Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Оренбургский государственный университет»

Кафедра управления и информатики в технических системах

В.А. Трипкош

## КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ УПРАВЛЕНИЯ В ТЕХНИЧЕСКИХ СИСТЕМАХ

Методические указания

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательным программам высшего образования по направлению подготовки 27.03.04 Управление в технических системах

Оренбург 2019 Рецензенты: профессор, доктор экономических наук В.Н. Шепель, доцент, кандидат технических наук М.Ю. Шрейдер

Трипкош, В.А.

Т67 Компьютерные технологии управления в технических системах: методические указания / В.А. Трипкош; Оренбургский гос. ун-т. – Оренбург : ОГУ, 2019. – 132 с.

Методические указания направлены на формирование общепрофессиональных и профессиональных компетенций, связанных с выполнением лабораторных работ по направлению подготовки бакалавров 27.03.04 Управление в технических системах. В методических указаниях приведены общие правила выполнения лабораторных работ, формируемые компетенции, а также описания лабораторных работ и указания по их выполнению, включая допуск к работе и защиту отчетов.

Методические указания предназначены для выполнения лабораторных работ по дисциплине «Компьютерные технологии управления в технических системах» обучающихся очной И заочной форм обучения для по образовательным программам высшего образования направлению по подготовки 27.03.04 Управление в технических системах.

> УДК 004.4(076.5) ББК 32.971.3я7

© Трипкош В.А., 2019 © ОГУ, 2019

### Содержание

Введение	4
1 Формируемые компетенции	6
2 Общие правила выполнения лабораторных работ и критерии оценки	8
3 Описания лабораторных работ и методические указания по их выполнению	13
3.1 Лабораторная работа № 1. Логическое и физическое проектирование	
компьютерной сети отдела предприятия	13
3.2 Лабораторная работа № 2. Знакомство с работой в среде имитационного	
моделирования GPSS World	17
3.3 Лабораторная работа № 3. Моделирование систем средствами	
имитационного моделирования GPSS World	26
3.4 Лабораторная работа № 4. Изучение комплекса CoDeSys для разработки	
прикладных программ для программируемых контроллеров	38
3.5 Лабораторная работа № 5. Создание визуализации проектов в среде	
программирования контроллеров CoDeSys	52
3.6 Лабораторная работа № 6. Изучение среды программных пакетов	
диспетчерского управления и сбора данных Trace Mode 6	70
3.7 Лабораторная работа № 7. Создание проекта системы автоматизации и	
управления в инструментальной среде Trace Mode	74
3.8 Лабораторная работа № 8. Создание статического и динамического	
изображения в среде Trace Mode	91
3.9 Лабораторная работа № 9. Программирование в среде Trace Mode на	
языках Texno ST и Texno FBD	100
3.10 Лабораторная работа № 10. Программирование в среде Trace Mode на	
языках Texno IL и Texno SFC	114
3.11 Лабораторная работа № 11. Создание интерфейса оператора и модели	
управления в инструментальной среде Trace Mode	. 123
Список использованных источников	. 132

#### Введение

Компьютерные технологии – это разновидность современных информационных технологий, основанных на использовании средств вычислительной техники (компьютеров).

В свою очередь компьютерные технологии управления в технических системах – это разновидность современных информационных технологий, основанных на использовании средств вычислительной техники (компьютеров) и направленных на решение задач управления техническими объектами (системами).

Поэтому учебная дисциплина «Компьютерные технологии управления в технических системах» посвящена изучению вопросов, связанных с решением задач управления техническими объектами и технологическими процессами на основе средств вычислительной техники.

Изучение дисциплины «Компьютерные технологии управления в технических системах» является важным этапом подготовки бакалавров по направлению 27.03.04 Управление в технических системах и в рамках этой дисциплины предусмотрено выполнение цикла лабораторных работ.

Лабораторные работы относятся к основным видам учебных занятий. Они обеспечивают экспериментальное подтверждение теоретических положений дисциплины и формирование учебных и профессиональных практических умений обучающихся. Лабораторные работы составляют важную часть теоретической и профессиональной практической подготовки обучающихся.

Лабораторные работы являются неотъемлемой частью изучения дисциплины «Компьютерные технологии управления в технических системах» и имеют целью закрепление практических навыков по использованию компьютерных технологий управления в технических системах.

Лабораторные работы по дисциплине относятся к средствам, обеспечивающим решение следующих основных задач:

– закрепление и углубление теоретических знаний, полученных на лекциях;

получение дополнительной информации по дисциплине;

4

 приобретение навыков научного исследования вопросов, изучаемых в рамках дисциплины, навыков аналитического мышления при обработке полученных результатов;

 приобретение навыков самостоятельной работы с программноаппаратными средствами, применяемыми для управления техническими системами и технологическими процессами.

Лабораторные работы выполняются по следующим разделам дисциплины:

Раздел № 1. Основы построения систем автоматизации и управления.

Раздел № 2. Задачи управления в технических системах и методы их решения с использованием компьютерных технологий.

Раздел № 3. Техническое обеспечение компьютерных технологий управления в технических системах.

Раздел № 4. Программное обеспечение компьютерных технологий управления в технических системах.

Цель настоящих методических указаний – способствовать формированию общепрофессиональных и профессиональных компетенций, связанных с выполнением лабораторных работ бакалавров, по дисциплине «Компьютерные технологии управления в технических системах».

Выполнение цикла лабораторных работ позволит расширить и углубить знания, умения, навыки и компетенции обучающихся по использованию современных компьютерных технологий при управлении техническими системами.

#### 1 Формируемые компетенции

В ходе выполнения цикла лабораторных работ по дисциплине «Компьютерные технологии управления в технических системах» обучающийся должен сформировать ряд общепрофессиональных компетенций (ОПК) и профессиональных компетенций (ПК), предусмотренных спецификой дисциплины:

 – ОПК-9 – способностью использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности;

– ПК-1 – способностью выполнять эксперименты на действующих объектах по заданным методикам и обрабатывать результаты с применением современных информационных технологий и технических средств.

В результате формирования ОПК-9 обучающиеся должны:

 – знать основы построения систем автоматизации и управления; задачи управления в технических системах и основные методы их решения с использованием компьютерных технологий;

– уметь создавать проекты систем в среде программных пакетов диспетчерского управления и сбора данных; разрабатывать средства автоматизации и управления в инструментальной среде для программирования контроллеров; осуществлять проектирование компьютерной сети отдела предприятия; соблюдать основные требования информационной безопасности на объектах автоматизации и управления;

 владеть способностью использовать навыки работы с компьютером; методами информационных технологий.

В результате формирования ПК-1 обучающиеся должны:

 – знать техническое и программное обеспечение компьютерных технологий управления в технических системах;

– уметь проводить моделирование систем в среде имитационного моделирования; выполнять эксперименты на действующих объектах по заданным методикам

6

и обрабатывать результаты с применением современных информационных технологий и технических средств;

– владеть навыками применения современных информационных технологий и технических средств систем автоматизации и управления.

### 2 Общие правила выполнения лабораторных работ и критерии оценки

Лабораторная работа – это разновидность учебного занятия, направленная на закрепление и углубление теоретических знаний, практических навыков и овладение современной методикой проведения эксперимента в соответствии с требованиями образовательной программы высшего образования по соответствующему направлению подготовки бакалавров. Лабораторная работа состоит из экспериментальной, аналитической частей и контрольных мероприятий.

Для проведения лабораторных занятий по дисциплине «Компьютерные технологии управления в технических системах» используется аудитория, оснащенная компьютерной техникой. Рабочим местом для выполнения лабораторных работ является лабораторная установка на базе персонального компьютера, оснащенного специальным программным обеспечением:

– Open Office/LibreOffice – свободный офисный пакет программ, включающий в себя текстовый и табличный редакторы, редактор презентаций и другие офисные приложения;

– CoDeSys – среда разработки прикладных программ для программируемых логических контроллеров;

– GPSS World Student Version – общецелевая система имитационного моделирования;

 – SCADA TRACE MODE – программная система для автоматизации технологических процессов, телемеханики, диспетчеризации, учета ресурсов и автоматизации зданий.

Для успешного выполнения лабораторных работ по дисциплине «Компьютерные технологии управления в технических системах» обучающийся должен выполнять следующие правила:

 – строго выполнять весь объем домашней подготовки, указанный в описаниях соответствующих лабораторных работ;  – знать, что выполнению каждой работы предшествует проверка готовности обучающегося (допуск к работе), которая проводится преподавателем;

– знать, что после выполнения работы, обучающиеся должны представить отчет с защитой полученных результатов и выводов.

При подготовке к очередной лабораторной работе необходимо знать, что на лекциях рассматривается далеко не весь материал. Поэтому основной объем учебного материала должен быть усвоен обучающимся в ходе самостоятельной работы с рекомендованной литературой. При этом необходимо обратить внимание на четкое уяснение практического приложения рассматриваемых теоретических вопросов дисциплины.

В ходе самостоятельной работы при подготовке к очередной лабораторной работе, обучающийся должен четко понять и твердо усвоить рассмотренный на предыдущих занятиях учебный материала и сопоставить его с конкретными заданиями, изложенными в описании соответствующей лабораторной работы. Описания всех лабораторных работ дисциплины приведены в настоящих методических указаниях, а также в электронном курсе «Компьютерные технологии управления в технических системах» в системе обучения Moodle [1].

При необходимости следует обратиться за консультацией к преподавателю. При этом перед консультацией требуется хорошо продумать вопросы, которые должны быть разъяснены.

Выполнение лабораторной работы осуществляется под руководством преподавателя кафедры, ведущего занятия по дисциплине.

В начале каждого лабораторного занятия проводится инструктаж и проверка знаний обучающихся (допуск к работе) – их теоретической готовности к выполнению задания.

Форму такой проверки (устный опрос, письменная «летучка» или тестирование), перечень контрольных вопросов, а также критерии принятия решения о допуске или недопуске обучающегося к выполнению работы преподаватель доводит до обучающихся на предшествующем лабораторном занятии. Содержание контрольных вопросов (тестов) позволяет оценить знания теоретических положений, которые предстоит экспериментально проверить обучающимся в ходе лабораторной работы, а также оценить знания аппаратных и программных средств лабораторной установки и умения их использовать в ходе работы.

Обучающиеся, не допущенные к выполнению лабораторной работы, устраняют указанные преподавателем пробелы в ходе текущего занятия. Повторная процедура допуска к работе и выполнение заданий лабораторной работы организуется в указанное преподавателем время.

Лабораторная работа выполняется каждым обучающимся индивидуально. Однако преподавателем может быть организована работа обучающихся в составе подгрупп или даже всей учебной группы, в соответствии с порядком и требованиями, изложенными в методических указаниях к выполнению конкретной лабораторной работы. На занятии не допускается осуществление обучающимися посторонних действий, непредусмотренных выполнением лабораторной работы.

Ответственность за соблюдение правил техники безопасности всеми участниками лабораторной работы возлагается на преподавателя, проводящего занятие.

Ответственность за порядок на учебном месте, за исправность и сохранность лабораторной установки несет обучающийся.

Отчет по результатам лабораторной работы выполняется в текстовом редакторе. Отчет должен содержать следующие структурные элементы:

- титульный лист;

– цель и задание на лабораторную работу;

основную часть, включающую описание хода выполнения задания и полученные результаты (таблицы, графики, скриншоты экрана компьютера и др.);

- основные выводы по работе;

– приложение (при необходимости).

Отчет по результатам лабораторной работы выполняется каждым обучающимся лично вне зависимости от того, осуществлялась ли работа индивидуально или в составе группы. В последнем случае в отчете обучающегося должны быть отражены результаты выполнения индивидуального задания или элементы лично выполненной им работы, в зависимости от порученной ему роли в группе. Содержание и оформление всех структурных элементов отчета по лабораторной работе должно соответствовать требованиям, предъявляемым к студенческим работам [2].

Отчет в распечатанном виде предъявляется преподавателю на этапе защиты лабораторной работы и/или выгружается в электронном курсе дисциплины в системе Moodle [1].

Результаты выполнения каждой лабораторной работы защищаются каждым обучающимся индивидуально (или группой при выполнении работы в группе) перед преподавателем, ведущим дисциплину. В ходе защиты обучающиеся должны продемонстрировать работу выполненного проекта, обосновать сформулированные в отчете выводы и ответить на контрольные вопросы преподавателя (примерный перечень вопросов приведен в описании каждой лабораторной работы).

Обучающиеся, выполнившие лабораторную работу, но не защитившие ее, имеют право на повторную защиту.

Результаты защиты лабораторной работы определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется за работу, в которой: полностью выполнено задание (в том числе индивидуальное), в отчете дано всестороннее и глубокое освещение практически исследуемых вопросов; исполнитель показал умение работать с лабораторной установкой и нормативными документами по проведению экспериментов, проводить научные исследования, делать теоретические и практические выводы; обучающийся дает исчерпывающие и аргументированные ответы на контрольные вопросы.

Оценка «хорошо» выставляется за работу, в которой: полностью выполнено задание (в том числе индивидуальное), отчет отвечает предъявляемым к нему требованиям; исполнитель умеет работать с лабораторной установкой и нормативными документами по проведению экспериментов, однако не на все вопросы дает аргументированные ответы.

Оценка «удовлетворительно» выставляется за работу, в которой: задание в основном выполнено, но имелись затруднения с выполнением индивидуального

11

задания, отчет отвечает основным, предъявляемым к нему требованиям; исполнитель в целом умеет работать с лабораторной установкой, но не проявил должного умения самостоятельно проводить эксперименты; обучающийся поверхностно отвечает на контрольные вопросы и допускает существенные недочеты.

Оценка «неудовлетворительно» выставляется за работу, в которой: задание (в том числе индивидуальное) не выполнено и/или отчет не отвечает основным, предъявляемым к нему требованиям; исполнитель не умеет работать с лабораторной установкой и не проявил должного умения самостоятельно проводить эксперименты; обучающийся не отвечает на контрольные вопросы или допускает в ответах грубые ошибки.

Обучающийся, не выполнивший в установленные сроки все лабораторные работы по дисциплине или не защитивший их, кроме случая неявки по уважительной причине, считается имеющим академическую задолженность и не допускается к сдаче экзамена по данной учебной дисциплине.

### 3 Описания лабораторных работ и методические указания по их выполнению

3.1 Лабораторная работа № 1. Логическое и физическое проектирование компьютерной сети отдела предприятия

Цели работы: приобрести практические навыки по проектированию компьютерной сети предприятия путем ее логической и физической структуризации; углубить и закрепить знания сетевых компьютерных технологий управления техническими системами.

Время: 4 часа.

Задание:

– в соответствии с заданным вариантом, разработать структурную схему локальной вычислительной сети (ЛВС) Ethernet отдела предприятия, включающую общий сервер и M рабочих групп из N компьютеров в группе;

– в соответствии с заданным вариантом предусмотреть расстояние между рабочими станциями в группе (L), расстояние между рабочими группами (S), среднюю интенсивность трафика одной рабочей станцией (k), соотношение трафиков внутригруппового и межгруппового 50 % на 50 %.

– подготовить и защитить отчет по лабораторной работе.

Примечание: работа выполняется по вариантам согласно таблице 1.1.

Таблица 1.1 – Варианты индивидуальных заданий на лабораторную работу № 1

		1		1	1
Вариант	М	Ν	S	L	k
1	2	3	4	5	6
1	3	3	200	4	0,165
2	2	4	250	4	0,125
3	3	5	300	3	0,1
4	2	6	350	3	0,08
5	3	7	400	3	0,07
6	2	8	450	4	0,06
7	3	9	475	4	0,06
8	2	10	475	4	0,05
9	3	11	475	5	0,045
10	2	12	475	5	0,04

Продолжение таблицы 1.1

1	2	3	4	5	6
11	4	6	200	4	0,1
12	3	10	250	5	0,08
13	4	7	300	4	0,07
14	5	6	350	5	0,06
15	5	7	300	5	0,06

Методические указания по выполнению задания.

3.1.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 1.

В начале занятия получите инструктаж преподавателя и допуск к работе. После чего получите индивидуальное задание на работу № 1.

3.1.2 Логическое проектирование сети (логическая структуризация и определение логической схемы ЛВС).

Рассчитайте трафик одного компьютера по формуле

$$C_i = k \cdot C$$
,

где *k* – коэффициент (берется из варианта задания);

*С* – равно 10 Мбит/с.

Определите суммарный трафик неструктурированной сети по формуле:

$$C_{\Sigma}^{H} = N \cdot M \cdot C_{i},$$

где *N*, *M* – величины, которые берутся из задания;

 $C_i$  – трафик одного компьютера.

Определите коэффициент нагрузки неструктурированной сети по формуле:

$$\rho_{H} = \frac{C_{\Sigma}^{H}}{C_{MAKC}},$$

где *С<sub>макс</sub>* – пропускная способность сетевой технологии (вначале необходимо выбрать 10 Мбит/с).

Проверьте выполнение условия допустимой нагрузки локальной сети (домена коллизий), используя выражение:

$$\rho_{\mu} \le \rho_{Ethermet} = 0.35, \tag{1.1}$$

где  $\rho_{\mu}$  – коэффициент нагрузки неструктурированной ЛВС (коэффициент нагрузки домена коллизий  $\rho_{Д.К}$ ).

Если не выполняется условие (1.1), то требуется произвести логическую структуризацию проектируемой локальной сети.

Для этого необходимо разбить ЛВС на домены коллизий (логические сегменты) по числу рабочих групп и количеству рабочих станций как указано в соответствующем варианте задания. Затем определите коэффициент нагрузки домена коллизий по формуле:

$$\rho_{\mathcal{J}.\mathcal{K}} = \frac{N_{\scriptscriptstyle \mathcal{J}.\mathcal{C}} \cdot C_i}{C_{\scriptscriptstyle \mathcal{M}\mathcal{A}\mathcal{K}\mathcal{C}}},$$

где  $N_{n.c}$  – количество компьютеров в логическом сегменте.

Проверьте выполнение условия (1.1).

Если условие не выполняется, то разбейте каждый домен коллизий на два. Повторяйте такое разбиение доменов до тех пор, пока не будет выполнено условие (1.1).

Определите межгрупповой трафик по формуле:

где  $C_{\Sigma}^{\mu}$  – суммарный трафик неструктурированной ЛВС.

Рассчитайте коэффициент нагрузки ЛВС по межгрупповому обмену по формуле:

$$\rho_{Me\#.rp}^{H} = \frac{C^{H}_{Me\#.rp}}{C_{Makc}},$$

где  $C^{H}_{Me\mathcal{K}.rp}$  – межгрупповой трафик;

 $C_{_{Makc}}$  – максимальная производительность выбранной базовой технологии Ethernet. Для выполнения условия (1.1) последовательно увеличивайте производительность базовой технологии, чтобы достичь нормальной нагрузки межгруппового соединения.

Произведите контрольную проверку проведенных расчетов.

3.1.3 Физическое проектирование сети.

Выберите тип соединительного кабеля для рабочих станций исходя из требований физического уровня стандартов Ethernet, Fast Ethernet, Gigabit Ethernet.

Выберите коммутационное оборудование. Такой выбор производится исходя из предложенного списка спецификаций. Выберите оптимально подходящее оборудование.

Согласно проведенному расчету графически выполните структурную схему ЛВС отдела предприятия. При соединении сетевого оборудования кабелем в соответствии с заданием установите длину сегментов.

3.1.4 Проверочный расчет спроектированной сети на корректность.

Производите расчет времени двойного оборота PDV (Path Delay Value) для доменов коллизий спроектированной ЛВС по формуле:

$$PDV = 2 \cdot \left[ \left( \frac{bt}{M} \cdot L \right) + \left( bt_r + bt_t \right) + \left( \frac{bt}{M} \cdot L \right) \dots \right],$$

где  $\frac{bt}{M}$  – задержка в кабеле на 1 метр (взять равным 1);

*L* – длина кабеля (м);

*bt*<sub>*r*</sub> – задержка на входном порту концентратора (взять равным 0);

 $bt_t$  – задержка на выходном порту концентратора (взять равным 0).

Проверьте допустимые размеры сети по формуле

$$PDV < 512bt$$
.

Если условие не выполняется, необходимо повторно произвести физическую структуризацию проектируемой ЛВС, изменив длину кабелей или спецификацию оборудования. 3.1.5 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет и представьте его для защиты лабораторной работы № 1. В ходе защиты отчета необходимо быть готовым к демонстрации корректности разработанной ЛВС, обоснованию принятых инженерных решений и к ответам на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) Какие этапы предусматривает проектирование компьютерной сети предприятия?

2) Содержание логического проектирования сети.

3) Содержание физического проектирования сети.

4) Что включает проверочный расчет спроектированной ЛВС на корректность?

## 3.2 Лабораторная работа № 2. Знакомство с работой в среде имитационного моделирования GPSS World

Цели работы: приобрести базовые навыки работы в среде имитационного моделирования GPSS World; углубить и закрепить знания по использованию GPSS World при решении профессиональных задач.

Время: 2 часа.

Задание:

– установить на персональный компьютер общецелевую систему имитационного моделирования GPSS World Student Version;

– изучить теоретические сведения, используя учебное руководство по системе GPSS World, ознакомиться с интерфейсом системы;

– практически выполнить загрузку в систему GPSS World Student Version примера модели на языке моделирования GPSS;

– подготовить и защитить отчет по лабораторной работе.

17

Методические указания по выполнению задания.

3.2.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 2.

В начале занятия получите инструктаж преподавателя и допуск к работе.

3.2.2 Установка на персональный компьютер и запуск GPSS World, ознакомление с его интерфейсом.

Одной из версий системы моделирования GPSS (General Purpose Systems Simulator) является система GPSS World. Для ее установки на компьютер вначале загрузите с сайта MinutemanSoftware.com [3] самораспаковывающийся архивный файл студенческой версии GPSS World. Также можно воспользоваться файлом, расположенным в указанной преподавателем папке компьютера.

Далее запустите этот файл. Программа GPSS World устанавливается на компьютер стандартно, так же как и другие программы в операционной системе Windows. После установки запустите программу GPSS World двойным щелчком мыши по ее значку или через мену Пуск. Появится окно Notices (Заметки) (рисунок 2.1).

CURRENT VERSIONS AVAILABLE: GPSS World Student Version 4.3.5
GPSS World Commercial Version 4.3.5
You should download notices from our Internet Web site when you install a new version of the software. ( File / Internet / Download Notices)
See General Information on our Web site for the new features in Version 4.3.5 Version 4.2.1 (and later) Objects are incompatible with those of 4.1.2, and earlier. You may
have to create new Model Objects using the Windows clipboard. Updates to Version 4.3.5 are free. A Commercial Version Upgrade (new Major Version Number
of the same product) or Reregistration requires a small charge. Visit the web site for details. Click: File / Internet / GPSS Web Page.
Stop showing these notices. Download Don't Download Cancel

Рисунок 2.1 – Окно Notices (Заметки)

В этом окне содержатся полезные сведения о программе и информация об обновлениях.

Главное окно GPSS World, которое появится после закрытия окна Notices, представлено на рисунке 2.2.

Brand Contract Contra		
<u>Eile E</u> dit <u>S</u> earch <u>V</u> iew <u>C</u> ommand <u>V</u>	<u>V</u> indow <u>H</u> elp	
	₩?	
For Help, press F1	Results	Clock

Рисунок 2.2 – Главное окно GPSS World

Главное окно программы содержит: заголовок, основное меню, панель инструментов, рабочую область и строку состояния. Строка состояния разделена на три части. В них представлены подсказки с информацией об используемых пунктах меню, сообщения об ошибках и модельное время (таймер) в процессе моделирования. Включение (выключение) таймера для каждого объекта «Процесс моделирования» производится установкой (сбросом) флажка View → Simulation Clock (Вид → Модельное время).

Кроме основного меню для каждого объекта GPSS World имеются собственные меню в их окнах, но большая часть действий запускается с помощью основного меню (File, Edit, View, Windows, Help).

В главном окне программы пункты основного меню имеют команды подменю, доступные для использования в определенных случаях. Для освоения приемов работы с программой имитационного моделирования предназначено учебное руководство по системе GPSS World, которое по приведенной в [4] ссылке необходимо загрузить на компьютер или открыть файл, в указанной преподавателем папке.

В данном руководстве изучите теоретические сведения по системе GPSS World, начиная с ознакомления с интерфейсом программы.

3.2.3 Загрузка в систему GPSS World Student Version примера на языке моделирования GPSS.

В соответствие с примером, приведенном в [5], в информационнотехническое подразделение ремонтного предприятия с одним каналом может поступать неисправное оборудование (НО) двух типов. НО первого и второго типов ремонтируются одними и теми же специалистами. Интервалы времени поступления НО первого типа распределены равномерно и составляют  $20\pm10$  часов. Распределение интервалов времени поступления НО второго типа –  $15\pm8$  часов. Поступающее НО ремонтируется по правилу «первым поступило – первым обслужено». На ремонт НО первого типа затрачивается  $6\pm2$  часа, а на ремонт НО второго типа –  $8\pm4$  часов.

Для описания работы информационно-технического подразделения будем использовать две независимые последовательности блоков (сегментов):

– первая последовательность моделирует ремонт НО первого типа;

– вторая – ремонт НО второго типа.

В качестве единицы модельного времени возьмем время, равное 1 минуте. Время моделирования – 72 часа.

При построении имитационной модели используются следующие операторы (идентификаторы):

- Rem1 – подразделение с одним каналом обслуживания;

- RemQ - общая очередь необслуженного HO,

- RemQ1 - очередь НО первого типа;

- RemQ2 - очередь НО второго типа;

- VrRem - имя таблицы, в которой табулируется общее время ремонта НО;

20

– VrReml – имя таблицы, в которой табулируется время ремонта НО первого типа;

– VrRem2 – соответственно, время ремонта НО второго типа.

Введите операторы модели в GPSS World для создания нового объекта «Модель». Выберите команду File  $\rightarrow$  New (Файл  $\rightarrow$  Новый), после чего появится окно Новый документ (рисунок 2.3).

Новый документ	
Создать	OK
Text File	Отмена
	<u>С</u> правка

Рисунок 2.3 – Окно выбора объекта типа «Модель»

Выберите Model, откроется окно текстового редактора. В это окно введите

программу рассматриваемого примера в части блока «Определение таблиц»:

;Пример ;Определение таблиц VrRem1TableM1,420,180,5 VrRem2TableM1,420,180,5 VrRemTableM1,420,180,5

Фрагмент программы определения таблиц в окне текстового редактора пока-

зан на рисунке 2.4.

Таким же образом введите остальные операторы модели:

GENERATE 600,300; Источник НО первого типа QUEUERemQ; Встать в общую очередь QUEUERemQ1; Встать в очередь НО первого типа SEIZERem1; Занять информационно-техническое подразделение DEPARTRemQ; Покинуть общую очередь DEPARTRemQ1; Покинуть очередь НО первого типа ADVANCE360,120; Имитация обслуживания RELEASERem1; Освободить информационно-техническое подразделение TABULATE VrRem1; Данные о НО первого типа TRANSFER ,Met1; Отремонтированное НО первого типа

Сегмент имитации ремонта НО второго типа GENERATE 720,480; Источник НО второго типа QUEUERemQ; Встать в общую очередь QUEUERemQ2; Встать в очередь НО второго типа SEIZERem1; Занять информационно-техническое подразделение DEPARTRemQ; Покинуть общую очередь DEPARTRemQ2; Покинуть очередь НО второго типа ADVANCE480,240; Имитация обслуживания RELEASERem1; Освободить информационно-техническое подразделение TABULATE VrRem2; Данные о НО второго типа в таблицу Met1 TABULATEVrRem; Данные о НО обоих типов в таблицу TERMINATE; Отремонтированное НО первого и второго типов

Сегмент задания времени моделирования GENERATE 4320 TERMINATE1

😹 GPSS World - Untitled Model 1		
<u> E</u> ile <u>E</u> dit <u>S</u> earch <u>V</u> iew <u>C</u> ommand <u>V</u>	<u>V</u> indow <u>H</u> elp	
	<b>№</b> ?	
Untitled Model 1		
;Пример ;Определение таблиц	20.100.5	
VrRemi IABLE MI44	(U,18U,5 20 120 5	
VrRem TABLE MI4	20.180.5	
For Help, press F1	Results	Clock //

Рисунок 2.4 – Фрагмент программы определения таблиц в окне текстового редактора

Однако лучше воспользоваться окнами создания блоков и окном Insert GPSS Blocks (Вставить блок GPSS). Для этого выберите команду Edit  $\rightarrow$  Insert GPSS Blocks... (Правка  $\rightarrow$  Вставить блок GPSS...). Щелкните левой кнопкой мыши по пункту GENERATE в меню блоков GPSS (рисунок 2.5).

Insert GPSS B	lock into Model	Object 🛛 🔀
ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	C WBITE	

Рисунок 2.5 – Окно меню блоков GPSS

В поле операнда А окна создания блока GENERATE введите число 600. Для перехода в поле операнда В воспользуйтесь клавишей Таb и введите в него число 300. В поле Comment введите комментарий: «Источник НО первого типа».

Окно создания блока GENERATE с заполненными данными представлены на рисунке 2.6.

Нажмите кнопку ОК, после чего в модели появится оператор

GENERATE 600, 300; Источник НО первого типа

Введите остальные операторы модели, как показано на рисунке 2.7.

Выберите команду File  $\rightarrow$  Save As... (Файл  $\rightarrow$  Сохранить как...) для создания папки Пример и сохранения объекта Модель. В окне вместо текста «Untitled» следует ввести текст «Пример» и нажать кнопку Сохранить (комбинация клавиш Ctrl + S).

Для открытия этой модели в дальнейшем нужно использовать команду File → Open (Файл → Открыть) (комбинация клавиш Ctrl + 0).

Далее необходимо создать объект Процесс моделирования.

Enter Block Information	
<b>(</b> )	GENERATE
GENERATE -	Create XN for future entry.
A: 600 B: 300 C: D: E: F: G: H:	Intergeneration time. Halfrange or Function Modifier. Start delay time. Creation limit. Priority.
Comment: Источник но	
Junio India India	
НО ОК	Cancel Help

Рисунок 2.6 – Окно создания блока GENERATE с заполненными данными

🔡 Пример.gp	15				×
;Пример					^
;Определе	ние таблиц	ы			
VrReml	Table	M1,420,180,5			
VrRem2	Table	M1,420,180,5			
VrRem	Table	M1,420,180,5			
;Сегмент	имитации р	емонта НО первого	TI	ипа	
	GENERATE	600,300	;	Источник НО первого типа	
	QUEUE	RemQ	;	Встать в общую очередь	
	QUEUE	RemQ1	;	Встать в очередь НО первого типа	
	SEIZE	Reml	;	Занять ремонтное подразделение	
	DEPART	RemQ	;	Покинуть общую очередь	
	DEPART	RemQ1	;	Покинуть очередь НО первого типа	
	ADVANCE	360,120	;	Имитация ремонта	
	RELEASE	Reml	;	Освободить подразделение	
	TABULATE	VrReml	;	Данные о НО первого типа	
	TRANSFER	,Metl	;	Отремонтированное НО первого типа	
; Сегмент и	MUTALUU PEM GENERATE QUEUE SEIZE DEPART DEPART ADVANCE RELEASE TERMINATE Metl TABU TERMINATE	oHTa HO второго типа 720,480 RemQ RemQ2 Rem1 RemQ2 480,240 Rem1 VrRem2 LATE	;;;;;;;;V;;	Источник НО второго типа Встать в общую очередь Встать в очередь НО второго типа Занять подразделение Покинуть общую очередь Покинуть очередь НО второго типа Имитация обслуживания Освободить подразделение Данные о НО второго типа в таблицу rRem ; Данные о НО обоих типов в таб Отремонтированное НО обоих типов	б
, cerment s	GENERATE	4320			
	TEDMINATE	1320			
1	TERMINALE	1			
1					~
<				>	·

Рисунок 2.7 – Модель Пример в окне текстового редактора

Объект Процесс моделирования представляет собой оттранслированный объект Модель. Трансляция объекта Модель осуществляется командой Command  $\rightarrow$  Create Simulation (Команда  $\rightarrow$  Создать процесс моделирования) (комбинацией клавиш Ctrl + Alt + S). В ходе своей работы транслятор GPSS проверяет нет ли в программе синтаксических ошибок. Если синтаксических ошибок нет, то транслятор активизирует все интерактивные команды и окна. Это позволяет пользователю следить за состоянием процесса моделирования и управления им.

При наличии синтаксических ошибок объект Процесс моделирования не будет создан. Поэтому необходимо воспользоваться списком сообщений об ошибках трансляции, который транслятор создает в окне Journal (Журнал).

Для поиска и исправления ошибок служит команда Search → Next Error (Поиск → Следующая ошибка) (комбинация клавиш Ctrl + Alt + N).

После исправления всех ошибок в окне Journal появится сообщение об этом. Далее выполните команду Command  $\rightarrow$  Retranslate (Команда  $\rightarrow$  Повторная трансляция) (комбинация клавиш Ctrl + Alt + R).

Запустите процесс моделирования, чтобы завершить создание объекта Процесс моделирования. Для этого выберите команду Command  $\rightarrow$  Start (Команда  $\rightarrow$  Старт). В диалоговом окне Start Command (рисунок 2.8) предоставляется возможность заменить один прогон (START 1) на любое их количество, например, на 100 или 1000. При вводе значения NP – стандартный отчет не выводится.

Start Command	
START 1	
ОК	Cancel

Рисунок 2.8 – Окно команды Start Command

Возможна ситуация, когда после нажатия кнопки ОК произойдет останов моделирования по ошибке выполнения. В этом случае будет выдано соответствующее сообщение в окне Journal и транслятор не создаст список ошибок выполнения, как это имеет место в случае ошибок трансляции. Поэтому строку с ошибкой выполнения потребуется искать самостоятельно, воспользовавшись командой Search  $\rightarrow$  Go To Line... (Поиск  $\rightarrow$  Перейти к строке...) (комбинация клавиш Ctrl + Alt + G).

После устранения ошибок объект Процесс моделирования будет создан.

3.2.4 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет и представьте его для защиты лабораторной работы № 2. В ходе защиты отчета необходимо быть готовыми к демонстрации загруженного в систему примера имитационной модели и к ответам на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) В каких случаях используются методы имитационного моделирования?

2) Что понимается под имитационной моделью?

3) Что представляет собой модель в системе GPSS World?

4) Что такое транзакт в системе GPSS World?

5) Что такое блок в системе GPSS World?

6) Дайте краткую характеристику основным типам блоков GPSS World (устройство, многоканальное устройство, логический ключ, очередь).

7) Дайте краткую характеристику основным типам блоков GPSS World (матрица, сохраняемое значение, таблица, переменная, поток данных).

5) Перечислите и опишите основные команды GPSS World.

# 3.3 Лабораторная работа № 3. Моделирование систем средствами имитационного моделирования GPSS World

Цели работы: приобрести базовые навыки работы в среде имитационного моделирования GPSS World; углубить и закрепить знания по использованию GPSS World при решении профессиональных задач.

Время: 4 часа.

26

Задание:

 отладить и протестировать модель, загруженную в ходе предыдущей лабораторной работы;

– практически выполнить пример имитационного моделирования на языке моделирования GPSS;

– подготовить и защитить отчет по лабораторной работе.

Методические указания по выполнению задания.

3.3.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 3.

В начале занятия получите инструктаж преподавателя и допуск к работе. После чего получите индивидуальное задание на работу № 3.

3.3.2 Отладка и тестирование имитационной модели.

Выберите команду File → Open (Файл → Открыть). Затем выберите файл Пример.gps paнee сохраненной модели Пример и нажмите Open (Открыть).

Далее потребуется закрепить команды SHOW за функциональными клавишами. Для этого откройте журнал настроек модели и выберите команду Edit  $\rightarrow$  Settings (Правка  $\rightarrow$  Настройки). Появится окно с открытой вкладкой Simulation (рисунок 3.1).

ntitled - SETTINGS	
Simulation Reports Random Nur	mbers Function Keys Expressions
Simulation Options	
Poll Count	50
Parameter Block Size	3
Parameter Block First	1
Max Evaluation Depth	2000
Stack Size	1000
Integration Tolerance	1.000000e-006
Max Memory Request	1500000
	□ Switch * and #
🔽 Single Calendar Day	1/0 Stream Error Stops
OK OTT	мена Применить Справка

Рисунок 3.1 – Окно журнала настроек модели с открытой вкладкой Simulation

Далее перейдите на вкладку Function Keys (Функциональные клавиши) для закрепления команд за функциональными клавишами. В полях рядом с клавишами F8 и F9 наберите следующие две команды:

- SHOW Q\$RemQ1;
- SHOW Q\$RemQ2.

Нажмите кнопку ОК. Заполненная данными вкладка Function Keys показана на рисунке 3.2.

Default F	unction Keys	
F1	HELP	F7 STOP "OFF
F2	CONTINUE	F8 SHOW1Q\$Rem
F3	EXIT	F9 SHOWDQ\$Rem
F4	HALT	F10
F5	STEP 1	F11
F6	STOP	F12
Model Ta	abstops	
10	20 37 44	50 📀 chars 🦳 twips

Рисунок 3.2 – Заполненная данными вкладка Function Keys

Теперь появилась возможность нажатием соответствующей клавиши во время выполнения моделирования интерактивно исследовать длины очередей RemQl и RemQ2.

Далее для запуска процесса моделирования выберите команду Command  $\rightarrow$  Create Simulation (Команда  $\rightarrow$  Процесс моделирования). Откройте окно Blocks (Блоки), для чего в главном меню задайте команду Window  $\rightarrow$  Simulation  $\rightarrow$  Blocks Window (Окно  $\rightarrow$  Окно процесса моделирования  $\rightarrow$  Окно Блоки).

Результат выполнения команд показан на рисунке 3.3.

🛃 GPSS World - Untitled.1		
Eile Edit Search Yiew Command Window H	elp	
🗄 Untitled		
Culotitlad Leine IOURMAL		
Ú V C		
V Location	Find Continue Halt Step Place Rer	3 move
Loc Block Type Current Co	o Entry Co Retry Ch Line Number Include-file	
For Help, press F1 **** Mod	el Translation Aborted ****	Clock

Рисунок 3.3 – Окна Blocks и Journal модели Пример

Задайте условие остановки моделирования на активный транзакт номер 20. Для этого выберите команду Command → Custom... (Команда → Ввести...), в диалоговом окне наберите STOP 20 и нажмите кнопку ОК.

В главном меню выберите команду Command  $\rightarrow$  START и в диалоговом окне вместо числа 1 наберите число 100, а затем нажмите кнопку ОК.

В процессе моделирования будет выдано сообщение о том, что транзакт номер 20 остановился при попытке войти в блок. В данном случае это первый блок GENERATE.

Теперь начните процесс моделирования в пошаговом режиме, используя клавишу F5 (в журнале настроек модели за ней закреплена команда STEP 1). Вначале снимите условие остановки. Для этого задайте команду Window  $\rightarrow$  Simulation Snapshot  $\rightarrow$  User Stops (Окно  $\rightarrow$  Кадр процесса моделирования  $\rightarrow$  Условия остановки) и удалите число 20, нажав команду Remove (Удалить) и кнопку ОК. Вызовите команду STEP 1, нажав функциональную клавишу F5 (комбинация клавиш Ctrl + Alt + S). Трассировочное сообщение окна Journal (Журнал) сообщит о том, что транзакт номер 20 вошел в блок GENERATE.

3.3.3 Практически выполнить пример имитационного моделирования на языке GPSS.

В процессе моделирования требуется определить некоторые показатели исследуемого объекта. Для этого задайте команду Command  $\rightarrow$  SHOW... (Команда  $\rightarrow$  SHOW...) и в диалоговом окне наберите Q\$RemQ. Нажмите кнопку OK.

В строке состояния общей очереди RemQ НО главного окна и в окне Journal (Журнал) отобразится ее длина.

Введите команды, закрепленные за функциональными клавишами F8 и F9. В строке состояния главного окна и в окне Journal отобразится длина очереди HO первого типа при нажатии на клавишу F8, а длина очереди HO второго типа – при нажатии на клавишу F9.

Для отображения окна с подробным обзором состояния очередей выберите команду Window  $\rightarrow$  Simulation Window  $\rightarrow$  Queues Window (Окно  $\rightarrow$  Окно процесс моделирования  $\rightarrow$  Окно Очереди).

Нажмите несколько раз на клавишу F5 для наблюдения за изменением параметров очередей. После чего закройте окно.

Для отображения окна с подробным обзором устройств выберите команду Window  $\rightarrow$  Simulation Window  $\rightarrow$  Facilities Window (Окно  $\rightarrow$  Окно процесса моделирования  $\rightarrow$  Окно Устройства). Понаблюдайте за изменением состояния устройств, для чего нажмите клавишу F2 (CONTINUE), а затем F4 (HALT).

Для перехода к общему обзору окна Facilities (Устройства) выберите команду View  $\rightarrow$  Entity Details (Вид  $\rightarrow$  Подробности). После того, как откроется окно Общий обзор устройств, нажмите клавишу F2 (CONTINUE), а затем F4 (HALT).

Состояние устройства Reml изменится. Цвет значка устройства меняется с серого – устройство свободно, на красный – устройство занято. Белая полоска в правой части значка устройства отражает размер очереди: небольшую очередь показывает маленькая полоска; очередь более десяти транзактов – большая полоска. Закройте окно Facilities (Устройства) и откройте окно Expression (Выражения). Для этого задайте команду Window  $\rightarrow$  Simulation Window  $\rightarrow$  Expression Window (Окно  $\rightarrow$  Окно процесса моделирования  $\rightarrow$  Окно Выражения). Появится окно Edit Expression (Редактор выражения) для введения выражений и их меток (рисунок 3.4).

Edit E	xpression Winde	w		×
	- New Expression- Label	1		View
	Expression			Memorize
	- Window Content	\$		
	Label	Expression	<u>_</u>	
	- Memorized Euror		M	Remove
	Label	Expression	<u>&lt; &gt;</u>	View Delete
		OK	Cancel	

Рисунок 3.4 – Окно Edit Expression

Метку и выражение, которое необходимо вычислить, запрашивает Редактор выражения. В поле Label (Метка) его диалогового окна введите текст: «Время». В поле Expression (Выражение) введите: «AC1».

Затем нажмите кнопки View (Просмотр) и Memorize (Запомнить). Это нужно для того, чтобы показывать значения этого выражения в текущем процессе моделирования и просматривать это выражение после сохранения процесса моделирования и повторного его запуска.

В поле Label (Метка) вместо текста «Время» введите текст: «Активный транзакт». Вместо «AC1» введите: «XN1».

Опять нажмите кнопки View и Memorize. Данные кнопки нужно нажимать каждый раз после нового ввода метки и выражения.

Введите метку и третье выражение: «Вероятность ремонта N\$Met1/QC\$RemQ». Нажмите кнопки View и Memorize.

После нажатия кнопки ОК в окне Expression (Выражения) можно увидеть абсолютное модельное время, номер активного транзакта и вероятность ремонта НО.

Последовательно нажимая клавиши F2 (CONTINUE) и F4 (HALT), можно увидеть изменения значений выражений.

Далее закройте и снова откройте окно Expression (Редактор выражения).

В открывшемся окне Edit Expression выражения и их метки сохранятся только в списке Memorized Expression (Запомненные выражения). В этом списке можно выбрать нужные для отображения выражения и нажать кнопку View (Просмотр).

Выберите строку «Время» и нажмите кнопку View (Просмотр), затем – строку «Вероятность ремонта» и кнопку View. Оба выражения появятся в поле Window Contents.

Нажмите кнопку ОК, при этом окно Expression (Выражения) покажет абсолютное модельное время и вероятность ремонта НО обоих типов. Закройте окно.

Закройте все окна, кроме окна с объектом Модель, для завершения процесса моделирования.

Для графического отображения результатов моделирования задайте команду Command  $\rightarrow$  Create Simulation (Команда  $\rightarrow$  Создать процесс моделирования), а затем – команду Window  $\rightarrow$  Simulation Window  $\rightarrow$  Plot Window (Окно  $\rightarrow$  Окно процесса моделирования  $\rightarrow$  Окно График).

Появится окно Edit Plot (Редактор графика), которое предназначено для задания осей, выражений и других общих данных, включаемых в график (рисунок 3.5).

Поле Label (Метка) окна должно содержать метку для пояснений к графику. В это поле введите текст: «Вероятность ремонта».

В поле Expression (Выражение) окна введите первую переменную для построения графика: «N\$Met1/QC\$RemQ».

	1		_	
Lab	el			Plot
Expression				Memorize
- Window Conte	nts			
Title		Time Ran	ge 100.00	00000
Min Value	0.000000	Max Value 1	00.000000	
Label	Expression			
			^	
			~	Remove
	vessions			
Memorized Exp	ressions			
Memorized Exp	Expression			Plot

Рисунок 3.5 – Диалоговое окно Edit Plot

Статистически рассчитать ожидаемую вероятность ремонта можно через отношение числа всего отремонтированного НО к числу всего поступившего в ремонт оборудования. Так как система одноканальная, то число всего поступившего в ремонт НО можно считать равным числу входов в общую очередь.

Для расчета и отображения в виде графика ожидаемых вероятностей ремонта и времени ремонта в поле Title (Заголовок) задайте имя графика, которое описывало бы обе переменные: вероятность выполнения ремонта и время ремонта.

Введите текст: «Вероятность ремонта и время ремонта».

Чтобы график можно было бы рассмотреть для каждого графика, должен быть указан достаточно большой интервал времени. Поэтому в поле Time Rang (Временной интервал) для временного интервала оси X наберите число 43200 (произведение времени моделирования на количество прогонов).

Для оси Y установите число 10, нажмите кнопку Plot (Построить) и кнопку Memorize (Запомнить).

В поле Label (Метка) введите вторую переменную: «Время ремонта».

В поле Expression (Выражение) введите формулу:

(QT\$RemQ+FT\$Rem1)/1000.

Для определения среднего времени ремонта надо просуммировать среднее время пребывания в очереди и среднее время непосредственного пребывания в ремонте. Так как вероятность не может превышать единицы, а время ремонта, в свою очередь, может превысить 1000, масштабный множитель для времени ремонта следует ввести равным 0,001. В этом случае оба графика будут наблюдаться в одном окне.

При нажатии кнопок Plot (Построить), Memorize (Запомнить) и ОК окно Plot (График) примет вид, как показано на рисунке 3.6.



Рисунок 3.6 – Окно Plot с двумя графиками

Теперь запустите процесс моделирования, а количество прогонов укажите равным 10. По завершению всех прогонов окно Plot (График) должно выглядеть, как показано на рисунке 3.7.

Закройте окно Plot (График) и задайте команду File  $\rightarrow$  Save (Файл  $\rightarrow$  Сохранить). По умолчанию будет задано имя Пример.sim. Далее нажмите кнопку Save (Сохранить) и закройте все окна.

Для продолжения процесса моделирования задайте команду File  $\rightarrow$  Open (Файл  $\rightarrow$  Открыть). В нижней части окна в выпадающем списке File of type (Тип файла) выберите тип файла Simulation (Процесс моделирования), а в верхней части окна выберите файл Пример.sim и нажмите кнопку Open (Открыть).

Untitled Model 1(1).9.sim:3	- TABLE WINDOW				
		VRREM	l.		
	Mean: 13789.153		S.	D.: 7218.894	
360 _	1				
-					
0 _	420	600 <sup>1</sup>	780 <sup>1</sup>	960	

Рисунок 3.7 – Процесс моделирования Plot Window

Открывается сохраненный процесс моделирования. Выберите команду Window  $\rightarrow$  Simulation Window  $\rightarrow$  Plot Window (Окно  $\rightarrow$  Окно процесса моделирования  $\rightarrow$  Окно График). В появившемся окне Edit Plot (Редактор графика) в списке Memorized (Сохраненные выражения) выделите первую метку и нажмите Plot (Построить), затем – вторую метку и также нажмите Plot.

Введите заголовок графика «Вероятность ремонта и время ремонта» в поле Title (Заголовок), значение 43200 – в поле Time Range (Временной интервал), число 10 – в поле Max Value (Максимальное значение). Для построения графиков теперь все готово.

После завершения всех указанных в команде START 10 прогонов процесс моделирования был сохранен. Поэтому можно запустить его снова, задав команду Command  $\rightarrow$  START и указав требуемое число прогонов. Если процесс моделирования будет остановлен и сохранен незавершенным, то его можно будет продолжить, нажав функциональную клавишу F2, с того же момента времени.

Закройте окно Plot (График).

Задайте команду Window  $\rightarrow$  Simulation Window  $\rightarrow$  Table Window (Окно  $\rightarrow$  Окно процесса моделирования  $\rightarrow$  Окно Таблица), в окне Open Table (Открыть таблицу) выберите переменную VrRem и нажмите кнопку ОК (рисунок 3.8).

Table	WAITTIME	•
ſ	OK	Cancel

Рисунок 3.8 – Окно Open Table

Появится отображение таблицы VrRem, но на диаграмме пока ничего нет кроме осей X и Y.

Указав в команде START значение 10, запустите процесс моделирования. В ходе прогонов будут наблюдаться изменения диаграммы. После завершения всех десяти прогонов в появившейся гистограмме среднее время ремонта НО обоих типов составит 5490,823 со стандартным отклонением 2521,028.

Закройте все окна и снова создайте процесс моделирования.

Задайте команду Window  $\rightarrow$  Simulation Window  $\rightarrow$  Blocks Window (Окно  $\rightarrow$  Окно процесса моделирования  $\rightarrow$  Окно Блоки). В окне Blocks (Блоки) щелкните на блоке TRANSFER, а затем нажмите кнопку Place (Поместить) на панели инструментов этого окна. В блок TRANSFER будет помещено условие остановки.

Запустите процесс моделирования, для этого выберите команду Command  $\rightarrow$  START (Команда  $\rightarrow$  START). В диалоговом окне наберите значение 10 и нажмите кнопку ОК. В этом случае, как только следующий транзакт попытается войти в блок TRANSFER, процесс моделирования остановится.

Удалите условие остановки из блока TRANSFER. Для этого задайте команду Window  $\rightarrow$  Simulation Snapshot  $\rightarrow$  User Stops (Окно  $\rightarrow$  Кадр процесса моделирования  $\rightarrow$  Условия остановки), нажмите кнопку Remove All (Удалить все) и кнопку ОК.

После завершения процесса моделирования система автоматически создаст отчет. В отчете статистическая информация выводится только о тех объектах, которые были заданы в модели.
В общем случае стандартный отчет, кроме общей характеристики модели, содержит большой перечень сведений, в том числе, например, сведения:

- об именах объектов модели;

– о блоках модели;

об устройствах;

- об очередях;

– о списке текущих событий;

– о списке будущих событий.

Создание стандартных отчетов можно отключить снятием флажка Create Standard Report (Создавать стандартные отчеты) на странице Reports (Отчеты) журнала настроек модели. Вывод отчета также может быть заблокирован записью значения NP в поле операнда В команды START.

Каждому отчету присваивается имя, которое по умолчанию состоит из имени файла модели, порядкового номера процесса моделирования и номера отчета. Номера отчетов всегда начинаются с единицы. Файлы с отчетами имеют расширение .qpr.

Пользователь не может изменить имя отчета, сформированное системой, так как отчеты являются специально отформатированными файлами. При необходимости, все же, сохранить какой-то отчет с именем пользователя, его нужно скопировать, а затем сохранить с требуемым именем. Отчеты можно распечатать с помощью команды File  $\rightarrow$  Print (Файл  $\rightarrow$  Печать) из окна Report (Отчет). Отчеты могут настраиваться на странице Reports (Отчеты), где по желанию можно изменить перечень объектов, сняв или установив соответствующие флажки.

По окончанию моделирования внесите в программу имитационной модели изменения в соответствие с индивидуальным заданием и повторно проведите моделирование.

3.3.4 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет по лабораторной работе № 3 и представьте его для защиты. При защите отчета производится демонстрация работы имитационной модели и анализ данных моделирова-

ния. Обучающиеся должны быть готовы к ответам на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) Какие задачи моделирования решаются на языке GPSS?

2) Поясните понятие «транзакт».

3) Назовите основные функции управляющих операторов/блоков языка GPSS.

4) Назовите основные группы объектов языка GPSS.

5) Назовите особенности синтаксиса языка GPSS.

6) Назовите и охарактеризуйте блоки языка GPSS.

## 3.4 Лабораторная работа № 4. Изучение комплекса CoDeSys для разработки прикладных программ для программируемых контроллеров

Цели работы: приобрести базовые навыки настройки и работы в среде CoDeSys; углубить и закрепить знания по использованию программного комплекса CoDeSys при решении профессиональных задач.

Время: 4 часа.

Задание:

– установить и настроить среду программирования CoDeSys v2.3;

- настроить связь компьютера с ПЛК и создать новый проект в CoDeSys;

– разработать проект логико-программного управления дискретным процессом на языке LD в среде CoDeSys;

- исследовать LD-программу в режиме эмуляции;

– подготовить и защитить отчет по лабораторной работе.

Методические указания по выполнению задания.

3.4.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 4.

В начале занятия получите инструктаж преподавателя и допуск к работе.

3.4.2 Установка и настройка среды программирования CoDeSys v2.3.

Для установки на персональный компьютер среды разработки прикладных программ для программируемых логических контроллеров (ПЛК) CoDeSys требуется регистрация на сайте <u>https://store.codesys.com</u> [6]. После чего по ссылке <u>https://www.codesys.com/download/download-center.html</u> загрузите файл инсталляции бесплатной версии программы CoDeSys v2.3.

Установка программы CoDeSys предусматривает распаковку архивного файла инсталляции программы и выбор компонентов. С учетом условий дальнейшего использования комплекса CoDeSys целесообразно выбрать стандартный набор компонентов.

Настройка пакета предполагает установку профиля целевой платформы, так называемого target-файла. Этот файл содержит всю информацию о контроллерах того или иного типа. Следовательно, для каждого типа контроллеров используется свой target-файл.

В комплексе CoDeSys предусмотрена специальная программа установки target-файлов. Способ ее запуска поясняется рисунком 4.1.

Основное окно программы установки target-файла приведено на рисунке 4.2.

В строке Installation directory основного окна программы, указывается путь к папке, в которой будут устанавливаться target-файлы.

В основном окне программы расположены:

- виды целевых устройств, информация о которых имеется в target-файлах;

– установленные виды целевых устройств, для которых можно создавать проекты на CoDeSys;

– кнопки открытия target-файлов и установки устройств.

Для открытия нужного target-файла требуется воспользоваться кнопкой Open. Потом надо перейти в папку или на диск с данным target-файлом и открыть его. При этом в окне появится информация о типах ПЛК, содержащихся в этом файле.

Далее требуется выбрать требуемое название целевой платформы ПЛК и нажать на кнопку Install.

ореина Кореина	IzeamViewer 12 Secure C Photosma						
=	3S Licensing Manager	События и общение		Развлечения и отды	ыX		
	CODESYS HMI - RoboDemo.pro						
	CODESYS OPC Configurator		<		$\textcircled{\textbf{o}}$		
	CODESYS SoftMotion V2.3	Календарь Почта		Компаньон к	Музыка Groo	Кино и ТВ	
	CODESYS SP PLCWinNT V2.4					_	
	CODESYS UserManual V2.3		A LEVEL	2	0		
	SCODESYS V2.3		Φοτοιοάφ	Финансы	Новости		
	CODESYS Visualization V2.3			_			
	CODESYS WebServer						
	ENI Admin		OneNote	Microsoft Store			
L.	ENI Control	2	_				
	ENI Explorer	Microsoft Edge	1000				
	FOSS components in CODESYS V2.3		-0	<b>N</b>			
<u>ين</u>	InstallTarget		Погода	CODESYS V2.3			
Ф	VersionInfo						
	a Ħ <b>= <u>9</u> = S</b>	<u>×</u>					

Рисунок 4.1 – К пояснению запуска программы установки target-файла

Installation directory: c:\program files (x86)\3s soft  Possible Targets: Installed Targets: Dpen Open
Possible Targets: Installed Targets:       Installed Targets:       Image: Description of the second
⊕- 3S-Smart Software Solutions GmbH     Open
Open
Install
Remove
Close

Рисунок 4.2 – Основное окно программы установки target-файла

Название выбранной целевой платформы, например, PLC150.U-L появится в аналогичном списке в соседней части основного окна. Для удаления ранее установленной целевой платформы ПЛК из списка используется кнопка Remove. Естественно, для этой платформы создавать проекты в CoDeSys уже будет нельзя.

3.4.3 Настройка связи компьютера с ПЛК.

После физического подключения ПЛК к компьютеру с помощью СОМ-порта интерфейса RS-232, USB-интерфейса или интерфейса локальной вычислительной сети Ethernet на основе витой пары требуется настройка подключения.

Такая настройка осуществляется после создания нового проекта, который содержит программы, выполняемые впоследствии ПЛК.

Для настройки подключения нужно вначале запустить программу CoDeSys. Затем в основном окне программы создайте новый проект, нажав на панели инструментов кнопку Новый или выбрав в меню команду Файл → Новый. После появления окна выбора целевой платформы для создания проекта нужно определить тип ПЛК, для которого создается проект (рисунок 4.3).



Рисунок 4.3 – Выбор целевой платформы для нового проекта CoDeSys

После выбора целевой платформы на экране появится новое окно, в котором содержатся параметры и настройки выбранного ПЛК. К ним относятся тип и тактовая частота процессора, адреса сегментов памяти, количества входов и выходов и др. Часть параметров и настроек при необходимости можно изменить.

Нажмите кнопку ОК, и система в окне Новый программный компонент (POU) предложит создать модуль с типом POU – Программа и Языком реализации – LD (рисунок 4.4).

Файл Правка Проект Вставить Дополнения Онлайн Окно Справка В С П П Ф Ф В В М А	
Новый программый компонент (РОИ) Иня нового РОU: Гип РОU С Программа С Фужщаствлений бок С Фужща Тир результатк ВООL С СFC С СFC	
	> OCMOTE
	E

Рисунок 4.4 – Создание нового проекта

В принципе имеется возможность создать любую программную единицу на любом языке.

Далее требуется настроить связь с контроллером, для чего надо в главном меню задать команду Онлайн  $\rightarrow$  Параметры связи.... Появится окно с уже существующими настройками. Если настроек нет, а имеется лишь строка «'localhost' via Tcp/Ip», то необходимо создать новое подключение. Для этого воспользуйтесь кнопкой New... и в появившемся окне выберите вид соединения с контроллером, например, Serial (RS232). Затем нажмите кнопку ОК (рисунок 4.5).

Communication Parameters		×
Channels - 'localhost' via Tcp/Ip localhost' via Tcp/Ip Name Port Baudrate Parity Stop bits Motorola Flow Con	232) Value Comment COM1 38400 No 1 byteorder No rol Off	OK Cancel New Remove Gateway Update

Рисунок 4.5 – Окно настройки связи с контроллером

После создания программы в CoDeSys можно будет проверить связь компьютера с контроллером.

3.4.4 Разработка в среде CoDeSys проекта логико-программного управления дискретным процессом на языке LD.

После создания нового проекта создадим первую программу для ПЛК (рисунок 4.6).

Программы для ПЛК создаются на языках программирования, стандартизированных Международной Электротехнической комиссией (МЭК) стандарта IEC61131-3 (МЭК61131-3). Самым простым языком программирования стандарта МЭК является язык Ladder Diagram (LD) – лестничная диаграмма, лестничная логика. В отечественном стандарте этот язык называется языком релейно-контактных схем (РКС).

Составим на языке LD первую программу для ПЛК [7], с помощью которой можно реализовать логико-программное управление дискретным процессом. Возьмем схему системы логико-программного управления дискретным процессом в релейно-контактном исполнении (рисунок 4.7).

🕏 CoDeSys - (Untitled)* Файл Правка Проект Вставить Дополнения Онлайн Окно Справка		٥	×
	> 		
			>
	1400 132	M Inpo	ICMOTE
vanje pva (b ±# ∧ =	14 01.09	:56	

Рисунок 4.6 – Новая программа, рабочее поле для программирования на LD



Рисунок 4.7 – Релейно-контактная схема системы логико-программного управления дискретным процессом

В состав схемы входят:

– три источника входных дискретных сигналов (кнопка управления «Пуск»
 SB, два электро-контактных дискретных датчика – датчик температуры SK и уровня SL);

- три промежуточных реле К1, К2, К3;

– два выходных устройства (обмотка магнитного пускателя КМ, включающего электродвигатель, и сигнальная лампа HL).

Требуется перенести эту схему в LD-диаграмму. В разрабатываемой многоступенчатой LD-программе потребуется пять цепей.

В первую цепь необходимо внести контакт SB и катушку реле K1.

Для этого наведите курсор на кнопку в панели инструментов с изображением замыкающего контакта, нажмите левую кнопку мыши, и этот контакт появится в цепи с тремя вопросительными знаками красного цвета (рисунок 4.8).



Рисунок 4.8 – Первая цепь с введенным контактом

Эти вопросительные знаки запрашивают имя или идентификатор внесенного в цепь компонента.

Наведите курсор на вопросительные знаки и щелкните левую кнопку мыши. Вопросы становятся белыми на фоне синего прямоугольника. Задайте имя компонента с помощью символов английского алфавита. В данном примере это имя SB.

Нажмите клавишу Enter. Откроется окно Объявление переменной, в которое необходимо занести класс переменных для этого компонента (рисунок 4.9).

Так как проектируемая схема создается в учебных целях и будет работать только в режиме эмуляции, то сразу нажмите левую кнопку мыши на ОК, и имя появится над элементом (рисунок 4.10).

Класс	<u>И</u> мя	<u>Т</u> ип	ОК
VAR	▼ SB	BOOL	
Список	<u>Н</u> ач. значение	Адрес	Отме <u>н</u> а
Global_Varables	*		
омментарии:			

Рисунок 4.9 – Окно объявления переменной



Рисунок 4.10 – Объявленный замыкающий контакт

Следует обратить внимание на пунктирный квадратик вокруг контакта SB и на исчезновение пунктирного прямоугольника в конце цепи. Это говорит о том, что активирован сам контакт SB, то есть можно последовательно и/или параллельно к этому контакту подключать другие контакты.

Завершение первой цепи осуществляется вводом в нее катушки реле К1. Сначала активируйте цепь, наведя курсор на линию цепи и щелкнув левой кнопкой мыши (появился пунктирный прямоугольник). Переведите курсор на кнопку  $\square$ , щелкните левую кнопку мыши, и в цепи появится катушка с тремя белыми вопросами в синем прямоугольнике. Присвойте катушке имя К1 по аналогии, как это было сделано для контакта SB.

Для обозначения функционального назначения цепи в LD-диаграммах часто задают комментарии на любом языке, в том числе и на русском. С этой целью наведите указатель на полоску, расположенную ниже полоски меток и щелкните левой кнопкой мыши, появится мигающий курсор. С клавиатуры задайте, например, «Knopka puska» и нажмите клавишу Enter.

Формирование первой цепи завершено (рисунок 4.11).

0001	Knopka_pusk	
	SB 	кı ()
		(7

Рисунок 4.11. Изображение первой цепи

Зона этой цепи очерчена двумя горизонтальными линиями. Для активации цепи достаточно навести курсор мыши в эту зону.

Создайте вторую цепь, для этого наведите курсор на кнопку Цепь (после) и щелкните левой кнопкой мыши. По аналогии с первой цепью внесите во вторую цепь замыкающий контакт датчика SL и катушку К2. При необходимости можно вписать комментарий, например, «Datchik\_urovnia» (рисунок 4.12).



Рисунок 4.12 – Изображение первой и второй цепи

Если ошибочно нажата не кнопка , а кнопка Цепь (перед) , то новая цепь станет впереди всех предыдущих и присвоит себе номер 0001, а следующим цепям увеличит их номера на единицу. При необходимости вставить новую цепь между двумя уже созданными, то нужно щелкнуть левой кнопкой мыши по цепи 0001, а затем по кнопке . Новая цепь станет после цепи 0001 и получит номер 0002, а вторая, ранее созданная цепь, станет и получит номер 0003.

По аналогии с предыдущими цепями создайте третью цепь, включите в ее состав замыкающий контакт датчика температуры SK и обмотку K3.

Создайте четвертую цепь, включив последовательно в цепь замыкающий контакт К1 и размыкающий К2.

Далее требуется создать цепочку из параллельно расположенных размыкающих контактов К2 и К3. Для этого щелкните левой кнопкой мыши на контакте К1. Контакт активирован, о чем свидетельствует пунктирный квадратик вокруг контакта. Наведите курсор на кнопку Параллельный контакт (инверсный) *щ*, щелкните левой кнопкой мыши, и соединение выполнено. Присвойте имя контакту К3. Внесите в цепь катушку КМ. Четвертая цепь завершена.

Создайте пятую цепь LD-программы. Для этого по аналогии с предыдущими цепями внесите в цепь размыкающие контакты КЗ и К1, замыкающий К2 и обмотку (катушку) HL. Объявлять переменные К1, К2 и К3 не требуется, так как они уже внесены в список булевых переменных.

Осталось только создать параллельную цепочку из замыкающего контакта К1 и размыкающего К2. Для этого активируйте размыкающий контакт К1, нажимая и не отпуская клавишу Shift, активируйте замыкающий контакт К2. Затем отпустите клавишу Shift, оба контакта будут охвачены пунктирным прямоугольником. Наведите курсор на кнопку **99**, щелкните левой кнопкой мыши, и параллельно с этими контактами появится значок  $-|\cdot|_{-}$ , которому присвойте имя К1. Активируйте новый контакт, наведя курсор на кнопку **99**, щелкните левой кнопкой мыши, и последовательно появится значок  $-|\cdot|_{-}$ . Присвойте имя К2 новому элементу.

Фрагмент полученной схемы в пятой цепи показан на рисунке 4.13.



Рисунок 4.13 – Изображение пятой цепи

Параллельная цепочка выполнена и, в целом, завершена вся многоступенчатая схема в виде LD-программы (рисунок 4.14).



Рисунок 4.14 – LD-программа системы логико-программного управления дискретным процессом

3.4.5 Исследование LD-программы в режиме эмуляции.

Разработанный в CoDeSys проект на языке LD проверьте на выполнение заданных условий срабатывания и на отсутствие ложных включений исполнительных элементов. В CoDeSys имеется возможность это сделать в режиме эмуляции, то есть не использовать реальные аппаратные средства, включая ПЛК.

Для этого наведите курсор на пункт меню Онлайн (Online) и щелкните левой кнопкой мыши. В открывшемся меню сместите курсор к строке Режим эмуляции (Simulation Mode) и щелкните левой кнопкой мыши, чтобы перед этой строкой появилась «галочка».

Снова откройте это меню и щелкните левой кнопкой мыши по строке Подключение (Login). Левая вертикальная «шина питания», размыкающие контакты реле, участки цепей от этой шины до какого-то «непроводящего» в этом положении элемента цепи (например, замыкающего контакта) окрашиваются в синий цвет.

Если в программе ошибок нет (в противном случае их надо устранить), опять откройте меню Онлайн и щелкните левой кнопкой мыши по строке Старт (Run). Программа готова к приему входных сигналов и можно начать ее исследование.

Начать исследование можно с комбинации входных сигналов, равных логическому нулю: SB = 0, SL = 0, SK = 0. То есть SB, SL и SK не сработали.

Откройте меню Онлайн и щелкните левой кнопкой мыши по строке Записать значения (Write Values). Убедитесь, что К1, К2, К3, КМ и HL не сработали, так как их вид не изменился.

Перейдите ко второй комбинации, где должен замкнуться контакт кнопки управления SB (SB = 1).

Наведите курсор на контакт SB, сделайте двойной щелчок левой кнопкой мыши. В полости этого контакта появится синий квадратик. Откройте меню Онлайн и щелкните по строке Записать значения левой кнопкой мыши. Контакт SB, катушка К1 и вся первая цепь закрасятся синим цветом. Контакты К1 в четвертой и пятой цепях замыкаются и также окрашиваются в синий цвет. Размыкающие контакты К3 и К4 в этих цепях остаются замкнутыми, то есть срабатывают КМ и HL.

Аналогично для всех других комбинаций входных сигналов исследуйте работу программы с целью контроля правильности работы алгоритма управления и выявления ложных срабатываний исполнительных элементов.

Например, для случая размыкания контакта SB и замыкания контакта SK наведите курсор на SB и щелкните левой кнопкой мыши. В полости контакта появится зеленый квадратик и останется часть синей заливки. Наведите курсор на SK и щелкните левой кнопкой мыши. Контакт заливается синим цветом. Откройте меню Онлайн и щелкните левой кнопкой мыши по строке Записать значения. Убедитесь, что выключилась катушка KM. Так последовательно пройдите по всем строкам таблицы состояний входных и выходных устройств. В случае простой системы для данного примера можно сразу проверить только условия срабатывания исполнительных элементов, замыкая и размыкая соответствующие входные элементы SB, SL и SK как показано выше.

Для исследования реакции LD-программы на следующую комбинацию состояний входных сигналов, требуется вернуть схему в исходное состояние (SB = 0, SL = 0 и SK = 0). Для этого достаточно щелкнуть левой кнопкой мыши по строке Отключение (Logout), то есть отключить режим Онлайн. Потом по описанной методике запустить режим эмуляции, создать нужную комбинацию состояний входных элементов и, нажав левой кнопкой мыши по строке Записать значения, зафиксировать поведение исполнительных элементов.

3.4.5 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет и представьте его для защиты лабораторной работы № 4. В ходе защиты отчета необходимо быть готовыми к демонстрации, установленной на компьютер, программы CoDeSys v2.3, разработанного проекта и к ответам на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) Назначение и особенности программного комплекса CoDeSys.

2) Назовите компоненты программного комплекса CoDeSys.

3) Перечислите языки программирования программного комплекса CoDeSys.

4) Текстовые языки МЭК, приведите пример.

5) Графические языки МЭК, приведите пример.

6) Поясните назначение базовых команд языка LD.

7) Поясните структуру LD-программы.

8) Как задают проверку состояния входных дискретных сигналов в LDпрограмме?

9) Какими командами формируют выходные дискретные сигналы в LDпрограмме?

10) Как можно фиксировать выходные дискретные сигналы в LD-программе?

11) Как выполняется конфигурация входных и выходных переменных в LDпрограмме?

12) Каким образом проверяется правильность LD-программы?

13) Каким способом загружается код программы в ПЛК?

14) Каким образом можно запустить программу в ПЛК?

## 3.5 Лабораторная работа № 5. Создание визуализации проектов в среде программирования контроллеров CoDeSys

Цели работы: приобрести практические навыки по созданию проектов и их визуализации в среде программирования контроллеров CoDeSys; углубить и закрепить знания по использованию программного комплекса CoDeSys при решении проектных задач.

Время: 4 часа.

Задание:

– создать в среде CoDeSys проект блока управления движением транспорта на перекрестке;

– создать визуализацию проекта блока управления движением транспорта;

– подготовить и защитить отчет по лабораторной работе.

Примечание: в работе требуется создать программу управления движением транспорта на перекрестке, оборудованном светофорами для двух пересекающихся направлений движения; управляемые по времени процессы управления движением транспорта необходимо представить средствами языков стандарта МЭК 61131-3, комбинируя различные языки в среде программирования контроллеров CoDeSys.

Методические указания по выполнению задания.

3.5.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 5.

В начале занятия после инструктажа преподавателя получите допуск к работе, а также индивидуальное задание на работу № 5.

3.5.2 Создание в среде CoDeSys проекта блока управления движением транспорта на перекрестке [7].

Создайте программный компонент Program Organization Unit (POU). Для этого запустите программу CoDeSys и задайте команду File New.

Появится окно Target Setting (Настройки целевого оборудования), в котором выберите тип контроллера, например, ПЛК 150 IM. Все предложенные настройки ПЛК оставьте без изменения. В окне диалога определите первый POU (умолчанию он получит наименование PLC\_PRG). Установите тип этого POU – Программа. Языком программирования для данного POU выберите язык Continuous Function Chart (CFC).

В главном меню задайте команду Project  $\rightarrow$  Object  $\rightarrow$  Add (Проект  $\rightarrow$  Объект  $\rightarrow$  Добавить) для создания еще трех объектов:

– программы с именем SEQUENCE на языке Sequential Function Chart (SFC);

– функционального блока с именем TRAFFICSIGNAL на языке Function Block Diagram (FBD);

– функционального блока с именем WAIT на языке Instruction List (IL).

В POU TRAFFICSIGNAL требуется сопоставить стадии работы светофора соответствующим цветам. То есть нужно установить, что красный свет горит в красной стадии и в желто-красной стадии, желтый свет – в желтой и желто-красной стадии и т.д.

В WAIT требуется создать простой таймер, который на вход получает длину стадии в миллисекундах и на выходе по завершении заданного периода времени выдает состояние Истина.

SEQUENCE должен устанавливать, чтобы нужный цвет зажигался в нужное время и на заданный период времени.

PLC\_PRG должен вводить входной сигнал для включения светофора и цветовые команды каждой его лампы, связанные с соответствующими выходами аппаратуры.

В редакторе объявлений для POU TRAFFICSIGNAL необходимо определить входную переменную (между ключевыми словами VAR\_INPUT и END\_VAR) по

53

имени STATUS типа INT. Переменная STATUS должна иметь четыре состояния, которые определяют соответствующие стадии – зеленую, желтую, желто-красную и красную.

Так как блок TRAFFICSIGNAL имеет три выхода, то необходимо определить еще три переменные: RED, YELLOW и GREEN (рисунок 5.1).

🧶 TRAFFICSIGNAL (FB-FBD)	
0001 FUNCTION_BLOCK TRAFFICSIGNAL	
0002VAR_INPUT	
0003 STATUS:INT;	
0004 END_VAR	
0005VAR_OUTPUT	
0006 GREEN:BOOL;	
0007 YELLOW:BOOL;	
0008 RED:BOOL;	
0009END_VAR	
0010VAR	
0011 OFF: BOOL;	
0012 END_VAR	
0013	
	· ·

Рисунок 5.1 – Раздел объявлений блока TRAFFICSIGNAL

Для программирования блока TRAFFICSIGNAL перейдите в раздел кода POU, выберите первую цепь и дайте команду Insert Box. В первой цепи будет вставлен прямоугольник с оператором AND и двумя входами. Замените текст AND на EQ.

Замените три знака вопроса около верхнего из двух входов на имя переменной STATUS. Для нижнего входа вместо трех знаков вопроса нужно поставить 1. Щелкните теперь на месте позади прямоугольника EQ, то есть выбран выход EQ.

Выполните команду Insert Assignment и измените три вопроса на GREEN. Переменная STATUS сравнивается с 1, а результат присваивается GREEN. Таким образом, GREEN будет включен, когда переменная STATUS равна 1.

Для других цветов блока TRAFFICSIGNAL требуются еще две цепи. Создаете их командой Insert Network (after). Законченный РОU показан на рисунке 5.2.



Рисунок 5.2 – Блок TRAFFICSIGNAL

Чтобы вставить оператор перед входом другого оператора выделите сам вход, а не текст, и далее используйте команду Insert Operator.

Таким образом, заканчивается программирование первого POU. Как и требовалось блок TRAFFICSIGNAL будет управлять включением выходов, по значениям переменной STATUS.

Для создания таймера в POU WAIT будет необходим POU из стандартной библиотеки. Откройте менеджер библиотек командами Window  $\rightarrow$  Library Manager (Окно  $\rightarrow$  Менеджер библиотек) и задайте команду Insert  $\rightarrow$  Additional library (Вставка  $\rightarrow$  Добавить библиотеку...). Откроется окно выбора файлов, в котором выберите из списка библиотеку standard.lib.

Теперь вернемся к POU WAIT, который будет работать таймером, задающим длительность стадий блока TRAFFICSIGNAL. POU WAIT должен иметь входную переменную TIME типа TIME и генерировать на выходе двоичную (Boolean) переменную, например, с именем ОК. Данная переменная должна принимать значение TRUE, когда заканчивается заданный период времени. Предварительно установите переменную TIME в FALSE в конце строки объявления « : = FALSE».

Далее будет нужен генератор времени РОU ТР, который имеет два входа (IN, PT) и два выхода (Q, ET). Генератор ТР должен выполнять следующие функции:

– пока вход IN установлен в FALSE выход ЕТ будет в состоянии 0, а выход Q будет FALSE;

– как только вход IN переключится в TRUE, выход ET начнет отсчитывать время в миллисекундах;

– когда выход ЕТ достигнет значения заданного входом РТ, счет будет остановлен;

– тем временем выход Q равен TRUE пока выход ET меньше входа PT;

– как только выход ЕТ достигнет значения входа РТ, выход Q снова переключится в FALSE.

Для использования генератора ТР в POU WAIT создайте его локальный экземпляр. Для этого объявите локальную переменную ZAB (отсчитанное время) типа ТР (между ключевыми словами VAR, END\_VAR). Вид раздела объявлений WAIT показан на рисунке 5.3.

Для программирования WAIT сначала проверьте, установлен ли выход Q в TRUE (возможно отсчет уже запущен). В этом случае не нужно изменять установки ZAB, надо только вызвать функциональный блок ZAB без входных переменных. Это необходимо для проверки завершения заданного периода времени (рисунок 5.4).

Установите переменную IN ZAB в FALSE и одновременно выход ET в состояние 0, а выход Q в FALSE. То есть установите все переменные в начальное состояние.

🌏 WAIT (FB-IL)	- O ×
0001 FUNCTION_BLOCK WAIT	
0002VAR_INPUT	
0003 TIME_IN:TIME;	
0004 END_VAR	
0005VAR_OUTPUT	
0006 OK:BOOL:=FALSE;	
0007 END_VAR	
0008VAR	
0009 ZAB:TP;	
0010 END_VAR	
0011	
	•

Рисунок 5.3 – Раздел объявлений блока WAIT

🍤 WAI	T (FB-IL	)	-OX
FU	JNCTIO	N_BLOCK WAIT	<u></u>
0001			
0002	LD	ZAB.Q	
0003	JMPC	mark	
0004			
0005	CAL	ZAB(IN:=FALSE)	
0006	LD	TIME_IN	
0007	ST	ZAB.PT	
0008	CAL	ZAB(IN:=TRUE)	
0009	JMP	end	
0010			
0011 m	ark:		
0012	CAL	ZAB	
0013 er	nd:		
0014	LDN	ZAB.Q	
0015	ST	OK	
0016	RET		
0017			
0018			
-0010			
-	II		<u> </u>

Рисунок 5.4 – Текст программы блока WAIT

Далее определите требуемое время TIME переменной PT и вызовите ZAB с IN: = TRUE. Функциональный блок ZAB будет работать до тех пор, пока не достигнет значения TIME и не установит выход Q в FALSE.

Инвертированное значение выхода Q будет сохраняться в переменной OK после каждого выполнения блока WAIT. Как только выход Q станет FALSE, переменная OK примет значение TRUE.

Для создания программы SEQUENCE объявите переменные: входную переменную START типа BOOL, две выходных переменных TRAFFICSIGNAL1 и TRAFFICSIGNAL2 типа INT и одну типа WAIT DELAY (рисунок 5.5).



Рисунок 5.5 – Объявления в программе SEQUENCE

Необходимо учитывать, что при создании SFC-диаграммы SFC-граф первоначально состоит из этапа Init, перехода Trans0 и возврата назад к Init. Поэтому SFC-граф придется дополнить.

Прежде чем программировать конкретные этапы и переходы, требуется выстроить структуру графа. Для этого понадобятся этапы для каждой стадии TRAFFICSIGNAL. Вставьте их, отмечая Trans0 и задав команду Insert  $\rightarrow$  Step transition (after). Повторите три раза данную процедуру.

Для редактирования названия нужного перехода или этапа щелкните на соответствующем тексте левой кнопкой мыши. Первый переход после Init назовите именем START, а все прочие переходы DELAY. Первый переход разрешается, когда START устанавливается в TRUE, все прочие – когда DELAY в переменной OK станет TRUE, то есть, когда закончится требуемый период.

Всем этапам (сверху вниз) дайте имена Switch1, Green2, Switch2, Green1.

Этап Init должен сохранить свое имя. Этап Switch2 должен включать желтую фазу. Во время этапа Green1 TRAFFICSIGNAL1 будет зеленым, а во время этапа Green2 TRAFFICSIGNAL2 будет зеленым.

В заключении измените адрес возврата Init на Switch1. В результате диаграмма примет вид как показано на рисунке 5.6.

Далее требуется запрограммировать все этапы. Двойным щелчком левой клавиши мыши на изображении этапа откройте диалог определения нового действия. В данном случае воспользуемся языком программирования IL.



Рисунок 5.6 – Диаграмма SEQUENCE

При программировании этапов и переходов следует учитывать, что во время действия этапа Init проверяется активность сигнала включения START.

Если сигнал не активен, то светофор выключается (рисунок 5.7).

🎭 Action Init (IL) - SEQUENCE (PRG-SFC)			
0001	LD	5	
0002	ST	TRAFFICSIGNAL1	
0003	LD	5	
0004	ST	TRAFFICSIGNAL2	
0005			
ann			
			,

Рисунок 5.7 – Программирование этапа Init

Для этого требуется записать число 5 в переменные TRAFFICSIGNAL1 и TRAFFICSIGNAL2.

Во время этапа Green1 TRAFFICSIGNAL1 будет зеленым (STATUS: = 1), TRAFFICSIGNAL2 будет красным (STATUS: = 3), задержка в 5000 миллисекунд (рисунок 5.8).



Рисунок 5.8 – Программирование этапа Green1

Этап Switch1 изменяет состояние TRAFFICSIGNAL1 на 2 (желтое) и, соответственно, TRAFFICSIGNAL2 на 4 (желто-красное). Кроме того, теперь устанавливается задержка в 2000 миллисекунд (рисунок 5.9).



Рисунок 5.9 – Программирование этапа Switch1

Этап Green2 включает красный в TRAFFICSIGNAL1 (STATUS: = 3) и зеленый в TRAFFICSIGNAL2 (STATUS: = 1). Задержка устанавливается в 5000 миллисекунд (рисунок 5.10).

Во время этапа Switch2 переменная STATUS в TRAFFICSIGNAL1 изменяется на 4 (желто-красный), соответственно, TRAFFICSIGNAL2 будет 2 (желтый) (рисунок 5.11). Задержка теперь должна быть в 2000 миллисекунд.



Рисунок 5.10 – Программирование этапа Green2



Рисунок 5.11 – Программирование этапа Green2

Теперь требуется проверить функционирование законченной программы в режиме эмуляции. Для этого откройте POU PLC\_PRG, вставьте в него компонент и замените AND на SEQUENCE. Все входы и выходы пока оставьте свободными.

Далее откомпилируйте Project Build и проверьте отсутствие в программе ошибок. В окне сообщений должен появиться текст: «0 Errors, 0 Warnings».

Теперь включите флажок Online  $\rightarrow$  Simulation (Онлайн  $\rightarrow$  Режим эмуляции) и дайте команду Online  $\rightarrow$  Login (Онлайн  $\rightarrow$  Подключение). Запустите программу на выполнение командой Online  $\rightarrow$  Run (Онлайн  $\rightarrow$  Старт).

Откройте программу SEQUENCE. После запуска программы она не будет работать, так как переменная START должна иметь значение TRUE. В последующем это будет делать PLC\_PRG, но сейчас надо изменить ее вручную. Для чего щелкните дважды левой кнопкой мыши по объявлению переменной START. После этого ее значение выделится цветом и станет равным TRUE.

Дайте команду Online → Write values (Онлайн → Записать значение) для записи значений переменных.

Теперь можно наблюдать за работой программы. Активные шаги диаграммы выделяются голубым цветом.

Для продолжения редактирования программы закройте режим Онлайн, выбрав команду Online → Logout (Онлайн → Отключение).

Таким образом, в блоке SEQUENCE определились два светофора, работа которых строго связана во времени.

Для завершения работы над программой нужно распределить входные и выходные переменные в блоке PLC\_PRG. Кроме того, для возможности запускать систему выключателем IN и для переключения всех шести ламп двух светофоров необходимо передавать команды переключения на каждом шаге SEQUENCE. Для этого объявите соответствующие Boolean переменные для всех шести выходов и одного входа, а затем создайте программу и сопоставьте переменные соответствующим адресам.

Далее в редакторе объявлений необходимо объявить переменные LIGHT1 и LIGHT2 типа TRAFFICSIGNAL (рисунок 5.12).

🍤 PLC_PRG (PRG-CFC)	_ 🗆 🗙
0001 PROGRAM PLC_PRG	
0002VAR	
0003 LIGHT1:TRAFFICS GNAL;	
0004 LIGHT2:TRAFFICS GNAL;	
0005 END_VAR	
0006	
•	•

Рисунок 5.12 – Объявление переменных LIGHT1 и LIGHT2

Для определения шести ламп светофоров нужны шесть переменных типа Boolean. Для этого используйте глобальные переменные (Global Variables) из ресурсов (Resources) и PLC Configuration.

Три переменные, управляющие лампами первого светофора, опишите в Global Variables как показано на рисунке 5.13.

Таким же образом должна быть определена двоичная входная переменная IN, необходимая для установки переменной START блока SEQUENCE в TRUE.



Рисунок 5.13 – Объявление глобальных переменных

Выберите вкладку Resources и откройте список Global Variables. Три переменные (L2\_GREEN, L2\_YELLOW, L2\_RED), управляющие лампами второго светофора, свяжите с реальными дискретными выходами ПЛК (рисунок 5.14).

fff PLC Configuration		_ 🗆 🗵
	Base parameters Module param	eters
L2_GREEN AT %QX1.0: BOOL; (* *) [CHANNEL (	Module id:	700
L2_YELLOW AT %QX1.1: BOOL; (* *) [CHANNEL L2_RED AT %QX1.2: BOOL; (* *) [CHANNEL (Q)]	Node id:	6
AT %QX1.3: BOOL; (* *) [CHANNEL (Q)]	Input address:	%IB6
⊕Special output(FIX)	Output address:	%QB6
Unifed signal sensor[SLOT]     Unifed signal sensor[SLOT]	Diagnostic address:	%MB6
⊞Unifed signal sensor[SLOT]	Comment:	
⊕—Analog output(FIX) ⊕—Analog output(FIX)		
۲ » (		

Рисунок 5.14 – Связывание переменных с выходами ПЛК

Для назначения имен выходам ПЛК дважды щелкните левой кнопкой мыши на слове «АТ» и введите имя.

Закончите PLC\_PRG.

Перейдите в окно редактора и выберите редактор Continuous Function Chart. После этого станет доступна соответствующая панель инструментов.

Щелкните правой кнопкой мыши в окне редактора и выберите элемент Вох. Щелкните на тексте AND и напишите SEQUENCE. Элемент преобразуется в SEQUENCE с уже определенными входными и выходными переменными. Далее вставьте два элемента и назовите их TRAFFICSIGNAL. Появившиеся три красных знака вопроса нужно заменить уже объявленными локальными переменными LIGHT1 и LIGHT2.

Далее создайте элемент типа Input, который получит название IN и шесть элементов типа Output, которым следует дать следующие имена: L1\_GREEN, L1\_YELLOW, L1\_RED, L2\_GREEN, L2\_YELLOW, L2\_RED.

Все элементы программы теперь на своих местах, и можно соединять входы и выходы. Для этого щелкните левой кнопкой мыши на короткой линии входа/выхода и тяните ее (не отпуская клавишу мыши) к входу/выходу нужного элемента. Программа должна принять вид, как показано на рисунке 5.15.



Рисунок 5.15 – PLC\_PRG

Теперь полностью завершенную программу необходимо проверить в режиме эмуляции. Убедитесь в правильности ее работы, контролируя последовательность выполнения и значения переменных в окнах редакторов CoDeSys.

3.5.3 Создание визуализации проекта блока управления движением транспорта.

С целью визуализации проекта нужно нарисовать два светофора и выключатель их блока управления.

Для создания визуализации в организаторе объектов выберите вкладку Visualizations. Выполните команду Project → Object Add и введите любое имя для визуализации, например Lights (рисунок 5.16).

New Visualization		×
Name of the new Visualization:	Ligths	ОК
		Cancel

Рисунок 5.16 – Создание новой визуализации

После нажатия кнопки ОК откроется окно для создания визуализации. В этом окне выберите команду Insert → Ellipse и нарисуйте окружность с диаметром около 2 см. Для чего щелкните левой кнопкой мыши на рабочем поле и, удерживая кнопку, растяните появившуюся окружность до требуемого размера. Для настройки элемента визуализации дважды щелкните на окружности левой кнопкой мыши.

В появившемся диалоговом окне выберите категорию Variables и в поле Change color введите имя переменной .L1\_RED (рисунок 5.17).

Рисунок 5.17 – Окно-1 конфигурации элемента визуализации

Глобальная переменная L1\_RED нужна для управления цветом нарисованной окружности.

Затем выберите категорию Color (рисунок 5.18), в ее области нажмите кнопку Inside и в появившемся окне выберите любой цвет, например, черный.

Regular Element Configuration (#1)	×
Category:          Shape       Color       Image: Color and the second sec	DK Incel

Рисунок 5.18 – Окно-2 конфигурации элемента визуализации

Нажмите кнопку Inside в области Alarm Color и выберите красный цвет. Полученная окружность будет черной, когда значение переменной ложно, и красной, когда переменная истинна. Таким образом, будет создан первый фонарь первого светофора.

Остальные цвета светофора создаются методом копирования. Для этого воспользуйтесь командой копирования Edit  $\rightarrow$  Copy (Ctrl + C) и командой вставки Edit  $\rightarrow$  Paste (Ctrl + V). В результате появятся две новые окружности, которые можно перемещать в нужные места экрана с помощью мыши. Расположите окружности в левой части окна редактора вертикально в ряд.

Настройте свойства соответствующих окружностей. В поле Change Color окон настройки свойств объектов введите для средней окружности переменную .L1\_YELLOW, для нижней окружности – .L1\_GREEN.

В категории Color в области Alarm color установите цвета окружностей (желтый и зеленый). Для того, чтобы нарисовать корпус светофора задайте команду Insert  $\rightarrow$  Rectangle. Вставьте прямоугольник так, чтобы введенные ранее окружности находились внутри него. Выберите цвет прямоугольника и командой Extras  $\rightarrow$  Send to back переместите его на задний план для того, чтобы окружности снова были видны.

Активизируйте режим эмуляции, поставив галочку перед пунктом меню Online  $\rightarrow$  Simulation.

Запустите программу путем выполнения команд Online  $\rightarrow$  Login и Online  $\rightarrow$  Run. Цвета светофора начнут меняться в соответствии с логикой его работы.

Создайте второй светофор с помощью самого простого способа его рисования – путем копирования всех элементов первого. Для этого воспользуйтесь командами Edit  $\rightarrow$  Copy и Edit  $\rightarrow$  Paste. Замените имена переменных, управляющих цветами, например, .L1\_RED на .L2\_RED. Таким образом будет создан второй светофор.

Для рисования переключателя ON вставьте прямоугольник, как было описано выше, установите его цвет и введите переменную .IN в поле Change Color категории Variables. В поле Content категории Text введите имя ON (рисунок 5.19).

Regular Element Configu	ration (#8)	×
Category: Shape Text Text variables Line width Colors Colorvariables Motion absolute Motion relative Variables Input Text for tooltip Security Programmability	Text Content: ON ? Horizontal C Left Center C Right Vertical C Top Center C Bottom Font Standard-Font	OK Cancel

Рисунок 5.19 – Окно-3 конфигурации элемента визуализации

Для того, чтобы переменная IN переключалась при щелчке левой кнопкой мыши на этом элементе в поле Toggle variable категории Input, введите переменную .IN (рисунок 5.20).

Regular Element Configu	iration (#8)	×
Category: Shape Text Text variables Line width Colors Colorvariables Motion absolute Motion relative Variables Input Text for tooltip Security Programmability	Input         Toggle variable       .IN         Tap variable          Tap FALSE          Zoom to vis.:          Execute program:          Text input of variable 'Textdisplay'          Text       Min:         Hidden       Max:         Dialog title:	OK Cancel

Рисунок 5.20 – Окно-4 конфигурации элемента визуализации

Созданный переключатель будет включать и выключать светофоры (рисунок 5.21).



Рисунок 5.21 – Визуализация проекта

Отобразить включенное состояние переключателя можно цветом также, как и для светофора. Для этого впишите переменную в поле Change Color.

3.5.4 Подготовка и защита отчета по лабораторной работе.

После выполнения заданий лабораторной работы № 5 подготовьте письменный отчет и представьте его для защиты. При защите отчета требуется продемонстрировать работу проекта и ответить на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) Дайте определение POU. Назовите типы POU, которые были использованы в программе.

2) Назовите языки программирования, которые использовались при написании программы.

3) Назовите переменные, которые были объявлены в секции VAR\_INPUT...END\_VAR раздела объявлений POU.

4) Назовите переменные, которые были объявлены в секции VAR OUTPUT... END VAR раздела объявлений POU.

5) Назовите переменные, которые были объявлены в секции VAR ... END VAR раздела объявлений POU.

6) Назовите способ объявления глобальных переменных проекта в CoDeSys.

7) Назовите способ связи переменных проекта CoDeSys с аппаратными входами и выходами ПЛК.

8) Охарактеризуйте назначение операторов LD, ST, CAL.

9) Охарактеризуйте назначение программы PLC\_PRG, ее отличия от других программ.

10) Назовите отличия функционального блока от программы в CoDeSys.

11) Охарактеризуйте назначение программы SEQUENCE.

12) Охарактеризуйте назначение функционального блока TRAFFICSIGNAL.

13) Охарактеризуйте назначение функционального блока WAIT.

14) Назовите библиотечный функциональный блок, который используется при написании функционального блока WAIT.

15) Назовите последовательность вызовов функций и функциональных блоков из программы PLC PRG.

16) Назначение компонента «Визуализация» в CoDeSys.

17) Назовите функции визуализации в реализованном проекте.

18) Назовите элементы визуализации, использованные в реализованном проекте?

## 3.6 Лабораторная работа № 6. Изучение среды программных пакетов диспетчерского управления и сбора данных Trace Mode 6

Цели работы: приобрести практические навыки по установке на персональный компьютер пакета Trace Mode 6; углубить и закрепить знания по использованию программных пакетов диспетчерского управления и сбора данных при решении профессиональных задач.

Время: 2 часа.

Задание:

– установить на персональный компьютер SCADA-пакет Trace Mode 6;

– изучить краткое описание системы Trace Mode 6;

– подготовить и защитить отчет по лабораторной работе.

Методические указания по выполнению задания.

3.6.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 6.

В начале занятия после инструктажа преподавателя получите допуск к работе.

3.6.2 Установка на персональный компьютер SCADA-пакета Trace Mode 6.

С официального сайта компании АдАстра [8] по ссылке <u>http://www.adastra.ru/products/dev/free\_SCADA/</u> скачайте инсталляционный файл бесплатной (после регистрации) программной системы Trace Mode 6 для автоматизации технологических процессов, телемеханики, диспетчеризации, учета ресурсов и автоматизации зданий. Запустите инсталляционный файл программы. В появившемся стартовом окне установки пакета Trace Mode 6 щелкните по кнопке Next (рисунок 6.1).



Рисунок 6.1 – Стартовое окно установки пакета Trace Mode 6

Появится окно, в котором надо принять лицензионное соглашение, выбрав I agree (рисунок 6.2).

🙀 Trace Mode IDE 6 Base		
License Agreement		
Please take a moment to read the Agree", then "Next". Otherwise o	a license agreement now. If you acce lick "Cancel".	pt the terms below, click "I
本产品的使用者或用户保	证已经读完本协议并理解了本	协议的内容。AdAstra 🔺
Research Group, Ltd俄多	9斯,107076, <mark>莫</mark> 斯科,a/ya 3	8
电话:007(095)737-5	9-33	
传真:007(095)232-0	0-92	
E-mail: ADASTRA@AD	ASTRA.RU,	
http://www.tracemode.co	om	-
🔘 I Do Not Agree	I Agree	
	Cancel	Back Next >

Рисунок 6.2 – Окно для принятия лицензионного соглашения

Для продолжения установки щелкните левой клавишей мыши по кнопке Next. В новом окне среди предложенных стран выберите Russian (рисунок 6.3).

Trace Mode IDE 6 Base			
Select Localization			TRACE MODE version
Available			
Russian			
C English			
Chinese 🖉			
	Cancel	< Back	Next >

Рисунок 6.3 – Окно выбора страны использования пакета

В следующем окне укажите папку для установки программы Trace Mode 6 (рисунок 6.4).

or click "Browse".
Browse
Disk Cost
iter:

Рисунок 6.4 – Окно выбора папки для установки программы
Выберите пункт Everyone (Все будут использовать программу) или Just me (Программу будет использовать только один пользователь). Для продолжения щелкните левой клавишей мыши по кнопке Next.

В окне Customer Information (рисунок 6.5) в поле Name введите имя пользователя, а в поле Organization – название организации.

🖟 Trace Mode IDE 6 Base	
Customer Information	TRACE MODE version
Enter your name and company or organization in the box be for subsequent installations. Name:	low. The installer will use this information
Admin	
Organization:	
Microsoft	
Cancel	< Back Next >

Рисунок 6.5 – Окно ввода сведений о пользователе

Снова щелкните левой клавишей мыши по кнопке Next. Начнется установка программного обеспечения, по завершению которого откроется соответствующее окно. Щелкнув левой клавишей мыши по кнопке Close этого окна завершите установку программного обеспечения.

3.6.3 Изучение краткого описания системы Trace Mode 6.

По ссылке <u>https://yadi.sk/d/Qfsbfvq3dtzNA</u> скачайте руководство пользователя Trace Mode 6 [9] и откройте его или откройте файл руководства в указанной преподавателем папке компьютера.

Изучите руководство пользователя в части следующих разделов:

- «Назначение и сферы применения»;

- «Основные термины и определения»;

- «Состав инструментальной системы»;

- «Принцип функционирования системы»;
- «Создание простейшего проекта».
- 3.6.4 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет и представьте его для защиты лабораторной работы № 6. В ходе защиты отчета необходимо быть готовыми к демонстрации установленного на компьютер пакета и к ответам на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

- 1) Дайте определение SCADA-системы.
- 2) Поясните основные понятия SCADA-систем.
- 3) Назовите основные функции SCADA-систем.
- 4) Назовите причины появления и развития SCADA-систем.
- 5) Поясните порядок установки SCADA-пакета Trace Mode 6.
- 6) Поясните назначение и сферы применения Trace Mode 6.
- 7) Назовите состав инструментальной системы Trace Mode 6.
- 8) Поясните принцип функционирования системы Trace Mode 6.
- 9) Поясните порядок создания простейшего проекта Trace Mode 6.

## 3.7 Лабораторная работа № 7. Создание проекта системы автоматизации и управления в инструментальной среде Trace Mode

Цели работы: приобрести практические навыки по созданию проекта системы автоматизации и управления в среде Trace Mode; закрепить и углубить знания по использованию программной среды при решении проектных задач.

Время: 4 часа.

Задание:

– в среде Trace Mode 6 создать проект системы автоматизации и управления;

– для разработанного проекта создать визуализацию;

– внести изменения в проект с учетом индивидуального задания, подготовить и защитить отчет по лабораторной работе.

Методические указания по выполнению задания.

3.7.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 7.

В начале занятия после инструктажа преподавателя получите допуск к работе и индивидуальное задание на работу № 7.

3.7.2 Создание в среде Trace Mode 6 проекта системы автоматизации и управления.

Разрабатываемый проект системы автоматизации и управления должен включать канал и генератор пилообразного сигнала [10]. В проекте необходимо провести привязку генератора к каналу и масштабирование сигнала в заданном диапазоне.

Для создания нового проекта запустите программу Trace Mode 6 и щелкните левой клавишей мыши на иконке <sup>№</sup> или выберите в меню команду Файл → Новый. В окне Навигатор проекта (рисунок 7.1) появятся элементы нового проекта.



Рисунок 7.1 – Новый проект Trace Mode

Сохраните созданный проект, щелкнув левой клавишей мыши на иконке или выберите в меню команду Файл → Сохранить или Сохранить как.

Для создания узла выделите в Навигаторе проекта строку Система (рисунок 7.2).



Рисунок 7.2 – Создание узла

Вызовите контекстное меню и в нем выберите строку Создать узел. Среди всех предлагаемых типов узлов выберите RTM. Навигатор проекта в разделе Система отобразит созданный узел. Для изменения имени узла надо выделить созданный узел RTM и щелкнуть левой клавишей мыши по нему.

Пример созданного узла RTM в окне Навигатора проекта приведен на рисунке 7.3.



Рисунок 7.3 – Отображение узла

Для создания канала выделите группу Каналы RTM узла. Вызовите контекстное меню и в нем выберите строку Создать компонент. Среди всех предлагаемых компонентов выберите Канал\_FLOAT (рисунок 7.4).



Рисунок 7.4 – Создание канала

Так как преобразование сигнала в канале будет производиться по формуле:

$$\mathbf{Y} = \mathbf{K} \cdot \mathbf{X} + \mathbf{Z},$$

где Ү – результат преобразования;

К – множитель;

Х – входное значение канала;

Z – смещение.

Требуется провести расчет К и Z.

Примем, что входное значение канала изменяется в диапазоне от 0 до 100 включительно. Тогда, чтобы обеспечить диапазон выходного сигнала от минус 10 до плюс 10, множитель К составит 0,2, а смещение Z составит минус 10.

Для редактирования созданного канала необходимо дважды щелкнуть левой клавишей мыши по имени Канал#1 созданного канала (рисунок 7.5).

Откроется окно Экран#1 для редактирования канала. В поле Имя введите новое имя канала. Установите флажок Использовать на панели Обработка.

Установите Апертуру равной 0, Пик равным 1, Сглаж. равным 0, а также вычисленные значения Множителя К и Смещения Z.

Убедитесь, что тип канала – Input (рисунок 7.6).



Рисунок 7.5 – Редактирование канала

Системные
Системные
Choronal Be
Основные
Tur
Размерность
Период Единица измерени
1 🚊 цикл САLС
Автопосылка
L DIGRAMTS
A
Мах Отработать
На старте

Рисунок 7.6 – Редактирование канала

Для генерации сигнала создайте группу Генераторы. Для этого выделите в Навигаторе проекта строку Источники/Приемники. Вызовите контекстное меню и в нем выберите строку Создать группу. Среди всех предлагаемых групп выберите группу Генераторы (рисунок 7.7).



Рисунок 7.7 – Создание группы для генераторов

Выделите созданную группу, вызовите контекстное меню и в нем выберите строку Создать компонент. Среди всех предлагаемых генераторов выберите требуемый тип генератора. В данном случае это тип Пила (рисунок 7.8).



Рисунок 7.8 – Создание генератора

Для привязки созданного генератора щелкните левой клавишей мыши по иконке 🚨. Откроется еще одно окно Навигатора проекта (рисунок 7.9).

В левой части верхнего Навигатора проекта выберите группу Каналы RTM узла. В левой части нижнего Навигатора проекта выберите группу Генераторы группы Источники/Приемники.



Рисунок 7.9 – Привязка генератора к каналу

Щелкните левой клавишей мыши по созданному ранее генератору. Не отпуская левой клавиши мыши, наведите курсор мыши на созданный канал и отпустите левую клавишу мыши. Иконка канала примет вид –

Закрыть одно из созданных окон навигатора проекта можно, щелкнув левой клавишей мыши по иконке 🗵.

3.7.3 Создание визуализации разработанного проекта.

В соответствие с заданием для визуализации проекта требуется создать экран и с помощью стрелочного прибора вывести на него отмасштабированный сигнал, тренд и текст.

Для создания экрана выделите группу Каналы, вызовите контекстное меню и в нем выберите строку Создать компонент. Среди всех предлагаемых компонентов выберите строку Экран (рисунок 7.10).



Рисунок 7.10 – Создание экрана

Откройте окно для редактирования шаблона экрана (рисунок 7.11), дважды щелкнув по созданному экрану левой клавишей мыши.

Для размещения объекта «стрелочный прибор» щелкните левой клавишей мыши по иконке 🐼. Если на панели инструментов нет этой иконки, а имеется только иконка 11., то щелкните правой клавишей мыши по ней. Среди предлагаемых объектов выберите стрелочный прибор (иконка 🐼) и щелкните левой клавишей мыши в том месте экрана, где должен располагаться один из углов этого прибора. Переместите курсор в противоположный угол расположения стрелочного прибора и щелкните левой клавишей мыши.

На экране будет создано изображение стрелочного прибора (рисунок 7.12).

Щелкните левой клавишей мыши по иконке выбора 💊 .



Рисунок 7.11 – Редактирование шаблона экрана



Рисунок 7.12 – Создание стрелочного прибора

Для настройки свойств созданного стрелочного прибора дважды щелкните левой клавишей мыши по нему.

Откроется окно Свойства объекта (рисунок 7.13).

Свойства объекта	×
Ø	Стрелочный прибор Справка
Свойство	Значение
Скрыть при старте	False
Отображаемая величина	50
Положение	По центру
Угол разворота	135
Заголовок	
Полоса	True
Шкала	True
Указатель	
Единицы	
Индикатор	True
Фон	
Рамка	

Рисунок 7.13 – Настройка стрелочного прибора

Раскройте раздел заголовка, сделав двойной щелчок левой клавиши мыши по подчеркнутой строке Заголовок. В появившемся поле Текст введите текст заголовка, который будет выведен на стрелочном приборе.

Двойным щелчком левой клавиши мыши по подчеркнутой строке Полоса раскройте раздел полосы.

Заполните поля окна Свойств объекта для стрелочного прибора, как показано на рисунке 7.14.

Для привязки объекта к каналу раскройте раздел Отображаемая величина и дважды щелкните левой клавишей мыши на подчеркнутой строке Отображаемая величина. Далее щелкните левой клавишей мыши в поле Привязка. Откроется окно свойств привязки с пустой таблицей.

Щелкните левой клавишей мыши на иконке <sup>5</sup>и установите тип IN у созданного аргумента ARG\_000.

Дважды щелкните левой клавишей мыши в столбце Привязка таблицы. Откроется окно привязки. В левой части открытого окна выделите Канал RTM узла, созданный ранее. В правой части окна выберите аргумент Реальное значение (рисунок 7.15).

()) ()	трелочный прибор Справка
Свойство	Значение
Скрыть при старте	False
Отображаемая величина	50
Положение	По центру
Угол разворота	135
<u>Заголовок</u>	
Текст	X
Цвет	
Шрифт	Arial,10
Полоса	True
Ширина	10
Верхний предел шкалы	10
P= <u>HL</u>	9.5
P <u>HA</u>	9
<u>► HW</u>	8
▶ <u>LW</u>	-8
P-LA	-9
▶ <u>LL</u>	-9.5
Нижний предел шкалы	-10
Цвет (HW, LW)	
Цвет (HA, HW), (LW, LA)	
Цвет (HL, HA), (LA, LL)	
Цвет >HL, <ll< td=""><td></td></ll<>	

Рисунок 7.14 – Пример заполнения полей свойств стрелочного прибора

<ul> <li>         Запалите Система     </li> </ul>		Атр	оибуты	Аргументы	
A ST RTM		ID	Полное	вимя	*
		0 Реальное значение			
<ul> <li>Каналы</li> <li>Ос. х</li> </ul>		1 Аппаратное значение			
		2 Входное значение			
FA	3 Состояние 4 Достоверность				
🖳 🥝 Экран					
			5 Период пересчета (значение)		
Источники/При	иемники	6 Тенденция 7 Интервал			
		8	Подклю	жение	
		9	Выходн	юе значение	*
		1			,
Атрибуты/Аргументы	Тип атрибита				

Рисунок 7.15 – Привязка

Щелкните левой клавишей мыши по кнопке Привязка, и окно Свойства привязки примет вид, как показано на рисунке 7.16.

è -	X	節	쿡쿳	素素	10	C	DA	Th.	86		
Maria	Tur	Tur									

Рисунок 7.16 – Результат привязки

При необходимости в поле Имя можно ввести новое имя канала аргумента. Щелкните по кнопке Готово левой клавишей мыши, и в поле Привязка будет выведено имя созданного аргумента.

Для создания тренда щелкните левой клавишей мыши по иконке 🛃. Если вместо нее на панели инструментов располагается другая иконка, например, тренд XY 🖾, архивный тренд 🕅 или архивная гистограмма 🏜, то щелкните правой клавишей мыши на соответствующей иконке и выберите тренд 🛃. Затем щелкните левой клавишей мыши по иконке 🗟

Разместите тренд в нужном месте экрана также, как это было сделано для стрелочного прибора.

Пример расположения тренда приведен на рисунке 7.17.

Настройте созданный тренд, для этого выделите его. В окне свойств тренда вначале будет открыта закладка основных свойств. В поле Заголовок введите подходящее название.

Перейдите на закладку Кривые (рисунок 7.18). Выделите строку Кривые, вызовите контекстное меню и в нем выберите Кривая.

В появившемся поле Имя введите подходящее имя для созданной кривой.

В поле Макс. значение введите значение верхней границы диапазона.

В поле Мин. значение – нижней границы диапазона.

Для привязки кривой щелкните левой клавишей мыши в поле Привязка, откроется окно Свойства привязки. Выберите созданный ранее аргумент и щелкните левой клавишей мыши по кнопке Готово.

Тренд примет вид, изображенный на рисунке 7.19.



Рисунок 7.17 – Создание тренда

Свойства объекта		×
	Тренд Справка	
R#		
Свойство	Значение	
Кривые	Коивая	
	- the second sec	

Рисунок 7.18 – Создание кривой

х				
X 9. 🗐	😭 🕑			
				<b>A</b>
1 A .				Ť
02.12.09 16:37:3	30 16:37:45	16:38:00	16:38:15	16:38:30
🔁 📰 🔛	🖇 💠 🔸		Þ	🄶 🦇 🏟
Кривые	Источник			Значение
	X			

Рисунок 7.19 – Созданный тренд

Для увеличения размеров графика наведите курсор на границу легенды и графика. Курсор примет форму двух параллельных линий со стрелками. Нажмите левую клавиши мыши и, не отпуская левой клавиши мыши, опустите курсор вниз и отпустите левую клавишу мыши.

Тренд примет вид, как показано на рисунке 7.20.



Рисунок 7.20 – Результат создания тренда

Для создания объекта Текст щелкните левой клавишей мыши на иконке **RBC**. Затем расположите объект Текст в нужном месте экрана так же, как это было сделано для стрелочного прибора и тренда. Рядом аналогично разместите еще один объект Текст (рисунок 7.21).



Рисунок 7.21 – Пример размещения объектов Текст

Для настройки объектов Текст выделите созданный объект Текст, расположенный слева. В поле Текст введите имя канала RTM узла. Выделите объект Текст, расположенный справа. Откройте раздел Текст двойным щелчком левой клавиши мыши по подчеркнутой строчке Текст. В поле Вид динамизации выберите Значение (рисунок 7.22).

Свойство	Значение
Контур	
Заливка	
Шрифт	MS Shell Dig.8
Зыравнивание	По центру
Гекст (X R)	<текст>
Вид индикации	Значение
Привязка	X_R
• Формат	Generic
Generic	%g
цвет текста	
Скрыть при старте	False
<u>Тодсказка</u>	
Выделение в МРВ	False

Рисунок 7.22 – Редактирование объекта Текст

Щелкните левой клавишей мыши в появившемся поле Привязка. Откроется окно Свойства привязки. Выберите созданный ранее аргумент и щелкните левой клавишей мыши по кнопке Готово.

Объекты Текст примут вид, как показано на рисунке 7.23.



Рисунок 7.23 – Объекты текст

Запустите проект, для чего щелкните левой клавишей мыши по иконке 🧖 или по строке Сохранить для МРВ в меню Файл.

В навигаторе проекта выделите созданный RTM узел и щелчком левой клавиши мыши по иконке 🐔 откройте профайлер. Если профайлер не запустился, то щелкните левой клавишей мыши по иконке 💘 или строке Открыть в меню Файл и запустите файл rtc.exe.

После запуска профайлера стрелка стрелочного прибора будет перемещаться, тренд строить график, а объект Текст выводить текущее значение созданного канала (рисунок 7.24).



Рисунок 7.24 – Результат запуска проекта

Внесите в разработанный проект изменения в соответствие с индивидуальным заданием и проведите его отладку.

3.7.4 Подготовка и защита отчета по лабораторной работе.

После выполнения заданий лабораторной работы № 7 подготовьте письменный отчет для его защиты. При защите отчета необходимо провести демонстрацию разработанного проекта и ответить на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

- 1) Поясните назначение и различие входных и выходных каналов.
- 2) Поясните назначение привязки и порядок ее проведения.
- 3) Поясните назначение навигатора проекта.
- 4) Поясните порядок импортирования изображения.
- 5) Поясните порядок создания компонентов базы каналов в Trace Mode.
- 6) Поясните порядок размещения объектов на экране.
- 7) Поясните назначение и функции окна свойств объекта.

## 3.8 Лабораторная работа № 8. Создание статического и динамического изображения в среде Trace Mode

Цели работы: приобрести практические навыки по работе со стандартными объектами Trace Mode, предназначенными для создания статических и динамических изображений; закрепить и углубить знания по использованию программной среды при решении проектных задач.

Время: 2 часа.

Задание:

– создать проект автоматизированной системы управления (АСУ) технической системой с визуализацией органов контроля и управления в Trace Mode 6;

 – создать статическое изображение элементов и структуры технической системы (объекта управления);

 – создать динамический объект для имитации технологического процесса объекта управления;

– с учетом индивидуального задания внести изменения в разработанный проект, подготовить и защитить отчет по лабораторной работе. Методические указания по выполнению задания.

3.8.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 8.

В начале занятия после инструктажа преподавателя получите допуск к работе и индивидуальное задание на работу № 8.

3.8.2 Создание проекта АСУ технической системой с визуализацией органов контроля и управления в Trace Mode 6.

В ходе работы необходимо создать новый проект. В данном проекте предусмотреть генератор, привязанный к каналу, значение которого определяется уровнем продукта (жидкости) в емкости [10]. Для визуализации проекта необходимо создать экран, на котором расположить тренд, отражающий зависимость уровня жидкости от времени.

По аналогии с предыдущей лабораторной работой в среде Trace Mode 6:

- создайте канал Уровень, показывающий уровень жидкости в емкости;

 – создайте генератор синусоидального сигнала и осуществите его привязку к каналу Уровень;

 – создайте экран и тренд, настройте кривую и осуществите их привязку к каналу Уровень (рисунок 8.1).

3.8.3 Создание статического изображения элементов и структуры технической системы (объекта управления).

В ходе работы необходимо создать статическое изображение емкости для жидкости в разрезе, трех труб насоса. Жидкость по одной трубе должна поступать в емкость, а по другой вытекать из нее. Вторая труба должна быть соединена с третьей трубой через насос.

Для создания статического изображения сначала создайте рамку, используя иконку . Если на панели инструментов этой иконки нет, то щелкните правой клавишей мыши по одной из иконок: , или . Далее выберите иконку рамки . С помощью мыши задайте противоположные углы рамки.

Для перехода в режим редактирования воспользуйтесь иконкой 🗟.

Пример созданной рамки показан на рисунке 8.2.



Рисунок 8.1 – Создание нового проекта

Для создания емкости воспользуйтесь иконкой **О**. Если на панели инструментов такой иконки нет, то щелкните правой кнопкой мыши по одной из иконок: **I**, **O**, **A**, **O**, **I**, **A**, **A**, **A**, **A**, **A** или **I**. Среди всех предложенных объектов выберите иконку емкости **O**.

Созданную емкость разместите на экране, задав противоположные углы емкости с помощью мыши. Пример размещения емкости показан на рисунке 8.3.

Выделите полученную емкость, откройте окно Свойства объекта и в поле Толщина стенок задайте толщину больше 0. Емкость будет изображена в разрезе, как показано на рисунке 8.4.

В поле Верхний и Нижний край выберите подходящие иконки, например **у** и **у**, для придания нужного вида краев емкости. В разделе Материал выберите подходящий материал емкости, дважды щелкнув левой клавишей мыши по подчеркнутой строке Материал. В поле Выбрать из списка выберите значение True. В поле Материал выберите необходимый материал, например Хром.



Рисунок 8.2 – Размещение рамки для визуализации проекта



Рисунок 8.3 – Создание емкости



Рисунок 8.4 – Задание толщины стенок емкости

В поле Стандартная текстура выберите нужную текстуру, например Гравировка. При необходимости можно добавить и другие объемные фигуры.

Для размещения конуса воспользуйтесь иконкой **О** и выберите инструмент **Л**. Разместите на экране конус, задав противоположные углы прямоугольника, в который будет вписан конус, и щелкните левой клавишей мыши по иконке **N**.

Выделите созданный конус и в поле Толщина стенок задайте ту же толщину, что и у емкости. Раскройте раздел Материал и выберите значение True в поле Выбрать из списка. В поле Материал выберите подходящий материал, например, Олово. В поле Стандартная гравировка задайте гравировку, к примеру, Шлифовка. Емкость примет вид, как показано на рисунке 8.5



Рисунок 8.5 – Пример визуализации емкости

Далее создайте насос, для чего щелкните правой клавишей мыши по иконке **Л**. Среди предложенных вариантов выберите инструмент **Л**. Для размещения насоса на экране с помощью мыши задайте противоположные углы прямоугольника, в который будет вписан насос. Щелкните левой клавишей мыши по иконке **R**.

Выделите насос и раскройте раздел Материал. В поле Выбрать из списка установите значение True. Выберите необходимый материал, например Пластик черный. Выберите нужную форму насоса в поле Форма насоса.

Пример статического изображения емкости и насоса показан на рисунке 8.6.

Для создания трубы щелкните правой клавишей мыши по иконке 🧟. Среди предложенных вариантов выберите инструмент **Г**.



Рисунок 8.6 – Размещение емкости и насоса

Создайте трубу, по которой жидкость поступает в емкость. Для этого с помощью мыши отметьте местоположение начала трубы и все точки изгиба трубы. Положение конца трубы отметьте щелчком правой кнопки мыши.

Аналогично создайте трубу для жидкости, поступающей в насос и вытекающей из него.

Отредактируйте свойства каждой трубы: толщину трубы и необходимый цвет, например коричневый

Пример статического изображения труб и других элементов проекта показан на рисунке 8.7.



Рисунок 8.7 – Статическое изображение труб, емкости и насоса

3.8.4 Создание динамического объекта для имитации технологического процесса объекта управления.

В ходе выполнения проекта, используя графический файл, необходимо создать динамический объект для имитации заполнения емкости жидкостью. Вначале надо провести импортирование подходящего изображения из графического файла.

Для этого выделите строку Ресурсы навигатора проекта, вызовите контекстное меню и в нем выберите строку Создать группу. Среди предложенных вариантов выберите строку Картинки (рисунок 8.8).



Рисунок 8.8 – Выбор группы Картинки

Далее выделите созданную группу Картинки, вызовите контекстное меню и в нем выберите строку Создать компонент. Среди предложенных компонентов выберите строку Библиотека\_Изображений (рисунок 8.9).



Рисунок 8.9 – Выбор библиотеки изображений

Для импортирования изображения дважды щелкните левой клавишей мыши по созданной библиотеке изображений. Откроется пустое окно, в котором вызовите контекстное меню и щелкните строку Импортировать. Появится диалоговое окно, в котором откройте подходящий графический файл изображения. Далее создайте динамическую заливку, для чего воспользуйтесь иконкой многоугольник  $\square$ . Многоугольник создается с целью отображения в динамике заполнения жидкостью внутренней полости емкости. Если на инструментальной панели иконки  $\square$  нет, то щелкните правой клавишей мыши по одной из следующих иконок:  $\square, \square, \square, \square, \square$ . Среди предложенных вариантов выберите инструмент  $\square$ .

Для размещения многоугольника с помощью мыши отметьте один из углов многоугольника и все требуемые точки излома. Последний угол многоугольника отметьте щелчком правой клавиши мыши.

Пример создания многоугольника для заливки на статическом изображении проекта показан на рисунке 8.10.



Рисунок 8.10 – Пример создания многоугольника для заливки

Выделите созданный многоугольник и раскройте раздел Заливка. В появившемся поле Стиль выберите строку Без заливки.

Перейдите на закладку Динамическая заливка 🏫 и поставьте флажок Разрешено. Раскройте раздел Слой и в появившемся поле Имя введите имя для слоя.

Произведите привязку слоя к аргументу, посредством которого произведена привязка тренда к каналу, хранящему значение уровня жидкости в емкости.

В поле Тип заливки выберите Изображение и щелкните в поле Изображение. Откроется окно, предлагающее выбор изображений, хранящихся в библиотеке изображений, созданной ранее (рисунок 8.11).

📓 Выбрать рес	урс	<u> </u>
Библиотека_Изоб	ражений	•
10000		
under 0		
Water U		
Готово	Отмена	Очистить
	19	E

Рисунок 8.11 – Выбор изображения для заливки

Выберите подходящее изображение и щелкните кнопку Готово. В поле Макс. установите значение верхней границы диапазона значений, хранимых в канале, а в поле Мин. – нижней границы.

Аналогично предыдущей работе запустите проект.

Пример исполнения созданного проекта приведен на рисунке 8.12.

Внесите в разработанную программу изменения в соответствие с индивидуальным заданием и проведите отладку приложения.

3.8.5 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет и представьте его для защиты лабораторной работы № 8. В ходе защиты отчета необходимо быть готовыми к демонстрации разработанного проекта и к ответам на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) Назовите отличия статического изображения от динамического в Trace Mode.

2) Поясните порядок создания статического изображения в Trace Mode.

3) Поясните порядок создания динамического изображения в Trace Mode.

4) Назовите элементы изображения, которые могут быть динамическими в Trace Mode.



Рисунок 8.12 – Пример исполнения проекта

## 3.9 Лабораторная работа № 9. Программирование в среде Trace Mode на языках Texno ST и Texno FBD

Цели работы: приобрести практические навыки по совершенствованию проекта системы АСУ технологическим процессом с использованием программной обработки в среде Trace Mode; закрепить и углубить знания по использованию языков Texno ST и Texno FBD при решении проектных задач.

Время: 2 часа.

Задание:

– в среде Trace Mode усовершенствовать проект АСУ технологическим процессом;

– реализовать дополнительные функции АСУ с использованием языков Texno FBD и Texno ST;

– внести изменения в разработанный проект с учетом индивидуального задания, подготовить и защитить отчет по лабораторной работе.

Методические указания по выполнению задания.

3.9.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 9.

В начале занятия после инструктажа преподавателя получите допуск к работе и индивидуальное задание на работу № 9.

3.9.2 Совершенствование проекта АСУ технологическим процессом в среде Trace Mode.

В процессе работы требуется усовершенствовать проект АСУ, разработанный в предыдущей лабораторной работе.

В проекте АСУ необходимо [10]:

 добавить каналы для передачи значений стоимости продукта (жидкости), расхода, суммарного расхода, суммарной стоимости продукта, периода генерации сигнала;

– удалить генератор, который использовался ранее;

 добавить кнопки для запуска ввода стоимости продукта, периода генерации уровня жидкости в емкости, расхода;

 добавить объекты Текст для вывода суммарного расхода продукта, стоимости израсходованного продукта, стоимости продукта, периода генерации уровня жидкости в емкости, расхода и стоимости.

Примечания:

1) Флаг Использовать на панели Обработка должен быть снят для всех каналов.

101

 Для всех каналов должен быть установлен одинаковый период 1 – 3 секунды.

Сохраните под новым именем проект, созданный при выполнении предыдущей лабораторной работы. Удалите генератор, который был привязан к каналу Уровень. Выделите объект Экран, откройте окно редактирования канала и выберите Период равным 1, единицу измерения – секунда. Аналогично внесите изменения в период пересчета канала, хранящего уровень жидкости в емкости.

Добавьте каналы, хранящие значения стоимости продукта, расхода продукта, периода генерации. Назовите канал, хранящий стоимость продукта, например, как Стоимость; канал, хранящий расход продукта – как Расход; канал, хранящий период генерации – как Период генерации. Установите тип каждого канала – Input.

Добавьте каналы, хранящие результаты вычислений: суммарный расход продукта, суммарная стоимость продукта. Назовите канал, хранящий суммарную стоимость продукта, как Суммарная стоимость, а суммарный расход продукта – Суммарный расход. Установите тип указанных каналов – Intput. Для всех каналов установите одинаковый период пересчета значения. Для этого вызовите окно редактирования для каждого канала и в поле Период установите период пересчета равным 1 и единицу измерения – секунда.

Для создания кнопок, задающих необходимые параметры, воспользуйтесь иконкой . Если данная иконка отсутствует на панели инструментов, то щелкните правой клавишей мыши по одной из иконок: **1** или **2**.

Задавая противоположные углы кнопок, разместите три кнопки на экране (рисунок 9.1).

Для каждой кнопки откройте окно свойств и введите назначение кнопки в поле Текст. Так, для кнопки, которая устанавливает период генерации можно ввести текст «Установить период»; для кнопки, которая устанавливает расход продукта – «Установить расход»; для кнопки, которая устанавливает стоимость продукта – «Установить стоимость».

102



Рисунок 9.1 – Пример размещения кнопок на экране

Перейдите на закладку Действия 🥌. Выделите строку mousePressed, вызовите контекстное меню и в нем выберите строку Передать значение (рисунок 9.2).

3	
Свойство	Значение
Код доступа	0
События	
MousePressed	Передать значение
mouseReleased	Показать/скрыть элементы
	Перейти на экран

Рисунок 9.2 – Создание события для передачи значения

В появившемся поле Тип передачи установите Ввести и передать. В поле Результат проведите привязку к входному значению канала, которому будет передаваться вводимое значение.

Пример настройки кнопки приведен на рисунке 9.3.

В результате привязки будут созданы четыре аргумента в таблице окна Свойства привязки (рисунок 9.4).

Для вывода значений, хранимых в каналах проекта, создайте объекты Текст. Пример создания объектов Текст изображен на рисунке 9.5.

Свойст	rB0	Значение
Код доо	ступа	0
Событи	R	
	usePressed	
	Подтверждение	False
	Сигнал	False
4	Передать значение	
	Тип передачи	Ввести и передать
	Значение	
	Результат	Стоимость_In
	Источник	
	Восстанавливать значение	False
P mou	useReleased	

Рисунок 9.3 – Настройка события для передачи значения

🍋 • 🗙 🔞 罌	話 り	e 🗈 🗈	A 🔽 🖌 第	
Имя	Тип 1	Тип данных	Привязка	Фла
Уровень_R	🛃 IN 🚦	REAL	Уровень:Реальное значение (Система.RTM.Каналы)	
Период_генерации_In	🔒 ООТ 🖁	REAL	Период генерации:Входное значение (Система.RTM.Каналы)	
Расход_In	🔒 ООТ 🖁	REAL	Расход:Входное значение (Система.RTM.Каналы)	
Стоимость_In	🚹 OUT 🔒	REAL	Стоимость:Входное значение (Система.RTM.Каналы)	

Рисунок 9.4 – Атрибуты для ввода данных

Период генерации, с	<текст>	Установить период
Расход, л/с	<текст>	Установить расход
Стоимость руб/л	<текст>	Установить стоимость
Суммарный расход, л	<текст>	
Суммарная стоимость, руб	<текст>	

Рисунок 9.5 – Пример создания объектов Текст

В данном примере слева расположены объекты Текст, которые дают подсказку (где какой параметр выведен), а справа – значения, хранимые каналами. Привязку объектов Текст, расположенных в правом столбце, производите к реальным значениям соответствующих каналов. При необходимости для привязки создайте дополнительные аргументы, как показано на рисунке 9.6.

🍋 • 🗙 🔞 躍	器 ウ	e 🗅 🗈	▲ 【 】 第 第
Имя	Тип	Тип данных	Привязка
Уровень_R	N 🗗	88 REAL	Уровень:Реальное значение (Система.RTM.Каналы)
Период_генерации_In	1 OUT	REAL	Период генерации:Входное значение (Система.RTM.Каналы)
Pacxog_In	10UT	REAL	Расход:Входное значение (Система.RTM.Каналы)
Стоимость_In	10UT	REAL	Стоимость:Входное значение (Система.RTM.Каналы)
Период_генерации_R	<mark>l]</mark> ∎N	REAL	Период генерации:Реальное значение (Система.RTM.Каналы)
Pacxog_R	🛃 IN	REAL	Расход:Реальное значение (Система.RTM.Каналы)
Стоимость_R	IN 🛃	REAL	Стоимость:Реальное значение (Система.RTM.Каналы)
Суммарный_расход_R	N 🗗	REAL	Суммарный расход:Реальное значение (Система.RTM.Каналы)
Суммарная стоимость	R	<b>REAL</b>	Суммарная стоимость:Реальное значение (Система. RTM. Каналы)

Рисунок 9.6 – Атрибуты экрана

3.9.3 Реализация в проекте дополнительных функций АСУ с использованием языка Texno FBD.

В ходе дальнейшей работы необходимо реализовать дополнительные функции АСУ путем добавления в проект FBD диаграммы, генерирующей уровень жидкости в емкости. Если период генерации меньше 10, то программа должна установить период равный 10. Программа должна возвращать отмасштабированный сигнал, то есть сигнал, изменяющийся в заданном диапазоне.

Примечание: у программы должен быть установлен период 1 – 3 секунды.

Проведите масштабирование сигнала в FBD-программе по формуле:

$$Y = K \cdot X + C,$$

где *Y* – результат преобразования;

K – множитель;

- Х входное значение канала;
- С смещение.

Производящие генерацию сигнала FBD блоки должны возвращать сигнал X, изменяющийся в диапазоне от минус единицы до плюс единицы. Поэтому, если необходимо обеспечить результат преобразования Y от нуля до единицы, множитель K составит 0,5 и смещение C также составит 0,5.

Для создания FBD диаграммы выделите группу Каналы RTM узла и вызовите контекстное меню, в котором выберите строку Создать компонент. Среди предложенных вариантов выберите строку Программа (рисунок 9.7).



Рисунок 9.7 – Создание объекта программа

Выделив созданную программу, измените имя программы, например на «Синусоида». Выделите созданный объект-программу и вызовите контекстное меню, в котором выберите строку Редактировать. Откроется окно, аналогичное редактированию канала Float. В поле Период выберите значение 1 и выберите единицу измерения – секунда. После настройки периода пересчета окно редактирования канала можно закрыть.

Двойным щелчком левой клавиши мыши по созданной программе вызовите окно редактирования шаблона программы (рисунок 9.8).

В структуре программы выделите строку Аргументы и заполните появившуюся таблицу, в которой каждая строка – аргумент программы.

Установите тип In для аргументов, которые будут передавать данные в программу. Установите тип Out для аргументов, которые будут передавать данные из программы. Установите тип In/Out для аргументов, которые будут передавать в программу и из нее.



Рисунок 9.8 – Окно редактирования шаблона программы

На рисунке 9.9 приведен пример создания атрибутов программы для генерирования синусоидального сигнала.



Рисунок 9.9 – Атрибуты программы на языке Texno FBD

Выделите строку Программа# в окне Структура программы. Среди предложенных языков программирования выберите FBD диаграмму (рисунок 9.10).

Воспользуйтесь иконкой 🧈 или выберите строку Палитра FBD блоков в меню Вид, если палитра FBD блоков (рисунок 9.11) не открыта.

🎽 Выбор Языка Программирования	x
Выберите язык программирования функции:	
ST программа	
SFC диаграмма	
FBD диаграмма	
🔿 LD диаграмма	
IL программа	
Справка Принять Отме	на

Рисунок 9.10 – Окно выбора языка программирования



Рисунок 9.11 – Окно выбора FBD блоков

Аналогично разместите на рабочем поле FBD:

- блок Выбор из двух (закладка Выбор);
- блок Синусоидальный сигнал (закладка Синусоидальный сигнал);

– блок Масштабирование (закладка Арифметические).

Программа, генерирующая сигнал, изображена на рисунке 9.12.

Произведите в программе привязку блоков между собой. Для привязки выхода одного блока к входу другого наведите курсор на соответствующий выход (вход), нажмите левую клавишу мыши и, не отпуская левой клавиши мыши, наведите курсор на необходимый вход (выход), отпустите левую клавишу мыши.


Рисунок 9.12 – Программа на языке Texno FBD

Для привязки аргумента программы к соответствующему входу (выходу) выделите соответствующий вход (выход), вызовите контекстное меню и в нем выберите строку Привязать. Среди предлагаемых аргументов выберите необходимый аргумент.

Для создания константы выберите вход, вызовите контекстное меню и выберите строку Привязать. В появившемся поле вместо выбора аргумента программы введите константу. Согласно заданию на вход IN2 блока Больше или равно и IN0 блока Выбор из двух следует подавать константу 10. Вход IN2 блока Больше или равно и IN0 блока Выбор из двух следует привязать к аргументу, который передает период генерации в программу.

Значения констант К и С, вычисленные ранее, следует подать на соответствующие входы блока Масштабирование.

После создания программы ее необходимо откомпилировать. Для этого воспользуйтесь иконкой Компиляция Ш. После чего в окне Сообщения будет выведен результат компиляции.

При отсутствии ошибок будет выведено, к примеру: «Программа#1.tms compiled successfully». При наличии ошибки будет выведено, к примеру: «E0011 Синтаксическая ошибка». В последнем случае ошибку потребуется устранить.

4.9.4 Реализация в проекте дополнительных функций АСУ с использованием языка Texno ST.

В процессе работы необходимо реализовать в проекте дополнительные функции АСУ путем добавления программы ST, которая должна проверять значения стоимости и расхода продукта и производить расчеты. Если значение стоимости или расхода меньше 1, то программа должна устанавливать соответствующий параметр равным 1. Кроме этого, необходимо, чтобы программа производила расчет суммарного расхода продукта и суммарной стоимости израсходованного продукта.

Примечания:

1) У программы должен быть установлен период 1 – 3 секунды.

2) Вычислять суммарный расход следует по формуле:

Суммарный\_расход = Предыдущий\_расход + Расход × Период\_Обработки,

где Суммарный\_расход – аргумент программы, возвращающий значение суммарного расхода входному значению канала, который передает значение суммарного расхода;

Предыдущий расход – глобальная переменная, содержащая значение суммарного расхода, вычисленное при предыдущем вызове программы;

Период\_Обработки – аргумент, привязанный к аргументу Период пересчета канала, который передает значение расхода.

3) После вычисления суммарного расхода следует глобальной переменной Предыдущий\_расход присвоить вычисленное значение Суммарный\_Расход, суммарную стоимость продукта вычислить как произведение вычисленного значения суммарного расхода на стоимость продукта.

110

Для создания ST-программы создайте в группе Каналы RTM узла объект Программа. Название программы можно изменить, к примеру, на «Обработка». Установите период пересчета равным 1 секунде.

Откройте программу и в окне Структура программы выберите строку Аргументы. Создайте аргументы, которые будут передавать в программу расход продукта, стоимость продукта, период пересчета каналов, а также аргументы, которые будут возвращать из программы суммарный расход продукта и суммарную стоимость всего израсходованного продукта (рисунок 9.13).

🍅 • 🗙 🔞 躍	語って	D 🗈 🖊	▼ 第 第
Имя	Тип	Тип данных	Привязка
Расход	🔰 IN/OUT	REAL	Pacxog:Peanshoe значение (Система.RTM.Каналы)
Стоимость	🔰 IN/OUT	REAL	Стоимость:Реальное значение (Система.RTM.Каналы)
Т	📥 IN	REAL	Уровень:Период пересчета (значение) (Система.RTM.Каналы)
Суммарный_расход	🕇 out	REAL	Суммарный расход:Входное значение (Система.RTM.Каналы)
Суммарная_стоимость	out 🔁	REAL	Суммарная стоимость:Входное значение (Система.RTM.Каналы)

Рисунок 9.13 – Аргументы программы на языке Texno ST

Выберите строку Глобальные переменные. Аналогично созданию аргументов создайте глобальную переменную, которая будет хранить значение суммарного расхода, вычисленного при предыдущем вызове программы, указав Начальное значение равным нулю (рисунок 9.14).



Рисунок 9.14 – Глобальная переменная

Выделите строку Программа# и в открывшемся окне наберите ST-программу. При указанных выше именах аргументов и глобальной переменной текст программы приведен на рисунке 9.15.

```
- 🏋 🎉 🍌 🏷 🏷
🔊 (° 🕌 🗋 🚺 👪 Документ_1_In
PROGRAM
   VAR INOUT Packog : REAL; END_VAR
   VAR INOUT CTOMMOCTE : REAL; END VAR
   VAR INPUT T : REAL; END VAR
   VAR OUTPUT Суммарный расход : REAL; END VAR
   VAR_OUTPUT Cymmaphas_crowmocrь : REAL; END_VAR
   If Pacxog < 1 Then
       Pacxog = 1;
   End if;
   If CTOMMOCTE < 1 Then
       Стоимость = 1;
   End if;
   Суммарный расход = Предыдущий суммарный расход + Расход * Т;
   Предыдущий суммарный расход = Суммарный расход;
   Суммарная_стоимость = Суммарный расход * Стоимость;
END PROGRAM
```

Рисунок 9.15 – Программа на языке Texno ST

Проверьте программу, воспользовавшись иконкой 👪.

Запустите проект на исполнение по аналогии с предыдущими работами.

Пример исполнения усовершенствованного проекта АСУ приведен на рисунке 9.16.

В соответствии с индивидуальным заданием внесите в усовершенствованный проект изменения и проведите его отладку.

3.9.5 Подготовка и защита отчета по лабораторной работе.

После выполнения заданий лабораторной работы № 9 подготовьте письменный отчет и представьте его для защиты. В ходе защиты отчета необходимо продемонстрировать усовершенствованную версию проекта АСУ и ответить на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) Поясните порядок создания объекта Кнопка и ее редактирование.

2) Поясните порядок создания программы в среде Trace Mode.

3) Поясните порядок создания атрибутов программы и переменных.

4) Поясните правила создания числовых констант.

5) Поясните правила создания строковых констант.

6) Поясните назначение и синтаксис оператора If в языке ST.

7) Поясните назначение и синтаксис оператора Case в языке ST.

8) Поясните назначение и синтаксис оператора Whule в языке ST.

9) Поясните назначение и синтаксис оператора Repeat в языке ST.

10) Поясните назначение и синтаксис оператора For в языке ST.

11) Опишите FBD-программу и поясните ее работу.

12) Поясните выбор FBD блока и его размещение на рабочем поле.

13) Поясните привязку FBD блока к переменным, атрибутам и создание констант.



Рисунок 9.16 – Пример исполнения усовершенствованного проекта

## 3.10 Лабораторная работа № 10. Программирование в среде Trace Mode на языках Texno IL и Texno SFC

Цели работы: приобрести практические навыки по совершенствованию проекта системы АСУ технологическим процессом с использованием программной обработки в среде Trace Mode; закрепить и углубить знания по использованию языков Texno IL и Texno SFC при решении проектных задач.

Время: 2 часа.

Задание:

- усовершенствовать проект АСУ технологическим процессом;

– реализовать дополнительные функции АСУ с использованием языков Texno SFC и Texno IL;

– внести изменения в разработанный проект с учетом индивидуального задания, подготовить и защитить отчет по лабораторной работе.

Методические указания по выполнению задания.

3.10.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 10.

В начале занятия после инструктажа преподавателя получите допуск к работе и индивидуальное задание на работу № 10.

3.10.2 Совершенствование проекта АСУ технологическим процессом.

В процессе работы необходимо усовершенствовать проект, разработанный в предыдущей лабораторной работе. Для совершенствования проекта требуется [10]:

– удалить обе программы, написанные на языках Texno ST и Texno FBD;

– добавить выключатель на экране;

 – создать канал, к которому производится привязка выключателя, для определения суммарного расхода и суммарной стоимости продукта.

Сохраните под новым именем проект, созданный при выполнении предыдущей лабораторной работы. Удалите обе программы.

Для создания выключателя создайте в проекте еще один канал. Назовите новый канал, например, Выключатель.

114

Установите период пересчета равным 1 секунде.

Для рисования выключателя воспользуйтесь одной из следующих иконок переключателей: Щ, Щ, Щ, Щ, 🥥, 🥥, 🤍 на панели инструментов. Возможности всех приведенных переключателей одинаковы, поэтому можно выбрать любой из них.

После выбора инструмента с помощью мыши разместите выключатель в нужном месте экрана.

Пример размещения выключателя на экране приведен на рисунке 10.1.



Рисунок 10.1 – Пример размещения выключателя

Откройте окно свойств объекта Выключатель. Произведите настройку объекта, как показано на рисунке 10.2, где Выключатель\_b1 – аргумент, который служит для привязки к первому биту канала Выключатель (рисунок 10.3).

3	
Свойство	Значение
Привязка	Выключатель_b1
Вид индикации	Агд & Конст
Инверсия	False
Константа	0x1
Код доступа	0
Значение	Ox1
Подсказка	
Выделение в МРВ	False

Рисунок 10.2 – Настройка выключателя

Имя	Тип	Тип данных	Привязка
Уровень_R	🛃 IN	REAL	Уровень:Реальное значение (Система.RTM.Каналы)
Период_генерации_In	100 💼	👪 REAL	Период генерации:Входное значение (Система.RTM.Каналы)
Pacxog_In	100T	REAL	Расход:Входное значение (Система.RTM.Каналы)
Стоимость_In	100T	👪 REAL	Стоимость:Входное значение (Система.RTM.Каналы)
Период_генерации_R	🛃 IN	REAL	Период генерации:Реальное значение (Система.RTM.Каналы)
Расход_R	🛃 IN	REAL	Расход:Реальное значение (Система.RTM.Каналы)
Стоимость_R	🛃 IN	REAL	Стоимость:Реальное значение (Система.RTM.Каналы)
Суммарный_расход_R	🛃 IN	REAL	Суммарный расход:Реальное значение (Система.RTM.Каналы)
Суммарная_стоимость_R	🛃 IN	REAL	Суммарная стоимость:Реальное значение (Система.RTM.Каналы)
Выключатель_b1	MIN/OUT	BOOL 🔡	Выключатель:Бит 1 (Система.RTM.Каналы)

Рисунок 10.3 – Аргументы экрана

3.10.3 Реализация дополнительных функций АСУ с использованием языков Texno SFC и Texno IL.

В процессе работы требуется реализовать дополнительные функции АСУ путем создания программы на языке Texno SFC, которая будет содержать следующие шаги и переходы:

– первый шаг – выполнение проверок значения периода дискретизации и значения расхода, стоимости продукта на языке Texno IL; если период дискретизации окажется меньше 10, то установите значение по умолчанию (10), если расход или стоимость продукта окажутся меньше 1, то установите соответствующий параметр равным 1, измененные значения программа должна возвращать соответствующим каналам;

– второй шаг – генерирование уровня в емкости на языке Texno FBD аналогично предыдущей лабораторной работе без проверки периода генерации (он проверен на первом шаге); в качестве условия перехода следует записать константу true;

– третий шаг, выполняемый, когда выключатель возвращает 1, реализованный, с использованием языка Texno IL: вычисление суммарного расхода, суммарной стоимости продукта аналогично предыдущей лабораторной работе с использованием глобальной переменной для хранения предыдущего значения суммарного расхода, вычисленные значения суммарного расхода, суммарной стоимости программа должна возвращать соответствующим каналам; – четвертый шаг, выполняемый, когда выключатель возвращает 0, реализованный с помощью языка Texno IL: вычисляется суммарный расход продукта, результат вычисления не возвращается программой соответствующему каналу, а только присваивается глобальной переменной, предназначенной для хранения предыдущего значения суммарного расхода.

Примечание: период пересчета у всех каналов и программ должен быть один и тот же в пределах 1 – 3 секунды.

Для создания SFC диаграммы создайте объект Программа. Создайте атрибуты. Кроме ранее использовавшихся атрибутов, потребуется атрибут, передающий в программу реальное значение канала, который хранит состояние выключателя.

Пример создания атрибутов приведен на рисунке 10.4.

🍋 • 🗙 🔞 緊調	8 9 C	D 🗅 🦂	▼ 將 將	
Имя	Тип	Тип данных	Привязка	4
Уровень	10UT	REAL	Уровень:Входное значение (Система.RTM.Каналы)	ar aar
Период_генерации	🚮 IN/OUT	REAL REAL	Период генерации:Реальное значение (Система.RTM.Каналы)	
Расход	🎝 IN/OUT	REAL REAL	Расход:Реальное значение (Система.RTM.Каналы)	
Стоимость	🚮 IN/OUT	REAL REAL	Стоимость:Реальное значение (Система.RTM.Каналы)	
Период_пересчета	🚽 IN	REAL REAL	Уровень:Период пересчета (значение) (Система.RTM.Каналы)	
Суммарный_расход	10UT	neal 👪	Суммарный расход:Входное значение (Система.RTM.Каналы)	
Суммарная_стоимость	TUO 💼	REAL REAL	Суммарная стоимость:Входное значение (Система.RTM.Каналы)	)
Выключатель	🛃 IN	BOOL	Выключатель: Бит 1 (Система. RTM. Каналы)	

Рисунок 10.4 – Аргументы программы

Создайте глобальную переменную для хранения результата вычисления суммарного расхода при предыдущем вызове программы. Для этого выделите строку Программа# в структуре программы. Среди предложенных языков программирования выберите SFC диаграмму.

На рабочем поле отображается только один шаг – начальный шаг (рисунок 10.5).

Дважды щелкните по данному шагу левой клавишей мыши. Введите новое имя шага, например, Проверка исходных данных.



Рисунок 10.5 – Новая программа на языке Texno SFC

Для создания нового шага выделите единственный существующий шаг. Вызовите контекстное меню на иконке 🛅 или и выберите команду Создать шаг/переход. Будет создан новый шаг и переход. Измените их имя: шаг, к примеру, можно назвать как Генерирование сигнала, а переход – true.

Выделите шаг Генерирование сигнала и создайте новый шаг и переход. Созданный шаг и переход можно назвать, к примеру, как Полный расчет и Реальное значение выключателя.

Выделите шаг Генерирование сигнала и создайте новый шаг и переход. Новый шаг и переход можно назвать, к примеру, как Расчет только предыдущего значения суммарного значения и Реальное значение выключателя с отрицанием.





Рисунок 10.6 – Программа на языке SFC

Создайте следующие шаги.

Шаг 1 – создание шага Проверка исходных данных.

Для этого в структуре программы раскройте раздел SFC диаграмма. Раскройте раздел Шаги и переходы. Выделите строку Проверка исходных данных. Среди предложенных языков программирования выберите язык Texno IL.

При указанных выше именах атрибутов и глобальной переменной текст программы на языке программирования Texno IL, производящей проверку исходных данных, примет вид:

SFC STEP "Проверка исходных данных" VAR OUTPUT Уровень : REAL; END VAR VAR\_INOUT Период\_генерации : REAL; END\_VAR VAR\_INOUT Pacxog : REAL; END\_VAR VAR\_INOUT Стоимость : REAL; END\_VAR VAR\_INPUT Период\_пересчета : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_OUTPUT Суммарная\_стоимость : REAL; END\_VAR VAR INPUT Выключатель : REAL; END VAR GE Расход 1//проверка расхода JMPC Проверка стоимости//переход к проверке стоимости при расходе > 1 LD 1 ST Расход//Расход по умолчанию Проверка стоимости: GE Стоимость 1 JMPC Проверка периода//переход к проверке периода генерации, когда стоимость > 1 LD 1 ST Стоимость//Стоимость по умолчанию Проверка периода: GE Период генерации 10 JMPC конец//выход при периоде > 10 LD 10 ST Период генерации//Период генерации по умолчанию конец: END\_SFC\_STEP

Шаг 2 – создание шага Генерирование сигнала.

Для этого выделите строку Генерирование сигнала. Среди предложенных языков программирования выберите язык FBD. Создайте FBD программу, как показано на рисунке 10.7.



Рисунок 10.7 – FBD программа шага для генерации сигнала

Для создания перехода к созданному шагу выделите строку true в разделе Переходы. Среди предложенных языков программирования выберите ST.

Текст программы на языке ST примет вид:

SFC\_TRANSITION "true" FROM( INITIAL\_STEP ) TO(STEP\_1 ) VAR\_OUTPUT Уровень : REAL; END\_VAR VAR\_INOUT Период\_генерации : REAL; END\_VAR VAR\_INOUT Pacxoд : REAL; END\_VAR VAR\_INOUT Стоимость : REAL; END\_VAR VAR\_INPUT Период\_пересчета : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_OUTPUT Суммарная\_стоимость : REAL; END\_VAR VAR\_INPUT Выключатель : REAL; END\_VAR true END\_SFC\_TRANSITION

Шаг 3 – создание шага Полный расчет.

Для этого выберите строку Полный расчет. Среди предложенных языков

программирования выберите язык IL.

Текст программы на языке IL примет вид:

SFC\_STEP "Полный расчет" VAR\_OUTPUT Уровень : REAL; END\_VAR VAR\_INOUT Период\_генерации : REAL; END\_VAR VAR\_INOUT Pacxoд : REAL; END\_VAR VAR\_INOUT Стоимость : REAL; END\_VAR VAR\_INPUT Период\_пересчета : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_INPUT Выключатель: REAL; END\_VAR MUL Период\_пересчета Pacxod//вычисление расхода за период пересчета ADD Предыдущий суммарный расход//Нахождение суммарного расхода ST Предыдущий\_суммарный\_расход//присвоение вычисленного значения суммарного расчета

ST Суммарный\_расход

MUL Стоимость//вычисление суммарной стоимости

ST Суммарная\_стоимость//присвоение аргументу результата вычисления суммарной сто-имости

END\_SFC\_STEP

Для создания перехода к созданному шагу выделите строку Реальное значе-

ние выключателя. Среди предложенных языков программирования выберите язык

ST. Текст программы на языке ST примет вид:

SFC\_TRANSITION "Peaльное значение выключателя" FROM(STEP\_1) TO(STEP\_2) VAR\_OUTPUT Уровень : REAL; END\_VAR VAR\_INOUT Период\_генерации : REAL; END\_VAR VAR\_INOUT Pacxod : REAL; END\_VAR VAR\_INOUT Стоимость : REAL; END\_VAR VAR\_INPUT Период\_пересчета : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_OUTPUT Суммарная\_стоимость : REAL; END\_VAR VAR\_INPUT Выключатель : BOOL; END\_VAR Bыключатель END\_SFC\_TRANSITION

Шаг 4 – создание шага Расчет только предыдущего значения суммарного

расчета.

Для этого выделите строку Расчет только предыдущего значения суммарного

расчета. Среди предложенных языков программирования выберите язык IL.

Текст программы на языке IL примет вид:

SFC\_STEP "Pacчет только предыдущего значения суммарного pacxoda" VAR\_OUTPUT Уровень : REAL; END\_VAR VAR\_INOUT Период\_генерации : REAL; END\_VAR VAR\_INOUT Pacxod : REAL; END\_VAR VAR\_INOUT Стоимость : REAL; END\_VAR VAR\_INPUT Период\_пересчета : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_OUTPUT Суммарная\_стоимость : REAL; END\_VAR VAR\_INPUT Выключатель : REAL; END\_VAR MUL Период\_пересчета Pacxod//вычисление расхода за период пересчета ADD Предыдущий\_суммарный\_расход//Нахождение суммарного расхода END\_SFC\_STEP

Для создания перехода к созданному шагу выделите строку Реальное значение выключателя с отрицанием. Среди предложенных языков программирования выберите язык ST. Текст программы на языке ST примет вид:

SFC\_TRANSITION "Реальное значение выключателя с отрицанием" FROM(STEP\_1) TO(STEP\_3)

VAR\_OUTPUT Уровень : REAL; END\_VAR VAR\_INOUT Период\_генерации : REAL; END\_VAR VAR\_INOUT Pacxoд : REAL; END\_VAR VAR\_INOUT Стоимость : REAL; END\_VAR VAR\_INPUT Период\_пересчета : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_OUTPUT Суммарная\_стоимость : REAL; END\_VAR VAR\_INPUT Выключатель : BOOL; END\_VAR !Выключатель END\_SFC\_TRANSITION

Запустите проект на исполнение.

При обоих положениях переключателя должно происходить заполнение ем-кости жидкостью.

При одном из положений переключателя должен производиться вывод новых полученных значений суммарного расхода и стоимости продукта, при другом – новые значения суммарного расхода и стоимости не выводятся (рисунок 10.8).

Внесите в усовершенствованный проект АСУ изменения в соответствие с индивидуальным заданием и проведите отладку проекта.

3.10.4 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет и представьте его для защиты лабораторной работы № 10. В ходе защиты отчета необходимо быть готовыми к демонстрации разработанного проекта АСУ и к ответам на контрольные вопросы преподавателя.

Примерный перечень контрольных вопросов:

1) Поясните SFC программу и порядок ее работы.

2) Поясните порядок создания нового шага, цикла, параллельных шагов.

3) Перечислите и поясните операнды и модификаторы языка IL.

4) Поясните назначение аккумулятора в языке IL и его работу.

5) Поясните назначение и синтаксис операторов для обмена данными с аккумулятором.

6) Поясните назначение и синтаксис арифметических операторов.

7) Поясните назначение и синтаксис операторов перехода и вызова.

- 8) Назовите группы языков программирования в среде SCADA-систем.
- 9) Приведите примеры графических языков программирования.



Рисунок 10.8 – Пример исполнения программы

## 3.11 Лабораторная работа № 11. Создание интерфейса оператора и модели управления в инструментальной среде Trace Mode

Цели работы: приобрести практические навыки по созданию интерфейсов оператора и моделей управления технологическим процессом, по созданию отчетов тревог и архивов значений; закрепить и углубить знания по использованию инструментальной среды Trace Mode при решении проектных задач.

Время: 4 часа.

Задание:

- усовершенствовать проект АСУ технологическим процессом;

–внести изменения в разработанный проект АСУ с учетом индивидуального задания, подготовить и защитить отчет по лабораторной работе.

Методические указания по выполнению задания.

3.11.1 Инструктаж и проверка готовности обучающихся к выполнению лабораторной работы № 11.

В начале занятия после инструктажа преподавателя получите допуск к работе и индивидуальное задание на лабораторную работу № 11.

3.11.2 Совершенствование проекта АСУ технологическим процессом.

В процессе работы необходимо усовершенствовать проект, разработанный в предыдущей лабораторной работе.

В усовершенствованном проекте требуется [10]:

– заменить тренд на архивный тренд;

 настроить анализ границ канала, передающего уровень жидкости в емкости;

– создать словарь сообщений для канала, передающего уровень жидкости в емкости, настроить узел для создания отчета тревог;

 изменить динамическую заливку емкости на заливку цветом, настроить выбор цвета в соответствии с принадлежностью значения уровня тому или иному диапазону;

– настроить систему промышленного архивирования данных (СПАД-архив) для архивирования канала, хранящего уровень продукта.

Сохраните под новым именем проект, созданный при выполнении предыдущей лабораторной работы. Удалите ранее созданный тренд. Разместите на экране архивный тренд, который настройте на вывод уровня жидкости в емкости.

Измените период колебаний по умолчанию с 10 на 40. Шаг, производящий проверку исходных данных примет вид:

SFC\_STEP "Проверка исходных данных" VAR\_OUTPUT Уровень : REAL; END\_VAR VAR\_INOUT Период\_генерации : REAL; END\_VAR VAR INOUT Pacxog: REAL; END VAR VAR\_INOUT Стоимость : REAL; END\_VAR VAR\_INPUT Период\_пересчета : REAL; END\_VAR VAR\_OUTPUT Суммарный\_расход : REAL; END\_VAR VAR\_OUTPUT Суммарная\_стоимость : REAL; END\_VAR VAR\_INPUT Выключатель : BOOL; END\_VAR GE Pacxoд 1 ЈМРС Проверка стоимости LD 1 ST Расход Проверка стоимости: GE Стоимость 1 JMPC Проверка периода LD 1 **ST** Стоимость Проверка периода: GE Период генерации 10 ЈМРС Конец LD 40 ST Период генерации Конец: END\_SFC\_STEP

Для настройки контроля границ в группе Каналы RTM узла выделите канал

Уровень. Вызовите контекстное меню и в нем выберите строку Редактировать.

На панели Границы поставьте флаг Использовать и флаг Контроль границ, установите границы диапазонов, как показано на рисунке 11.1.

Границы	
🗸 Использ	овать
BΠ	1
BA	0.9
ВΓ	0.8
нг	0.2
HA	0.1
нп	0
Гистерезис	0
🗸 Контроль	праниц

Рисунок 11.1 – Задание границ канала

Для создания словаря сообщений выделите узел RTM и вызовите контекстное меню. В контекстном меню выберите строку Создать группу. Среди предложенных групп выберите строку Словари\_сообщений (рисунок 11.2).



Рисунок 11.2 – Создание группы для словарей сообщений

Выделите созданную группу Словари\_сообщений и вызовите контекстное меню. Среди предложенных словарей выберите Словарь\_для\_FLOAT (рисунок 11.3).



Рисунок 11.3 – Создание словаря сообщений

Вызовите контекстное меню на созданном словаре сообщений и откройте окно для редактирования словаря. В поле Имя введите имя словаря. Двойным щелчком левой клавиши мыши на каждой строке откройте окно для редактирования сообщения (рисунок 11.4).

Редактируя все сообщения, настройте словарь сообщений, как показано на рисунке 11.5.

Направление:	AR+G	-	Категория:	<m> Сообщение</m>
Редактировать	ь текст сообще	ения:		
Все в норме				

Рисунок 11.4 – Окно для задания сообщения

Імя	Для_уровня	
револи		
Направление	Категория	Текст
AR+G	<М> Сообщение	Все в норме
AR+G	<w> Предупреждение</w>	Возможно переполнение емкости
AR+G	<w> Предупреждение</w>	Продукт кончается
AR+G	<А> Тревога	Бак переполнен
AR+G	<А> Тревога	Продукта не достаточно
AR+G	<Е> Ошибка	Датчик не исправен
AR+G	<Е> Ошибка	Датчик неисправен

Рисунок 11.5 – Редактор словаря сообщений

Для редактирования узла выделите узел RTM, вызовите контекстное меню и выберите Редактировать. Перейдите на вкладку Отчет тревог/дамп/параметры. Произведите заполнение полей, как показано на рисунке 11.6.

Основные	Архивы	Отчет тревог / Дамп / Параметры	Таймауты	аймауты Дополнительно			
Отчет тре	вог	*					
Имя файла т		тревоги.bd					
Максимум записей 1000		1000	Состо	ояние	TRUE	-	
Формат	даты	%x %H:%M:%S	Счите	ывать при старте	TRUE	-	

Рисунок 11.6 – Редактирование узла на вкладке Отчет тревог/дамп/параметры

Перейдите на вкладку Архивы и произведите заполнение полей, как показано на рисунке 11.7.

Основные	Архивы	Отчет	тревог /	Дамп / Параметры	Таймауты	Дополнительн	10
СПАД 1							
Имя файл	а	data.rep					
Состояни	е архива	TRUE	-		Pas	мер файла, Мб	128
Кэш Период	сохранен	ия 300	Pase	ер,Мб 48 👘	Максимум оче	реди записей	450000

Рисунок 11.7 – Редактирования узла на вкладке Архивы

В окне редактирования канала Уровень задайте границы на вкладке Архивация настройте архивацию и создание отчета тревог, как показано на рисунке 11.8.

Имя Уро	ень Кодировка Т	C5
Комментарий		
Границы	Обработка	Системные
📝 Использовать	Использовать	Основные
вп	1 Апертура 0	Архивация
BA	.9 Пик 0	СПАД 1
ВГ	.8 Сглаж. 0	Регистратор
нг	.2 Множитель 0.01	
HA	.1 Смещение 0	Синур/Ламп
НП	0 🗸 Масштабирование	

Рисунок 11.8 – Редактирование канала Уровень

Для настройки динамической заливки емкости откройте окно свойств объекта Многоугольник. Для получения зависимости цвета заливки от состояния контролируемого процесса произведите настройку, как показано на рисунке 11.9.

-	🚸 🖸			
💶 Pa	зрешено			
Свой	ство	Значение		
Hanpa	вление	Вверх		
Слои				
<u>⊿</u> <u>Cr</u>	юй (Уровень R)			
	Имя	Уровень		
	Привязка	Уровень_R		
	Тип заливки	Цвет		
	Цвет заливки			
	Стиль заливки			
Макс.		1		
Мин.		0		
Мин =	НП, Макс = ВП	False		
Цвета	для диапазонов	True		
Преду	преждение			
Авари	я			
Вне гр	аниц			

Рисунок 11.9 – Настройка изменения цвета динамической заливки

Запустите проект на исполнение.

Результат запуска проекта на исполнение будет отличаться тем, что будет выводиться результат предыдущих запусков проекта, периодически будет происходить обновление тренда, который строит зависимость на основе данных архива.

Цвет заливки емкости будет определяться уровнем жидкости в емкости (рисунок 11.10).

Проанализируйте созданный отчет тревог, который находится в файловой системе компьютера в папке с именем узла, размещенной в папке проекта.

Внесите в усовершенствованный проект АСУ изменения в соответствие с индивидуальным заданием и проведите его отладку.

3.11.3 Подготовка и защита отчета по лабораторной работе.

По результатам выполнения всех заданий подготовьте письменный отчет и представьте его для защиты лабораторной работы № 11. В ходе защиты отчета необходимо быть готовыми к демонстрации разработанного проекта АСУ и к ответам на контрольные вопросы преподавателя.

)8 - )6 - )4 -	=			
12 - 0 - 11.10 17:51:00 Марин Кривье +	17:51:15	17:51:30	17.51.45	17.52
	Период генерации, о	40	Установить период	
ALCONE, MARKA	Расход лис	1	y grand bitts packag	
	Стримо ать руб/п	Ť	Установить стои не сть	

Рисунок 11.10 – Пример исполнения усовершенствованного проекта АСУ

Примерный перечень контрольных вопросов:

1) Охарактеризуйте отчеты в SCADA-системах.

2) Охарактеризуйте отчет тревог в среде Trace Mode.

3) Поясните порядок создания отчета тревог в среде Trace Mode.

4) Поясните назначение границ канала и порядок их анализа в среде Trace Mode.

5) Поясните назначение словаря сообщений в среде Trace Mode.

6) Поясните порядок изменения цвета динамической заливки.

7) Поясните назначение СПАД-архива и порядок его создания в среде Trace Mode.

8) Поясните отличия события от тревоги в среде Trace Mode.

9) Поясните назначение и особенности аналоговых тревог SCADA-систем.

10) Поясните назначение и особенности дискретных тревог SCADA-систем.

- 11) Поясните назначение и особенности отчета тревог.
- 12) Поясните порядок получения сведений о сгенерированных тревогах.
- 13) Назовите условия SCADA-систем, позволяющие сформировать отчет.
- 14) Поясните отчет в виде html-страницы.
- 15) Поясните табличную форму отчета.

## Список использованных источников

1 Трипкош, В.А. Компьютерные технологии управления в технических системах [Электронный ресурс] : электронный курс в системе Moodle / В.А. Трипкош; М-во науки и высш. образования Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования "Оренбург. гос. ун-т". – Оренбург : ОГУ. – 2019. – 8 с. – Режим доступа: <u>https://moodle.osu.ru/course/view.php?id=820</u>.

2 Стандарт организации. СТО 02069024. 101 – 2015. Работы студенческие. Общие требования и правила оформления [Электронный ресурс]. – Режим доступа: http://www.osu.ru/docs/official/standart/standart\_101-2015\_.pdf. – 85 с.

3 <u>http://www.minutemansoftware.com/downloads.asp</u> – сайт «Текущие загрузки GPSS World» (дата обращения: 8.09.2019 г.).

4 <u>https://docplayer.ru/27102104-Uchebnoe-rukovodstvo-po-sisteme-gpss-</u> world.html – учебное руководство по системе GPSS World (дата обращения: 8.09.2019 г.).

5 Боев, В.Д. Моделирование систем. Инструментальные средства GPSS World / В.Д. Боев. – Санкт-Петербург: BHV-Санкт-Петербург, 2004. – 368 с.

6 <u>https://store.codesys.com</u> – официальный сайт разработчика инструментального программного комплекса промышленной автоматизации CoDeSys компании 3S-Smart Software Solutions GmbH (дата обращения: 8.09.2019 г.).

7 Рыжов, Д.Р. Автоматизация контроля технологических процессов : лаб. практикум / Д.Р. Рыжов, И.В. Трифанов ; Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2016. – 68 с.

8 <u>http://www.adastra.ru</u> – официальный сайт компании АдАстра (дата обращения: 8.09.2019 г.).

9 Руководство пользователя Trace Mode 6. Быстрый старт. – Москва: AdAstra Research Group, Ltd; изд. шестое. – 2006, – 114 с.

10 Ефимов, И.П. SCADA-система Trace Mode : методические указания к лабораторным работам / И.П. Ефимов, Д.А. Солуянов. – Ульяновск: УлГТУ, 2010. – 158 с.