

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования**

**«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Н.А. Долгий, Е.П. Шамаев**

**Микропроцессорные системы автоматизации и управления**

Учебно-методическое пособие по курсовой работе для студентов бакалавриата,  
обучающихся по направлению подготовки

15.03.04 «Автоматизация технологических процессов и производств»

Калининград

Издательство ФГБОУ ВО «КГТУ»

2020

УДК 681.382

РЕЦЕНЗЕНТ

– Арунянц Г.Г., д.т.н., профессор кафедры систем управления и вычислительной техники Калининградского государственного технического университета

Долгий Н.А., Шамаев Е.П.

Микропроцессорные системы автоматизации и управления. Учебно-методическое пособие- Калининград: Издательство ФБГОУ ВО «КГТУ», 2020 г., 53 с.

Учебно-методическое пособие по курсовой работе «Микропроцессорные системы автоматизации и управления» для студентов бакалавриата содержит задание на курсовую работу, порядок выполнения курсовой работы, рекомендации по анализу объекта автоматизации, разработке алгоритма и программы управления.

Рис. 30, табл. 6, приложений 5, список лит.- 9 наименований.

Учебно-методическое пособие рассмотрено и одобрено кафедрой автоматизации производственных процессов ФБГОУ ВО «Калининградский государственный технический университет» 19 мая 2020 г., протокол № 7.

Учебно-методическое пособие рекомендовано к изданию методической комиссией факультета автоматизации производства и управления «10» сентября 2020 г., протокол № 1.

© Федеральное государственное бюджетное образовательное учреждение высшего образования «Калининградский государственный технический университет», 2020 г.

© Долгий Н.А., Шамаев Е. П., 2020

## Содержание

Введение.....	4
1.Выбор технологического объекта управления (ТОУ). .....	5
2. Задание на курсовую работу. ....	5
3. Методические указания по выполнению курсовой работы. ....	6
3.1. Анализ объекта автоматизации.....	6
3.2. Перечень входных и выходных сигналов объекта.....	10
3.3. Алгоритмы управления и функционирования. ....	13
3.4. Принципиальная электрическая схема.....	20
3.5. Разработка программы управления .....	21
3.6. Отладка программы на симуляторе.....	29
3.7. Визуализация проекта.....	30
4. Оформление пояснительной записки. ....	35
Список использованной литературы.....	37
Приложение А. ....	39
Приложение Б.....	41
Приложение В.....	42
Приложение Г.....	45
Приложение Д. ....	47
Приложение Е.....	52

## Введение

Целью курсового проектирования по дисциплине «Микропроцессорные системы автоматизации и управления» является практическое применение знаний, полученных при изучении теории, приобретение навыков в программировании, отладке программ управления в интегрированной среде разработки (Integrated Development Environment) приложений для программируемых логических контроллеров (ПЛК) CoDeSys (акроним английского Controller Development System).

Интегрированная среда разработки CoDeSys является freeware, т.е. бесплатно распространяемым программным обеспечением и доступна на официальном сайте фирмы OWEN [owen.ru](http://owen.ru). Для установки данной среды на компьютере возможен следующий путь: [owen.ru](http://owen.ru) => каталог продукции => среда программирования CoDeSys => CoDeSys V2 – подробнее => CoDeSys V2.3.9.41 (русифицированная версия) => Скачать.

В процессе проектирования студенты выполняют анализ технологического объекта, составляют список входных и выходных сигналов ПЛК, разрабатывают блок-схему алгоритма управления и принципиальную электрическую схему устройства управления, содержащую микроконтроллер и радиоэлементы, позволяющие получать и передавать сигналы с объекта управления. Учебно-методическое пособие составлено в соответствии с учебным планом по для студентов бакалавриата, обучающихся по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств» и учебной программой по указанной дисциплине. Учебно-методическое пособие содержит рекомендации по выбору и анализу объекта автоматизации, разработке алгоритма и программы управления, принципиальной электрической схемы системы управления объекта автоматизации.

Важным этапом работы является проверка разработанной в курсовой работе программы на симуляторе виртуального контроллера, входящего в состав среды CoDeSys.

Учебное пособие может быть использовано не только в курсовом проектировании по данной дисциплине, но и при выполнении выпускной квалификационной работы (ВКР) для студентов бакалавриата, обучающихся по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств» в разделах, связанных с моделированием и программированием микропроцессорных систем управления.

## 1. Выбор технологического объекта управления (ТОУ)

Студент может самостоятельно предложить тему курсовой работы (название ТОУ) для разработки программного обеспечения объекта автоматизации, например, связанную с его производственной деятельностью, выбранную по интернет-ресурсам или литературным источникам и согласовать ее с руководителем. Выбор ТОУ основывается, во-первых, на результатах производственной технологической практики, во-вторых, желательно, чтобы тема курсовой работы совпадала с темой ВКР. Руководитель может назначить тему объекта управления из списка тем, предложенных в Приложении А.

Рекомендуется выбирать такие объекты автоматизации, которые могут быть использованы в других курсовых проектах в следующих семестрах и при выполнении ВКР. Предпочтение следует отдавать темам, которые не использовались ранее при выполнении ВКР бакалавров кафедры АПП. Выбрав таким образом технологический объект (процесс), студенты уточняют состав и функционирование объекта по литературным источникам, интернет-ресурсам, на предприятиях рыбной и пищевой промышленности, на которых они проходили практику или работают (студенты заочного отделения).

## 2. Задание на курсовую работу

2.1. Выбрать технологический процесс для автоматизации.

2.2. Провести анализ технологического процесса.

2.3 Составить таблицы выходных и входных сигналов управляющего контроллера.

2.4 Разработать алгоритм управления объектом автоматизации и алгоритмы имитации ТОУ в виде блок-схемы.

2.5 Выбрать управляющий контроллер с необходимыми входными и выходными модулями.

2.6 Разработать принципиальную электрическую схему, отражающую соединения контроллера с датчиками и исполнительными механизмами объекта управления.

2.7 Разработать программу управления технологическим объектом.

2.8 Разработать программы имитации отдельных параметров технологического объекта.

2.9 Разработать мнемосхему объекта и произвести ее анимацию.

2.10 Произвести совместную отладку программы управления технологическим объектом и программ имитации параметров технологического объекта.

2.11 Оформить пояснительную записку и чертежи.

2.12 Пояснительная записка и чертежи сдаются руководителю в бумажном виде и в электронном виде.

2.13 Разработанный в среде CoDeSys проект защищается перед руководителем на персональном компьютере. Положительная оценка ставится в случае, если проект полностью работоспособен и на мнемосхеме отражаются все процессы выбранного технологического объекта.

### **3. Методические указания по выполнению курсовой работы**

После выбора темы курсовой работы и согласования с руководителем необходимо тщательно проработать источники информации по данному объекту управления (учебники, научно-технические журналы, интернет, техническая документация с производственной практики). Результатом данного анализа должны быть рисунки, фотографии, технологические схемы объекта управления и его описание.

Во введении следует описать место объекта управления в промышленности или сельском хозяйстве. Роль автоматизации для повышения производительности данного объекта, примеры существующих технологических объектов.

#### **3.1. Анализ объекта автоматизации**

Анализ объекта автоматизации следует начинать с описания технологического процесса. Определение функций, задач и технических средств автоматизации базируется на детальном изучении характеристик объекта автоматизации.

Современное предприятие характеризуется развитой производственной структурой, многотипностью связей между организационно-управленческими и технологическими подразделениями, многокритериальностью оценок результатов его работы. Управление такими объектами осуществляется по иерархическому принципу, в соответствии с которым система управления разделяется на ряд подсистем – уровней, находящихся между собой в определенной информационной, временной, функциональной взаимосвязи и подчиненности. Иерархический принцип построения системы управления (СУ) находит отражение в организационной структуре предприятия, включающей в себя следующие основные уровни управления: рабочее место – первый уровень управления, производственный участок – второй уровень, цех – третий уровень, предприятие – верхний уровень. Каждому уровню соответствуют свои объекты, являющиеся подсистемами производственно-технологической структуры предприятия.

Объектом автоматизации для задач, решаемых на первом уровне управления, является технологический модуль (ТМ), под которым понимается отдельный аппарат (дефростер, бланширователь, рыбообжарочная печь), машина (рыборазделочная, набивочная), робот или другое оборудование, предназначенное для выполнения по крайней мере одной технологической операции. Целью управления является оптимизация или обеспечение заданных режимов выполнения технологических операций. Цель достигается автоматизацией контроля и управления технологическим оборудованием в темпе технологического процесса.

Объектом автоматизации второго уровня является технологическая система, представляющая собой совокупность функционально взаимосвязанных технологических модулей, предметов производства и исполнителей, работающих в пределах одного производственного участка. Технологическими системами являются и технологические (поточные) линии, используемые в рыбной промышленности. На уровне участков и линий цель управления состоит в определении и обеспечении оптимального режима совместного функционирования (координации) элементов технологической системы. Управление на первых двух уровнях относится к управлению технологическим процессом. На уровне технологического процесса образуется информация, характеризующая фактический расход и движение по технологическому маршруту материалов, сырья и полуфабрикатов. Технологический процесс является базовым для всего материального производства и требует проведения входного, текущего, выходного и контроля.

На уровне цеха, как и на уровне участка, решаются задачи оперативного (диспетчерского) управления. Оперативное управление предполагает непрерывное и регламентированное слежение за состоянием ОУ. На верхнем уровне осуществляется управление предприятием как производственно-экономической системой.

Качество конечного продукта формируется в ходе всего технологического процесса, а контролироваться может в конце технологического цикла.

Результатом анализа объекта автоматизации в курсовой работе является технологическая схема и описание процесса его функционирования. Необходимо выделить все контролируемые, измеряемые и управляющие сигналы, которые должны быть сведены в табличном виде в перечень входных и выходных сигналов объекта.

Методику выполнения курсовой работы рассмотрим на примере проекта автоматизации поточной линии производства мороженого в вафельных стаканчиках [1].

Несмотря на значительное разнообразие в ассортименте, производство мороженого с некоторыми изменениями осуществляется по общей технологической схеме и состоит

из следующих операций: приемка сырья, подготовка сырья, составление смеси, пастеризация смеси, гомогенизация смеси, охлаждение и созревание смеси, фризирование смеси, фасование и закалывание мороженого, упаковывание и хранение мороженого. Технологическая схема производства мороженого в вафельных стаканчиках приведена на рисунке 1.

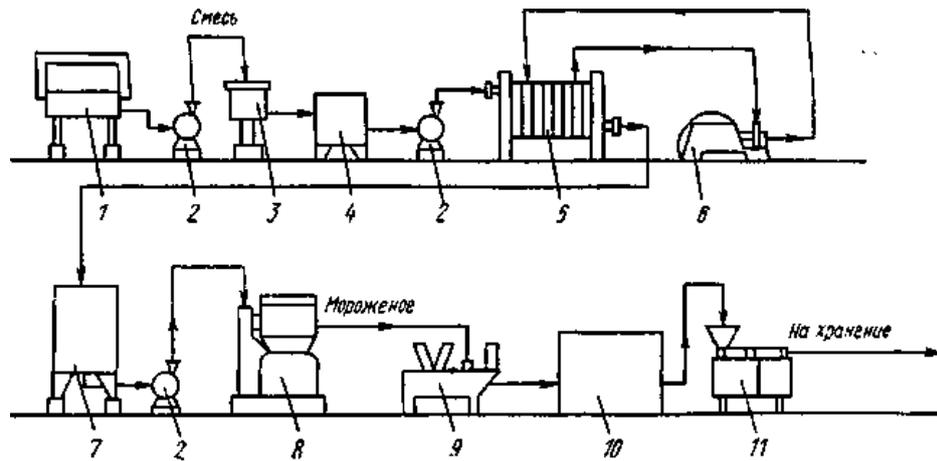


Рисунок 1 – Технологическая схема производства мороженого в вафельных стаканчиках  
 1 — ванна для приготовления смеси; 2 — насос; 3 — фильтр; 4 — уравнительный бак (уровень LE1); 5 — пластинчатая пастеризационно-охладительная установка (температура пастеризатора TE1, охладителя TE2); 6 — гомогенизатор; 7 — резервуар для смеси (уровень LE2); 8 — фризер (температура TE3); 9 — автомат для фасования мороженого в вафельные стаканчики; 10 — морозильный аппарат; 11 — автомат для закрутки мороженого.

Смесь мороженого готовят в ванне 1. Первыми в смесительную ванну вносят сгущенные продукты и сливочное масло, затем сухие компоненты: сахар, сухое молоко, порошок какао и яичные продукты. В последнюю очередь вносят стабилизаторы. Некоторые вкусовые вещества, а также ароматизаторы добавляют непосредственно перед фризированием в емкость 7.

Приготовленная смесь насосом 2 подается в фильтр 3 и через балансировочный бак 4 в пастеризационно-охладительную установку пластинчатого типа. При такой тепловой обработке уничтожаются все микроорганизмы, попавшие в смесь с компонентами, кроме этого, пастеризация способствует более полному растворению всех её составных частей. Смеси для мороженого имеют высокую вязкость. В связи с этим установлены более строгие режимы тепловой обработки смесей: длительная пастеризация - при 68 °С

с выдержкой 30 мин, кратковременная - при 75 °С с выдержкой 15 мин и моментальная - при температуре 85 °С с выдержкой 10-15 с.

Степень насыщения смеси воздухом оценивается по взбитости, которая представляет собой отношение объема воздуха в мороженом к первоначальному объему смеси, выраженное в процентах. Минимальная взбитость должна быть не ниже 50 % (молочное мороженое), 60% (сливочное мороженое и пломбир), 35 - 40 % (плодово-ягодное мороженое). Взбитость повышается при увеличении содержания СОМО (сухой обезжиренный молочный остаток), количества стабилизатора и дисперсности жира, а также при уменьшении содержания жира и сахара. Смеси, приготовленные с использованием сухих молочных продуктов, взбиваются лучше, чем с применением жидкого молока. Взбитость мороженого зависит также от конструктивных особенностей фризеров. В хорошо взбитом мороженом средний размер воздушных пузырьков не должен превышать 60—70 мкм. При взбитости 100 % в 1 г мороженого содержится около 8,3 млн. воздушных пузырьков с общей поверхностью 0,1 м<sup>2</sup>.

Мороженое, вышедшее из фризера, по консистенции и внешнему виду напоминает крем. После фризирования мороженое фасуется и замораживается (закаливается) от -15°С до -18 °С. Закаливание следует осуществлять интенсивно, чтобы не допустить увеличения размеров кристаллов льда более 60— 80 мкм.

Мороженое фасуется в брикеты на вафлях по 100 г, стаканчики из вафель и бумажные по 100 г, вафельные рожки по 100 г, пачки по 250 г и брикеты на палочке (эскимо) по 50 и 100 г. Торты из мороженого выпускают массой 0,25—3 кг, кексы 0,5—1 и пирожные 0,1 кг.

В процессе фризирования смеси происходит превращение свободной влаги в кристаллики льда размером (50÷100) мкм, насыщение мороженого воздухом, который распределяется по всей массе в виде пузырьков, в результате чего объем смеси увеличивается в 1,5...2 раза.

Выходящее из фризера мороженое с температурой (- 3...- 5 °С) необходимо быстро расфасовать (поз. 9) и направить на закалку в аппарат 10. Закаливание производят в специализированных камерах или скороморозильных аппаратах.

Готовое мороженое хранится в холодильных камерах при температуре воздуха (-20 ...- 23) °С. Допустимые колебания температуры не должны превышать ±2°С. Резкие колебания температуры мороженого приводят к укрупнению в нем кристаллов льда, в результате чего ухудшается его качество. Допустимая продолжительность хранения 1-1,5 мес (молочное мороженое), 1,5-2 мес (сливочное) и 2-3 мес (пломбир). Длительность

хранения снижается для мороженого с повышенным содержанием влаги, мелкофасованного и при наличии наполнителя.

### 3.2. Перечень входных и выходных сигналов объекта

Во всех инструментальных системах программирования ПЛК входными сигналами считаются сигналы, поступающие на входы контроллера, а выходными – на его выходы. Соответственно, сигналы датчиков объекта автоматизации будут считаться входными сигналами ПЛК, а управляющие сигналы на исполнительные механизмы объектов автоматизации, выходными. Сигналы по своему виду могут быть дискретными (булевыми), аналоговыми (целыми и вещественными), импульсными и передаваемыми по стандартным интерфейсам (например, RS-485, CAN), сигналами сообщений.

Для рассматриваемого примера линии производства мороженого заполним таблицу входных и выходных сигналов (таблица 1). Первые шесть столбцов таблицы заполняются по результатам анализа объекта автоматизации. Столбец «сигнал» заполняется после выбора датчика.

Цифровые ссылки на место установки датчика берутся с рисунка 1, а столбец модуль PLC заполняется после разработки принципиальной электрической схемы. Например, AI:27,28 – обозначает аналоговый вход ПЛК-150, контакты 27 и 28.

Имя переменной обозначается в соответствии с ее функцией. Имя переменной (идентификатор) не должно содержать пробелов и спецсимволов, не должно объявляться более одного раза и не должно совпадать с ключевыми словами. Регистр символов не учитывается, это означает, что VAR1, Var1 и var1 - это одна и та же переменная.

Символ подчеркивания является значимым, т.е. "A\_VCD" и "AB\_CD" - это разные имена. Имя должно включать не более одного символа подчеркивания. Ограничений на длину имени нет.

Таблица 1 – Входные аналоговые сигналы

№	Наименование	Место установки	Обозначение на Т.С.	Диапазон	Номинальное значение	Сигнал	Модуль PLC	Переменная
1	Уровень смеси в уравнительном баке, %	Внутри уравнительного бака	LE1	0÷100%	100%	4...20	AI:27,28	ZnachLevelUravnitelBak
2	Температура смеси в пастеризаторе, °С	Внутри пастеризатора	TE1	0÷100 °С	86°С	4...20	I:29,30	ZnachTempPasterizator
3	Температура смеси в охладителе, °С	Внутри охладителя	TE2	0÷100°С	6°С	4...20	AI:31, 32	ZnachTempOhladitel

Продолжение таблицы 1 – Входные аналоговые сигналы

4	Уровень смеси в резервуаре, %	Внутри резервуара для смеси	LE2	0÷100	%	4...20	AI:33,34	ZnachLevelRezervuar
5	Температура смеси в фризере, °С	Внутри фризера	TE3	10÷-30	-5	4...20	AI:35, 36	ZhachTempFrizer

К входным дискретным сигналам относятся сигналы датчиков, выдающих 2 значения (ДА –НЕТ, ВЫШЕ –НИЖЕ, и.т.п.) и сигналы органов управления. В данном примере это сигнал кнопки «Пуск», состояния насосов пастеризатора и гомогенизатора.

Таблица 2 - Входные дискретные сигналы

	Наименование сигнала	Уровень	Состояние	Модуль подключения ПЛК 160	Переменная
	Включение/выключение системы	«0» - разомкнутый контакт кнопки; «1» - замкнутый контакт кнопки.	FALSE- Выключено  TRUE- Включено	DI1:7	Pusk
	Состояние насоса пастеризатора	(0 – 1)В – «0» (3,5 – 24)В – «1»	Отключен  Работает	DI1:8	NasosPast
	Состояние насоса гомогенизатора	(0 – 1)В – «0» (3,5 – 24)В – «1»	Отключен  Работает	DI1:9	NasosGomog

В таблицу 3 занесены выходные сигналы ПЛК. При анализе объекта заполняется только первый столбец. Столбцы выходной сигнал и сигнал управления исполнительным механизмом заполняются после выбора исполнительного механизма, в данном случае клапана или двигателя насоса. Столбец «Модуль подключения» заполняется после разработки принципиальной электрической схемы, а переменная в соответствии с правилами, указанными выше.

Таблица 3 – Выходные дискретные сигналы ПЛК

№	Наименование	Сигнал управления исполнительным механизмом	Выходной сигнал ПЛК	Переменная	Модуль подключения ПЛК 160
1	Клапан подачи пара	220В,50мА	«Сухой» контакт	KIParPasterizator	DO1:52
2	Клапан подачи холодной воды	220В,50мА	«Сухой» контакт	KIHolodVodi	DO2:54
3	Клапан подачи смеси	220В,50мА	«Сухой» контакт	KISmesy	DO3:56
4	Клапан подачи хладагента	220В,50мА	«Сухой» контакт	KlapanHladogen	DO4:58
5	Насос уравнильного бака	220В,2А	«Сухой» контакт	Na- sosUravnitel'nogo- Baka	DO5:60
6	Насос пастеризатора	220В,2А	«Сухой» контакт	NasosPasterizatora	DO6:62
7	Насос гомогенизатора	220В,2А	«Сухой» контакт	NasosGomogeniza- tor	DO7:64
8	Насос резервуара	220В, 2А	«Сухой» контакт	NasosRezervuar	DO8:66
9	Конвейер (до морозильника)	220В,1А	«Сухой» контакт	Konveer1	DO9:68
10	Конвейер (от морозильника до заправки)	220В,1А	«Сухой» контакт	Konveer2	DO10:69
11	Конвейер (от заправки до хранения)	220В,1А	«Сухой» контакт	Konveer3	DO11:71
12	Мешалка	220В,3А	«Сухой» контакт	Meshalka	DO12:72
13	Дозатор смеси	220В,100мА	«Сухой» контакт	Dozator	МУ210- D01: В

Для наблюдения оператором хода процесса по мнемосхеме на экране компьютера в визуализацию введены текстовые сообщения, которые появляются по ходу процесса. Сообщения выводятся с помощью переменных типа STRING (см. таблицу 4).

Таблица 4 – Сигналы сообщений

№	Переменная сообщения	Текст сообщения
1	Soobshenie	Охлаждение смеси
2	Soobshenie1	Пастеризация смеси
3	Soobshenie2	Гомогенизация смеси
4	Soobshenie3	Созревание смеси
5	Soobshenie4	Фризерование смеси
6	Soobshenie5	Дозирование смеси
7	Soobshenie6	Закаливание мороженого
8	Soobshenie7	Упаковка мороженого

### 3.3. Алгоритмы управления и функционирования

Для разработки, отладки и проверки функционирования программы управления объектом автоматизации создадим 2 типа алгоритмов:

- алгоритм управления объектом, в соответствии с которым будут написаны программы управления на языках CoDeSyS;

- алгоритмы функционирования (имитации) объекта управления, который будет использоваться для разработки программ имитации реакции объекта управления на сигналы ПЛК с выдачей сигналов имитаторов датчиков объекта.

В общем виде структура алгоритмов проекта показаны на рисунке 2.

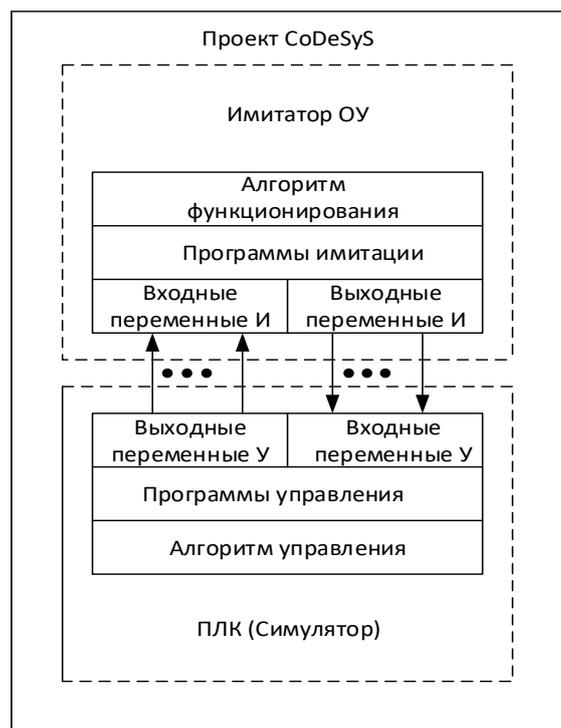


Рисунок 2 - Структура проекта CoDeSyS

Алгоритм функционирования совместно с программами имитации объекта управления служит для отладки основных программ управления, формируя сигналы датчиков (выходные переменные И) в ответ на сигналы ПЛК (выходных переменных У), которые подаются на исполнительные механизмы реального объекта, а при проверке соединятся со входными переменными И.

Правила выполнения алгоритмов изложены в Приложении Б.

Для рассматриваемого примера технологической линии производства мороженого алгоритм управления приведен на рисунках 3 -6.

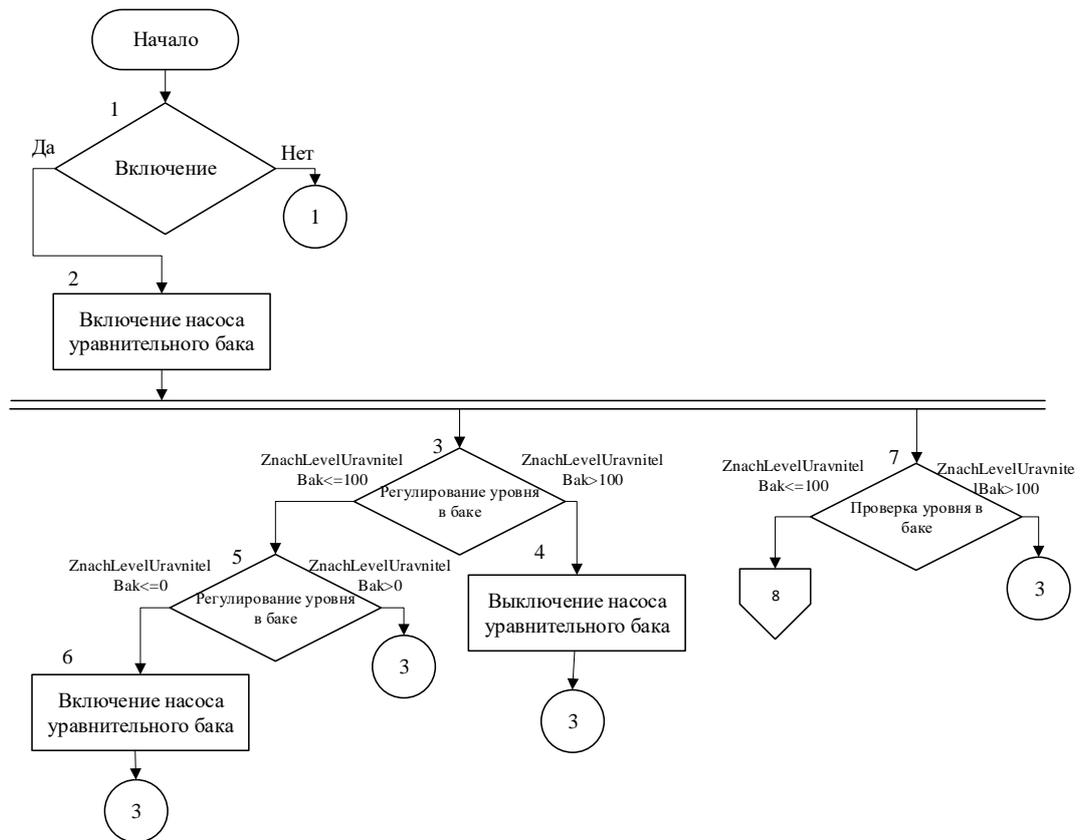


Рисунок 3 - Алгоритм управления

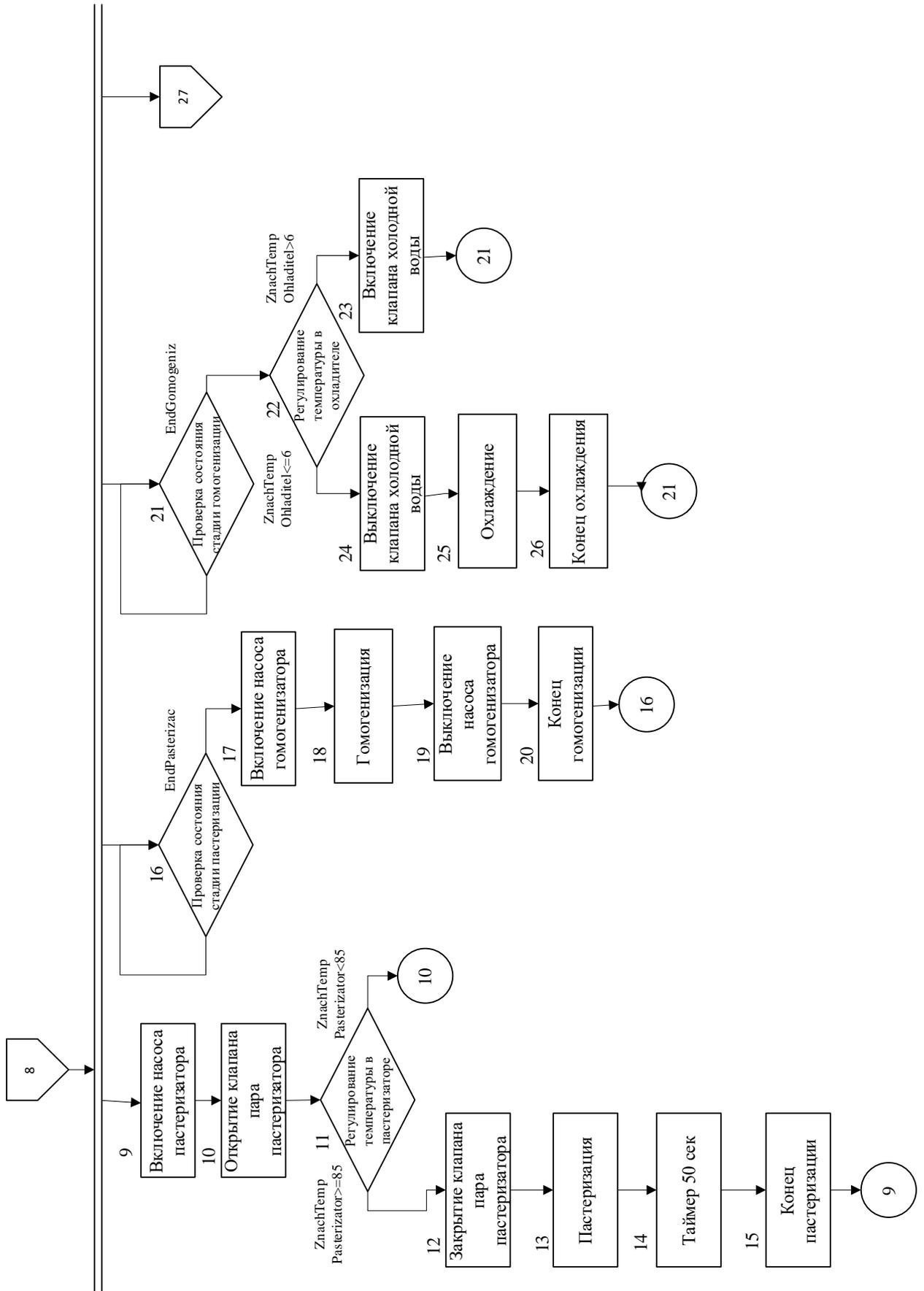


Рисунок 4 - Алгоритм управления. Продолжение



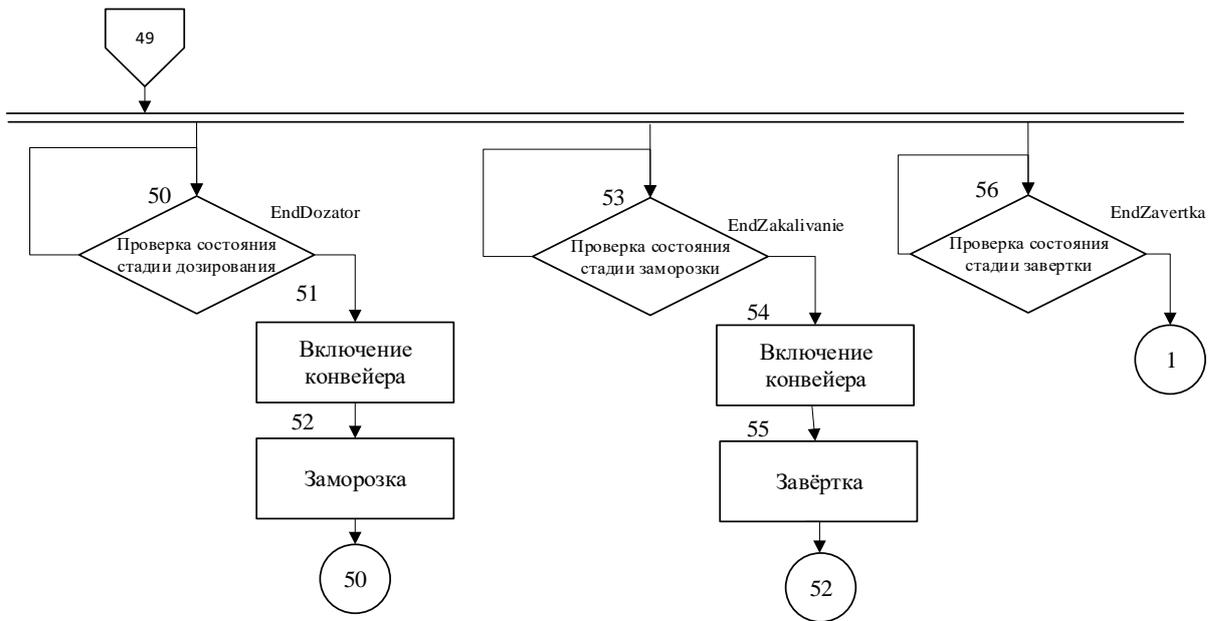


Рисунок 6 - Алгоритм управления. Конец

Укрупненно расшифровка работы алгоритма приведена ниже. В первом блоке происходит проверка состояния системы. Во втором блоке включается насос, который перекачивает смесь из ванны в уравнительный бак. С третьего по седьмой блок регулируется уровень смеси в уравнительном баке. Далее в параллельной ветви происходит несколько процессов, первый из которых - пастеризация. Включается насос пастеризатора (Блок 9), открывается клапан подачи пара (Блок 10), происходит проверка температуры (Блок 11). Если температура равна  $85^{\circ}\text{C}$ , закрывается клапан пара (Блок 12), включается таймер и смесь пастеризуется 50 секунд (Блок 13-15). В следующей параллельной ветви проверяется состояние стадии пастеризации (Блок 16) и если она закончена, то включается насос гомогенизатора (Блок 17) и смесь гомогенизируется (Блок 18-20). После в параллельной ветви проверяется состояние стадии гомогенизации (Блок 21). Если стадия завершена, то происходит проверка температуры в охладителе (Блок 22). При условии, что температура больше  $6^{\circ}\text{C}$ , включается клапан холодной воды (Блок 23). В обратном случае, клапан холодной воды выключается (Блок 24) и происходит процесс охлаждения смеси мороженого (Блок 25-26). Далее в параллельной ветви происходит проверка состояния стадии охлаждения (Блок 28). Включается клапан подачи смеси в резервуар для созревания смеси (Блок 29).

Проверяется уровень смеси в резервуаре (Блок 30) и, если смесь достигла нужного уровня, клапан подачи смеси закрывается (Блок 31) и смесь созревает в течение 4-х часов (Блок 32-34). После в параллельной ветви проверяется состояние стадии созревания (Блок 35). Если стадия завершена, то происходит включение насоса подачи смеси в фризёр (Блок 37), включение шнека (Блок 38), открывается клапан подачи хладагента (Блок

39). Происходит проверка температуры в фризере (Блок 40). Если она равна  $-5^{\circ}\text{C}$ , закрывается клапан подачи хладагента (Блок 42) и происходит процесс фризирования (Блок 43-44). Далее в параллельной ветви происходит проверка состояния стадии фризирования (Блок 45). Включается дозатор (Блок 46) и происходит процесс дозирования смеси в вафельные стаканчики (Блок 47-48). После в параллельной ветви проверяется состояние стадии дозирования (Блок 50). Включается конвейер (Блок 51) и происходит процесс заморозки мороженого в морозильной камере (Блок 52). Далее проверяется состояние стадии заморозки (Блок 53), включается конвейер (Блок 54), и происходит завертка мороженого в упаковку (Блок 55). После проверяется состояние стадии завертки (Блок 56) и все описанные выше процессы повторяются снова.

Алгоритмы функционирования (имитации объекта управления). Программы имитации объекта автоматизации служат только для проверки основной программы управления, поэтому принято решение о том, что сигнал любого датчика меняется по линейному закону во времени. Поскольку исполнительная система CoDeSys работает циклически, то в имитаторе датчика выходную переменную можно увеличивать или уменьшать в каждом цикле на постоянное значение, принимаемое скоростью изменения выходного сигнала. Функциональная схема имитатора любого датчика показана на рисунке 7.

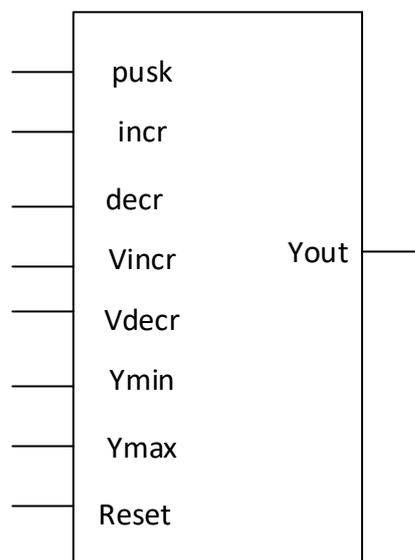


Рисунок 7 - Функциональная схема имитатора любого датчика

`pusk`, `incr`, `decr` - переменные типа `BOOL`, соответственно, запуск, увеличение и уменьшение значения выходной переменной `Yout`.

$V_{incr}$ ,  $V_{decr}$ ,  $Y_{min}$ ,  $Y_{max}$  – переменные типа INT или REAL, соответственно скорость увеличения, уменьшения, минимальное и максимальное значение выходной переменной  $Y_{out}$  такого же типа.

Блок-схема алгоритма имитации сигнала датчика изображена на рисунке 8. Операции  $incr$  & NOT ( $decr$ ) соответствует логическое произведение переменной  $incr$  с инверсией переменной  $decr$ . В блоках 5, 7 и 9 происходит изменение в каждом цикле исполнительской системы CoDeSys выходной переменной  $Y_{out}$  в зависимости от значения переменных  $incr$  и  $decr$ . Действия в блоках с 10 по 13 не дают выйти переменной  $Y_{out}$  за значения  $Y_{min}$  и  $Y_{max}$ .

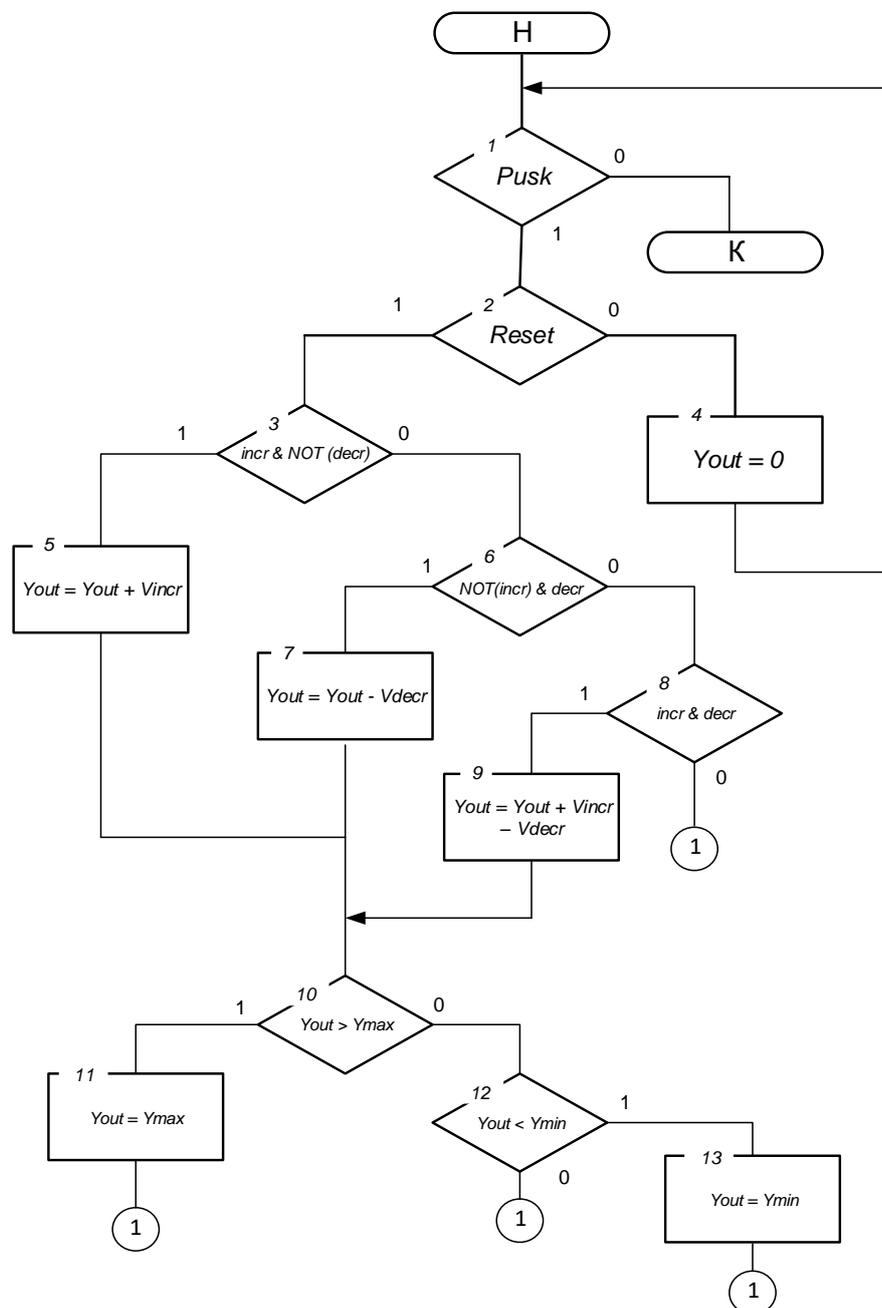


Рисунок 8 - Блок-схема алгоритма имитации сигнала датчика

### 3.4. Принципиальная электрическая схема

На принципиальной электрической схеме необходимо отразить все элементы, используемые в система автоматизации и связи между ними. В рассматриваемом примере используются 8 аналоговых датчиков: 2 датчика уровня (BL1, BL2) и 6 датчиков температуры (BK1 – BK6). Входной дискретный сигнал *pusk* формируется кнопкой SB3. Принципиальная электрическая схема приведена в Приложении В, перечень элементов схемы указан в Приложении Г.

Программируемый логический контроллер выбирается из каталога фирмы OWEN по следующим критериям [2, 3]:

- число аналоговых входов и дискретных входов должно быть не меньше, чем требуется в проектируемой системе управления. Аналоговые входы должны обеспечивать прием сигналов выбранных датчиков (стандартные токи и напряжения, возможность подключения датчиков сопротивления и термоэлектрических термометров, при необходимости);

- выходы контроллера и их число должны обеспечивать электрические параметры исполнительных механизмов проекта;

- интерфейсы контроллера должны обеспечивать связь с компьютером, на котором разрабатывается проект и с датчиками и исполнительными механизмами, которые управляются интерфейсными сигналами (RS-232, RS-485, Ethernet и т.п.);

- стоимость ПЛК должна быть минимальной из выпускаемых фирмой.

Для данного проекта по числу входов и интерфейсам подходит ПЛК-160 (на схеме А1), однако число дискретных выходов 12 недостаточно. Поэтому дополнительно выбран недорогой модуль дискретного вывода МУ210-401 (на схеме А2), соединенный с основным ПЛК по Ethernet.

В качестве датчиков уровня выбраны поплавковые датчики уровня ПДУ-И ТМ «OWEN» с токовым выходом 4 - 20 мА. Датчики температуры - термопреобразователь сопротивления OWEN ДТС-И с аналогичным выходным сигналом. Подключение ко входам контроллера идет по интерфейсу «токовая петля». Питание датчиков осуществляется от одноканального блока питания БП30Б-ДЗ-24, ТМ «OWEN» (на схеме G1).

В качестве исполнительных механизмов используются трехфазные электродвигатели насосов АИР71В6 (на схеме М1-М8). Магнитные пускатели КМИ-10910 (на схеме КМ1- КМ9). КМ1 с кнопками SB1 и SB2 подключает и отключает трехфазной питание на все элементы. Для включения и выключения клапанов используется электромагнитный клапан EV25В.

Дискретные выходы контроллера по своей нагрузочной способности (220В, 3А) значительно превосходят потребности исполнительных механизмов.

### 3.5. Разработка программы управления

Разработка программы управления начинается с создания проекта. Подробно создание проекта, состав проекта, языки программирования рассмотрено в разделе «Справка» системы программирования CoDeSys [4,5].

Проекту необходимо дать новое имя, оно же послужит и названием файла проекта.

В пробных проектах это может быть группа и фамилия студента, в окончательном варианте имя проекта - это название объекта автоматизации.

Проект включает в себя программные компоненты. Первый программный компонент (POU - Program Organization Unit) помещается в новый проект автоматически и получает название PLC\_PRG. Именно с него и начинается выполнение процесса (по аналогии с функцией main в языке C), из него будут вызываться другие программные блоки (программы, функциональные блоки и функции). Следует отметить, что все другие POU должны быть обязательно включены в PLC\_PRG, иначе при отладке они не запускаются. Любой из POU можно программировать на одном из шести языков стандарта IEC 61131-3 (CFC, SFC, FBD, LD, ST, IL). Описание языков указано в разделе «Справка» системы программирования CoDeSys.

Программу PLC\_PRG также можно разрабатывать на любом языке, но следует иметь в виду, что в программах на SFC в каждом машинном цикле выполняются действия только в тех шагах, которые активны. Если вызовы каких-либо POU будут записаны в произвольных шагах, то они не будут гарантированно запущены. Для студенческих проектов язык SFC не рекомендуется для PLC\_PRG. Достаточно удобным для этой главной программы будет язык CFC [7,8,9].

Проект включает следующие объекты: POU, типы данных, визуализации, ресурсы, библиотеки. Каждый проект сохраняется в отдельном файле.

В рассматриваемом примере линии производства мороженого в POU содержатся: алгоритм основной программы, имитатор движения шляпки (видимой над стаканчиком части мороженого), имитатор мешалки, имитатор движения конвейера, основная программа PLC\_PRG, имитатор движения вафельных стаканчиков (Pруг) и имитатор регулирования уровня и температуры. Программные компоненты POU представлены на рисунке 9.

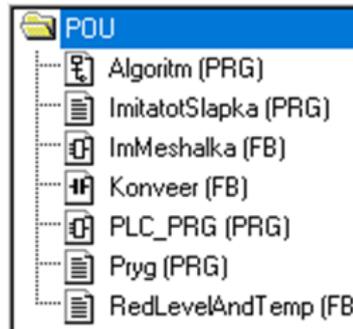


Рисунок 9 - Программные компоненты POU

Программа PLC\_PRG - это специальный POU, который должен быть в каждом проекте. Эта программа вызывается один раз за цикл управления. В данной работе программа PLC\_PRG (вызов главной программы и всех имитаторов) написана на языке CFC.

В данной работе программа Algoritm (PRG) написана на языке SFC. Программа реализует алгоритм, указанный в подразделе 3.2 на рисунках 3-6. (Приложение Д). Действия внутри шагов алгоритма реализованы на языке IL (Рисунок 10). В шаге VklNasos2 выполняется действие:

```
LD TRUE
ST NasosPasterizatora
```

Включается насос пастеризатора (NasosPasterizatora) на 2 секунды, так как установлен атрибут шага минимальное время T#2s. Через две секунды происходит переход на шаг выключения насоса (VyklNasos2), где на IL выполняется действие – выключение насоса пастеризатора:

```
LD FALSE
ST NasosPasterizatora
```

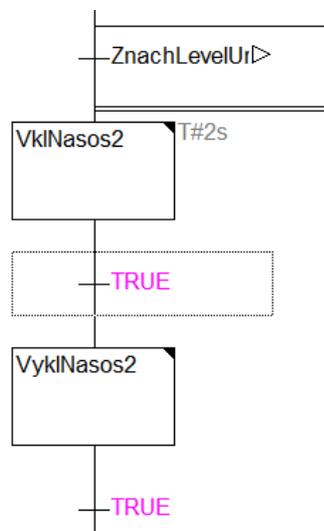


Рисунок 10 - Фрагмент программы Algoritm (PRG)

Сообщения на визуализации по ходу процесса поясним на примере сообщения о гомогенизации. Фрагмент программы с шагами, в которых указаны действия по выводу сообщений приведен на рисунке 11. В шаге VklGomogizer приведена следующая программа на языке IL:

```
LD      TRUE
ST      Gomogenizator
LD      'Гомогенизация смеси'
ST      Soobshenie2
```

В результате переменная Soobshenie2, размещенная над гомогенизатором, принимает значение «Гомогенизация смеси». В шаге VyklGomogenizator программа

```
LD      FALSE
ST      Gomogenizator
LD      "
ST      Soobshenie2
```

придает пустое значение переменной Soobshenie2, так как предыдущий оператор LD загружает отсутствие символов между апостофами (").

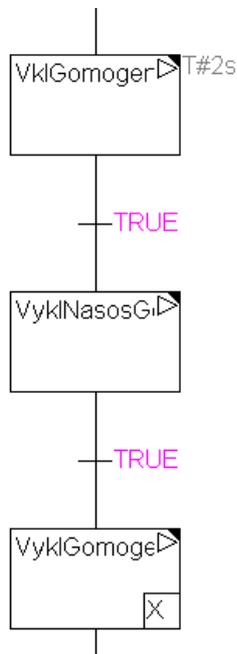


Рисунок 11 - Фрагмент программы. Algoritm (PRG)

На языке ST написаны следующие POU: имитатор регулирования температуры и уровня RedLevelAndTemp (FB), движение стаканчиков по конвейеру Prg (PRG), визуализация наполнения вафельного стаканчика ImitatotSlapka (PRG).

На рисунке 12 рассмотрена программа, в которой организована видимость наполнения вафельного стаканчика мороженым. В разделе объявлений программы указана одна локальная переменная Slapka: BOOL, состояние которой TRUE соответствует отсутствию мороженого в стаканчике на конвейере, FALSE – наличию мороженого. Значения переменной PeremesenieXGI, указанные в программе соответствуют видимости мороженого в стаканчике.

```

0001 PROGRAM ImitatotSlapka
0002 VAR
0003     Slapka: BOOL;
0004 END_VAR
0005
0006 <
0007
0001 Slapka:=NOT((PeremesenieXGI=11) OR (PeremesenieXGI=14) OR (PeremesenieXGI=17)OR (PeremesenieXGI=20)
0002 OR (PeremesenieXGI=23)OR (PeremesenieXGI=26)OR (PeremesenieXGI=29)OR (PeremesenieXGI=32)OR (PeremesenieXGI=35)
0003 OR (PeremesenieXGI=38)OR (PeremesenieXGI=41)OR (PeremesenieXGI=44)OR (PeremesenieXGI=47)OR (PeremesenieXGI=50)
0004 OR (PeremesenieXGI=53)OR (PeremesenieXGI=56)OR (PeremesenieXGI=59)OR (PeremesenieXGI=62)OR (PeremesenieXGI=65)
0005 OR (PeremesenieXGI=68)OR (PeremesenieXGI=71)OR (PeremesenieXGI=74)OR (PeremesenieXGI=77));
0006
0007

```

Рисунок 12 - Программа видимости наполнения вафельного стаканчика мороженым

В программе имитатора движения вафельных стаканчиков (Pryg) формируется переменная PeremesenieXGI (рисунок 13). В разделе объявлений объявлены промежуточные булевские переменные A, B и локальная переменная PeremesenieX целого типа. Также объявлены экземпляры функциональных блоков BLINK1 и F\_TRIG1. Переменная PeremesenieXGI используется для передачи значения переменной PeremesenieX в другие программы. Когда переменная Dozator = TRUE, BLINK1 вырабатывает на выходе A импульсы с периодом 1 секунда. Блок F\_TRIG1 вырабатывает на выходе B импульс TRUE длительностью в один цикл CoDeSys. В результате переменная PeremesenieX увеличивается на единицу каждую секунду. При превышении ее значения 93 она обнуляется. Это обеспечивает видимость движения стаканчиков с мороженым в визуализации по конвейеру.

```

0001 PROGRAM Prvg
0002 VAR
0003   A: BOOL;
0004   B: BOOL;
0005   PeremesenieX: INT;
0006   BLINK1: BLINK;
0007   F_TRIG1: F_TRIG;
0008 END_VAR
0009
0001 IF Dozator THEN
0002 BLINK1(ENABLE:= TRUE, TIMELOW:=T#500ms , TIMEHIGH:=T#500ms , OUT=>A);
0003 F_TRIG1(CLK:=A, Q=>B);
0004 IF B THEN
0005 PeremesenieX:=PeremesenieX+1;
0006 PeremesenieXGI:=PeremesenieX;
0007 END_IF;
0008 IF (PeremesenieX>93) THEN
0009 PeremesenieX:=0;
0010 END_IF;
0011 END_IF;
0012
0013

```

Рисунок 13 - Программа имитатора движения вафельных стаканчиков

На рисунке 14 приведен пример разработки функционального блока, а именно имитатора мешалки, установленной во фризере. Функциональный блок отличается от программы тем, что его экземпляры могут быть использованы в других программах проекта без ограничений, при этом входные и выходные переменные этих экземпляров будут разными. В этом блоке одна входная переменная StartMeshalka и две выходные переменные Lopast1 и Lopast2. Здесь используется экземпляр G1 стандартного функционального блока BLINK. В результате формируются два противофазных сигнала Lopast1 и Lopast2 с периодом 350ms.

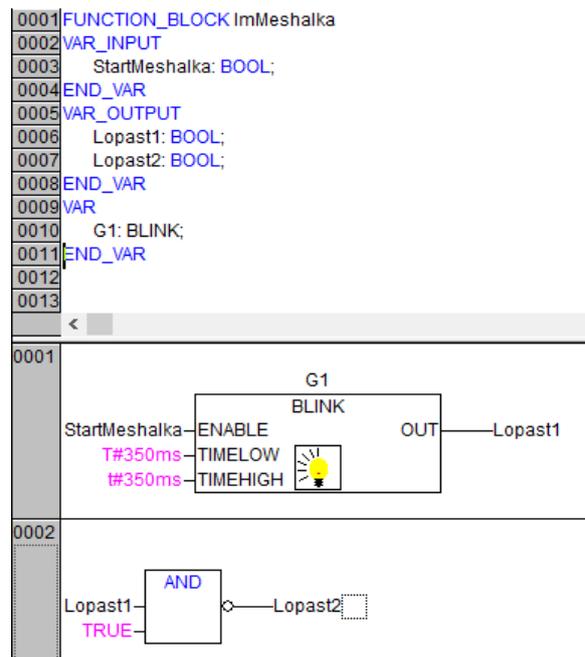


Рисунок 14 - Функциональный блок имитатора мешалки

На рисунке 15 изображен функциональный блок имитатора конвейера, написанный на языке FBD. Для выполнения входная переменная StartConveer должна принять значение TRUE. Если выходная переменная Angle больше – 360 (цепь 1), то она уменьшается на 10 (цепь 2). Когда она достигает значения -360 и меньше (цепь 3), то она обнуляется (цепь 4). Этот функциональный блок используется для имитации вращения приводов конвейеров на визуализации.

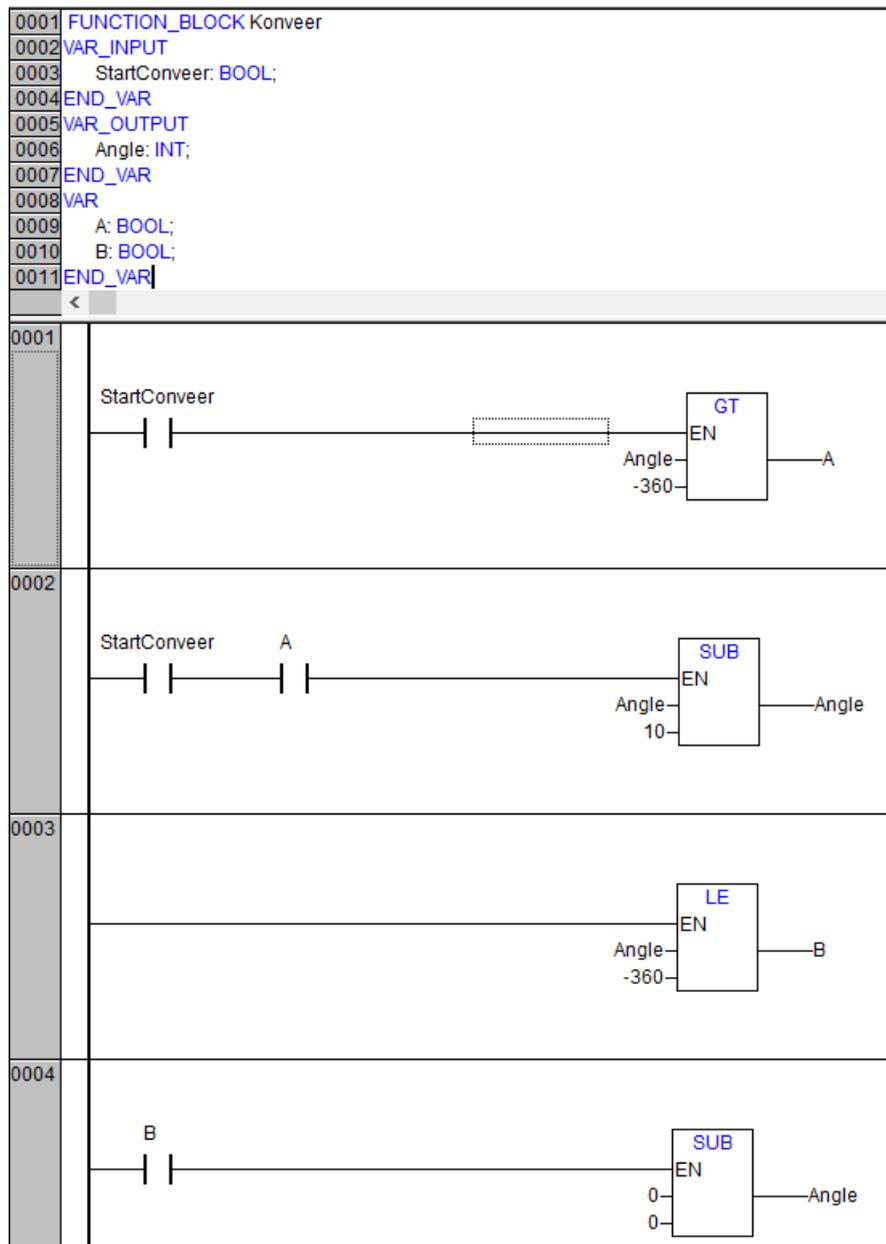


Рисунок 15 - Функциональный блок имитатора конвейера

На рисунке 16 представлен функциональный блок имитатора аналогового значения датчика. Данный функциональный блок написан на языке ST и реализует алгоритм, показанный на рисунке 8. Обозначениям переменных Vol и Men соответствуют incr и decr. На языке ST блоки алгоритма (3, 5-9) реализованы в строке 0001 в разделе кода (сверху

с 0001 по 0014 – раздел объявлений переменных, за ним, после разделительной черты с 0001 по 0008 следует раздел кода. Функция `BOOL_TO_REAL(Bol)` переводит значение переменной `Bol` булевского типа в действительное (`TRUE` в 1.0, `FALSE` в 0.0). Операторы `IF` в строках 0002 – 0006 реализуют блоки алгоритма 10-13 (рисунок 8), выполняя проверку выходы переменной `Yout` за допустимые значения.

```

0001 FUNCTION_BLOCK RedLevelAndTemp
0002 VAR_INPUT
0003     Vbol: REAL;
0004     Bol: BOOL;
0005     Vmen: REAL;
0006     Men: BOOL;
0007     Ymax: REAL;
0008     Ymin: REAL;
0009 END_VAR
0010 VAR_OUTPUT
0011     Yout: REAL;
0012 END_VAR
0013 VAR
0014 END_VAR
0015
0001 Yout:=Yout+Vbol*BOOL_TO_REAL(Bol)-Vmen*BOOL_TO_REAL(Men);
0002 IF Yout>=Ymax THEN
0003     Yout:=Ymax;
0004 END_IF;
0005 IF Yout<=Ymin THEN
0006     Yout:=Ymin;
0007 END_IF;
0008
0009

```

Рисунок 16 - Функциональный блок имитатора аналогового значения датчика

На рисунке 17 приведена основная программа `PLC_PRG`, вызываемая исполнительной системой `CoDeSys`. Данная программа написана на языке `SFC`, на котором все элементы диаграммы выполняются в следующем порядке: слева направо и сверху вниз в каждом цикле исполнения. Сначала выполняется программа `Algoritm` (верхний индекс 0 на диаграмме), затем программа `Pryg` (индекс1), затем экземпляр `VklDozatorKonveer` функционального блока `Konveer` (2). Экземпляры функционального блока `Konveer` используются еще два раза `VklMorozilnikKonveer` (4) и `VklZavertkaKonveer` (6). Экземпляры функционального блока `RedLevelAndTemp` пять раз:

- `RegTempPasterizator(9)` – имитирует температуру пастеризатора (переменная `ZnachTempPasterizator`);
- `RegLevelUrovBak(11)` – имитирует уровень в уравнительном баке (переменная `ZnachLevelUravnitelBak`);

- RegTempOhladitel(13) – имитирует температуру охладителя (переменная ZnachTempOhladitel);
- RegLevelRezervuar(15) – имитирует уровень в резервуаре (переменная ZnachLevelRezervuar);
- RegTempFrizer(17) – имитирует температуру во фризере (переменная ZhachTempFrizer).

Программа заканчивается вызовом экземпляра MeshalkaFrizera функционального блока ImMeshalka.

```

0001 PROGRAM PLC_PRG
0002 VAR
0003   VklDozatorKonveer: Konveer;
0004   VklMorozilnikKonveer: Konveer;
0005   VklZavertkaKonveer: Konveer;
0006   RegTempPasterizator: RedLevelAndTemp;
0007   RegLevelUrovBak: RedLevelAndTemp;
0008   RegTempOhladitel: RedLevelAndTemp;
0009   RegLevelRezervuar: RedLevelAndTemp;
0010   RegTempFrizer: RedLevelAndTemp;
0011   MeshalkaFrizera: ImMeshalka;
0012 END_VAR
0013 VAR_OUTPUT
0014   ZnachUgla: INT;
0015   ZnachUglaMorozilnik: INT;
0016   ZnachUglaZavertka: INT;
0017   Pozic2: BOOL;
0018   Pozic1: BOOL;
0019 END_VAR

```

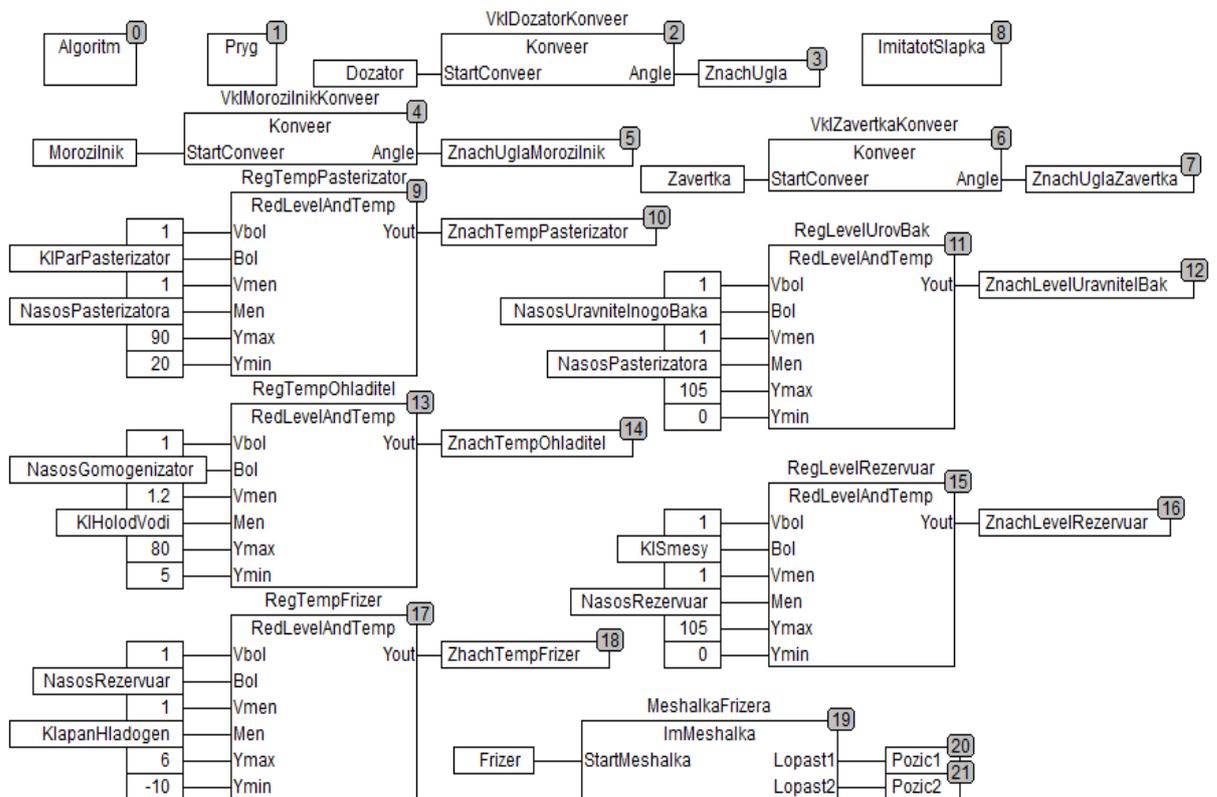


Рисунок 17 - Программа PLC\_PRG

### 3.6. Отладка программы на симуляторе

Отладку программу всегда рекомендуется проводить по частям. Каждую программу, функциональный блок или функцию следует отлаживать отдельно. Проверка синтаксиса программы осуществляется путем выбора команды «Компилировать» во вкладке «Проект» (или клавиша F11). При отсутствии ошибок выводится сообщение «0 ошибок, 0 предупреждений». При наличии ошибок выводится сообщение об ошибке и указывается место ее нахождения. После исправления ошибок можно перейти к эмуляции программы. Во вкладке «Онлайн» устанавливаем флажок «Режим эмуляции», вызываем команду «Подключение» и команду «Старт». Все переменные можно задать, вызвав процедуру изменения щелчком левой кнопки мыши. После задания переменной на диаграмме или в таблице выполнить команду «Записать значение» во вкладке «Онлайн» (Ctrl+F7).

Отладку всего проекта можно начинать только после отладки всех входящих в него программных компонентов. При этом необходимо помнить, что все программные компоненты должны вызываться (быть включены) из программы PLC\_PRG.

Программное обеспечение CoDeSys располагает несколькими инструментами отладки проекта. Их краткие описания приведены ниже.

В режиме отладки ПЛК работает существенно медленнее. Из-за этого возможны прерывания связи по таймауту. Поэтому при отладке следует перенастроить таймауты в окне настройки, вызываемом командой «Проект | Опции (Project | Options)» главного меню ПО CoDeSys: на вкладке «Рабочий стол (Desktop)», значение параметра «Таймаут связи (Communication timeout)» увеличить до 15000.

Опция отладки ПО CoDeSys заставляет компилятор формировать дополнительный код, упрощающий поиск ошибок. Опция «Отладочный код (Debugging)» включается установкой флажка переключателя «Отладочный код (Debugging)» в окне «Опции (Options)», вызываемом командой «Проект | Опции (Project | Options)» главного меню, на вкладке «Генератор кода (Build)».

Точки останова – это места, в которых выполнение программы будет приостанавливаться, что позволяет просмотреть значения переменных на определенном этапе работы программы. Точки останова можно задавать во всех редакторах. В текстовом редакторе точка останова устанавливается на номер строки, в FBD и LD – на графический элемент, в SFC – на шаг. Точки останова можно включить в режиме эмуляции.

Пошаговое выполнение позволяет проверить логическую правильность программы. Под «шагом» на разных языках подразумевается:

IL: Выполнить программу до следующего оператора CAL, LD or JMP.

ST: Выполнить следующую инструкцию.

FBD, LD: Выполнить следующую цепь.

SFC: Продолжить действие до следующего шага.

Программу можно выполнить по циклам. Команда «Онлайн | Один цикл (Online | Single Cycle)» выполняет один рабочий цикл и останавливает контроллер после выполнения.

Режим эмуляции последовательно включается и отключается выбором команды «Онлайн | Режим эмуляции (Online | Emulation)» главного меню. Включенный режим маркируется установленным флажком в строке главного меню и записью «Эмул.» в строке состояния главного окна. Во время эмуляции созданная программа выполняется не в ПЛК, а в компьютере, на котором запущено ПО CoDeSys. В этом режиме допустимы все функции онлайн, что позволяет проверить логическую правильность программ, не используя контроллер.

Бортжурнал (Log)» хронологически записывает действия пользователя, внутренние сообщения системы исполнения, изменения состояния и исключения в режиме онлайн. Это позволяет анализировать условия возникновения ошибки при отладке программы. Просмотр записей «Бортжурнала (Log)» производится в режиме, вызываемом выбором строки «Бортжурнал (Log)» дерева ресурсов проекта на вкладке «Ресурсы» организатора объектов CoDeSys.

### 3.7. Визуализация проекта

Визуализация предназначена для графического представления объекта управления и непосредственно связана с созданной в CoDeSys программой контроллера. Редактор визуализации CoDeSys предоставляет набор готовых графических элементов, которые могут быть связаны соответствующим образом с переменными проекта [6].

В Online режиме представление элементов на экране изменяется в зависимости от значений переменных.

Свойства отдельных элементов визуализации, а также визуализации в целом устанавливаются в соответствующих диалогах конфигурации и диалоге свойств объекта. Здесь определяется начальный вид элементов и выполняется привязка динамических свойств к значениям переменных проекта.

Дополнительные возможности открывает программируемость свойств элементов посредством специальных структурных переменных.

Используя заместители в диалогах конфигурации можно облегчить работу и сэкономить время, в случаях, когда созданный объект визуализации должен использоваться неоднократно. Например, при визуализации значений нескольких экземпляров одного функционального блока.

В приведенном примере проекта визуализация будет играть роль единственного пользовательского интерфейса для контроля и управления работой ПЛК программы в режиме эмуляции и при наличии контроллера в рабочем режиме. В этом случае ввод данных для программы должен выполняться исключительно посредством элементов визуализации.

При отладке отдельных компонентов проекта следует использовать клавиши быстрого ввода для булевских переменных и слайдеры для ввода аналоговых переменных. Для контроля выходных параметров можно использовать изменение надписи и цвета, переменные столбцы, индикаторы и т.п. Такую возможность обеспечивают специальные возможности ввода, задаваемые в процессе конфигурации.

На рисунке 18 приведен снимок с экрана визуализации проекта линии производства мороженого.

Все элементы визуализации (прямоугольники, круги, трубопроводы, насосы, клапаны, транспортеры, стаканчики мороженого и т.п.) созданы при помощи инструментов визуализации, расположенных в закладке «Вставка» или в строке инструментов. Конфигурирование элемента, то есть придания ему анимационных свойств при изменении связанных с ним переменных поясним на примере пастеризатора. Двойным щелчком по изображению пастеризатора вызывается опция конфигурирования элемента. Появляется окно конфигурирования на рисунке 19. В категории «Цвета» задаются «Цвет» и «Тревожный цвет» пастеризатора соответствующие значению FALSE или TRUE переменной *Pasterizacia*. В категории «Переменные» в графе «Изм. цвета» заносится переменная, которая будет менять цвет. Вносить переменную рекомендуется с помощью ассистента ввода вызываемого клавишей F2.

Для вывода сообщения в левом верхнем углу рисунка 18 необходимо конфигурировать прямоугольник следующим образом: в категории «Текст» проставить %s, в категории «Переменные» занести переменную типа *STRING*. В данном проекте *Algoritm.Soobshenie1* – это локальная переменная *Soobshenie1* программы *Algoritm*.

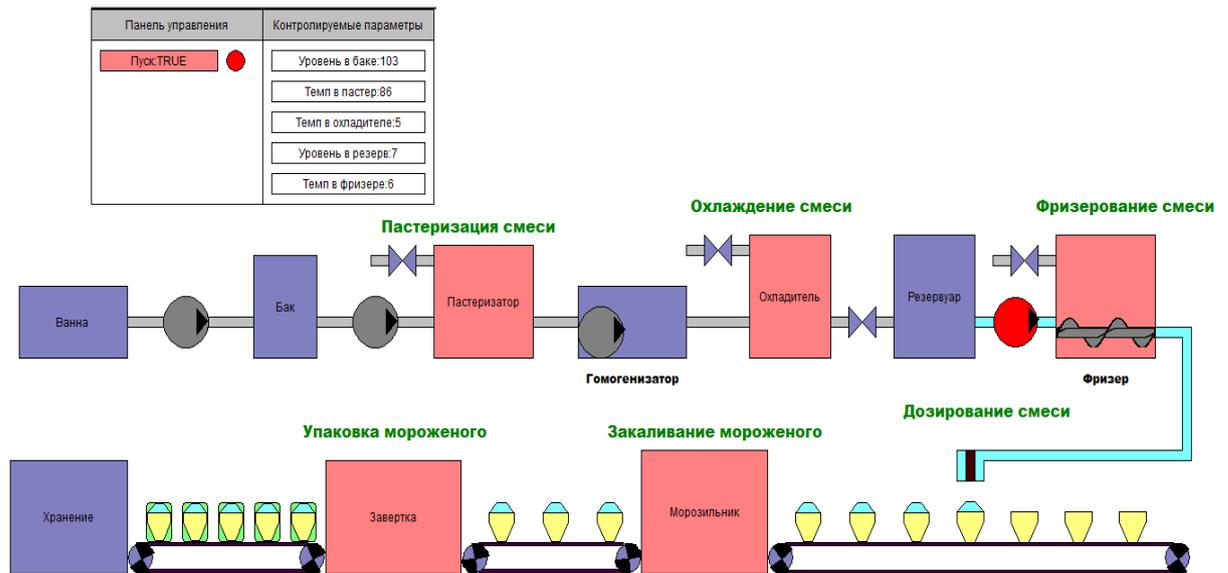


Рисунок 18 - Визуализация проекта

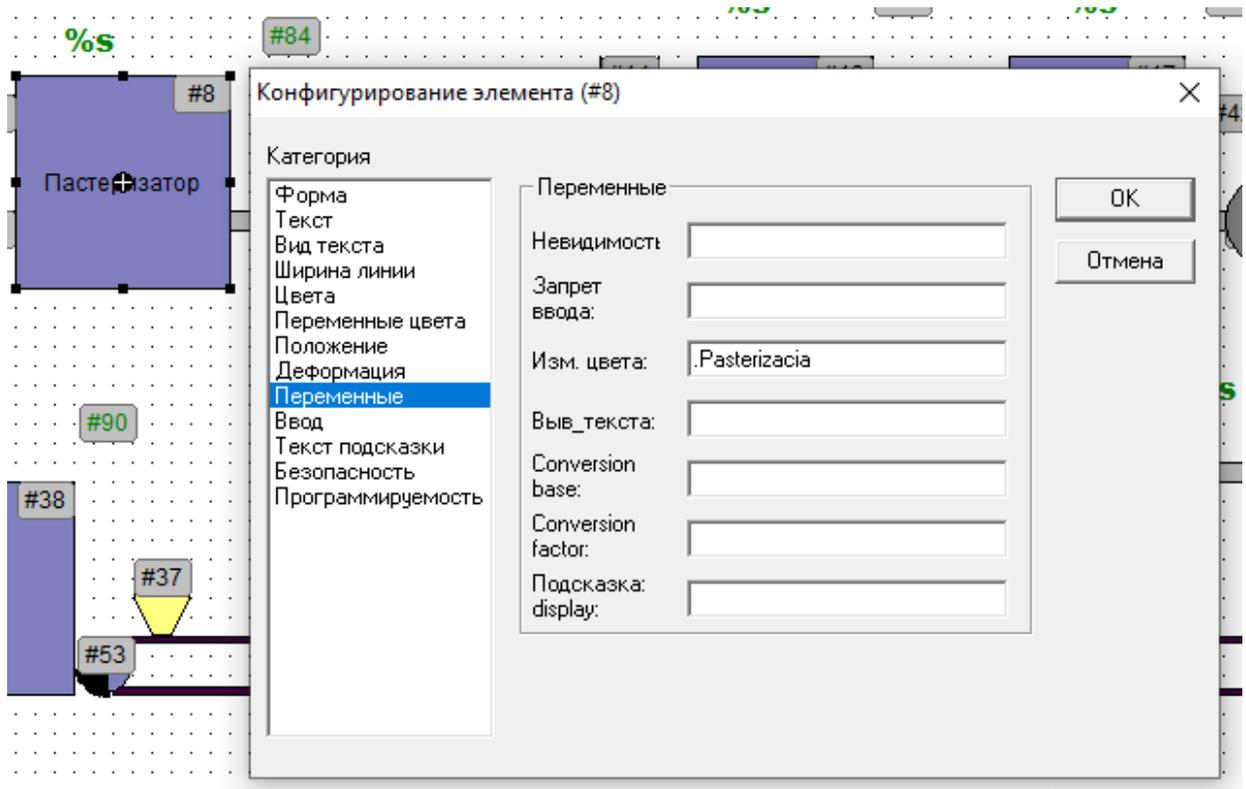


Рисунок 19 - Конфигурирование пастеризатора

Имитация движения шнека во фризере реализована следующим образом: подобраны два рисунка шнека в положении отличающиеся друг от друга поворотом на 180°

(шнек1.bmp и шнек2.bmp). Эти рисунки внесены в категорию «Растровый рисунок» (рисунок 20). В категорию «Переменные» - «Невидимость» внесены значения PLC\_PRG.Pozic2 и PLC\_PRG.Pozic1 соответственно для каждого рисунка. Эти переменные формируются в программе PLC\_PRG экземпляром функционального блока MeshalkaFrizera.

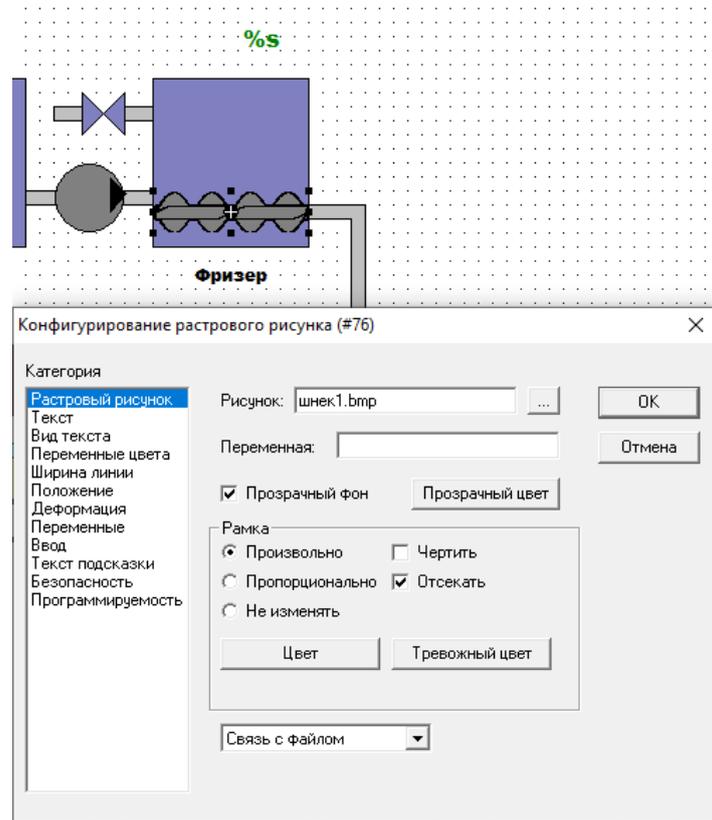


Рисунок 20 - Конфигурирование шнека фризера

Имитация движения ролика транспортера реализована следующим образом: использовались два рисунка секторов в положении отличающимся друг от друга поворотом на 180°. При конфигурировании сектора во вкладке «Положение» указывается значение угла поворота PLC\_PRG.ZnachUgla одинаковое для обоих секторов. Раздел «Угол» категории «Конфигурирование сектора» содержит числа или переменные проекта, которые определяют начальный и конечный углы элемента кругового сектора в градусах. Значение угла поворота от 0 до 360 градусов (с шагом 10 градусов) формируются в экземпляре VklDozatorKonveer функционального блока Konveer. На рисунке 21 показано конфигурирование вращения ролика конвейера.

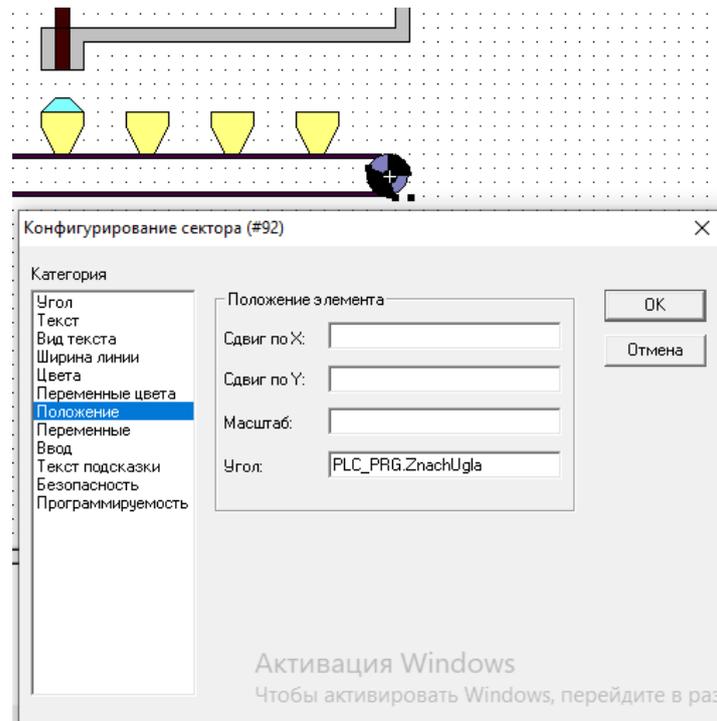


Рисунок 21 - Конфигурирование вращения ролика конвейера

Перемещение вафельных стаканчиков по транспортерам визуализировано следующим образом. Все стаканчики формы «Полигон» размещены на всех трех конвейерах. В категории «Переменные» - «Невидимость» каждого стаканчика внесено выражение  $.PeremesenieXG1 < STEP$ , где  $STEP$  – числа  $\{3, 6, 9, \dots, 48\}$ . Значение 3 – для второй справа позиции стаканчика, значение 48 – для последней (крайней слева). В результате при исполнении стаканчик появляется, когда переменная  $PeremesenieXG1$  достигает числа  $STEP$  для данной позиции. На рисунке 22 показана конфигурация для появления пятого стаканчика, когда переменная  $PeremesenieXG1$  достигнет 12.

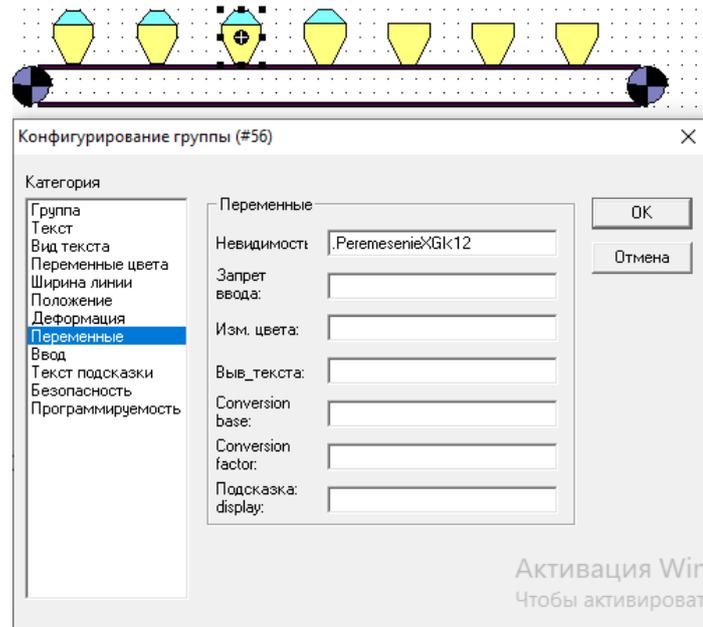


Рисунок 22 - Конфигурирование движения вафельных стаканчиков

#### 4. Оформление пояснительной записки

Работа выполняется на стандартных листах бумаги формата А4. Образец титульного листа приведен в Приложении Е.

Текст записки необходимо набирать в редакторе Microsoft Office Word. Принципиальная электрическая схема выполняется на листе формате А3 в приложении Microsoft Office Visio с условными графическими обозначениями электро-радиоэлементов согласно требованиям ЕСКД. Перечень электро-радиоэлементов можно разместить на принципиальной электрической схеме или на отдельном листе (листах) расчетно-пояснительной записки (Приложение Д). Необходимо строго придерживаться установленных ГОСТ буквенных обозначений каждой физической величины.

При разработке блок схемы алгоритма пользоваться условными графическими обозначениями, указанными в Приложении Б. Алгоритм допускается размещать на листах формата А4. При необходимости размещения на нескольких страницах использовать межстраничные переносы.

При оформлении программ на текстовых языках следует придерживаться следующих правил: в одной строчке – один оператор; все действия сопровождаются комментариями, при этом комментарий должен отражать не действие команды, а выполняемое действие по алгоритму управления; для удобства чтения используют пробелы и табуляция.

Расчетно-пояснительная записка должна содержать введение, в котором излагается цель проводимых расчетов, примеры использования ПЛК в системах автоматике.

В заключении следует сформулировать выводы по работе: отметить, насколько разработанные программа и схема устройства удовлетворяют требованиям технического задания, раскрыть достоинства и недостатки разработанного проекта, указать на трудности в работе.

Список литературы оформляется с учетом библиографических требований. Оглавление должно быть согласовано с указанием номеров разделов, подразделов и страниц.

Общий объем расчетно-пояснительной записки не должен превышать 20-30 страниц.

### Список использованной литературы

1. Арсентьева Т.П. Справочник технолога молочного производства. Технологии и рецептуры. Т.4. Мороженное/ Т.П. Арсентьева – СПб.:ГИОРД, 2002.- 184с.
2. Сергеев, А.И. Программирование контроллеров систем автоматизации: учебное пособие / А.И. Сергеев, А.М. Черноусова, А.С. Русяев; Министерство образования и науки Российской Федерации, Оренбургский Государственный Университет. – Оренбург: ОГУ, 2017. – 126 с.: схем., табл., ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=481806> – Библиогр. в кн. – ISBN 978-5-7410-1649-7. – Текст: электронный.
3. Документация по CoDeSys. [http://www.owen.ru/catalog/codesys\\_v2/51162335](http://www.owen.ru/catalog/codesys_v2/51162335)
4. Первые шаги с CoDeSys; Русская редакция ПК «Пролог». – 2004. [Электронной ресурс] – URL:[http://www.kipshop.ru/CoDeSys/steps/first\\_steps\\_with\\_codesys\\_ru.pdf](http://www.kipshop.ru/CoDeSys/steps/first_steps_with_codesys_ru.pdf)
5. Руководство пользователя по программированию ПЛК в CoDeSys 2.3; Русская редакция ПК «Пролог». 2006. [Электронной ресурс] – URL: [http://www.kipshop.ru/CoDeSys/steps/codesys\\_v23\\_ru.pdf](http://www.kipshop.ru/CoDeSys/steps/codesys_v23_ru.pdf).
6. Визуализация CoDeSys. Дополнение к руководству пользователя по программированию ПЛК в CoDeSys 2.3; Русская редакция ПК «Пролог». 2006. [Электронной ресурс] – URL: [http://www.kipshop.ru/CoDeSys/steps/codesys\\_visu\\_v23\\_ru.pdf](http://www.kipshop.ru/CoDeSys/steps/codesys_visu_v23_ru.pdf).
7. Свободно программируемые устройства в автоматизированных системах управления: учебное пособие / И.Г. Минаев, В.В. Самойленко, Д.Г. Ушкур, И.В. Федоренко. – Ставрополь: Ставропольский государственный аграрный университет, 2016. – 168 с.: ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=484913> (дата обращения: 07.08.2020). – Библиогр. в кн. – ISBN 978-5-9596-1222-1. – Текст: электронный.
8. Петров, И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования: практическое пособие / И.В. Петров; ред. В.П. Дьяконов. – Москва: СОЛОН-ПРЕСС, 2004. – 254 с. – (Библиотека инженера). – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=117671> (дата обращения: 07.08.2020). – ISBN 5-98003-079-4. – Текст: электронный.
9. Основы программирования микропроцессорных контроллеров в цифровых системах управления технологическими процессами / В.С. Кудряшов, А.В. Иванов, М.В. Алексеев и др.; науч. ред. В.К. Битюков; Министерство образования и науки РФ, ФГБОУ ВПО «Воронежский государственный университет инженерных техно-

логий». – Воронеж: Воронежский государственный университет инженерных технологий, 2014. – 144 с.: ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=336026> (дата обращения: 07.08.2020). – Библиогр. в кн. – ISBN 978-5-00032-054-9. – Текст: электронный.

**Приложение А**  
**(рекомендуемое)**  
**Примерный список тем курсовой работы**

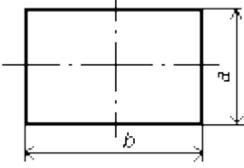
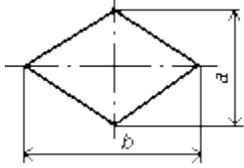
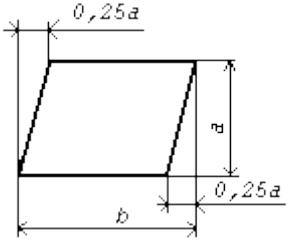
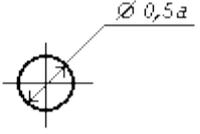
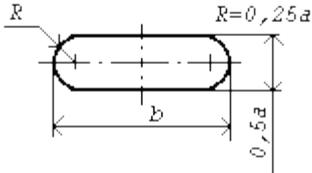
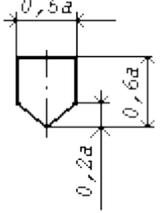
1. Автоматизация бланширователя рыбных консервов.
2. Автоматизация установки подготовки газа на компрессорной станции.
3. Автоматизация подготовки питьевой воды.
4. Автоматизация насосной станции второго подъема городского водоснабжения.
5. Автоматизация процесса очистки промышленных стоков рыбоконсервного комплекса.
6. Автоматизация хлебопекарного производства.
7. Автоматизация дефростера непрерывного действия.
8. Автоматизация процесса производства крабовых палочек.
9. Автоматизация рыбонабивочной установки.
10. Автоматизация линии производства рыбных пресервов.
11. Автоматизация холодильной камеры с рассольным охлаждением.
12. Автоматизация процесса посола рыбы.
13. Автоматизация мазутного терминала.
14. Автоматизация линии брожения виноградного сусла.
15. Автоматизация линии производства сливочного масла.
16. Автоматизация камеры горячего копчения мясопродуктов.
17. Автоматизация коптильной установки производства мясных полуфабрикатов.
18. Автоматизация процесса производства сыровяленых мясных продуктов.
19. Автоматизация газовой котельной.
20. Автоматизация процесса производства сыра.
21. Автоматизация процесса вакуумной упаковки продукта.
22. Автоматизация линии производства карамельных конфет.
23. Автоматизированная система управления тепличным хозяйством.
24. Автоматизация процесса полива и поддержания температуры в теплице.
25. Автоматизации производства шоколадной глазури.
26. Автоматизация однокорпусной выпарной установки для приготовления томатной пасты.
27. Автоматизация линии розлива пива.
28. Автоматизация камеры холодного копчения рыбы.
29. Автоматизация вакуумно-выпарной установки производство сгущенного молока.

30. Автоматизация линии производства йогурта.
31. Автоматизации линии производства мороженого.
32. Автоматизация сепаратора-сливкоотделителя.
33. Автоматизация паромасляной рыбообжарочной печи.
34. Автоматизация конвекционной обжарочной печи.
35. Автоматизация цеха производства кормов.
36. Автоматизация процесса мойки пастеризационной установки.
37. Автоматизация системы пожаротушения на станции хранения нефтепродуктов.
38. Автоматизация процесса производства творога.
39. Автоматизация газовой котельной.
40. Автоматизация термомасляного котла.
41. Автоматизация линии покраски кузовов автомобиля.
42. Автоматизация установки отбраковки дефектного участка доски.
43. Автоматизация буферного склада соевого масла.
44. АСУ ТП газораспределительной станции.
45. Модель экономичной системы охлаждения газа.
46. Система автоматизации реактора по производству биодизельного топлива.
47. Система управления линии термообработки колбасных изделий.
48. Система управления аппаратом воздушного охлаждения газа.
49. Система управления парокотельной установки ДЕ16/14 – Г.
50. Система управления котельной установки КВГМ – 50.
51. Система управления линии приготовления водки.
52. Система управления хлебопекарной печью.
53. Система управления вентиляцией зерносушилки.
54. САУ тепловым пунктом на примере КГТУ.
55. САУ тоннельной хлебопекарной печи.
56. САУ газоперекачивающим агрегатом.
57. Регулирование климата, умный дом.

Для обеспечения сквозного курсового и дипломного проектирования желательно выбирать тему проекта, которая будет являться и темой выпускной квалификационной работы бакалавра, согласованной с руководителем ВКР. Новый технологический объект, не указанный в вышеприведенном списке, выбранный самостоятельно в качестве темы курсовой работы и согласованный с преподавателем в бюджет иметь более высокий приоритет по сравнению с типовым.

**Приложение Б**  
**(рекомендуемое)**

Таблица 5. Обозначение условные графические БСА

Наименование	Обозначение и размеры ( $a/b=2/3$ )	Функция
1. Процесс		Выполнение операций или группы операций, в результате которых изменяется значение данных
2. Решение		Выбор направления выполнения алгоритма или программы в зависимости от условий
3. Ввод-вывод		Ввод или вывод информации
4. Соединитель		Указание связи между блоками на одной странице
5. Пуск - остановка		Начало, конец, прерывание процесса обработки данных или выполнения программы
6. Межстраничный соединитель		Указание связи между частями алгоритмов, расположенных на разных листах

**Приложение В**  
**(рекомендуемое)**  
**Принципиальная электрическая схема. Часть 1**

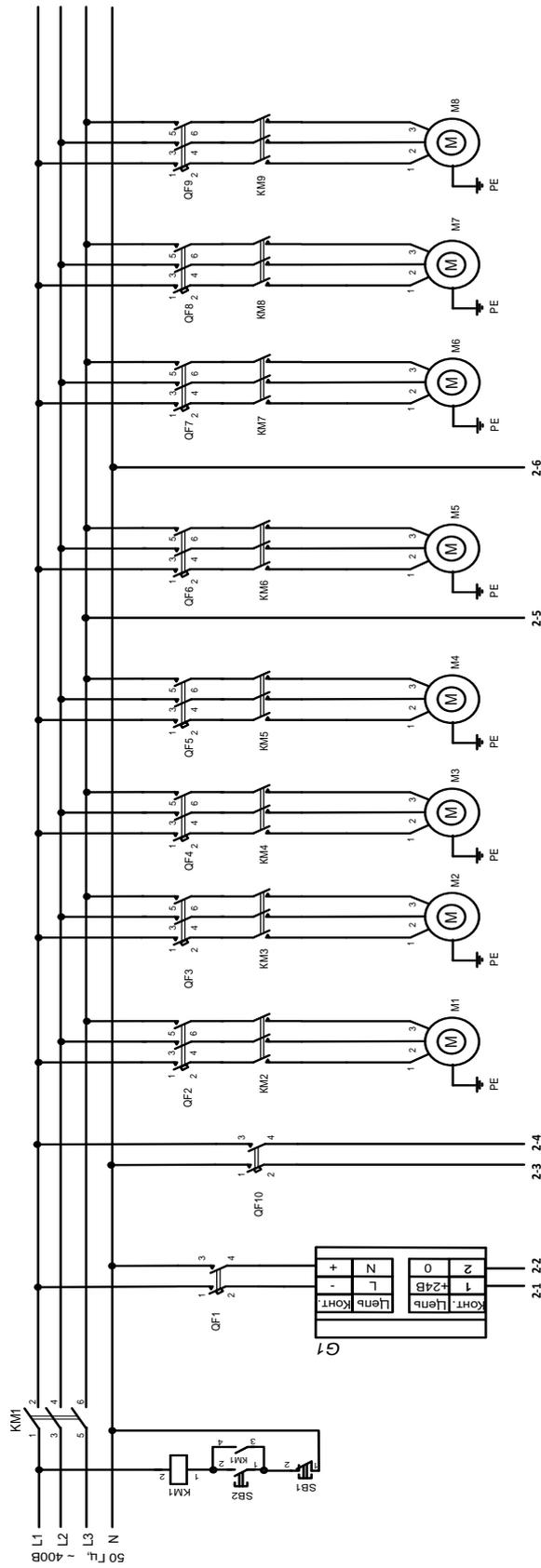


Рисунок 23 - Принципиальная электрическая схема. Часть 1

**Продолжение приложения В  
(рекомендуемое)  
Принципиальная электрическая схема. Часть 1**

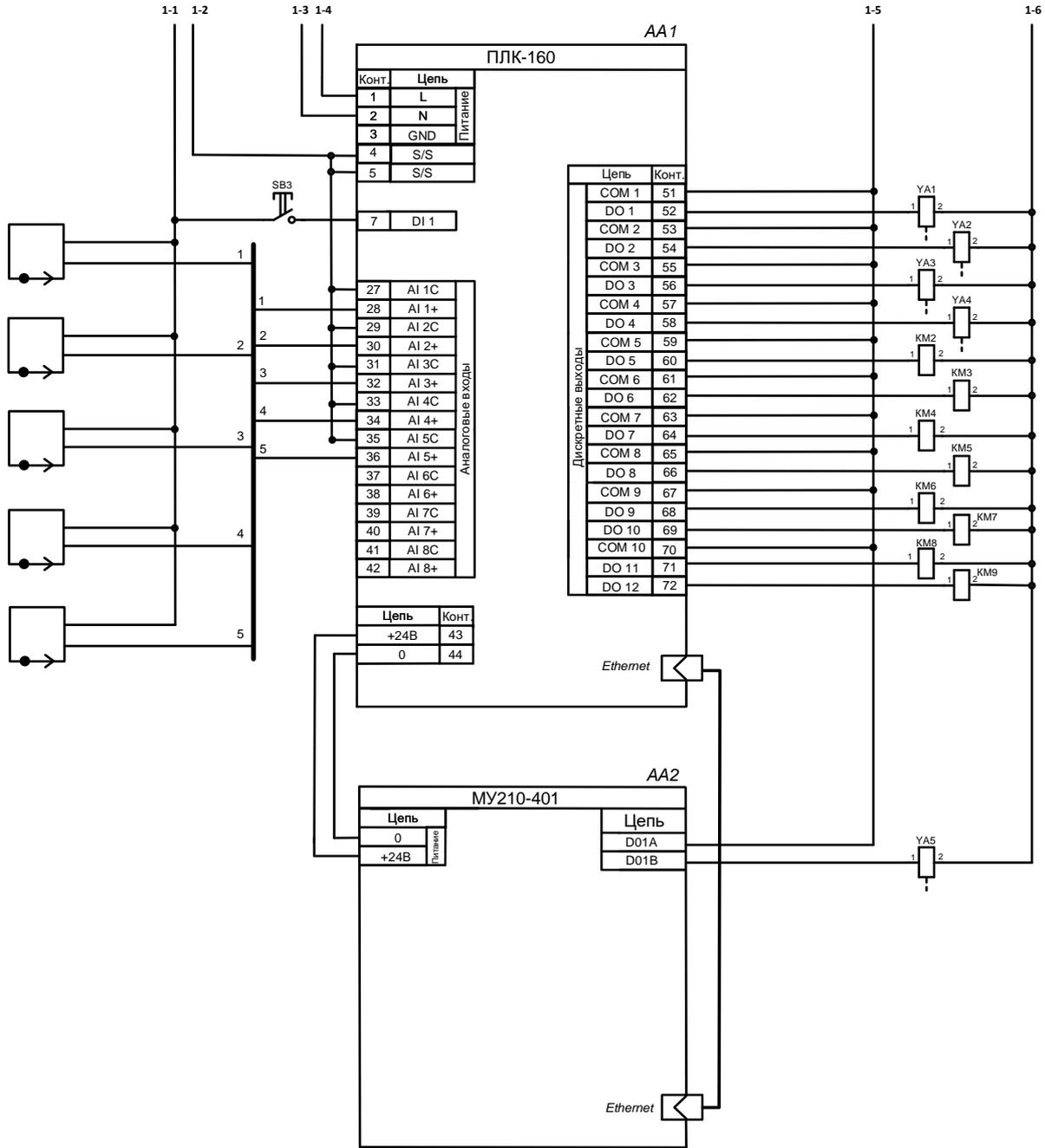


Рисунок 24 - Принципиальная электрическая схема. Часть 2



**Приложение Г  
(рекомендуемое)**

Таблица 6. Перечень элементов принципиальной электрической схемы

<b>Поз. обозначение</b>	<b>Наименование</b>	<b>Кол.</b>	<b>Примечание</b>
A1	Программируемый логический контроллер ПЛК 160 ОВЕН[МО2]	1	
A2	Модули дискретного вывода (Ethernet) МУ210-401	1	
A3	Персональный компьютер	1	
A4	Клапан запорно-регулирующий КРЗ РУ41 С	1	
A5	Автомат ОР1С для фасовки мороженого в вафельные стаканчики	1	
A6	Автомат ОЗС для заправки мороженого в вафельных стаканчиках	1	
G1, G2	Одноканальный блок питания БП30Б-ДЗ-24	2	
ВК1, ВК2, ВК3, ВК4	Термопреобразователь сопротивления ДТС 045М ОВЕН	1	
ВР1	Датчик давления ПД100И-ДВ ОВЕН	1	
ВЛ1	Поплавковый датчик уровня ПДУ-И, ТМ ОВЕН	1	
ВЛ2, ВЛ3	Ультразвуковой уровнемер Rosemount™ 3100	2	
М1-М3, М5	Винтовой пищевой насос ОНВ-М1	4	
М4, М6-М11	Электропривод 4МЦ2С-100	7	

**Продолжение приложения Г  
(рекомендуемое)**

Продолжение таблицы 6. Перечень элементов принципиальной  
электрической схемы

YA1	Клапан соленоидный ASCO/JOUCOMATIC	1	
YA2	Клапан соленоидный SMART SM55635S WATERVALVE	1	
YA3	Клапан соленоидный AR-SB125	1	
KK1-KK22	Тепловое реле ТРТ - 100	22	
QF1, QF13- QF19	ABB S210 Двухполюсной автоматиче- ский выключатель 1P 2A	7	
HL1-HL3	Сигнальная лампа красная, 220V AC/DC	3	
KM1, KM13- KM17	Магнитный пускатель ПМЕ121	6	
KM2-KM12	Магнитный пускатель ПМЕ222	11	
SB1, SB2, SB4-SB35	Красная кнопка без фиксации СК- 25312/R	34	
SB3	Красная кнопка с фиксацией	1	
SA1	Автоматический переключатель черный, 3 положения 2NO, с фиксацией МТВ2- BJZ133	1	

Приложение Д  
(рекомендуемое)  
Программа algorithm. Часть 1

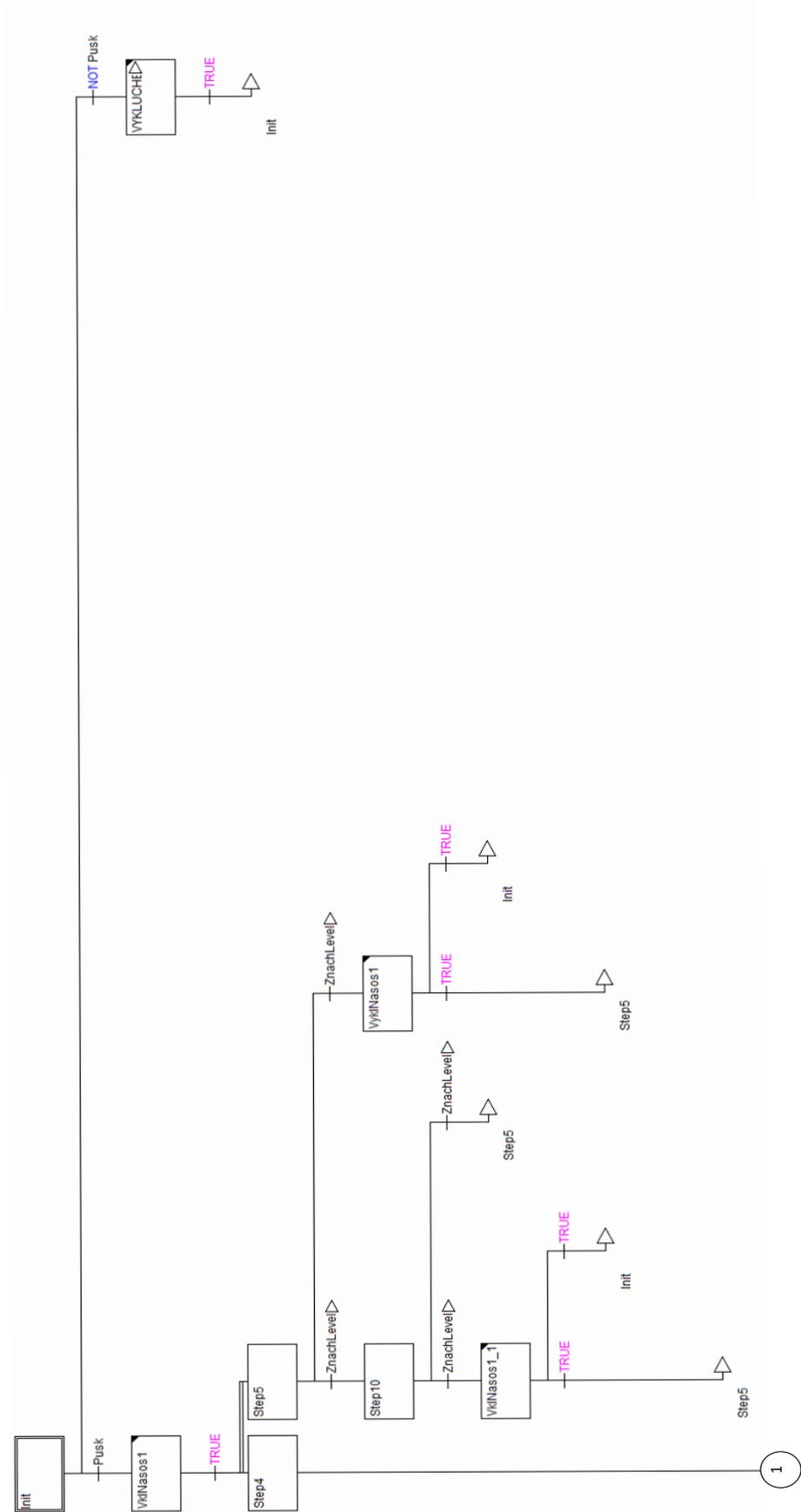


Рисунок 26 -.Программа algorithm. Часть 1

Продолжение приложения Д  
(рекомендуемое)  
Программа algorithm. Часть 2

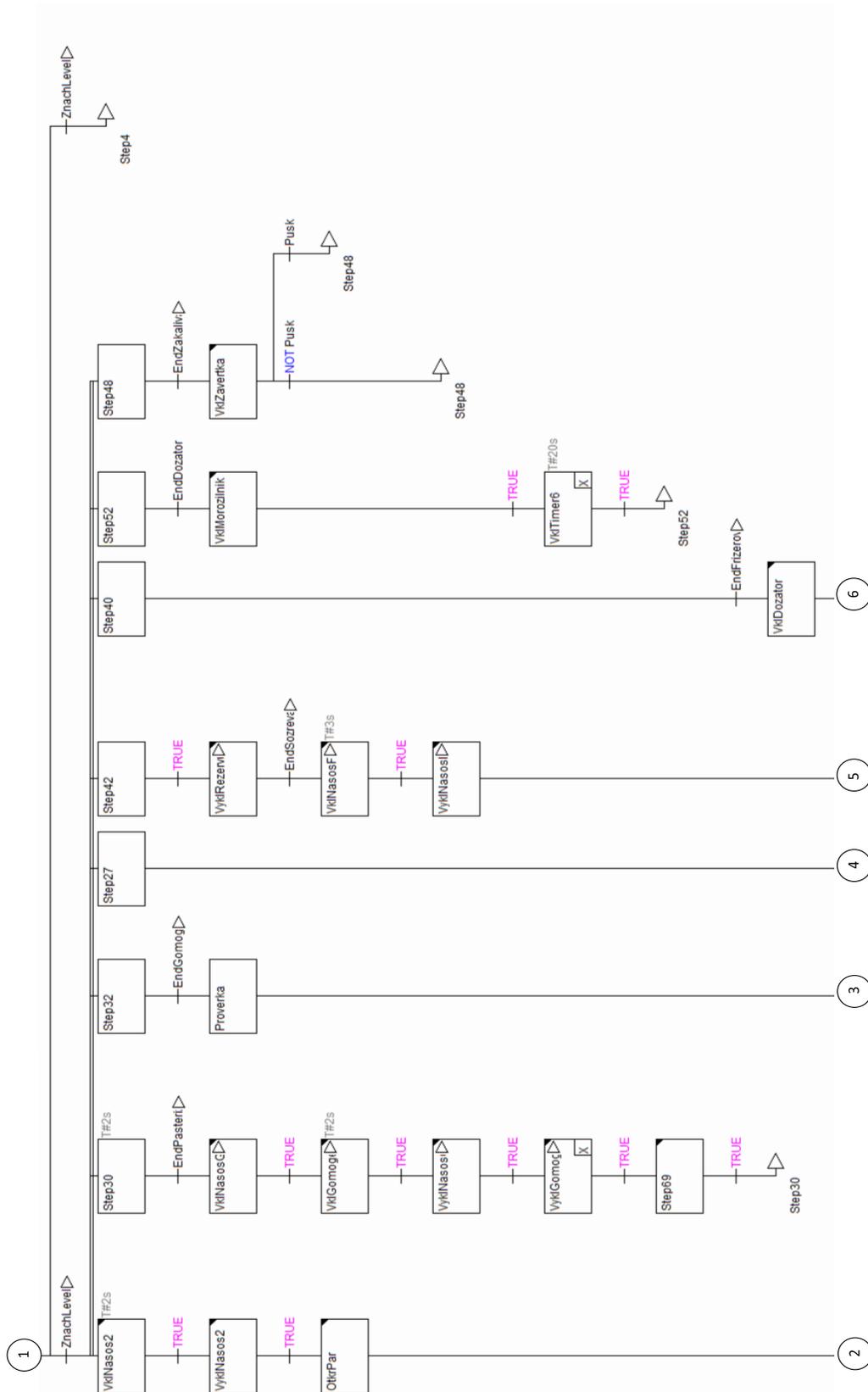


Рисунок 27 -.Программа algorithm. Часть 2

Продолжение приложения Д  
(рекомендуемое)  
Программа `algorithm`. Часть 3

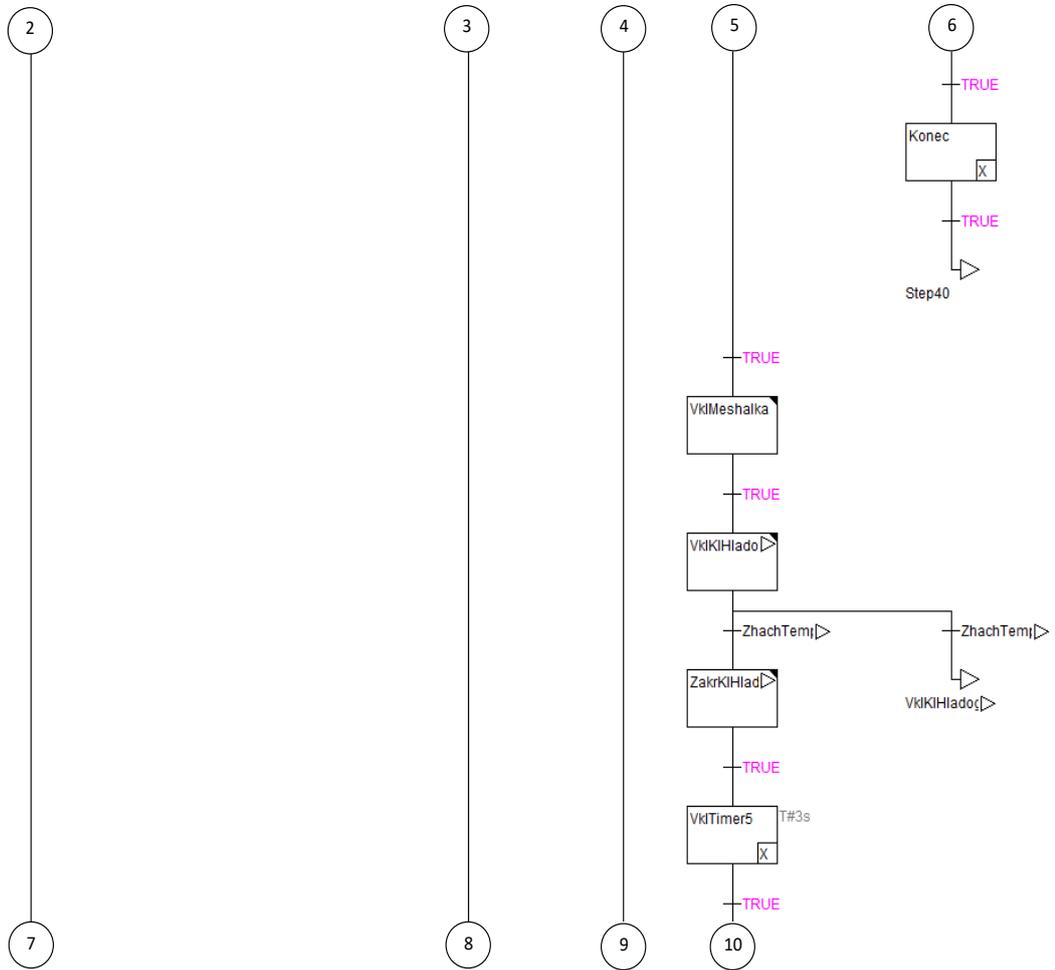


Рисунок 28 -Программа `algorithm`. Часть3

Продолжение приложения Д  
(рекомендуемое)  
Программа algorithm. Часть 4

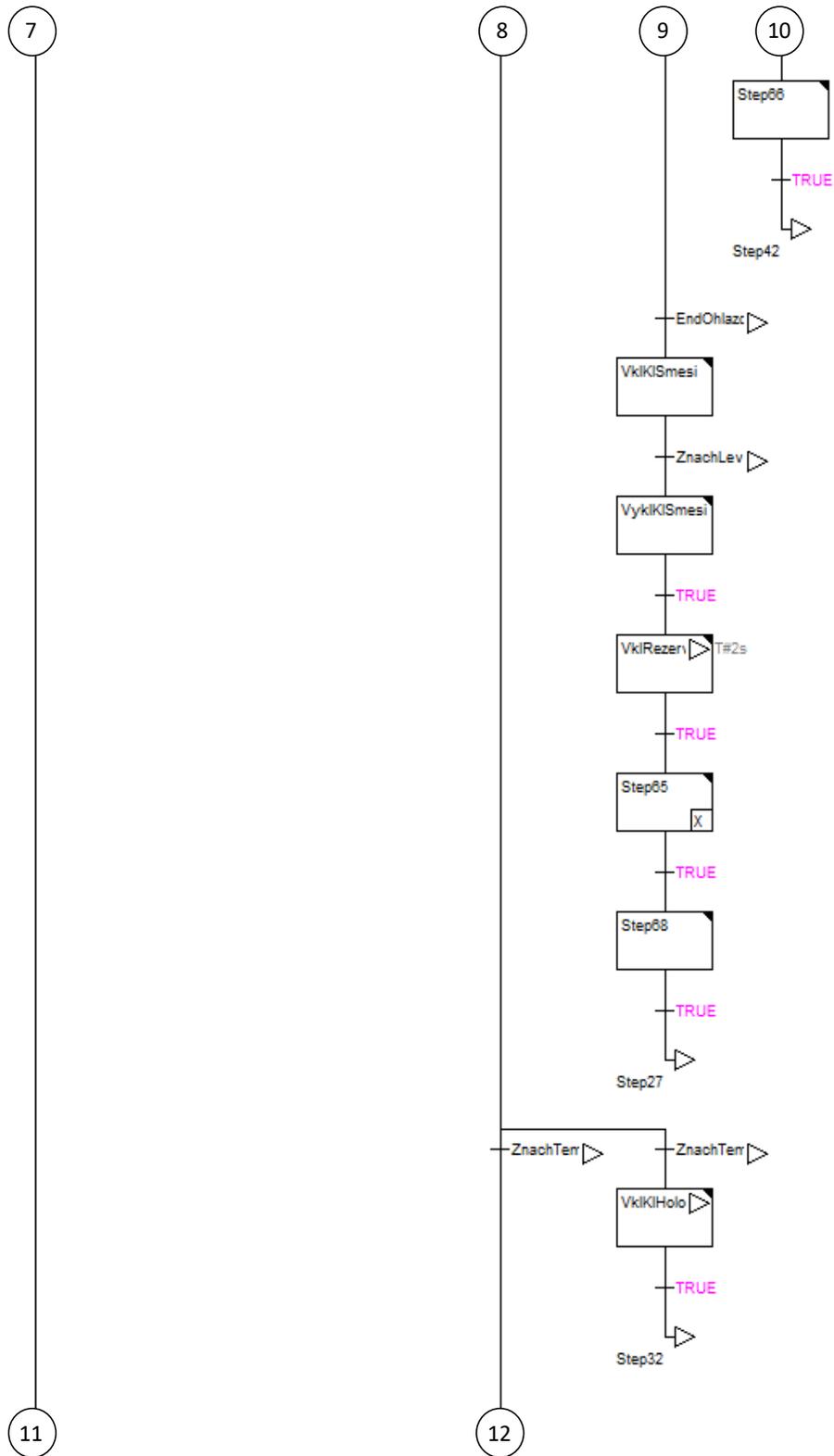


Рисунок 29 - Программа algorithm. Часть 4

Продолжение приложения Д

(рекомендуемое)  
Программа algorithm. Часть 5

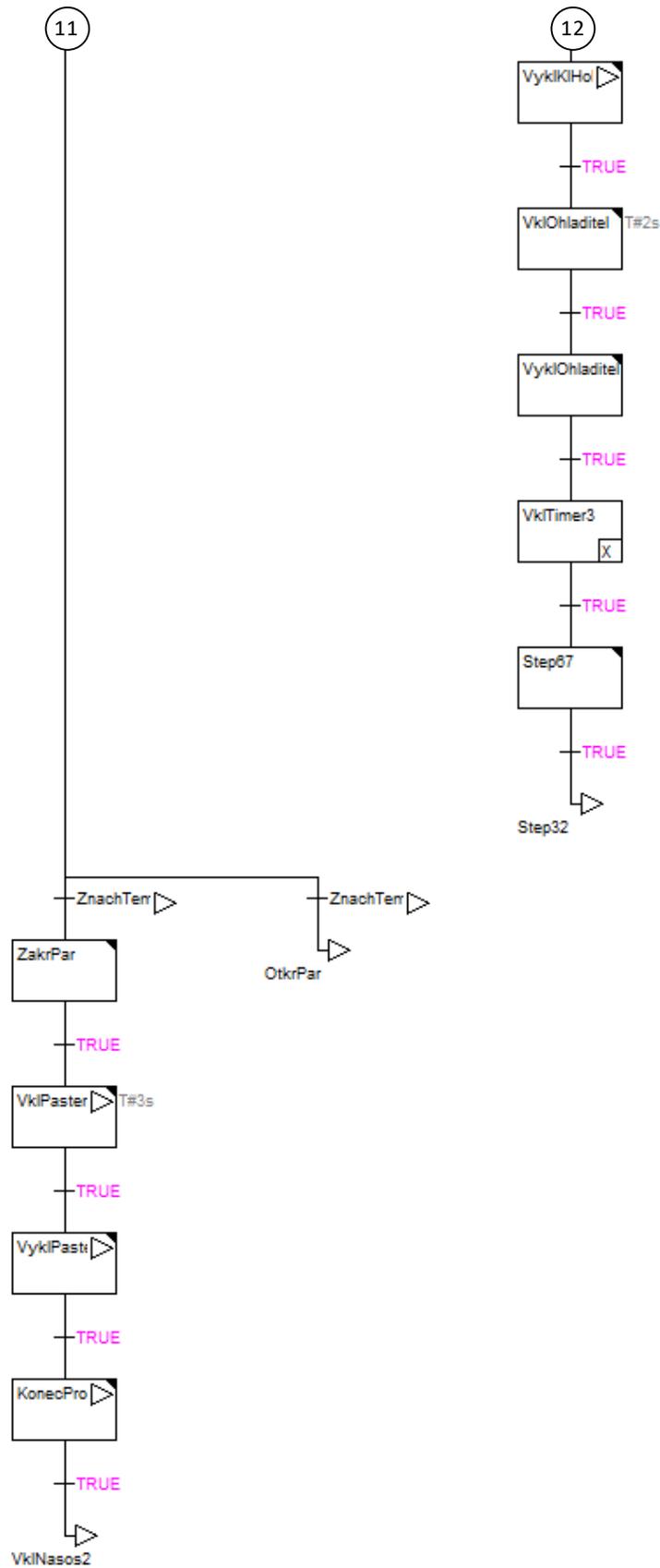


Рисунок 30 - Программа algorithm. Часть 5

(рекомендуемое)

Пример оформления титульного листа курсовой работы

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ

КАФЕДРА АВТОМАТИЗАЦИИ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

Курсовая работа  
допущена к защите

\_\_\_\_\_  
Руководитель  
«\_\_» \_\_\_\_\_ 20\_\_ г.

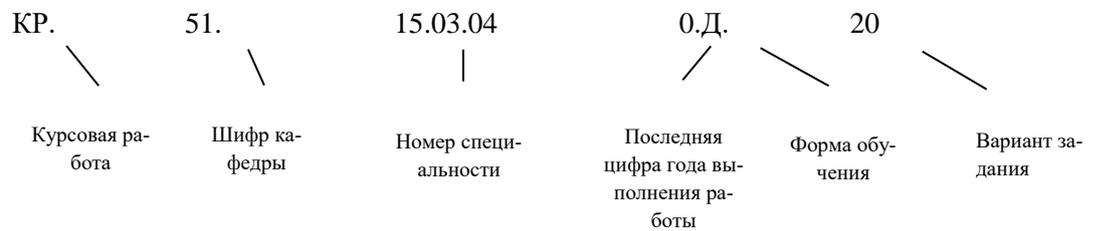
Курсовая работа защищена  
с оценкой \_\_\_\_\_

\_\_\_\_\_  
Руководитель  
«\_\_» \_\_\_\_\_ 20\_\_ г.

КУРСОВАЯ РАБОТА

по дисциплине

**Разработка программного обеспечения систем автоматизации и  
управления технологическими процессами.**



Пример записи: КР.51.15.03.04.0.Д.20

Работу выполнил  
студент гр. 17-АП  
Иванов А.А.

«\_\_» \_\_\_\_\_ 20\_\_ г.

«\_\_\_\_\_»

подпись

Калининград

2020

Учебное издание

Долгий Николай Алексеевич, Шамаев Евгений Петрович

**Н.А. Долгий, Е.П. Шамаев**

**Разработка программного обеспечения систем автоматизации и  
управления технологическими процессами.**

Учебно-методическое пособие по курсовой работе для студентов бакалавриата,  
обучающихся по направлению подготовки

15.03.04 «Автоматизация технологических процессов и производств»

*Редактор А. Иванова*

Подписано в печать \_\_. \_\_.2020 г. Формат 60x90 1/16. Уч.-изд. л. \_\_.

Печ. л. \_\_. Тираж 100 экз. Заказ №\_\_

Издательство федерального государственного бюджетного

образовательного учреждения

высшего образования

«Калининградский государственный технический университет».

236022, Калининград, Советский проспект,1