при запуске проекта.

Создание графика технологического параметра. ГЭ «Тренд». Чтобы показывать динамику изменения значений технологических параметров во времени используются тренды.

Выберите на панели ГЭ группу «Тренды»  – по умолчанию будет выбран нужный ГЭ «Тренд». Размещение и задание размера тренда производится также, как и других ГЭ (см. создание ГЭ «Емкость»). Разместите тренд на все свободное пространство экрана (рис. 1.2).

На вкладке «Общие свойства»  ГЭ «Тренд» в свойстве «Заголовок» введите текст «Уровень воды Е-101» и нажмите *Enter*. Нажатием ЛКМ на параметрах шрифта в свойстве «Шрифт» установите жирное начертание, 10 кегль, выбрав их в списке. Разверните свойство «Легенда» и в подпункте «Показывать» выберите из списка значение «False».

Перейдите на вкладку «Кривые» . Нажмите ПКМ на свойстве «Кривые» и в контекстном меню выберите пункт «Кривая». Среди появившихся подпунктов данного свойства нажмите ЛКМ в свойстве «Привязка» справа на «...». Откроется *табличный редактор аргументов*. Нажатием клавиши *Insert* или кнопки «Создать новый аргумент»  выполните создание нового аргумента (у каждого компонента проекта должны быть свои аргументы. Экран без аргументов не будет вызываться вовсе). Переименуйте аргумент в «Уровень». Тип аргумента в данном случае можно оставить «IN». Оставьте в «Тип данных» пункт «Real». После завершения задания свойств аргумента, нажмите кнопку «Готово» в нижней части *табличного редактора аргументов*.

В подпункте «Толщина линии» введите «3». Значения в подпунктах «Макс. значение» и «Мин. значение» определяют пределы шкалы тренда – оставьте их без изменений, т.к. шкала соответствует измерению уровня в процентах.

В результате экран примет вид как на рис. 1.2. Проверьте работу тренда в режиме эмуляции.

Создание кнопки вызова всплывающего окна. ГЭ «Кнопка». Для вызова всплывающего окна будем использовать кнопку. Откройте на редактирование шаблон экрана «Мнемосхема» через навигатора проекта или просто переключившись на соответствующую вкладку РПД.

Выберите на панели ГЭ инструмент «Кнопка» . Разместите кнопку под насосом (элемент 16), как показано на рисунке 1.1. Размещение и задание размера кнопки производится также, как и других ГЭ (см. создание ГЭ «Емкость»).

В пустом поле свойства «Текст» на вкладке «Общие свойства»  введите надпись «ТРЕНД», нажмите *CTRL+Enter*, введите надпись «уровень воды». Нажатием ЛКМ на параметрах шрифта в свойстве «Шрифт» установите жирное начертание, 18 кегль, выбрав их в списке.

Переключитесь на вкладку «События» . Затем нажмите ПКМ на свойстве *MouseDown* и в контекстном меню выберите пункт «Перейти на экран». Появятся дополнительные пункты свойств. В подпункте «Перейти на экран» нажмите ЛКМ справа и выберите экран «Тренд».

Учтите, что в режиме эмуляции всплывающее окно не появляется, т.к. является самостоятельным компонентом проекта, отличным от исполняемого в режиме эмуляции.

Изменение размеров основного окна «Мнемосхема». Для этого нажмите дважды ЛКМ на любом свободном пространстве экрана. Выберите пункт «Произвольно» (в этом случае свойство «Размеры» можно будет развернуть) и введите размеры экрана в пикселях вручную либо используйте скроллинг колесом мыши так, чтобы размер экрана соответствовал размещенным на нем ГЭ.

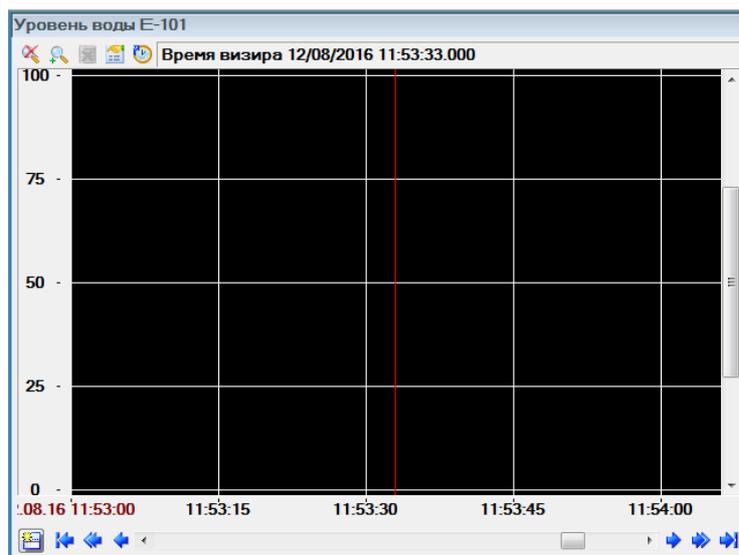


Рис. 1.2. Экран «Тренд»

Добавление в проект ресурсов. Картинки. К ресурсам проекта относятся тексты, картинки, анимация и готовые графические объекты (емкости, насосы, теплообменники и т.д.)

Добавим картинку логотипа Trace Mode 6 на экран оператора. Для этого нажмите ПКМ на слое «Ресурсы» в навигаторе проекта и выберите в контекстном меню «Создать группу → Картинки». Затем нажмите ПКМ на пункте «Картинки» и выберите в контекстном меню «Создать компонент → Библиотека_изображений».

Дважды нажмите ЛКМ на созданной группе *Библиотека_изображений*, открыв ее, и затем в рабочем поле самой библиотеки нажмите ПКМ и в контекстном меню выберите пункт «*Импортировать*»  либо используйте соответствующую кнопку на панели ресурсов. Откроется окно проводника, где необходимо указать путь к папке «C:\Program Files (x86)\AdAstra Research Group\Trace Mode IDE 6\Lib\Logotip\» (на примере Windows 7) и затем выбрать файл *logo_tm6_100*. Картинка будет добавлена в библиотеку.

Точно также в проект могут быть добавлены любые картинки форматов *bmp*, *png*, *jpg* или *xpm*.

Чтобы использовать добавленный в библиотеку логотип на экране, переключитесь на вкладку РПД для шаблона экрана «*Мнемосхема*». На панели ГЭ нажмите кнопку «*Ресурсы*» , что откроет ресурсную библиотеку растровых изображений. Нажмите ЛКМ на картинке в библиотеке и, удерживая ее, перетащите картинку на экран, разместив в требуемом месте (элемент 17).

Для того чтобы выбрать иные виды ресурсов – тексты, видеоклипы, векторные изображения – нужно, как и в случае с другими инструментами, дважды нажать ЛКМ на треугольнике в правом нижнем углу кнопки «*Ресурсы*»  на панели ГЭ и выбрать нужный вид ресурса. Откроются соответствующие ресурсные библиотеки.

Разработка графической части проекта завершена. Приступим к разработке прикладного программного обеспечения проекта.

Разработка программы-эмулятора изменения уровня в емкости. Нажмите ПКМ на созданном узле *RTM_1* и выберите в контекстном меню «*Создать компонент → Программа*».

При этом в узле создается канал класса «*Вызов*» типа *Input*, вызывающий соответствующий компонент – шаблон программы, в котором на языке *Structured Text* напишем программу изменения уровня воды в емкости *E-101* в зависимости от состояния исполнительных устройств. Программы проекта находятся (группируются) в слое «*Шаблоны_программ*».

Нажмите ЛКМ на слое «*Шаблоны_программ*» в навигаторе проекта и затем нажмите ПКМ в его правой части на шаблоне «*Программа#1*» и в контекстном меню выберите пункт «*Переименовать*». Введите название «*Эмуляция_уровень*» и нажмите *Enter*. Вернитесь обратно в слой «*Система*» в узел *RTM_1*. Обратите внимание, что канал вызова шаблона программы также сменил свое название на указанное вами.

Дважды нажмите левой кнопкой мыши (ЛКМ) на канале вызова  Эмуляция_уровень:3 в правой части навигатора проекта. Откроется редактор шаблонов программ (РШП).

Написание программ всегда следует начинать с предварительного создания аргументов (переменных).

Нажмите ЛКМ на пункте «Аргументы» РШП и создайте в табличном редакторе аргументов программы три аргумента аналогично тому, как это делалось при разработке шаблона экрана (см. Настройка отображения состояния насоса, Создание динамических надписей. ГЭ «Текст»).

Имена аргументов сохраним такими же, как и при разработке шаблонов экранов. Тип данных выберем их тех же соображений, что и при разработке шаблонов экранов.

Тип аргументов определяется направлением потоков данных. Те аргументы, без значений которых программа не сможет работать и которые необходимо вводить в качестве исходных данных, т.е. значения которых могут изменяться оператором, должны иметь тип *IN*. Те же, значения которых вычисляются в программе и не должны быть изменены кроме как в результате пересчета самой программой – *OUT*.

Поскольку для расчета значения уровня необходимо знать состояния насоса и клапана и оператор может влиять на значения этих аргументов, нажимая кнопки передачи соответствующих значений на экране, то эти аргументы должны иметь тип *IN*, а т.к. значение уровня вычисляется самой программой и на него оператор не может оказывать влияния, то аргумент «Уровень» должен иметь тип *OUT*.

Таким образом, аргументы должны иметь следующие параметры: *Насос IN USINT*; *Клапан IN USINT*; *Уровень OUT REAL*.

Нажмите ЛКМ на заголовке программы «Эмуляция_уровень». В появившемся окне выбора языка программирования выберите язык *ST* – паскалеподобный текстовый высокоуровневый язык программирования, обеспечивающий максимально гибкие возможности для написания программ. Нажмите в этом окне кнопку «Принять».

Обратите внимание, что *РШП* уже «прописал» ключевую конструкцию *Program ... End Program*, обозначающие точку входа в программу и выхода из нее, а также созданные ранее аргументы с указанием типа и типа данных.

Алгоритм работы программы-эмулятора заключается в следующем. Пока значение уровня меньше 100% (есть куда наполнять), включение насоса приводит к увеличению уровня в емкости на 0,5% за каждый пересчет программы;

при выключенном насосе уровень не увеличивается (увеличение 0%). В противном случае (уровень больше либо равен 100%) увеличения уровня не происходит – он равен максимальному значению 100% (емкость полностью заполнена).

Пока значение уровня больше 0% (есть что сливать), открытие клапана приводит к уменьшению уровня в емкости на 1,0% за каждый пересчет программы (пренебрежем гидростатическим давлением – скорость слива считаем постоянной); при закрытом клапане уровень не уменьшается (уменьшение 0%). В противном случае (уровень меньше либо равен 0%) уменьшения уровня не происходит – он равен минимальному значению 0% (емкость пуста).

Для проверки условий соответствия уровня граничным значениям и будем использовать оператор *if-then-else*. Оператор имеет конструкцию

```
if {выражение} then {последовательность предложений1};  
else {последовательность предложений2};  
end_if;
```

Действие: *если {выражение1} истинно, то выполняется {последовательность предложений1}, иначе выполняется {последовательность предложений2}.*

Таким образом, проверка уровня будет иметь вид

```
if Уровень < 100 then {последовательность предложений1};  
else Уровень = 100;  
end_if;  
if Уровень > 0 then {последовательность предложений2};  
else Уровень = 0;  
end_if;
```

{*Последовательность предложений1*} и {*последовательность предложений2*} – это блоки, в которых будет размещено описание действий по изменению уровня в зависимости от состояния насоса и клапана соответственно.

Для описания изменения уровня в зависимости от состояния насоса и клапана будем использовать оператор *Case* (хотя можно было бы использовать и примененный ранее оператор *if-then-else*), имеющий конструкцию

```
case {выражение} of  
{список значений}: {последовательность предложений};  
...  
{список значений}: {последовательность предложений};  
else {последовательность предложений};  
end_case;
```

Действие: если результат вычисления {выражения} принадлежит множествам, заданным {списками значений}, выполняется соответствующая {последовательность предложений}. Если результат вычисления {выражения} не принадлежит ни одному из заданных множеств, выполняется {последовательность предложений}, следующая за ключевым словом *else*.

В рассматриваемом случае часть оператора, следующую после *else*, можно опустить. Тогда изменение уровня в зависимости от состояния насоса и клапана будет иметь вид

```
case Насос of
1: Уровень=Уровень+0.5;
0: Уровень=Уровень;
end_case;
case Клапан of
1: Уровень=Уровень-0.5;
0: Уровень=Уровень;
end_case;
```

Обратите внимание, что в качестве разделителя целой и дробной части используется точка, а не запятая.

Итоговый вид (листинг) программы-эмулятора показан на рисунке 1.3.

Компиляция и отладка. После написания (или в процессе) программы необходимо выполнить ее компиляцию, включающую в себя проверку синтаксической правильности.

Для этого нажмите кнопку «Компиляция»  на панели инструментов отладчика или воспользуйтесь клавишей *F7*. Все пункты, касающиеся компиляции и отладки доступны также в меню «Программа» строки меню.

В случае успешной компиляции не выводится сообщений об ошибках. Убедиться в успешной компиляции (compiled succesfully) можно также открыв *окно сообщений* кнопкой  на панели инструментов отладчика.

```
PROGRAM
VAR_OUTPUT Уровень : REAL; END_VAR
VAR_INPUT Насос : USINT; END_VAR
VAR_INPUT Клапан : USINT; END_VAR

if Уровень<100 then
case Насос of
1: Уровень=Уровень+0.5;
0: Уровень=Уровень;
end_case;
else Уровень=100;
end_if;
if Уровень>0 then
case Клапан of
1: Уровень=Уровень-0.5;
0: Уровень=Уровень;
end_case;
else Уровень=0;
end_if;

END_PROGRAM
```

Рис. 1.3. Листинг программы

Сообщения делятся на относящиеся к *компиляции* и к *отладке*. Переключение между ними выполняется соответствующими вкладками внизу *окна сообщений*.

При наличии ошибок *окно сообщений* откроется автоматически. В нем будет указан код ошибки и текстовое пояснение к ней. При двойном нажатии ЛКМ на коде ошибки в самом тексте программы строка, в которой предположительно возникла ошибка, будет выделена серым треугольником. Следует учесть, что компилятор выделяет не строку, а предложение, поэтому если вы забыли разделитель (например, точку с запятой) предложений, то выделена может быть и предыдущая строка.

После успешной *компиляции* всегда выполняйте *отладку* программы, чтобы убедиться, что программа не просто сможет запуститься и работать, но и будет выполнять именно требуемые действия и должным образом.

Запустите программу на циклическое выполнение, нажав кнопку «*Старт*»  на панели инструментов отладчика или воспользуйтесь клавишей *F5*.

Откройте окно переменных, нажав кнопку «*Переменные*»  на панели инструментов отладчика. Вводя значения «1», соответствующие пуску насоса и открытию клапана, и «0», соответствующие останову насоса и закрытию клапана, в поля значений (можно использовать любое поле: десятичное, шестнадцатеричное, двоичное) аргументов «*Насос*» и «*Клапан*» наблюдайте за изменением значения аргумента «*Уровень*». Проверьте корректность работы программы при достижении граничных значений 0% и 100%.

Проверив работу программы, остановите ее выполнение кнопкой «*Стоп*»  на панели инструментов отладчика или воспользуйтесь сочетанием клавиш *SHIFT+F5*.

Создание базы каналов. Для того чтобы передавать данные между программой и экранами проекта, потребуется создать каналы. Для этого воспользуемся процедурой *автопостроения* каналов по аргументам шаблонов программы или экрана.

Нажмите ПКМ на канале вызова шаблона экрана «*Мнемосхема*» (процедуру можно проводить с любым шаблоном вызова), раскрыв узел *RTM_1* в навигаторе проекта, и выберите в контекстном меню пункт «*Свойства*» . В открывшемся окне переключитесь на вкладку «*Аргументы*». Далее нажмите ЛКМ на кнопку «*Создать по аргументам каналы с привязкой*»  на панели работы с аргументами.

Если нужно создать каналы не по всем, а только по конкретным аргументам и привязать их, то сначала нужно выделить аргументы, по которым будут

создаваться и привязываться каналы нажатием ЛКМ с зажатой клавишей *CTRL* для выборочного выделения или *SHIFT* для выделения диапазона. Обратите внимание, что при создании каналов указанным образом (непосредственно в узле) узел должен быть выделен в навигаторе проекта. Иначе каналы не будут принадлежать узлу, а будут сгруппированы до последующего распределения по узлам в слое «База каналов», видимость которого настраивается в настройках ИС.

При правильном выполнении действий в узле появятся три канала с именами, идентичными аргументам, по которым они были созданы. Аргументам с типом данных *USINT* соответствуют каналы класса *HEX16* ; с типом данных *REAL* – класса *FLOAT* . Также обратите внимание, что в таблице привязок аргументов у аргументов типа *IN* привязка каналов выполняется к атрибуту «Реальное значение». Для аргументов типа *OUT* привязка каналов выполняется к атрибуту «Входное значение».

Привязка к неверному атрибуту может вызвать неверный результат пересчета канала. Поэтому при изменении типа данных аргумента необходимо удалить ранее созданный канал, имеющий уже неверный класс, и создать его повторно классом, соответствующим новому типу данных аргумента.

При изменении типа аргумента можно пойти более простым путем. В таблице аргументов в свойствах канала, подобной той, что была открыта выше, нажмите дважды ЛКМ в поле «Привязка» нужного аргумента и выберите из списка на вкладке «Атрибуты» соответствующий типу аргумента атрибут (реальное или входное значение).

Теперь необходимо связать созданные каналы со вторым шаблоном экрана. Аналогично нажмите ПКМ на канале вызова шаблона экрана «Тренд» и выберите в контекстном меню пункт «Свойства» . В открывшемся окне переключитесь на вкладку «Аргументы». Выделите ЛКМ канал  Уровень в навигаторе проекта и, не отпуская ЛКМ, протяните его на строку аргумента  Уровень   REAL в табличном редакторе привязок аргументов. Привязка будет создана автоматически. Редактировать ее не требуется.

Повторите процедуру с перетаскиванием каналов с целью создания привязок для канала вызова шаблона программы «Эмуляция_уровень».

Сохранение и подготовка проекта к запуску. Нажмите ЛКМ кнопку «Сохранить для MPB»  на главной панели инструментов или выберите аналогичный пункт в меню «Файл». После этого в папке, где был сохранен проект, появится папка с файлами (ее имя совпадает с именем проекта), сгруппирован-

ными по вложенным папкам, соответствующим узлам проекта. Таким образом, файлы удобно разделены для загрузки – каждому узлу как отдельному устройству соответствует отдельная папка.

Если в проект были внесены изменения, то перед экспортом будет выдан запрос на сохранение внесенных изменений.

Типы файлов, создаваемых при экспорте узлов проекта, имеют следующее назначение: *.cnv – файл текстового формата, содержащий служебную информацию; *.dbb – бинарный файл узла (загружается МРВ); *.dbx – бинарный файл узла (используется для конвертации узла проекта в узел МРВ); *.rtp – вспомогательный текстовый файл узла (загружается МРВ); addr.ind – описание параметров узлов (этот файл создается в папке каждого узла); Эмуляция_уровень.tmsd – программа, вызываемая каналами узла (код для отладки в ИС); *.res – файлы, содержащие исполняемый код для МРВ, нумерация файлов начинается с 0.

Запуск проекта на исполнение. Отладку проекта будем выполнять в *профайлере с поддержкой графических экранов*. Профайлер является исполнительным модулем одного узла проекта (при наличии нескольких сетевых плат или последовательных портов можно запустить несколько профайлеров для обмена данными по сети даже на одном компьютере) и соответствует реальному МРВ, но входит в состав ИС и не требует отдельного приобретения. За исключением ограничения в продолжительности работы (1 час для базовой версии и 2 часа для профессиональной), профайлер работает точно также, как и реальный исполнительный модуль. Во время работы профайлер исполняет файлы *.dbb и *.rtp.

Откройте профайлер кнопкой «*Запустить профайлер*»  на главной панели инструментов или выберите пункт «*Отладка*» в меню «*Файл*». Для того чтобы данная кнопка или пункт были активны, нужно обязательно предварительно выделить в навигаторе проекта узел, который будет открываться в профайлере, нажатием ЛКМ. Если профайлер не запускается таким образом, убедитесь, что в имени проекта отсутствуют пробелы.

В открывшемся профайлере еще раз нажмите ЛКМ кнопку «*Запуск/Останов*»  или сочетание клавиш *CTRL+R*.

Проверьте работоспособность проекта, включая (отключая) насос и открывая (закрывая) сливной клапан соответствующими кнопками. Откройте всплывающий экран «*Тренд*» и проследите динамику изменения уровня в емкости в зависимости от состояния исполнительных устройств.

Итоговый результат разработки проекта, запущенный на исполнение в профайлере, показан на рис. 1.4.

Для останова профайлера нажмите ЛКМ кнопку «Запуск/Останов»  или сочетание клавиш *CTRL+R*. После этого профайлер можно будет закрыть.

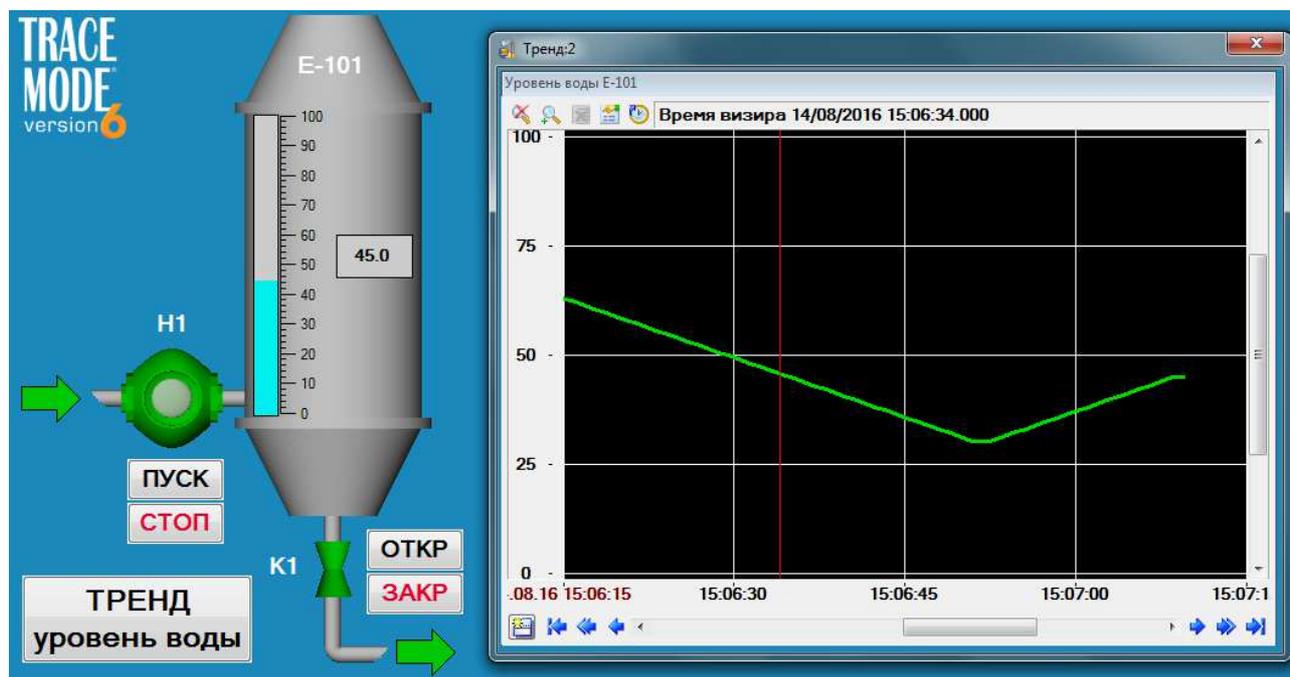


Рис. 1.4. Проект, запущенный на исполнение в профайлере

1.4. Содержание отчета

Отчет должен включать:

- 1) скриншоты экранов проекта, запущенных на исполнение в профайлере, с указанием типов использованных ГЭ и настраиваемых свойств этих ГЭ;
- 2) разработанный алгоритм программы эмулятора по ГОСТ 19.701-90;
- 3) листинг программы-эмулятора с текстовым пояснением ее работы и описанием синтаксиса использованных операторов;
- 4) скриншот базы каналов узла с описанием классов каналов и их связи с типами и типами данных аргументов, по которым они были построены;
- 5) вывод по проделанной работе.

1.5. Контрольные вопросы

2. Дайте определения следующим терминам и понятиям: SCADA, интегрированная среда разработки проекта, исполнительные модуль, узел проекта, канал.
3. Какие свойства канала известны вам? Поясните их.
4. Что такое класс канала. Какие классы были использованы в лабораторной работе?
5. Какие атрибуты называются аргументами?
6. Что определяет тип аргумента для программы? Какие типы аргументов вам известны?
7. Что определяет тип данных аргумента/атрибута? Какие типы данных аргументов/атрибутов вам известны?
8. Для чего служит режим эмуляции?
9. Для чего выполняется отладка программы?
10. Какие типы файлов формируются при экспорте узлов проекта?
11. Что такое профайлер и каковы его отличия от монитора реального времени?

Тема 2. ПОЗИЦИОННЫЕ РЕГУЛЯТОРЫ

ЛАБОРАТОРНАЯ РАБОТА №2

ДВУХПОЗИЦИОННОЕ РЕГУЛИРОВАНИЕ ПАРАМЕТРОВ

Цель работы: научиться программно реализовывать двухпозиционные регуляторы и выполнять их настройку с учетом требуемых показателей качества процесса.

Задача работы: разработать технологический экран управления нагревателем теплового агента и комбинированную программу эмулятор-регулятор температуры агента в емкости на языке Function Block Diagram.

Программное обеспечение: инструментальная система Trace Mode 6.

2.1. Порядок выполнения работы:

1. Изучите теоретические сведения, необходимые для выполнения лабораторной работы.
2. Выполните практические задания согласно методическим рекомендациям, описывающим содержание и порядок работы.
3. Составьте отчет по проделанной работе, включив в него результаты в соответствии с требованиями к содержанию отчета.
4. Подготовьте ответы на контрольные вопросы.

2.2. Теоретические сведения

Двухпозиционные регуляторы обеспечивают хорошее качество регулирования для инерционных объектов с малым запаздыванием. Двухпозиционные регуляторы используются для управления переключательными элементами – дискретными исполнительными устройствами: электромеханическими реле, контакторами, транзисторными ключами, симисторными или тиристорными устройствами, твердотельными реле и др.

В простейшем случае (без обратной связи) двухпозиционный регулятор работает как двухпозиционный переключатель.

Выход (физический или программный) данного регулятора может иметь только два состояния – «включен» (логическая единица) и «выключен» (логический ноль), что и обуславливает название.

К основным параметрам настройки регулятора относятся *заданное значение (уставка) SP* (англ. *Set point* – заданная точка); *гистерезис H* ; *алгоритм (направление) действия регулятора* – прямой или обратный (инверсный).

Назначение гистерезиса H – предотвращение «дребезга» управляющего выходного устройства (например, реле) вблизи задания SP от слишком частого включения и отключения. Гистерезис может принимать как положительные, так и отрицательные значения.

Эти типы гистерезиса применяются для учета инерционности объектов регулирования. Представление зоны гистерезиса описывается в руководстве по эксплуатации на соответствующий тип регулятора или программный блок (рис. 2.1).

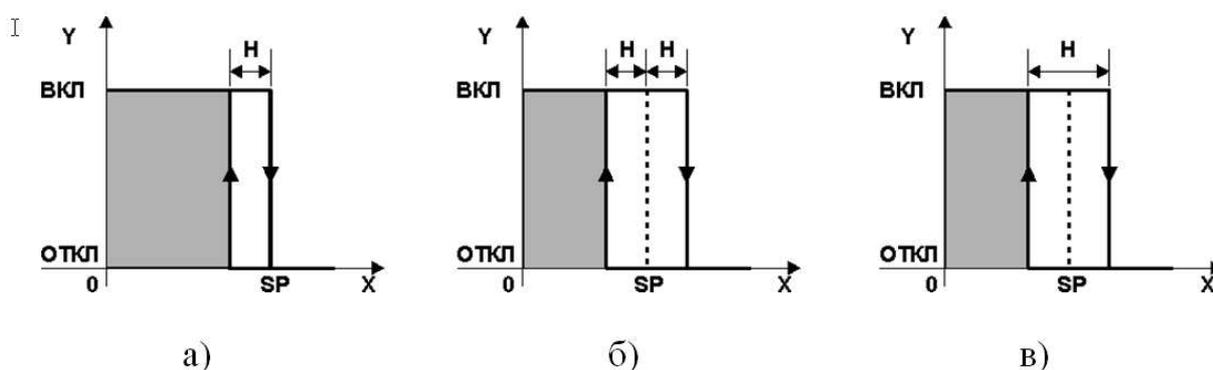


Рис. 2.1. Варианты представления зоны гистерезиса

Алгоритм двухпозиционных регуляторов определяется статической характеристикой: зависимостью выходного сигнала регулятора Y от входного значения технологического параметра с объекта управления X (рис. 2.1, рис. 2.2).

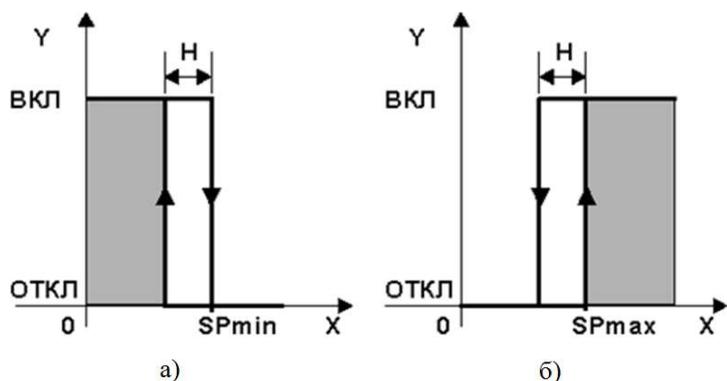


Рис. 2.2. Статические характеристики двухпозиционных регуляторов

Вид статической характеристики, представленный на рис. 2.2-а, называется *обратным регулятором* и обычно применяется в различных процессах управления нагревом – нагревательных приборах, печах, термошкафах, теплообменниках и т.п. При использовании в системах сигнализации

данная логика работы выходного устройства носит название «*меньше установленного значения*» или «*меньше минимума*».

Вид статической характеристики, представленный на рис. 2.2-б, называется *прямым регулятором* и обычно применяется в различных процессах управ-

ления охлаждением – в системах вентиляции, в холодильных установках и т.п. При использовании в системах сигнализации данная логика работы выходного устройства носит название «*больше установленного значения*» или «*больше максимума*».

Рассмотрим влияние параметров объекта управления и настроек регулятора на процесс регулирования. Определение параметров объекта управления производят чаще всего экспериментально по *переходной характеристике*, которую снимают, подавая на вход объекта ступенчатое воздействие (рис. 2.3). В случае наличия у объекта свойства *самовыравнивания* (свойство после выхода объекта из установившегося состояния вернуться к этому состоянию самостоятельно, без внешнего воздействия), большинство технологических объектов регулирования с явно выраженной преобладающей постоянной времени может быть описано передаточной функцией *инерционного звена первого порядка с запаздыванием* (2.1)

$$W(s) = \frac{K}{Ts + 1} \cdot e^{-\tau s}, \quad (2.1)$$

где K – коэффициент передачи, T – постоянная времени, τ – запаздывание объекта управления.

Определение динамических характеристик объектов по кривой разгона производится *методом касательной к точке перегиба* переходной характеристики (кривой разгона). В данном случае точка перегиба соответствует переходу кривой от режима ускорения к режиму замедления темпа нарастания выходного сигнала.

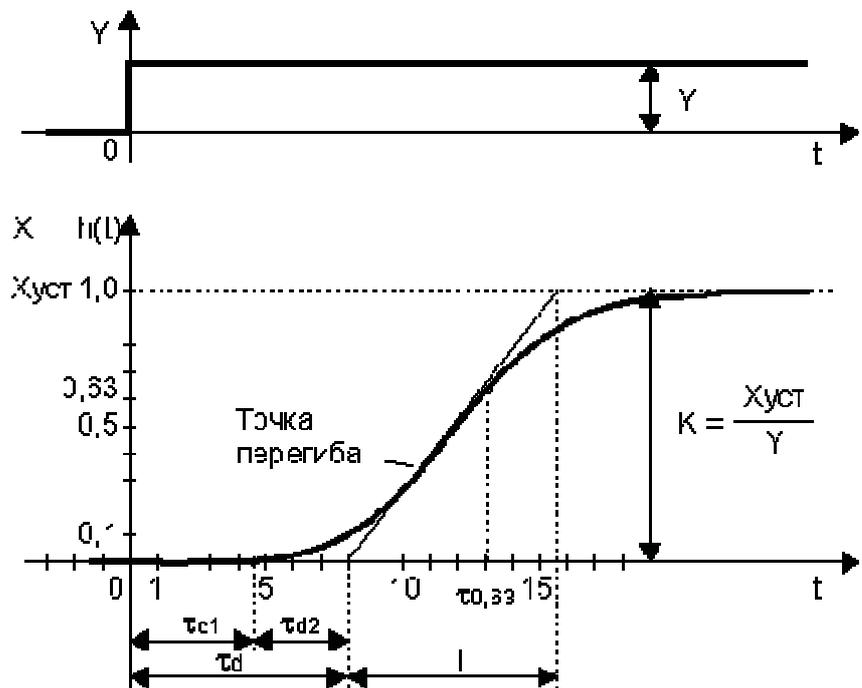


Рис. 2.3. Определение динамических характеристик объектов по кривой разгона

Перед началом обработки переходную характеристику (кривую разгона) рекомендуется пронормировать (привести диапазон изменения нормированной кривой к 0...1, где 1 соответствует установившемуся значению) и выде-

лить из ее начального участка величину времени **транспортного запаздывания** τ_{d1} (см. далее).

Динамический коэффициент передачи K_0 – величина, равная отношению амплитуды технологического параметра на выходе объекта управления $X_{уст}$ в установившемся режиме к амплитуде поданного входного воздействия Y (для нормированной переходной характеристики с отсчетом значения технологического параметра от нуля) (2.2), или это же иначе при ненулевых начальных условиях K_1 – отношение приращения выходного сигнала объекта $\Delta X_{уст}$ в окрестности рабочей точки к приращению входного ΔY (рис. 2.3).

$$K_0 = \frac{X_{уст}}{Y}; K = \frac{\Delta X_{уст}}{\Delta Y} \quad (2.2)$$

Запаздывание τ_d называется **полным запаздыванием** (рис 2.3). Оно состоит из двух составляющих: **транспортного (передаточного, также называемого «чистым»)** τ_{d1} , т.е. периода времени, когда изменение входного сигнала Y не приводит к изменению регулируемого параметра X , и **емкостного (переходного, динамического) запаздывания** τ_{d2} .

Транспортное запаздывание обусловлено наличием задержки в передаче сигналов (энергии) или веществ, например, удалением измерительного преобразователя от места, где осуществляется непосредственное воздействие на объект, т.е. установлено исполнительное устройство системы управления. Увеличение транспортного запаздывания отрицательно сказывается на качестве регулирования.

Емкостное запаздывание процесса регулирования – это запаздывание, зависящее от емкости объекта регулирования. Для медленно изменяющихся процессов емкостное запаздывание благоприятствует регулированию; это объясняется тем, что чем больше запаздывание, тем при нарушении равновесия медленнее изменяется регулируемая величина. Наибольшим емкостным запаздыванием, при прочих равных условиях, обладают тепловые объекты регулирования, наименьшим – объекты, в которых регулируется расход жидкости.

На практике емкостное запаздывание определяют как время достижения регулируемой величиной X приблизительно 5...10% по амплитуде от установившегося значения за вычетом времени чистого (транспортного) запаздывания τ_{d1} (2.3)

$$T = t_{X0.1} - \tau_{d1}. \quad (2.3)$$

Однако поскольку емкостное запаздывание индивидуально для каждого процесса, вызвавшего задержку, его учет неоправданно сложен. Ввиду этого

его заменяют моделью транспортного запаздывания, дающей большую погрешность на начальном участке, но являющейся более простой, что допустимо, поскольку данный участок кривой далек от участка, соответствующего номинальному режиму («рабочей точке»). Таким образом, полное запаздывание при определении параметров объекта управления считают идентичным чистому (транспортному) запаздыванию, а динамические (инерционные) свойства емкостного запаздывания учитывают в постоянной времени объекта T .

Постоянная времени T характеризует инерционность объекта и определяется по проекции касательной к точке перегиба кривой переходного процесса на ось времени как интервал между пересечением касательной уровня, соответствующего установившемуся значению, и пересечением ей самой оси времени.

Постоянная времени T представляет собой интервал времени, за который регулируемый параметр X достигает 63,2% от установившегося значения без учета транспортного запаздывания τ_{d1} . Поэтому на рис. 2.3 постоянная времени объекта достаточно точно может быть определена по (2.4):

$$T = t_{X0.63} - \tau_{d1} \quad (2.4)$$

Чем больше постоянная времени объекта, тем медленнее изменяется регулируемый параметр при воздействии на объект, что упрощает управление им.

Приближенно постоянную времени объекта управления можно определить также по формуле (2.5)

$$T \approx \frac{t_{X0.95} - \tau_{d1}}{3}, \quad (2.5)$$

где $t_{X0.95}$ – время переходного процесса, τ_{d1} – чистое (транспортное) запаздывание. Под временем переходного процесса в данном случае подразумевается интервал времени, за который регулируемый параметр X достигает 95% от установившегося значения. 100% от установившегося значения достигается за время $5T$.

Процесс двухпозиционного регулирования является автоколебательным – регулируемая величина как в переходном, так и в установившемся режиме периодически изменяется относительно заданного значения (рис. 2.4), т.е. регулируемая величина X подвержена незатухающим колебаниям.

Показателями автоколебательного режима являются амплитуда автоколебаний A_k и период автоколебаний T_k .

Частота и амплитуда колебаний зависят и определяются следующими величинами: временем транспортного запаздывания τ_d , постоянной времени объекта T (т.е. инерционностью объекта), максимальной скоростью R изменения параметра X ($R=X_{уст}/T$), величиной гистерезиса H переключающего элемента регулятора.

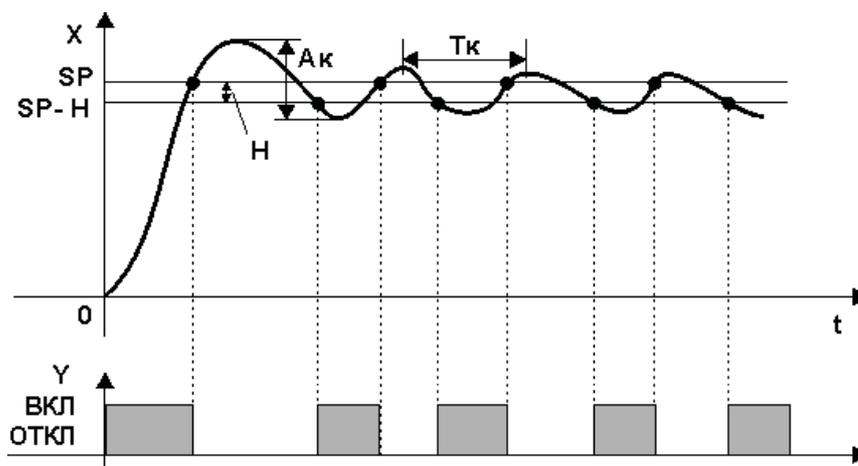


Рис. 2.4. Процесс регулирования с двухпозиционным законом (обратный регулятор)

Для объектов с большой инерционностью (большим значением постоянной времени объекта T) и с малым запаздыванием τ_d регулирование происходит с постоянными колебаниями до 5...15% от задания SP .

Чем больше гистерезис H , отношение τ_d/T , R - тем больше амплитуда колебаний A_k . Чем больше время запаздывания τ_d и постоянная времени объекта T , тем больше период колебаний T_k (рис. 2.3).

Точность регулирования технологического параметра зависит от величины гистерезиса H . Чем меньше гистерезис, тем точнее регулирование, но тем чаще включается исполнительное устройство и тем самым больше его износ или износ коммутационных элементов (например, реле). Уменьшая гистерезис, можно повысить качество регулирования до некоторого предела, определяемого параметрами объекта регулирования.

Отметим, что двухпозиционные регуляторы следует применять при выполнении условия $0 < \tau_d/T < 0,2$. При $\tau_d/T > 0,2$ данные регуляторы не обеспечивают требуемого качества регулирования.

2.3. Содержание и выполнение работы

Разработаем проект системы двухпозиционного регулирования температуры жидкого агента с ТЭНом в качестве исполнительного устройства. Регулирование температуры выполняется автоматически на основании уставки, заданной оператором (номинально 65°C), с гистерезисом $\pm 2^\circ\text{C}$. Начальная температура агента составляет 40°C . Максимальная температура нагрева составляет 100°C . Нагрев происходит за 2 минуты.

Задание требуемой температуры осуществляется оператором ползунком в диапазоне от 50 до 90°C. Введенное значение уставки отображается цифровым табло. Состояние ТЭНа отображается анимацией (включен/ выключен) типа сигнальная лампа.

На емкости располагается цифровой указатель температуры агента в градусах Цельсия. Кроме того, динамика изменения температуры агента, задание по температуре агента и состояния ТЭНа отображается на графике (тренде).

Проект включает в себя один шаблон экрана и одну комбинированную программу эмулятор-регулятор температуры агента в емкости на языке Function Block Diagram.

Создание проекта, узла, шаблона экрана. Откройте инструментальную систему Trace Mode 6 и создайте новый проект, нажав кнопку «Создать новый проект»  на панели инструментов «Главная». Сохраните проект, нажав кнопку «Сохранить текущий проект»  на панели инструментов «Главная». Выберите местоположение проекта и задайте его имя (см. Создание и сохранение проекта, ЛР№1).

Создайте узел проекта типа RTM, нажав ПКМ на слое «Система» в навигаторе проекта и в контекстном меню выбрав «Создать узел → RTM». (см. Создание узла проекта, ЛР№1). Создайте шаблон экрана, аналогично нажав ПКМ на созданном узле RTM_1 и выбрав в контекстном меню «Создать компонент → Экран» (см. Создание шаблона экрана оператора, ЛР№1).

Добавление ресурса «Графические элементы». Графические элементы представляют собой готовые наработки типовых элементов мнемосхем – емкостей, теплообменников, насосов и т.д. Их использование сокращает затраты времени на разработку экрана оператора.

Первоначально необходимо при закрытой инструментальной системе добавить библиотеку компонентов tmdevenv.tmul. Для этого ее нужно скопировать из папки «%TRACE_MODE%\LIB\» в папку «C:\Users\All Users\AdAstra\Trace Mode IDE 6\» при использовании ОС Windows 7 или в папку «C:\Documents and Settings\All Users\Application Data\AdAstra\Trace Mode IDE 6\» под ОС Windows XP и заменить расположенный там одноименный файл.

Помните, что библиотеки компонентов базовой и профессиональной версий ИС различаются и несовместимы. При необходимости эксплуатации на одном ПК ИС профессионального и базового форматов для обеспечения совместимости форматов надо заменять соответствующий файл пользовательской библиотеки компонентов.

Библиотека компонентов базовой версии содержит минимальное количество элементов, в то время как профессиональная – намного больше.

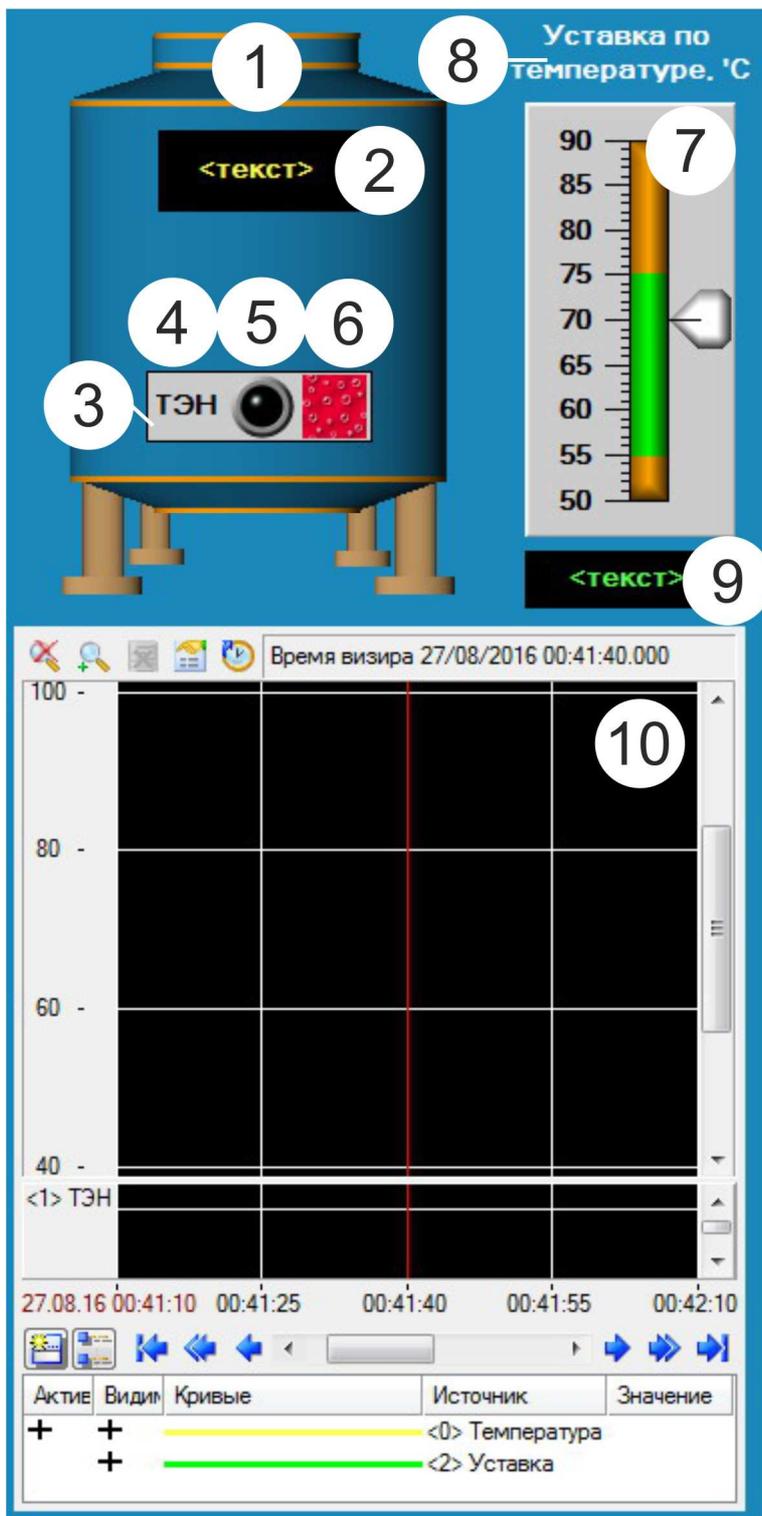


Рис. 2.5. Экран оператора

созданную группу «Графические_элементы». Теперь данный объект добавлен в ваш проект.

Снова откройте инструментальную систему и откройте сохраненный ранее проект данной лабораторной работы.

Создайте в проекте группу «Графические_элементы». Для этого переключитесь в слой «Ресурсы» в навигаторе проекта, нажмите ПКМ на нем и выберите в контекстном меню «Создать группу → Графические_элементы».

Последовательно разверните пункты слоя «Библиотека компонентов → Пользовательская → Library_1 → Object_1 → Resources → Tanks». В этом пункте библиотеки компонентов собраны шаблоны графических объектов емкостей. Соответственно, *Valves* – шаблоны арматуры (клапанов), *Motors* – электродвигатели, *Cooling&Heating* – теплообменники и чиллеры, *Pumps* – насосы, *Industrials* – колонны.

Выберите подходящий шаблон графического объекта (ГО) из группы «Tanks» и перетащите его, зажав ЛКМ,

Откройте шаблон экрана на редактирование в РПД и произведите размещение и настройку свойств ГЭ экрана оператора таким образом, чтобы он выглядел, как показано на рисунке 2.5.

Использование ресурса ГО «Емкость». Нажмите ЛКМ на ГО «Графический_объект_1» и, не отпуская ее, перетащите ГО на открытый на редактирование в РПД шаблон экрана. Измените размер и, при необходимости, положение ГО (см. Создание ГЭ «Емкость» ЛРН№1) «Емкость» (элемент 1, рис. 2.5).

Создание индикатора температуры агента. ГЭ «Текст». Разместим цифровой индикатор, отображающий температуру агента в емкости с помощью ГЭ «Текст» (см. Создание динамических надписей. ГЭ «Текст» ЛРН№1). Разместите текст на емкости, как показано на рисунке 2.5 (элемент 2).

Для того, чтобы упорядочить расположение элементов можно использовать команды панели «Топология» (включите ее, дав команду «Вид → Топология экрана»). Например, чтобы расположить цифровой указатель по середине емкости, нужно нажать ЛКМ на емкости (объект, относительно которого осуществляется позиционирование), затем на ГЭ «Текст» (объект, который позиционируется) и затем кнопку  «Центрировать в группе горизонтально».

Установите жирное начертание шрифта текста, 10 кегль. Измените цвет текста на желтый, цвет заливки – на черный. Создайте и выполните привязку к созданному аргументу «Температура» с типом *IN* и типом данных *REAL*. Установите формат отображения численного значения «*Float*» (численный) «%.lf».

Добавление анимационного клипа в проект и размещение на экране. Видеоклип. Анимация также относится к ресурсам проекта и используется для повышения наглядности процесса и упрощения его восприятия.

Нажмите ПКМ на слое «Ресурсы» и в контекстном меню выберите «Создать группу → Анимация». Затем также нажмите ПКМ на группе «Анимация» и выберите в контекстном меню «Создать компонент → Библиотека_видеоклипов».

Двойным нажатием ЛКМ откройте библиотеку видеоклипов на редактирование. Затем нажмите ПКМ в рабочем поле библиотеки и в контекстном меню выберите пункт «Импортировать» . Откроется окно проводника, где необходимо указать путь к папке «C:\Program Files (x86)\AdAstra Research Group\Trace Mode IDE 6 Base\Lib\Animation\» (на примере Windows 7) и затем сначала выбрать папку «\Lamps» и в ней файл lamp_green_top, а затем папку «\Boiling_fluid» и в ней файл fluid_boiling_red. Оба анимационных клипа будут добавлены в библиотеку.

Чтобы использовать добавленные в библиотеку видеоклипы на экране, переключитесь на вкладку *РПД* для разрабатываемого шаблона экрана. На панели ГЭ нажмите ЛКМ на треугольнике в правом нижнем углу кнопки «Ресурсы»  и выберите ресурсную библиотеку видеоклипов .

Нажмите ЛКМ на каждом видеоклипе в библиотеке и, удерживая ее, перетащите анимацию на экран, разместив пока в произвольном месте экрана.

Создание анимации работы ТЭНа. Разместите на емкости ГЭ «Прямоугольник» (элемент 3) (см. Создание смотрового окна. ГЭ «Прямоугольник» ЛРН№1). Отцентрируйте его относительно емкости, как было описано выше. Над прямоугольником разместите ГЭ «Текст» (элемент 4) и анимационные клипы «Зеленая лампа» (элемент 5) и «Красная кипящая жидкость» (элемент 6). Для облегчения позиционирования используйте кнопки «Выровнять влево» , «Выровнять вправо» , «Выровнять вверх» , «Выровнять вниз» , «Центрировать в группе вертикально» , «Выровнять по высоте» , «Выровнять по ширине» , «Выровнять по размеру»  панели «Топология экрана». В случае, если один из элементов закрывает другой, используйте кнопки «Переместить вниз»  и «Переместить вверх»  для изменения порядка элементов.

В свойстве «Текст» элемента текст введите значение «ТЭН» (см. Создание статических надписей. ГЭ «Текст» ЛРН№1).

В свойствах видеоклипа «Зеленая лампа» создайте и выполните привязку (свойство «Привязка») к созданному аргументу «ТЭН» с типом *IN* и типом данных *USINT*. Остальные свойства оставьте без изменений. При значении аргумента «ТЭН» отличном от нуля, анимационный клип будет запущен.

Аналогично выполните привязку видеоклипа «Красная кипящая жидкость» к этому же аргументу. Кроме того, в свойстве «Показывать при остановке» выберите из списка значение «False», что позволит не только не воспроизводить, но и не показывать данную анимацию вообще при неактивном ТЭНе, т.е. при ТЭН=0.

Появляющиеся при определенных обстоятельствах элементы экрана не только способствуют акцентированию внимания оператора, но и позволяют не перегружать экран не активными элементами.

Проверьте работоспособность элементов в режиме «Эмуляция» (см. Проверка правильности задания свойств ГЭ (эмуляция) ЛРН№1), вводя значения 0 или 1 в аргумент «ТЭН» – в активном состоянии лампа должна начать мигать, а жидкость – стать видимой и «кипеть», и значения температуры в одноименный аргумент – они должны отображаться цифровым индикатором на емкости.

Создание задатчика температуры. ГЭ «Ползунок». Разместите справа от емкости ГЭ «Ползунок» (элемент 7) (см. Создание шкалы. ГЭ «Ползунок» ЛРН№1). Развернув свойство «*Отображаемая величина*», в пункте «*Привязка*» создайте и выберите привязку к созданному аргументу «*Уставка*» с типом *IN* и с типом данных *REAL*. Аналогично выберите привязку к аргументу «*Уставка*» в свойстве «*Задаваемая величина*».

Раскройте свойство «*Шкала*». Измените цвет текста – выберите черный, нажав на раскрывающийся список пункта «*Цвет текста*» данного свойства.

Раскройте пункт «*Уровень 2*» свойства «*Шкала*» и выберите значение «*False*» в подпункте «*Использовать*». Разверните пункт «*Уровень 1*» и введите значение «8» в подпункт «*Число делений*», нажав после *Enter*. В подпункт «*Десятичные знаки*» введите значение «0», также нажав после *Enter*. Измените параметры шрифта в одноименном свойстве, нажав ЛКМ на нем и задав в списке кегль «10» и жирное начертание.

Разверните свойство «*Ползунок*» и выберите в пункте «*Тип*» из списка значение «*Домик*». В свойстве «*3D-эффекты*» выберите из списка значение «*True*».

Раскройте свойство «*Полоса*» и в пункте «*Ширина*» введите значение «20». В свойстве «*Верхний предел шкалы*» введите значение «90», а в свойстве «*Нижний предел шкалы*» – «50». В свойство «*HW*» – «75», а в свойство «*LW*» – «55». В свойстве «*Цвет >HL, <LL*» выберите из палитры оранжевый (при этом может потребоваться сменить тип палитры с «*Standart*» на «*Trace Mode 6*», выбрав из списка).

Создание подписи к задатчику температуры. ГЭ «Текст». Разместите над ГЭ «Ползунок» ГЭ «Текст» (элемент 8). В свойстве «*Текст*» ГЭ «Текст» введите значение «*Уставка по*», нажмите *CTRL+Enter*, и далее введите «*температуре, 'C*» (см. Создание статических надписей. ГЭ «Текст» ЛРН№1).

Создание цифрового индикатора уставки. ГЭ «Текст». Разместим цифровой индикатор, отображающий задание (уставку) по температуре агента в емкости с помощью ГЭ «Текст» (см. Создание динамических надписей. ГЭ «Текст» ЛРН№1). Разместите ГЭ «Текст» под ГЭ «Ползунок», как показано на рисунке 2.5 (элемент 9).

Установите жирное начертание шрифта текста, 10 кегль. Измените цвет текста на зеленый, цвет заливки – на черный. Выполните привязку к созданному ранее аргументу «*Уставка*». Установите формат отображения численного значения «*Float*» (численный) «*%.1f*».

Создание графиков по уставке и температуре агента, состоянию ТЭНа. ГЭ «Тренд». Под емкостью и ГЭ «Ползунок» разместите ГЭ «Тренд» (элемент 10) (См. Создание графика технологического параметра. ГЭ «Тренд» ЛР№1).

Перейдите на вкладку «Кривые» . Нажмите ПКМ на свойстве «Кривые» и в контекстном меню выберите пункт «Кривая». В свойстве «Привязка» выберите аргумент «Температура». В свойстве «Цвет» выберите желтый. В свойстве «Толщина линии» введите «3». В свойстве «Макс. значение» оставьте значение «100». В свойстве «Мин. значение» введите значение «40».

Аналогично создайте еще одну кривую с такой же толщиной линии и пределами изменения значений. В свойстве «Привязка» выберите аргумент «Уставка». В свойстве «Цвет» выберите зеленый.

В свойствах третьей созданной таким же образом кривой в пункте «Интерпретировать как» выберите «Статус» и выполните привязку к аргументу «ТЭН».

Изменение размеров окна. Для этого нажмите дважды ЛКМ на свободном пространстве экрана. Выберите пункт «Произвольно» и введите размеры экрана в пикселях вручную либо используйте скроллинг колесом мыши так, чтобы размер экрана соответствовал размещенным на нем ГЭ.

Разработка графической части проекта завершена. Приступим к разработке прикладного программного обеспечения проекта.

Разработка программы эмулятора-регулятора температуры агента в емкости. Нажмите ПКМ на созданном узле RTM_1 и выберите в контекстном меню «Создать компонент → Программа». Двойным нажатием ЛКМ на канале вызова программы, откройте ее на редактирование в РШП (См. Разработка программы-эмулятора изменения уровня в емкости ЛР№1).

Напишем комбинированную программу регулирования температуры агента в емкости, совмещенную с эмулятором объекта управления, на языке Function Block Diagram.

Нажмите ЛКМ на пункте «Аргументы» РШП и создайте в табличном редакторе аргументов программы три аргумента аналогично тому, как это делалось при разработке программы-эмулятора изменения уровня в емкости в ЛР№1, с параметрами *Уставка IN REAL, ТЭН OUT USINT Температура OUT REAL*.

Нажмите ЛКМ на заголовке программы. В появившемся окне выбора языка программирования выберите язык *FBD* – графический язык, где программа формируется последовательностью функциональных блоков, исполняемых слева направо сверху вниз. Каждый функциональный блок также может представ-

лять собой подпрограмму или функцию. Написание программы представляет собой использование библиотечных блоков, сгруппированных по разделам, и выполняющих определенные функции. При необходимости программистом могут быть разработаны собственные блоки. FBD-программа может выступать в роли основной программы, функции и функции-блока.

Нажмите в окне выбора языка кнопку «Принять».

В РШП нажмите на панели кнопку «Показать/скрыть палитру FBD блоков» . Откроется библиотека функциональных блоков.

Откройте в библиотеке вкладку «Управление» и, нажав ЛКМ на блоке «Гистерезис» и не отпуская ее, перетащите блок из библиотеки в рабочее поле РШП. Двойное нажатие ЛКМ на блоке в библиотеке открывает справку. Прочтите справку по блоку «Гистерезис» *HSTR*.

Аналогично перетащите блок «Модель объекта» *OBJ* из раздела библиотеки «Регулирование» и блок «Сложение» $X+Y$ из раздела «Арифметические». Изучите справку по данным блокам.

Перемещение блоков в РШП выполняется также выделением блока однократным нажатием ЛКМ (он меняет цвет) с последующим протягиванием и отпусканьем ЛКМ в нужном новом месте расположения блока. Также выделение можно выполнить рамкой.

В верхней части блока выводится обозначение функции, выполняемой блоком (например, сложение $X+Y$). Слева располагаются входы блока, к которым могут быть привязаны (присвоены значения) аргументы с типом *IN* (*IN/OUT*) или константы. Вход без имени (в Trace Mode носит название RUN) управляет выполнением блока. Блок выполняется, если $RUN=0$ (не присвоенное значение также равно нулю).

Отрезки, примыкающие к блоку справа, обозначают выходы блока (возвращаемые функцией значения). К ним могут быть привязаны аргументы с типом *OUT*.

Кроме входов и выходов, некоторые FBD-блоки имеют внутренние переменные, недоступные пользователю. Переменные FBD-блока (входы/выходы и внутренние) являются глобальными, т.е. сохраняют свое значение между вызовами программы, в том числе при $RUN=1$.

В нижней части блока выводится его номер и, после двоеточия, номер следующего выполняемого блока. Номера блоков задаются последовательно при их размещении в рабочем поле редактора; номера следующих выполняемых блоков определяются автоматически при соединении входов и выходов блоков (образовании диаграммы). На блоке, который выполняется первым в програм-

ме, после его номера отображается символ В; на блоке, который выполняется последним, – символ Е.

Таким образом, разрабатываемая программа состоит из трех функциональных блоков.

Блок *HSTR* «Гистерезис» представляет собой двухпозиционный регулятор с симметричным гистерезисом (рис.2.1 б) и логикой «прямого регулятора» (рис. 2.2 б). Функцией данного блока является формирование единичного выходного сигнала (*Q*), т.е. включение ТЭНа, при увеличении входного значения (*INP*), поступившего от измерительного преобразователя температуры агента в емкости, больше уставки (*PV*) с учетом значения гистерезиса (*DLT*). Т.е. *if INP > PV + DLT then Q := 1*, а *if INP < PV - DLT then Q := 0*.

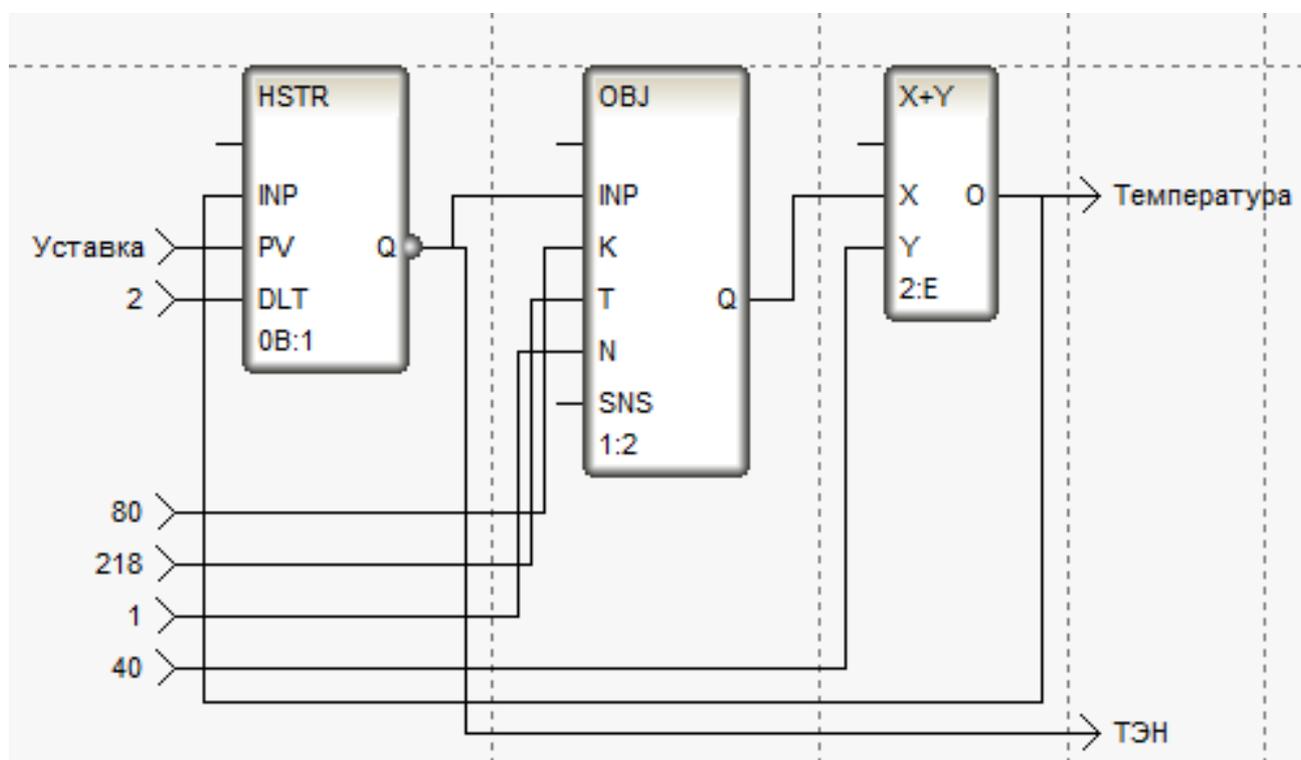


Рис. 2.6. Комбинированная программа эмулятор-регулятор температуры агента в емкости на языке FBD

Присвойте соответствующим входам аргументы (рис. 2.6): входу (*PV*) – аргумент «Уставка», входу (*DLT*) – константу «2» (значение гистерезиса по условию), входу (*INP*) – обратную связь с «измерительного преобразователя», т.е. объекта управления, представленного блоками «Модель объекта» *OBJ* и блока сложения *X+Y* (см. далее).

Для присвоения входу соответствующего аргумента дважды нажмите ЛКМ на нем и выберите нужный аргумент из списка (для входов отображаются только аргументы с типом *IN*, выходов – *OUT*).

Для соединения входов и выходов двух блоков, необходимо привести указатель на выход блока и, когда он примет форму , нажать ЛКМ и, не отпуская ее, протянуть указатель ко входу нужного блока и отпустить ЛКМ. Будет создана связь в виде линии. При перемещении блоков связи сохраняются.

Так как для управления нагревом необходимо использовать логику «обратного регулятора», инвертируем выход регулятора (Q). Для этого нажмите ПКМ на выходе блока (Q) и в контекстном меню выберите пункт «Инвертировать».

Выход (Q) блока *HSTR* соедините со входом (INP) блока *OBJ*. К выходу (Q) блока *HSTR* привяжите аргумент «ТЭН», чтобы показывать состояние исполнительного устройства на экране.

Блок *OBJ* «Модель объекта» моделирует объект управления для отладки алгоритмов регулирования при подготовке проектов. Он представляет собой комбинацию апериодического (инерционного) звена первого порядка и звена запаздывания, т.е. передаточная функция блока имеет вид (2.1), где k и T – соответственно коэффициент передачи и постоянная времени инерционного звена первого порядка, а $N=\tau$ – время полного запаздывания.

Входным по отношению к моделируемому объекту является вход (INP). Входы (K), (T) и (N) используются для задания соответственно коэффициента передачи, постоянной времени и времени запаздывания. Последние два параметра задаются в тактах пересчета (по умолчанию длительность такта пересчета 550 мс), максимальное значение времени запаздывания – 4.

Кроме того, на выходной сигнал блока можно наложить помеху в виде случайной составляющей, синусоидального сигнала или случайных бросков. Здесь же можно задать случайное колебание динамических характеристик объекта. Для этого необходимо подать соответствующее значение на вход (SNS). Значение 1 отдельных битов этого входа включает следующие помехи: 1 бит – добавление к выходному сигналу случайной величины в диапазоне от 0 до 1%; 2 бит – формирование пика величиной 25% от значения выхода с вероятностью 0,01; 3 бит – добавление к выходу синусоидального сигнала с амплитудой 2% от значения выхода; 5 бит – случайное увеличение коэффициента усиления в диапазоне от 0 до 2%; 6 бит – случайное увеличение постоянной времени в диапазоне от 0 до 2%; 7 бит – случайное изменение запаздывания на 1.

Вход (SNS) блока *OBJ* оставьте пустым. На вход (K) блока *OBJ* подайте константу «80», что означает увеличение температуры на 80°C при активном выходе регулятора, т.е. включенном ТЭНе. На вход (T) блока *OBJ* подайте константу «218», т.к. 120 секунд/0,55 секунды (длительность цикла пересчета) \approx

218 циклов. На вход (N) блока OBJ подайте константу «1» (1 цикл запаздывания).

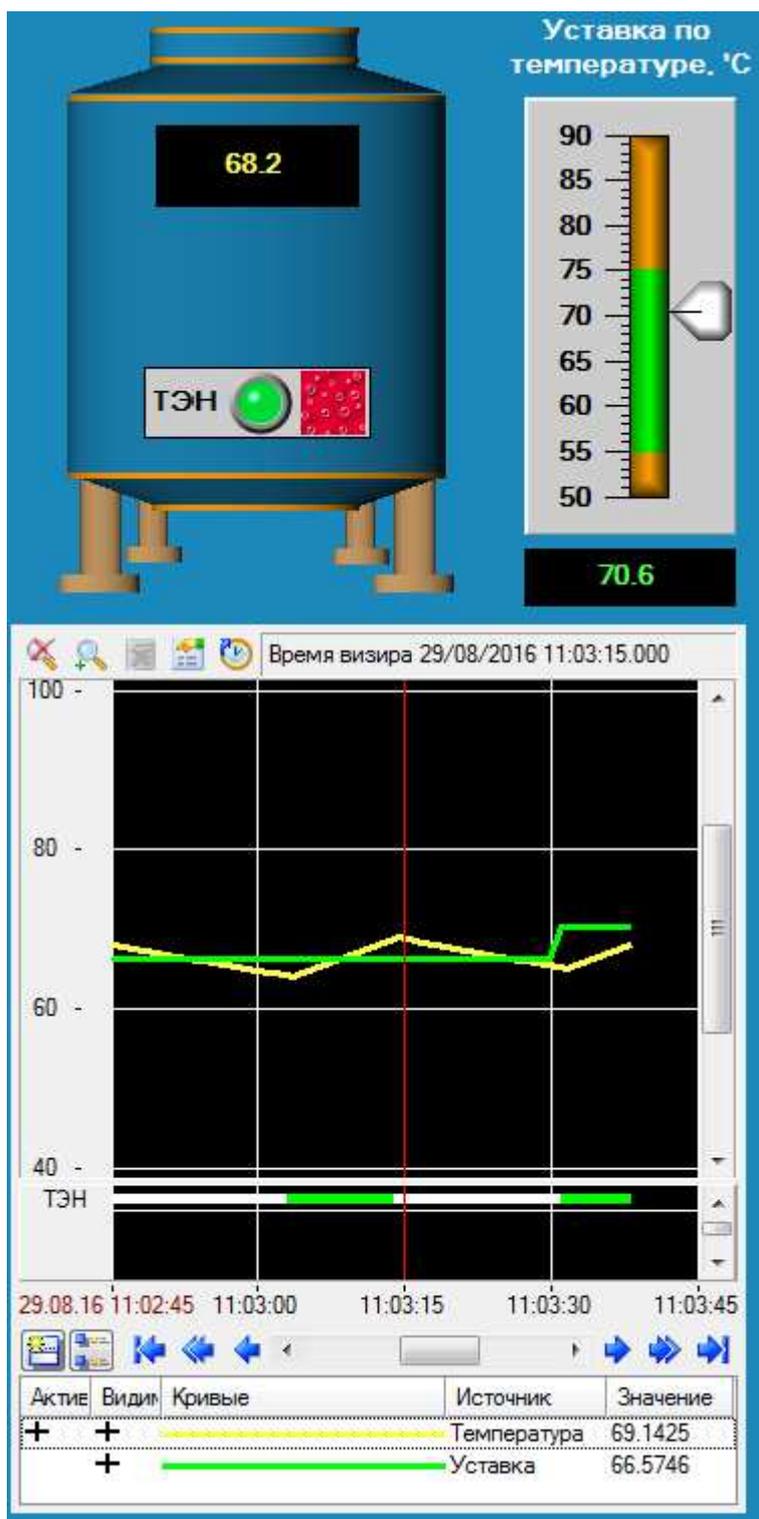


Рис. 2.7. Проект, запущенный на исполнение в профайлере

Введите в поле аргумент «Уставка» значение «65» (уставка по условию). Наблюдайте как выход регулятора и аргумент «ТЭН» принимают единичное

Выход (Q) блока OBJ соедините со входом (X) блока $X+Y$.

Блок «Сложение» $X+Y$ необходим для учета (добавления) начальной температуры, от которой нагревается агент. На вход Y блока $X+Y$ подайте константу «40» (начальная температура агента). К выходу (O) блока $X+Y$ привяжите аргумент «Температура», чтобы показывать значение температуры агента на экране.

Итоговый вид комбинированной программы эмулятор-регулятора температуры агента в емкости на языке FBD представлен выше на рисунке 2.6.

Компиляция и отладка. Выполните компиляцию программы клавишей $F7$ или нажмите кнопку «Компиляция»  на панели инструментов отладчика. Откройте окно переменных, нажав кнопку «Переменные»  на панели инструментов отладчика, и запустите программу на циклическое выполнение, нажав кнопку «Старт»  здесь же или воспользуйтесь клавишей $F5$ (см. Компиляция и отладка ЛРН№1).

значение и как при этом увеличивается температура (аргумент «*Температура*»). Заметьте при каких значениях температуры происходит смена состояния ТЭНа. Сделайте вывод о назначении гистерезиса регулятора. Измените значение уставки.

Связывание компонентов при помощи каналов. Воспользуйтесь процедурой автопостроения каналов по аргументам шаблонов программы или экрана, чтобы связать их между собой (см. Создание базы каналов ЛРН№1).

Сохранение проекта и запуск на исполнение в профайлере. Нажмите ЛКМ кнопку «*Сохранить для MPB*»  на главной панели инструментов (см. Сохранение и подготовка проекта к запуску ЛРН№1). Откройте профайлер кнопкой «*Запустить профайлер*»  на главной панели инструментов. В открывшемся профайлере еще раз нажмите ЛКМ кнопку «*Запуск/Останов*» .

Проверьте работоспособность проекта, изменяя значение уставки ползунком. Введенное значение должно отображаться на индикаторе уставки. Температура агента должна выходить на заданное значение и периодически изменяться относительно его на величину гистерезиса. Рост температуры должен происходить при включенном ТЭНе, состояние которого должно проследиваться по анимации на экране. Значение температуры агента, а также уставки по данному параметру и состояние ТЭНа должны отображаться на тренде.

Итоговый результат разработки проекта, запущенный на исполнение в профайлере, показан на рисунке 2.7.

Для останова профайлера нажмите ЛКМ кнопку «*Запуск/Останов*» . После этого закройте профайлер.

2.4. Содержание отчета

Отчет должен включать:

- 1) скриншот экрана проекта, запущенного на исполнение в профайлере, с указанием типов использованных ГЭ и настраиваемых свойств ГЭ, использованных в данной работе;
- 2) разработанный алгоритм программы эмулятора-регулятора температуры агента по ГОСТ 19.701-90;
- 3) программу эмулятор-регулятор температуры агента на языке FBD с описанием использованных блоков и алгоритма их работы, а также текстовым пояснением работы всей программы;
- 4) *самостоятельное задание.* Программу регулятора температуры агента на языке ST (без эмуляции объекта);

5) вывод по проделанной работе.

2.5. Контрольные вопросы

1. Каково назначение двухпозиционных регуляторов?
2. Для управления какими исполнительными устройствами предназначены двухпозиционные регуляторы?
3. Каковы основные настроечные параметры двухпозиционных регуляторов?
4. Что такое гистерезис и для чего он нужен?
5. Какие статические характеристики двухпозиционных регуляторов вам известны?
6. Каковы параметры объекта управления, описываемого передаточной функцией инерционного звена первого порядка с запаздыванием?
7. Что такое коэффициент передачи объекта управления и как он рассчитывается?
8. Каков физический смысл постоянной времени объекта и как определить ее графически и аналитически?
9. Из каких составляющих складывается полное запаздывание и как оно определяется графически?
10. Поясните алгоритм работы функционального блока «Гистерезис».

ЛАБОРАТОРНАЯ РАБОТА №3

ТРЕХПОЗИЦИОННОЕ РЕГУЛИРОВАНИЕ ПАРАМЕТРОВ. ПЕРЕКЛЮЧЕНИЕ РЕЖИМА УПРАВЛЕНИЯ

Цель работы: научиться программно реализовывать трехпозиционные регуляторы и выполнять их настройку, а также реализовывать переключение режимов управления «автоматический-ручной».

Задача работы: модифицировать технологический экран (мнемосхему) участка емкости E-101, разработать программу-регулятор уровня в емкости E-101 с выбором режима управления «автоматический-ручной».

Программное обеспечение: инструментальная система Trace Mode 6.

3.1. Порядок выполнения работы:

1. Изучите теоретические сведения, необходимые для выполнения лабораторной работы.
2. Выполните практические задания согласно методическим рекомендациям, описывающим содержание и порядок работы.
3. Составьте отчет по проделанной работе, включив в него результаты в соответствии с требованиями к содержанию отчета.
4. Подготовьте ответы на контрольные вопросы.

3.2. Теоретические сведения

Трехпозиционные регуляторы, как и двухпозиционные, обеспечивают хорошее качество регулирования для инерционных объектов с малым запаздыванием и применяются для управления теми же исполнительными устройствами в силу схожести принципа работы. По сути, трехпозиционный регулятор может быть получен объединением двух двухпозиционных регуляторов с противоположными статическими характеристиками.

Трехпозиционные регуляторы используются для систем управления уровнем различных веществ, для систем управления нагреванием-охлаждением различных тепловых процессов, холодильных установок, регулирования микроклимата подогревателем и вентилятором, для систем распределения и смешивания различных потоков веществ с помощью трехходовых клапанов, кранов, смесителей, реверсивных электродвигателей, сервоприводов и др.

Данный тип регуляторов имеет два выхода, каждый из которых также может иметь только два состояния – «включен» (логическая единица) и «выключен» (логическая единица).

чен» (логический ноль). Пара выходов используется для управления либо двумя исполнительными устройствами с противоположным воздействием на регулируемый технологический параметр (обязательно на один и тот же), например, ТЭН (нагреватель) и вентилятор (охладитель), либо одним реверсивным исполнительным устройством, которое может принимать три состояния (условно «прямо», «стоп», «реверс»), например, реверсивный электродвигатель. Таким образом, комбинация выходов включает три варианта: 1 и 0; 0 и 1; 0 и 0, что определило название данного типа регуляторов. Комбинация 1 и 1 не осуществима, что достигается программной или аппаратной защитой.

К основным параметрам настройки регулятора относятся *заданное значение (уставка) SP* (англ. *Set point* – заданная точка); *зона нечувствительности* («мертвая зона») *DB*; *гистерезис H*. Ряд настроечных параметров относится к видам представления «мертвой зоны» и гистерезиса, а также изменению направления действия (см. далее).

Термин «зона нечувствительности» или «мертвая зона» относится к величине изменений регулируемой переменной, которые могут происходить в процессе, не вызывая действия регулятора. Кроме того, зона нечувствительности препятствует одновременному включению обоих выходов, что, например, в случае использования реверсивного электродвигателя, приведет к его выходу из строя.

У трехпозиционного регулятора (рис. 3.1), по сути, не одна, а две уставки, которые отстоят друг от друга на величину зоны нечувствительности *DB*. Максимальное «нормальное» значение уставки получают прибавлением к номинальной уставке, задаваемой оператором, половины величины зоны нечувствительности, т.е. $SP_{max} = SP + DB/2$. Соответственно, минимальное «нормальное» значение получают вычитанием из номинальной уставки, задаваемой оператором, половины величины зоны нечувствительности, т.е. $SP_{min} = SP - DB/2$. Таким образом, нахождение регулируемой величины в пре-

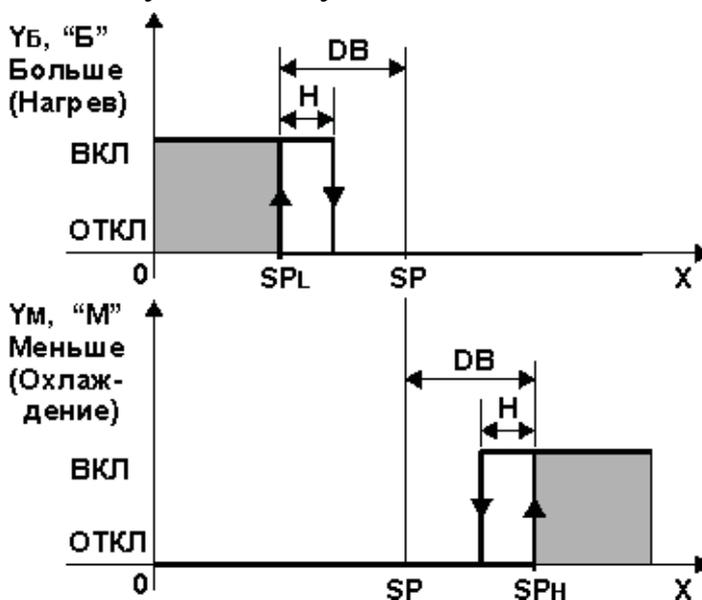


Рис. 3.1. Статическая характеристика обратного трехпозиционного регулятора с зоной регулирования без перекрытия

делах зоны нечувствительности (нормы), т.е. между вычисленными SP_{max} и SP_{min} , не требует от регулятора каких либо действий и оба его выхода не активны. Вполне очевидно, что увеличение зоны нечувствительности снижает точность регулирования, но позволяет избежать выработки регулятором управляющих воздействий при незначительных колебаниях регулируемого параметра.

Следует отметить, что наряду с представлением зоны нечувствительности с центром в заданной точке (уставке), т.н. «полная зона нечувствительности», когда относительно уставки откладывается половина величины DB как описано выше, существует представление в виде «половинного значения зоны нечувствительности», когда относительно уставки откладывается само заданное значение DB . О том, какой конкретно вариант представления «мертвой зоны» используется в том или ином случае, можно узнать просмотрев статическую характеристику регулятора.

Для того чтобы избежать путаницы с представлением зоны нечувствительности и вычисленными в связи с этим значениями максимума и минимума уставки, некоторые разработчики и производители вместо пары параметров «уставка (номинальная) – зона нечувствительности» используют пару абсолютных значений уставки «максимальное значение параметра» и «минимальное значение параметра».

Точно также, как и при использовании двухпозиционного регулятора, гистерезис служит для предотвращения слишком частой смены состояний выходных устройств («дребезга») вблизи граничных значений (максимума и минимума уставки). Гистерезис (в некоторых типах регуляторов) может принимать как положительные, так и отрицательные значения (рис. 2.1 ЛР№2).

Состояние выходов регулятора при нахождении регулируемого параметра за пределами зоны нечувствительности определяется его статической характеристикой – зависимостью выходных сигналов Y_M (меньше) и Y_B (больше) от входного значения регулируемого параметра X . Наиболее распространен трехпозиционный регулятор с **обратной статической характеристикой и зоной регулирования без перекрытия** (рис. 3.1).

Выход «больше» $Y_B=1$ активен при $X < SP - DB$, где X – значение регулируемой величины, SP – величина уставки, DB – значение ширины зоны нечувствительности. Выход «больше» $Y_B=0$ не активен при $X > SP - DB + H$, где H – значение гистерезиса. Выход $Y_M=1$ активен при $X > SP + DB$. Выход $Y_M=0$ не активен при $X < SP + DB - H$.

Процесс трехпозиционного регулирования также, как и при двухпозиционном регулировании, является автоколебательным – регулируемая величина

$PV(X)$ подвержена незатухающим колебаниям (рис. 3.2). Для объектов с большой инерционностью (большим значением постоянной времени объекта T) и с малым запаздыванием τ_d регулирование происходит с постоянными колебаниями до 3...10% от задания SP .

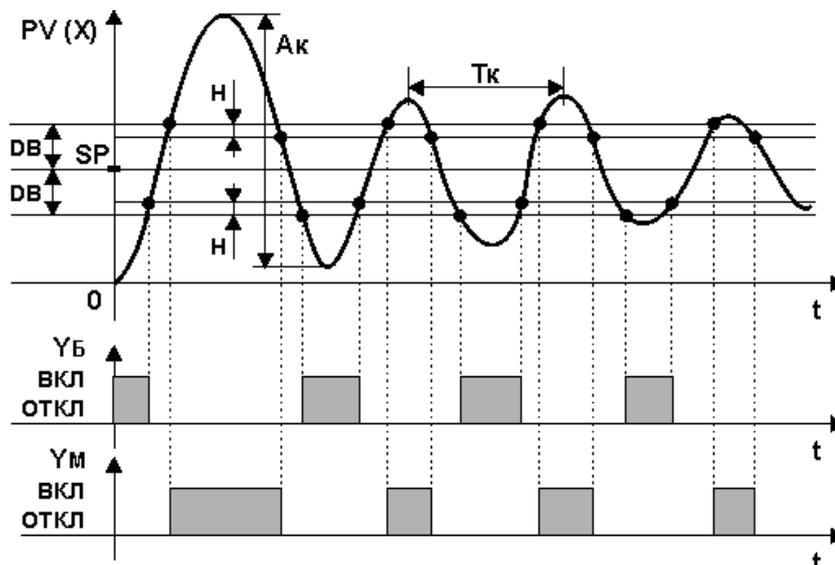


Рис. 3.2. Процесс регулирования с обратным трехпозиционным законом

Показателями автоколебательного режима являются амплитуда автоколебаний A_k и период автоколебаний T_k (рис. 3.2). Частота и амплитуда колебаний зависят и определяются следующими величинами: временем транспортно запаздывания τ_d , постоянной времени объекта T (определяется инерционностью объекта), максимальной скоростью R изменения параметра X ($R=X_{ycm}/T$), величины зоны нечувствительности DB регулятора и гистерезиса H переключательного элемента.

Чем больше гистерезис H , отношение τ_d/T , скорость изменения параметра R , зона нечувствительности DB , тем больше амплитуда колебаний A_k . Чем больше время запаздывания τ_d , постоянная времени объекта T , тем больше период колебаний T_k (рис. 3.2).

Отметим, что как и двухпозиционные, трехпозиционные регуляторы следует применять при выполнении условия $0 < \tau_d/T < 0,2$. При $\tau_d/T > 0,2$ данные регуляторы не обеспечивают требуемого качества регулирования.

3.3. Содержание и выполнение работы

Модифицируем проект, разработанный в лабораторной работе №1, добавив функцию автоматического трехпозиционного регулирования уровня с выбором способа управления – автоматически или в реализованном в первой работе ручном режиме. В качестве основного режима будет использоваться автоматическое управление. Уставка составляет 70% и может изменяться оператором, зона нечувствительности – 10%, гистерезис переключения – 2%.

Открытие проекта и шаблона экрана. Запустите инструментальную систему Trace Mode 6. Откройте проект, созданный в ходе выполнения лабора-

торной работы №1, используя пункт меню «Файл → Открыть» либо нажмите кнопку «Открыть существующий проект»  на панели инструментов «Главная» и выберите расположение файла проекта.

Сохраните проект под другим именем, используя пункт меню «Файл → Сохранить как» либо сочетание клавиш *CTRL+SHIFT+S*. Выберите местоположение проекта и задайте его имя. Разверните узел RTM_1 и откройте на редактирование шаблон экрана «Мнемосхема».

Выбор режима управления. ГЭ «Группа кнопок». Создадим элемент для выбора режима управления между автоматическим и ручным. Раскройте группу «Кнопки» на панели ГЭ, нажав на треугольник справа внизу инструмента «Кнопка»  и выберите ГЭ «Группа кнопок» .

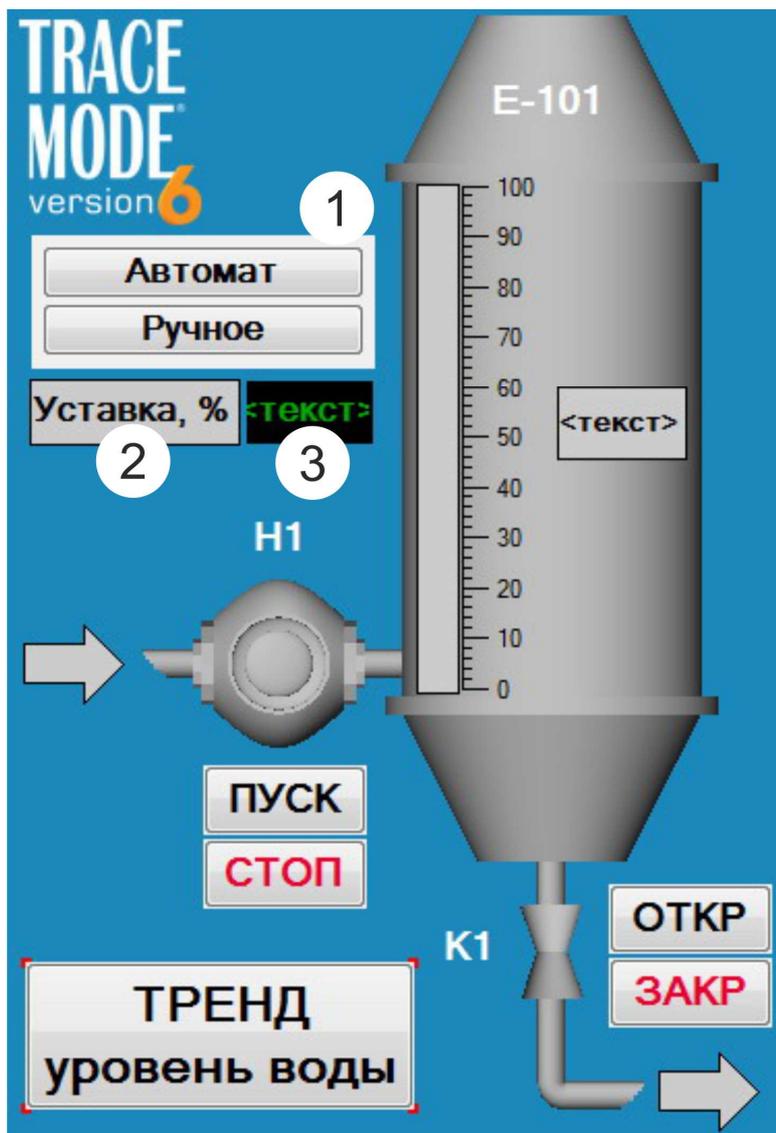


Рис. 3.3. Модифицированный технологический экран «Мнемосхема»

Разместите ГЭ «Группа кнопок» на экране, как показано на рисунке 3.3 (элемент 1). Размещение и задание размера ГЭ «Группа кнопок» производится также, как и других ГЭ (см. создание ГЭ «Емкость» ЛРН№1).

В свойстве «Привязка» данного ГЭ создайте привязку к вновь созданному аргументу «Режим» с типом *IN* и типом данных *USINT*. В свойстве «Исключающий» выберите из списка значение «True», что позволит при нажатии (активации) одной из кнопок автоматически деактивировать остальные. В свойстве «Режим привязки» выберите из списка «Управление и индикация», что позволит видеть какая из кнопок активна.

Нажмите ПКМ на свойстве «Кнопки» и далее ЛКМ на пункт «Кнопка». В подпункте «Текст» свойства «Кнопки» вместо «Элемент» введите «Автомат», а в подпункте «Значение» введите «0». Повторите действия для создания второй кнопки с текстом «Ручное» и значением «1».

Нажмите ЛКМ справа от свойства «Шрифт» и выберите 10 кегль и жирное начертание.

Включите режим эмуляции, нажав кнопку «Эмуляция» , и проследите, что аргумент «Режим» при нажатии кнопок «Автомат» или «Ручное» принимает те значения, что были введены при создании данных кнопок.

Создание задатчика уставки. Статический и динамический текст. ГЭ «Текст». Воспользуемся ГЭ «Текст» для организации ввода значения уставки. Для этого создайте статический текст «Уставка, %» с помощью инструмента «Текст»  (см. Создание статических надписей. ГЭ «Текст» ЛРН№1) и разместите его под группой кнопок выбора режима (элемент 2). Установите жирное начертание, 12 кегль в свойстве «Шрифт» и черный цвет текста в одноименном свойстве.

Затем создайте динамическое поле для ввода значения уставки (элемент 3) также с помощью инструмента «Текст» , который разместите справа от созданной ранее подписи и выровняйте относительно ее и группы кнопок с помощью команд панели «Топология экрана».

Обеспечим индикацию значения уставки данным ГЭ, как это было сделано в ЛРН№2. В свойствах динамического текста задайте привязку в подпункте «Вид индикации» свойства «Текст» к значению вновь созданного аргумента «Уставка» с типом *IN* и типом данных *REAL* (см. Создание динамических надписей. ГЭ «Текст» ЛРН№1). В свойстве «Заливка» выберите из списка подпункта «Цвет заливки» черный цвет. В свойстве «Цвет текста» выберите зеленый цвет. Установите жирное начертание, 12 кегль в свойстве «Шрифт».

В данной лабораторной работе будем использовать ГЭ «Текст» не только для индикации, но и для ввода значения уставки. Для этого в свойствах данного ГЭ переключитесь на вкладку «События» . Затем нажмите ПКМ на свойстве *MousePress* (См. Создание кнопок управления исполнительными устройствами. ГЭ «Кнопка» ЛРН№1) и в контекстном меню выберите пункт «Передать значение». Появятся дополнительные пункты свойств. В свойстве «Тип передачи» выберите из списка значение «Ввести и передать». Свойства «Значение» оставьте пустым, т.к. значение будет вводиться с клавиатуры. В свойстве «Результат» нажмите на «...» и выберите в редакторе аргументов созданный ра-

нее аргумент «Уставка», подтвердив выбор нажатием кнопки «Готово» в редакторе.

Редактирование аргументов экрана. Поскольку в данной лабораторной работе фактическое состояние исполнительных устройств (клапана и насоса) определяется режимом управления, то необходимо разделить аргументы для команд ручного управления ими и определяющие их фактическое состояние.

Откроем аргументы экрана командой «Вид → Аргументы экрана» в сроке меню. В открывшемся табличном редакторе аргументы «Клапан» и «Насос» переименуем в «Клапан_упр» и «Насос_упр». Также создадим два новых аргумента «Клапан_ручн» и «Насос_ручн» с типом IN и типом данных USINT.

В свойствах кнопок «ПУСК», «СТОП» изменим привязку на вкладке «События» на аргумент «Насос_ручн», а в свойствах кнопок «ОТКР», «ЗАКР» – на аргумент «Клапан_ручн».

Создание программы трехпозиционного регулирования уровня с выбором способа управления на языке FBD.

Напишем данную программу регулирования. Для этого нажмите ПКМ на созданном узле *RTM_1* и выберите в контекстном меню «Создать компонент → Программа». Переключитесь в слой «Шаблоны программ» и переименуйте ее в «Регулятор». Переключитесь обратно в узел *RTM_1*.

Двойным нажатием ЛКМ на канале вызова программы, откройте ее на редактирование в *РШП* (См. разработка программы эмулятора-регулятора температуры агента в емкости ЛР№2).

Нажмите ЛКМ на пункте «Аргументы» *РШП* и создайте в табличном редакторе аргументов программы шесть аргументов с параметрами *Уставка IN REAL*, *Уровень IN REAL*, *Насос_ручн IN USINT*, *Клапан_ручн IN USINT*, *Режим IN USINT*, *Насос_упр OUT USINT*, *Клапан_упр OUT USINT*.

Нажмите ЛКМ на заголовке программы «Регулятор». В появившемся окне выбора языка программирования выберите язык *FBD*. Нажмите в окне выбора языка кнопку «Принять».

В *РШП* нажмите на панели кнопку «Показать/скрыть палитру *FBD* блоков» . Откроется библиотека функциональных блоков.

Откройте в библиотеке вкладку «Арифметические» и, нажав ЛКМ на блоке «Сложение» $X+Y$ и не отпуская ее, перетащите блок из библиотеки в рабочее поле *РШП* (рис. 3.4). Аналогично перетащите блок «Вычитание» $X-Y$.

Откройте в библиотеке вкладку «Регулирование» и перетащите в рабочее поле *РШП* блок «Трехпозиционный регулятор» *PREG*. Аналогично перетащите

блок «Выбор из двух» *SEL* из раздела библиотеки «Выбор». Изучите справку по данным блокам.

Блок «Выбор из двух» *SEL* скопируйте (*CTRL+C*) и вставьте здесь же (*CTRL+V*).

Блок трехпозиционного регулятора *PREG* (рис. 3.4) сигнализирует о выходе контролируемой величины, подаваемой на вход (*INP*), за границы диапазона [*MIN*, *MAX*]. Блок может быть использован в программе, с помощью которой значение некоторого сигнала удерживается в заданном диапазоне. Вход (*DLT*) блока предназначен для задания величины гистерезиса на отключение сигналов управления (*QL*) и (*QH*).

Блок работает по следующему алгоритму. Пока $MIN < INP < MAX$, $QL = QH = 0$. Если $INP > MAX$, то выход $QL = 1$, но $QL = 0$ при $INP < MAX - DLT$. Аналогично, если $INP < MIN$, выход $QH = 1$, но $QH = 0$ при $INP > MIN + DLT$.

Присвойте соответствующим входам блока «Сложение» $X+Y$ аргументы (рис. 3.4): входу (X) – аргумент «Уставка», входу (Y) – константу «5» (значение половины зоны нечувствительности по условию). Выход (O) блока $X+Y$ соедините со входом (MAX) блока *PREG* (См. разработка программы эмулятора регулятора температуры агента в емкости ЛРН₂).

Аналогично поступите со входами блока «Вычитание» $X-Y$. Выход (O) блока $X-Y$ соедините со входом (MIN) блока *PREG*.

Таким образом, с помощью блоков сложения и вычитания реализовано вычисление пороговых (максимального и минимального) значений уставки с учетом зоны нечувствительности на основании номинального значения уставки, задаваемого оператором.

К входу (*INP*) блока «Трехпозиционный регулятор» *PREG* привяжите аргумент «Уровень», который хранит соответствующее значение, рассчитанное программой-эмулятором. На вход (*DLT*) подайте значение константы «2» (значение гистерезиса по условию).

Выход (*QL*) блока *PREG*, активизирующийся в случае необходимости уменьшения регулируемой величины, соедините со входом (*IN0*) одного из блоков выбора *SEL*. А выход (*QH*), активизирующийся в случае необходимости увеличения регулируемой величины, – со входом (*IN0*) другого блока выбора *SEL*.

Блок «Выбор из двух» *SEL* предназначен для выбора одного из двух значений, поступающих на входы (*IN0*) и (*IN1*) соответственно, чей номер (0 или 1) поступает на вход (*IG*), и выдачи его на выход (*SEL*), т.е. $SEL = IN_k$, ($k = 0,1$), если $IG = k$. В нашем случае на вход (*IN0*) подадим значения с автоматического

регулятора, а на (*INI*) – команды ручного управления. Таким образом, при подаче на вход (*IG*) значения «0», будет активирован автоматический режим управления и будут взяты значения со входов (*INO*), на которые подает команды управления регулятор, а при подаче на вход (*IG*) значения «1», будет активирован ручной режим управления и будут взяты значения со входов (*INI*), на которые подадим команды управления оператора. Два блока нужны постольку поскольку команд управления также две (больше и меньше), и в отношении каждой из них следует делать выбор.

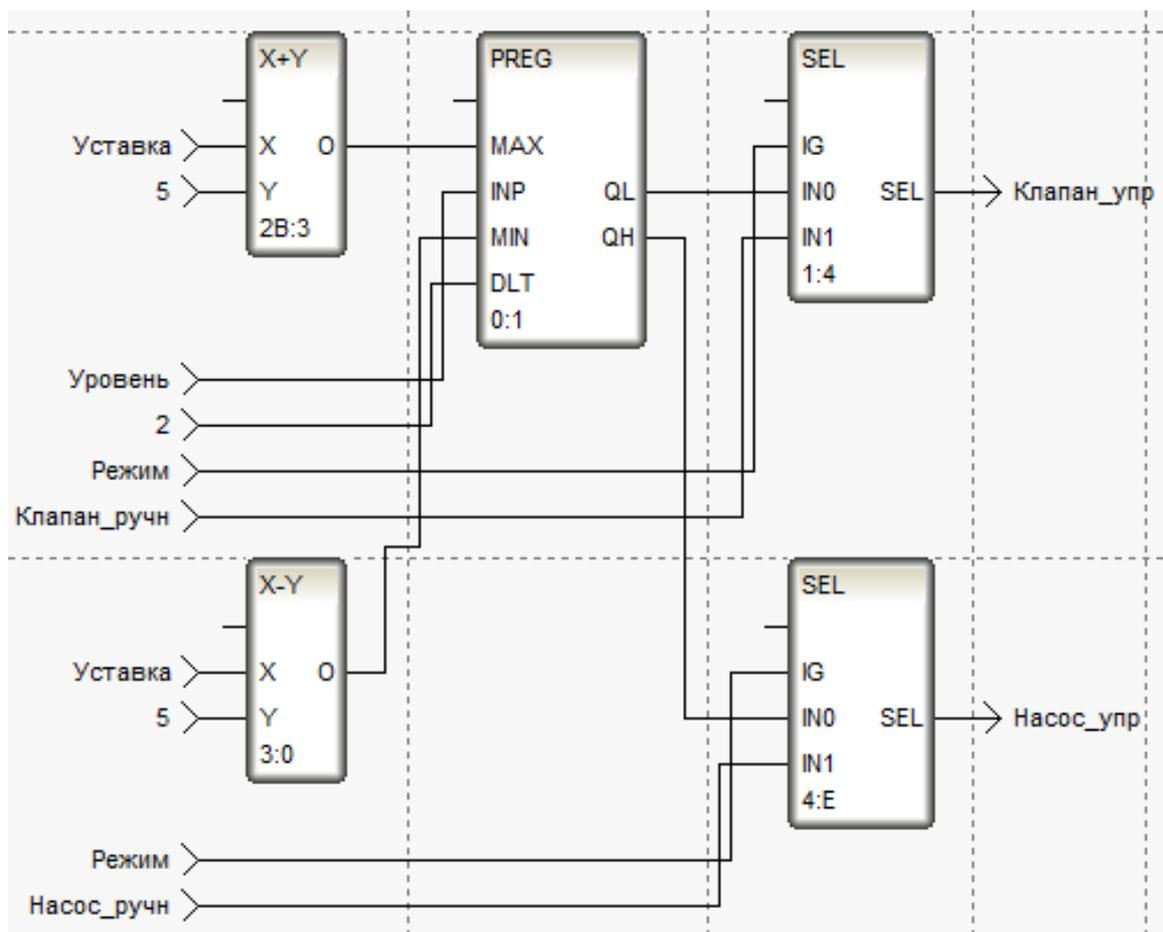


Рис. 3.4. Программа трехпозиционного регулирования уровня с выбором режима управления

Руководствуясь вышесказанным, ко входу (*INI*) блока *SEL*, отвечающего за формирование команды «меньше» (выход (*QL*) регулятора), привяжите аргумент «*Клапан_ручн*», т.к. именно клапан в открытом состоянии способствует уменьшению уровня. К выходу (*SEL*) данного блока привяжите аргумент «*Клапан_упр*».

Соответственно, ко входу (*INI*) второго блока *SEL*, отвечающего за формирование команды «больше» (выход (*QH*) регулятора), привяжите аргумент

«Насос_ручн», т.к. именно включенный насос способствует увеличению уровня. К выходу (*SEL*) данного блока привяжите аргумент «Насос_упр».

На входы (*IG*) обоих блоков *SEL* привяжите аргумент «Режим».

Компиляция и отладка. Выполните компиляцию программы клавишей *F7*. Откройте окно переменных, нажав кнопку «Переменные»  на панели инструментов отладчика, и запустите программу на циклическое выполнение, нажав клавишу *F5* (см. Компиляция и отладка ЛРН№1).

Введите в поле аргумент «Уставка» значение «70» (уставка по условию). Поскольку программа-эмулятор в данный момент не исполняется, то значение уровня в соответствующий аргумент для проверки введите вручную – например, «50». Обратите внимание, что регулятор выработал команду на запуск насоса «Насос_упр»=1, т.к. требуемое значение «70» больше текущего уровня «50» даже с учетом зоны нечувствительности «5». Измените значение уровня на, скажем, «90». Проследите за реакцией регулятора – теперь выработана команда открытия клапана «Клапан_упр»=1, а насос отключен. Измените значение уровня на число в диапазоне [65...75], т.е. в пределах зоны нечувствительности. Как можно видеть, регулятор отключил оба исполнительных устройства.

Введите в аргумент «Режим» значение «1», тем самым перейдя на ручное управление. Вводите команды пуска/открытия («1») или останова/закрытия («0») в аргументы «Насос_ручн» и «Клапан_ручн» и наблюдайте, что в данном режиме команды, выработанные регулятором, игнорируются и на выход программы в аргументы «Насос_упр» и «Клапан_упр» поступают введенные вами значения управляющих команд.

Редактирование шаблона программы «Регулятор». Комбинирование языков МЭК 61131-3. Функции. В настоящий момент проект содержит две программы: программу-эмулятор на языке ST и программу-регулятор на языке FBD. Создадим комбинированную программу, сочетающую в себе оба языка. Программу-эмулятор при этом представим в виде функции (пользовательского блока).

Для этого откройте на редактирование программу «Регулятор» в *РШП* и выделите ЛКМ пункт «Функции» в дереве структуры программы. Аналогично тому, как мы создавали ранее аргументы (см. Настройка отображения состояния насоса ЛРН№1), создайте функцию с именем «LEVEL_EMUL».

В дереве структуры программы разверните функцию «LEVEL_EMUL» и нажмите ЛКМ на пункте «Аргументы» функции. Создайте три новых аргумента также и с теми же параметрами, как делали это при разработке программы-

эмулятора в первой лабораторной работе (Разработка программы-эмулятора изменения уровня в емкости ЛР№1).

Откройте на редактирование шаблон программы-эмулятора из навигатора проекта и скопируйте текст программы на ST из программы-эмулятора в буфер обмена, а затем вставив в тело разрабатываемой функции. Открыть функцию на редактирование можно, нажав на заголовке функции «*LEVEL_EMUL*» в структуре программы и выбрав язык ST в диалоге.

Снова нажмите на заголовок программы «*Регулятор*» в структуре программы. Откройте палитру блоков FBD кнопкой  на панели и перейдите в раздел «*Пользовательские*». Единственный пользовательский блок «*LEVEL_EMUL*» (блок получает имя функции) и представляет созданную нами функцию. Аргументы функции с типом *IN* являются входами блока, а аргумент с типом *OUT* – выходом блока.

Перетащите блок в рабочее поле редактора и разместите, как показано на рисунке 3.5. Выходы (*SEL*) блоков выбора *SEL* с аргументами «*Насос_упр*» и «*Клапан_упр*» соедините с соответствующими входами (*Насос*) и (*Клапан*) блока «*LEVEL_EMUL*».

Переключатель в аргументы программы «*Регулятор*» в структуре программы и измените тип аргумента «*Уровень*» на *OUT*. Присвойте выходу (*Уровень*) блока «*LEVEL_EMUL*» соответствующий аргумент. Также удалите привязку входа (*INP*) блока *PREG* к аргументу «*Уровень*», в который поступает значение, рассчитанное внешней программой-эмулятором, и подайте на него значение с выхода (*Уровень*) блока «*LEVEL_EMUL*» в качестве обратной связи.

Выполните компиляцию и отладку программы. Убедитесь, что функция регулирования и выбора режима управления работает также, но теперь программа вычисляет и значение уровня.

Редактирование шаблона экрана «Тренд». ГЭ «Тренд». Добавим возможность отображения уставки на тренде. Для этого откройте шаблон экрана «Тренд» на редактирование в РПД. Зайдите в свойства ГЭ «Тренд» и, перейдя на вкладку «*Кривые*» , нажмите ПКМ на свойстве «*Кривые*» и в контекстном меню выберите пункт «*Кривая*». В свойстве «*Привязка*» выберите вновь созданный аргумент «*Уставка*» с типом *IN* и типом данных *REAL*. В свойстве «*Цвет*» выберите желтый. В свойстве «*Толщина линии*» введите «3». В свойстве «*Макс. значение*» оставьте значение «100», в свойстве «*Мин. значение*» – «0». (См. Создание графиков по уставке и температуре агента, состоянию ТЭНа. ГЭ «Тренд»).

Редактирование базы каналов. Удалите не нужную теперь программу-эмулятор из узла. Для этого нажмите ПКМ на канале вызова шаблона программы «Эмуляция_уровень» и выберите в контекстном меню пункт «Уничтожить» . Подтвердите данное действие. Сама программа при этом остается в проекте, в чем можно убедиться переключившись в слой «Шаблоны_программ» в навигаторе проекта.

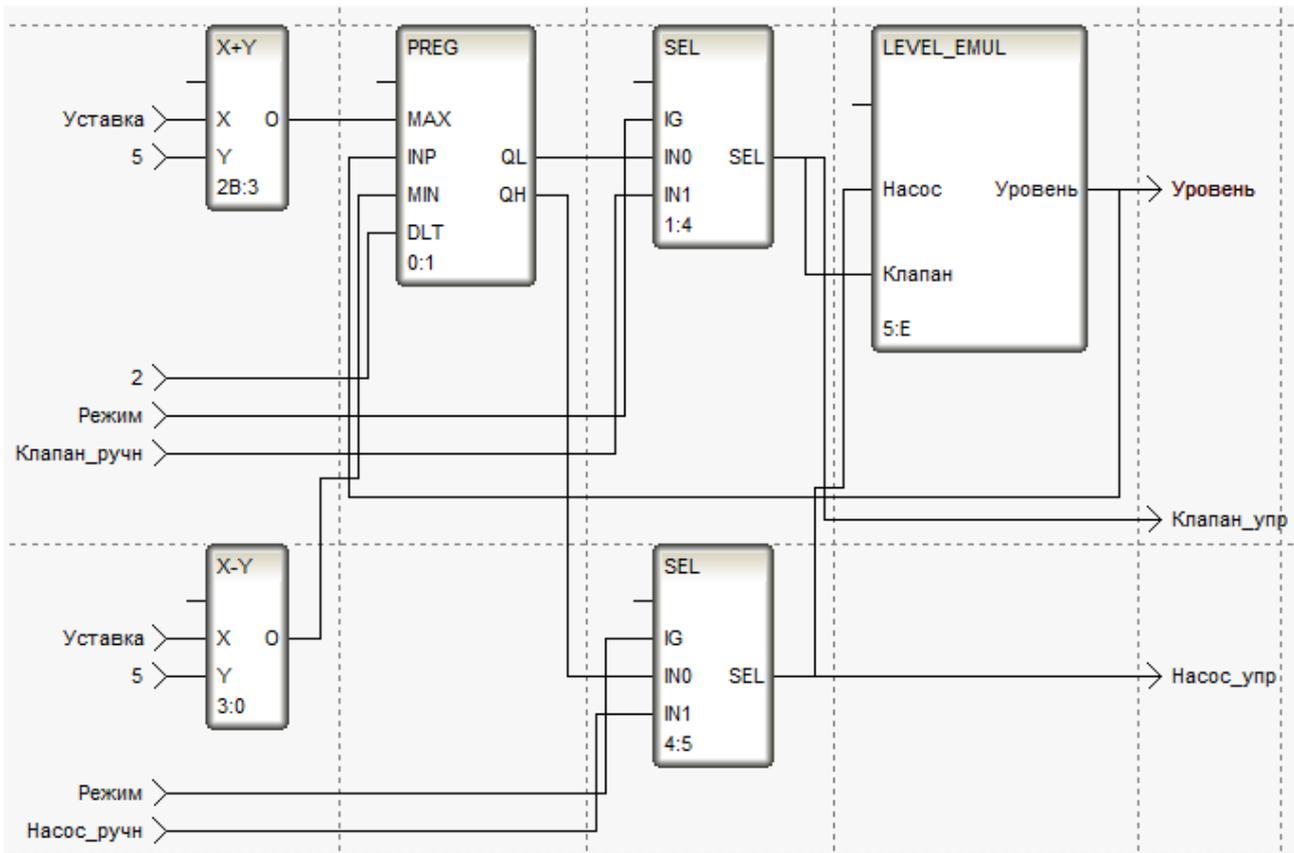


Рис. 3.5. Комбинированная программа эмулятор-регулятор уровня

Для того чтобы не было путаницы, удалите аналогично из базы каналов узла *RTM_1* также каналы «Насос» и «Клапан». В отличие от команды «Удалить» , команда «Уничтожить»  производит удаление канала не только из узла, но и из базы каналов проекта.

Воспользуйтесь процедурой автопостроения каналов по аргументам шаблонов программы или экрана, чтобы связать их между собой (см. Создание базы каналов ЛРН№1). Для этого сначала создайте каналы по аргументам шаблона экрана «Мнемосхема» (так как его аргументы мы отредактировали) в добавление к имеющемуся каналу «Уровень», а затем привяжите созданные каналы к одноименным аргументам других шаблонов методом перетаскивания.

Для того чтобы оператору не требовалось сразу при запуске проекта в профайлере каждый раз вводить значение уставки, зададим его по умолчанию.

Для этого в навигаторе проекта нажмите ПКМ на канале «Уставка» и выберите в контекстном меню пункт «Редактировать» . Значок канала изменит вид на  Уставка. В бланке свойств канала установите флаг в поле «Отработать» и введите в поле «На старте» значение «70».

Сохранение проекта и запуск на исполнение в профайлере. Нажмите ЛКМ кнопку «Сохранить»  и затем «Сохранить для МРВ»  на главной панели инструментов (см. Сохранение и подготовка проекта к запуску ЛР№1). Откройте профайлер кнопкой «Запустить профайлер»  на главной панели инструментов. В профайлере еще раз нажмите ЛКМ кнопку «Запуск/Останов» .

Проверьте работоспособность проекта, вводя различные значения уставки в автоматическом режиме, и управление исполнительными устройствами в ручном режиме. Итоговый результат разработки, запущенный на исполнение в профайлере, показан на рисунке 3.6.

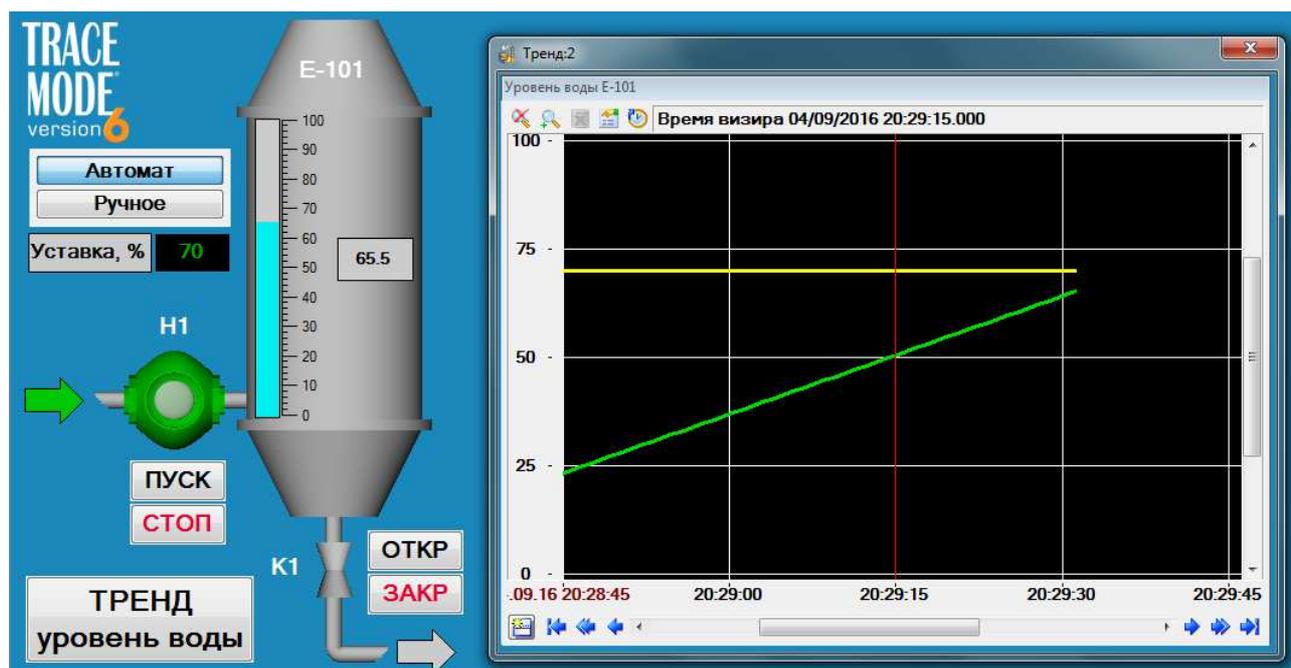


Рис. 3.6. Проект, запущенный на исполнение в профайлере

Для останова профайлера нажмите ЛКМ кнопку «Запуск/Останов» . После этого закройте профайлер.

3.4. Содержание отчета

Отчет должен включать:

- 1) скриншот экрана проекта, запущенного на исполнение в профайлере, с указанием типов использованных ГЭ и настраиваемых свойств ГЭ, использованных в данной работе;
- 2) разработанный алгоритм программы трехпозиционного регулирования уровня воды (без эмуляции) по ГОСТ 19.701-90;
- 3) комбинированную программу эмулятор-регулятор уровня воды на языке FBD с описанием использованных блоков и алгоритма их работы, а также текстовым пояснением работы всей программы;
- 4) *самостоятельное задание.* Программу трехпозиционного регулятора уровня на языке ST (без эмуляции объекта);
- 5) *самостоятельное задание.* Программу-эмулятор уровня на языке FBD (без регулятора);
- 6) вывод по проделанной работе.

3.5. Контрольные вопросы

1. Каково назначение трехпозиционных регуляторов?
2. Для управления какими исполнительными устройствами предназначены трехпозиционные регуляторы?
3. Каковы основные параметры настройки трехпозиционных регуляторов?
4. Каково условие применения двухпозиционных и трехпозиционных регуляторов по отношению к параметрам объекта управления?
5. Поясните статическую характеристику обратного трехпозиционного регулятора с зоной регулирования без перекрытия.
6. Поясните алгоритм работы функционального блока «Трехпозиционный регулятор».
7. Поясните алгоритм работы функционального блока «Выбор из двух».
8. Что такое «функция» и какова их роль в программах?
9. Каково назначение пользовательских функциональных блоков?
10. Поясните порядок создания пользовательских функциональных блоков в Trace Mode.

ЗАКЛЮЧЕНИЕ

Системы сбора данных и диспетчерского управления (SCADA-системы) являются неотъемлемым компонентом современной автоматизированной системы управления (АСУ). В пособии рассмотрено применение SCADA Trace Mode 6 для решения задач управления технологическим процессом и разработки программ регулирования технологических параметров, являющихся важнейшей частью прикладного программного обеспечения автоматизированных систем управления.

В данном пособии была рассмотрена реализация программ двухпозиционного и трехпозиционного регулирования на языках программирования промышленных контроллеров стандарта МЭК 61131 Structured Text (ST) и Function Block Diagram (FBD), а также должное внимание уделено освещению теоретических аспектов выбора конкретного регулятора в соответствии с решаемой задачей и параметрами объекта управления; расчету параметров регуляторов и их влиянию на показатели качества переходного процесса. Так же освещены порядок и принципы разработки интерфейса оператора.

Надеемся, что данное пособие будет полезным не только студентам направления «Автоматизация технологических процессов и производств», но и действующим специалистам АСУТП, занятым в сфере разработки прикладного программного обеспечения и работающим со SCADA-системами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. / В.И. Денисенко. – М.: Горячая Линия-Телеком, 2009 – 608 с.: ил. - ISBN 978-5-9912-0060-8
2. Trace Mode 6. Руководство пользователя. Интегрированная SCADA/HMI-SOFTLOGIC-MES-EAM-HRM-система для разработки АСУ ТП, АСКУЭ и систем управления производством. 14-е изд. – М.: «AdAstra Research Group», Ltd, 2011. – 619 с. ил.
3. Симановский А.Ю. Типы регуляторов. Методика настройки регуляторов / А.Ю.Симановский. – Ивано-Франковск: Изд-во «МИКРОЛ», 2011. – 63 с
4. Копелович, А. П. Инженерные методы расчета при выборе автоматических регуляторов / А. П. Копелович; ред. А. Л. Малый. - М. : Metallurgizdat, 1960. - 190 с.: ил.
5. Ротач В.Я. Теория автоматического управления: учебник для вузов / В. Я. Ротач. — 5-е изд., перераб. и доп. — М.: Издательский дом МЭИ, 2008. — 396 с., ил. - ISBN: 978-5-383-00326-8
6. Андреев Е.Б. Программные средства систем управления технологическими процессами в нефтяной и газовой промышленности: учеб. пособие для вузов/ Е.Б.Андреев, В.Е.Попадько.-М.:ФГУП Изд-во "Нефть и газ" РГУ нефти и газа им. И.М.Губкина.Ч.1.-2005.-268с. - ISBN 978-5-8365-0316-1
7. Орлов С.А. Технологии разработки программного обеспечения: Разработка сложных программных систем: Учебник для вузов. - 3-е изд.-СПб.:Питер,2004.-527с.:ил.-(Учебник для вузов) - ISBN 5-94723-145-X
8. Харазов В.Г. Интегрированные системы управления технологическими процессами: учеб. пособие для вузов.-СПб.:Профессия, 2009.- 592с. :ил. - ISBN 978-5-93913-176-6

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Тема 1. Основы разработки проекта в TRACE MODE	4
Лабораторная работа №1. Основы разработки проекта АСУТП в TRACE MODE.....	4
Тема 2. Позиционные регуляторы	29
Лабораторная работа №2. Двухпозиционное регулирование параметров.....	29
Лабораторная работа №3. Трехпозиционное регулирование параметров. Переключение режима управления.....	47
ЗАКЛЮЧЕНИЕ	61
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	62

Практикум

ШКРОМАДО Антон Алексеевич
ШЕСТОВ Руслан Владимирович
БИРЮКОВ Алексей Николаевич

SCADA-СИСТЕМЫ

Редакторы:

Е. С. Захарова
И. А. Назарова

Подписано в печать 27.02.17 г.
Формат 60x84 1/16. Бумага офсетная
Усл. п. л. 3,8 Уч.-изд. л. 3,8
Тираж 100 экз. Рег. № 3/17sf

Федеральное государственное бюджетное образовательное
Учреждение высшего образования
«Самарский государственный технический университет»
443100, г. Самара, ул. Молодогвардейская, 244. Главный корпус

Отпечатано в типографии
Самарского государственного технического университета
Филиал в г. Сызрани, 446001, г. Сызрань, ул. Советская 45