

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 15.03.04 Автоматизация технологических процессов и производств
 Отделение школы (НОЦ) – Отделение автоматизации и робототехники

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Кодогенератор из графического представления детерминированного конечного автомата алгоритма управления

УДК 04.896:004.415.2:004.4'422

Студент

Группа	ФИО	Подпись	Дата
8Т8Б	Гительман Владислав Сергеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Зебзеев А. Г.	к.т.н., доцент		

Со-руководитель (по разделу «Концепция стартап-проекта»)

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ШИП	Селевич О.С.	к.э.н., доцент		

КОНСУЛЬТАНТЫ:

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Мезенцева И. Л.	—		

Нормоконтроль

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Кузьминская Е. В.	к.т.н., доцент		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Е. И.	к.т.н., доцент		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код компетенции	Наименование компетенции
Общекультурные (универсальные) компетенции	
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов
УК(У)-9	Способен проявлять предприимчивость в практической деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи
УК(У)-10	Способен принимать обоснованные экономические решения в различных областях жизнедеятельности
УК(У)-11	Способен формировать нетерпимое отношение к коррупционному поведению
Общепрофессиональные компетенции	
ОПК(У)-1	Способен использовать основные закономерности, действующие в процессе изготовления продукции требуемого качества, заданного количества при наименьших затратах общественного труда
ОПК(У)-2	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-3	Способен использовать современные информационные технологии, технику, прикладные программные средства при решении задач профессиональной деятельности
ОПК(У)-4	Способен участвовать в разработке обобщенных вариантов решения проблем, связанных с автоматизацией производств, выборе на основе анализа вариантов оптимального прогнозирования последствий решения
ОПК(У)-5	Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью
Профессиональные компетенции	
ПК(У)-1	Способен собирать и анализировать исходные информационные данные для проектирования технологических процессов изготовления продукции, средств и систем автоматизации, контроля, технологического оснащения, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством; участвовать в работах по расчету и проектированию процессов изготовления продукции и указанных средств и систем с использованием современных информационных технологий, методов и средств проектирования
ПК(У)-2	Способен выбирать основные и вспомогательные материалы для изготовления изделий, способы реализации основных технологических процессов, аналитические и численные методы при разработке их математических моделей, методы стандартных испытаний по определению физико-механических свойств и технологических показателей материалов и готовых изделий, стандартные методы их проектирования, прогрессивные методы эксплуатации изделий
ПК(У)-3	Готов применять способы рационального использования сырьевых, энергетических и других видов ресурсов, современные методы разработки малоотходных, энергосберегающих и экологически чистых технологий, средства автоматизации технологических процессов и производств
ПК(У)-4	Способен участвовать в постановке целей проекта (программы), его задач при заданных критериях, целевых функциях, ограничениях, разработке структуры его взаимосвязей, определении приоритетов решения задач с учетом правовых и нравственных аспектов профессиональной деятельности, в разработке проектов

Код компетенции	Наименование компетенции
	изделий с учетом технологических, конструкторских, эксплуатационных, эстетических, экономических и управленческих параметров, в разработке проектов модернизации действующих производств, создании новых, в разработке средств и систем автоматизации, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством в соответствии с техническими заданиями и использованием стандартных средств автоматизации расчетов и проектирования
ПК(У)-5	Способен участвовать в разработке (на основе действующих стандартов и другой нормативной документации) проектной и рабочей технической документации в области автоматизации технологических процессов и производств, их эксплуатационному обслуживанию, управлению жизненным циклом продукции и ее качеством, в мероприятиях по контролю соответствия разрабатываемых проектов и технической документации действующим стандартам, техническим условиям и другим нормативным документам
ПК(У)-6	Способен проводить диагностику состояния и динамики производственных объектов производств с использованием необходимых методов и средств анализа.
ПК(У)-7	Способен участвовать в разработке проектов по автоматизации производственных и технологических процессов, технических средств и систем автоматизации, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством, в практическом освоении и совершенствовании данных процессов, средств и систем
ПК(У)-8	Способен выполнять работы по автоматизации технологических процессов и производств, их обеспечению средствами автоматизации и управления, готовностью использовать современные методы и средства автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством
ПК(У)-9	Способен определять номенклатуру параметров продукции и технологических процессов ее изготовления, подлежащих контролю и измерению, устанавливать оптимальные нормы точности продукции, измерений и достоверности контроля, разрабатывать локальные поверочные схемы и выполнять проверку и отладку систем и средств автоматизации технологических процессов, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством, а также их ремонт и выбор; осваивать средства обеспечения автоматизации и управления
ПК(У)-10	Способен проводить оценку уровня брака продукции, анализировать причины его появления, разрабатывать мероприятия по его предупреждению и устранению, по совершенствованию продукции, технологических процессов, средств автоматизации и управления процессами, жизненным циклом продукции и ее качеством, систем экологического менеджмента предприятия, по сертификации продукции, процессов, средств автоматизации и управления
ПК(У)-11	Способен участвовать: в разработке планов, программ, методик, связанных с автоматизацией технологических процессов и производств, управлением процессами, жизненным циклом продукции и ее качеством, инструкций по эксплуатации оборудования, средств и систем автоматизации, управления и сертификации и другой текстовой документации, входящей в конструкторскую и технологическую документацию, в работах по экспертизе технической документации, надзору и контролю за состоянием технологических процессов, систем, средств автоматизации и управления, оборудования, выявлению их резервов, определению причин недостатков и возникающих неисправностей при эксплуатации, принятию мер по их устранению и повышению эффективности использования
ПК(У)-18	Способен аккумулировать научно-техническую информацию, отечественный и зарубежный опыт в области автоматизации технологических процессов и производств, автоматизированного управления жизненным циклом продукции, компьютерных систем управления ее качеством,
ПК(У)-19	Способен участвовать в работах по моделированию продукции, технологических процессов, производств, средств и систем автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством с использованием современных средств автоматизированного проектирования, по разработке алгоритмического и программного обеспечения средств и систем автоматизации и управления процессами
ПК(У)-20	Способен проводить эксперименты по заданным методикам с обработкой и анализом их результатов, составлять описания выполненных исследований и подготавливать данные для разработки научных обзоров и публикаций
ПК(У)-21	Способен составлять научные отчеты по выполненному заданию и участвовать во внедрении результатов исследований и разработок в области автоматизации

Код компетенции	Наименование компетенции
	технологических процессов и производств, автоматизированного управления жизненным циклом продукции и ее качеством
ПК(У)-22	Способен участвовать: в разработке программ учебных дисциплин и курсов на основе изучения научной, технической и научно-методической литературы, а также собственных результатов исследований; в постановке и модернизации отдельных лабораторных работ и практикумов по дисциплинам профилей направления; способностью проводить отдельные виды аудиторных учебных занятий (лабораторные и практические), применять новые образовательные технологии, включая системы компьютерного и дистанционного обучения

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 15.03.04 Автоматизация технологических процессов и производств
 Уровень образования – Бакалавриат
 Отделение школы (НОЦ) – Отделение автоматизации и робототехники
 Период выполнения – Весенний семестр 2021/2022 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
...	Основная часть	60
...	Концепция стартап-проекта	20
	Социальная ответственность	20

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Зебзеев А. Г.	к.т.н., доцент		

Консультант (при наличии)

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОАР ИШИТР	Тутов И. А.	–		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Е. И.	к.т.н., доцент		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 15.03.04 «Автоматизация технологических процессов и производств»
 Отделение (НОЦ) – Отделение автоматизации и робототехники

УТВЕРЖДАЮ:
 Руководитель ООП
Громаков Е.И.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8Т8Б	Гительман Владислав Сергеевич

Тема работы:

Кодогенератор из графического представления детерминированного конечного автомата алгоритма управления	
Утверждена приказом директора (дата, номер)	№4713/с от 16.02.2022

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объект автоматизации – фрезер наклонных поверхностей надressорной балки железнодорожной тележки грузового вагона</p> <p>Цель работы: увеличение производительности разработки программного обеспечения автоматизированных систем управления.</p>
---	---

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ul style="list-style-type: none"> – описание методологии автоматного программирования; – разработка интерфейса программного обеспечения для генерации кода; – описание возможностей программного обеспечения – создание графической схемы алгоритма работы станка – применение разработанного кодогенератора для получения кода на основе графической схемы алгоритма – создание макроса на языке ST в среде Owen Logic с использованием сгенерированного кода – разработка программы на языке FBD с применением макроса на языке ST в Owen Logic для управления станком – тестирование разработанной программы в симуляции и на реальном ПР200 – анализ результатов работоспособности ПО
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<ul style="list-style-type: none"> – интерфейс окна проектирования алгоритмов программного обеспечения; – интерфейс окна отображения таблиц на основе графов-переходов; – результаты создания алгоритмов в виде скриншотов; – граф-схема алгоритма для фрезера наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона; – схема в Owen Logic с применением сгенерированного кода в виде макроса на языке ST;
<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Концепция стартап-проекта</p>	<p>Селевич Ольга Семеновна</p>
<p>Социальная ответственность</p>	<p>Мезенцева Ирина Леонидовна</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Заключение (Conclusion)</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОАР ИШИТР	Тутов И. А.	–		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Т8Б	Гительман Владислав Сергеевич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «КОНЦЕПЦИЯ СТАРТАП-ПРОЕКТА»

Студенту:

Группа 8Т8Б	ФИО Гительман В.С.
----------------	-----------------------

Школа	ИШИТР	Направление	15.03.04 «Автоматизация технологических процессов и производств»
Уровень образования	Бакалавр		

Перечень вопросов, подлежащих разработке:	
Проблема конечного потребителя, которую решает продукт, который создается в результате выполнения НИОКР	Программное обеспечение предназначено для автоматической генерации кода, повышения производительности разработки программного обеспечения автоматизированных систем управления
Способы защиты интеллектуальной собственности	Регистрация программы для ЭВМ
Объем и емкость рынка	Объем российского рынка: 161 млн. долларов; Потенциальная емкость российского рынка: примерно 290 млн. долларов).
Современное состояние и перспективы отрасли, к которой принадлежит представленный в ВКР продукт	Наблюдается увеличение спроса на разработчиков программного обеспечения для АСУ ТП и, как следствие, увеличение спроса на программные продукты, способные их заменить либо полностью, либо частично
Себестоимость продукта	963200 рублей
Конкурентные преимущества создаваемого продукта	Уменьшение времени разработки программ для систем автоматики, низкая стоимость, высокая доступность, возможность адаптации генератора кода под конкретное предприятие, низкие системные требования
Сравнение технико-экономических характеристик продукта с отечественными аналогами	На основании конкурентных преимуществ
Целевые сегменты потребителей продукта	Применение при проектировании и создании автоматизированных систем управления технологическими процессами, использование при разработке алгоритмов и программ для систем автоматики
Бизнес-модель проекта	Модель по А. Остервальдеру и И. Пинье
Производственный план	Каждые полгода обновления программного продукта
План продаж	Лицензия на ПО + подписка на обновления (либо просто лицензия на ПО)

Перечень графического материала:	
При необходимости представить эскизные графические материалы	Матрица по А. Остервальдеру и И. Пинье

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант по разделу «Концепция стартап-проекта» (со-руководитель ВКР):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ШИП	Селевич О.С.	к.э.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Т8Б	Гительман В.С.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8Т8Б		Гительман Владислав Сергеевич	
Школа	ИШИТР	Отделение (НОЦ)	ОАР
Уровень образования	Бакалавриат	Направление/специальность	<i>15.03.04. Автоматизация технологических процессов и производств</i>

Тема ВКР:

<i>Кодогенератор из графического представления детерминированного конечного автомата алгоритма управления</i>	
Исходные данные к разделу «Социальная ответственность»:	
<p>Введение</p> <ul style="list-style-type: none"> – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации 	<p><i>Объект исследования: программное обеспечение для генерации программного кода из алгоритма управления.</i></p> <p><i>Область применения: разработка программ для систем автоматики и для станков на предприятиях.</i></p> <p><i>Рабочая зона: лаборатория отделения ОАР, ИШИТР.</i></p> <p><i>Размеры помещения: 7.5*5.5м</i></p> <p><i>Количество и наименование оборудования рабочей зоны: ноутбук, лабораторный стол, Ethernet-кабель, лампа настольная, блок питания.</i></p> <p><i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне: разработка программного обеспечения для генерации программного кода, написание программы на языке FBD в среде программирования Owen Logic с использованием сгенерированного кода, проверка работоспособности программного обеспечения.</i></p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<p>ГОСТ 12.0.002-2014 ССБТ. Термины и определения.</p> <p>Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ стоя. Общие эргономические требования. ГОСТ 12.2.033-78.</p> <p>Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования. ГОСТ 12.2.032-78.</p>
<p>2. Производственная безопасность при разработке проектного решения</p> <ul style="list-style-type: none"> – Анализ выявленных вредных и опасных производственных факторов; 	<p>Вредные факторы:</p> <ul style="list-style-type: none"> – Умственное перенапряжение; – Отклонение показателей микроклимата; – Перенапряжение зрительного анализатора; – Превышение уровня шума; – Недостаточная освещенность рабочей зоны;

<p>– Обоснование мероприятий по снижению воздействия.</p>	<p>– Эмоциональные перегрузки; – Отсутствие естественного света. Опасные факторы: – Повышенное значение напряжения в электрической цепи, которое может пройти через тело человека;</p>
<p>3. Экологическая безопасность при разработке проектного решения</p>	<p>Воздействие на литосферу: при замене комплектующих ноутбука утилизация неисправных или устаревших деталей; Воздействие на гидросферу: попадание отходов при утилизации деталей в сточные воды; Воздействие на атмосферу: вредные выбросы на электростанциях, вырабатывающих используемое электричество, питающее ноутбук, а также загрязнение воздуха при производстве оборудования рабочей зоны.</p>
<p>4. Безопасность в чрезвычайных ситуациях при разработке проектного решения</p>	<p>Возможные ЧС: пожар, разрушение зданий в результате разрядов атмосферного электричества, ураган, землетрясение. Наиболее типичная ЧС: пожар.</p>
<p>Дата выдачи задания для раздела по линейному графику 02.04.2022</p>	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
<p>Старший преподаватель ООД ШБИП</p>	<p>Мезенцева Ирина Леонидовна</p>	<p>-</p>		<p>02.04.2022</p>

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
<p>8Т8Б</p>	<p>Гительман Владислав Сергеевич</p>		<p>02.04.2022</p>

Реферат

Пояснительная записка содержит 120 страниц машинописного текста, 13 таблиц, 26 рисунков, 65 использованных источников, 10 приложений.

Цель работы – разработка программного инструмента упрощения разработки ПО для систем автоматизации с функцией генерации кода из графического представления детерминированного конечного автомата. На примере станка, фрезера наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона, подтверждается работоспособность данного инструмента.

В результате было разработано программное обеспечение для генерации кода, используемого с целью управления фрезерным станком на базе программируемого реле ОВЕН ПР200.

Разработанное программное обеспечение позволит увеличить производительность разработки кода для автоматизированных систем управления, сократить объём трудозатрат программиста по написанию кода вручную.

Пояснительная записка выполнена с помощью текстового редактора Microsoft Word 2019. Помимо этого, для выполнения работы использовались такие программные продукты как: Embarcadero RAD Studio 2010, Microsoft Visio 2019, OwenLogic.

Ниже представлен перечень ключевых слов:

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИНТЕРФЕЙС, СТАНОК, ФРЕЗЕР, АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ, ГЕНЕРАЦИЯ КОДА, ПРОГРАММИРУЕМОЕ РЕЛЕ, ГРАФ, АЛГОРИТМ, КОНЕЧНЫЙ АВТОМАТ, СОСТОЯНИЕ, ПЕРЕХОД, КОДОГЕНЕРАЦИЯ.

Содержание

Оглавление	
Реферат	13
Содержание	14
Термины и определения.....	17
Обозначения и сокращения	18
Введение	19
1 Техническое задание.....	22
1.1. Назначение программного обеспечения.....	22
1.2. Основные цели и задачи разработки ПО	22
1.3. Требования к программному обеспечению.....	24
2. Основная часть.....	25
2.1. Описание принципов построения программного обеспечения на основе автоматного программирования	25
2.2. Пользовательский интерфейс программного обеспечения	29
2.3. Обеспечение вывода таблицы списка ребер на основе графов.....	36
2.3.1. Генерация кода на языке ST.....	39
2.4. Применение разработанного программного обеспечения.....	39
2.4.1. Реализация алгоритма работы станка	39
2.4.2. Генерация кода из алгоритма работы станка	46
2.4.3. Применение сгенерированного кода на языке ST в программном обеспечении Owen Logic.....	46
2.4.4. Перспективы дальнейшего развития программного обеспечения	48
3. Концепция стартап-проекта	50
3.1 Основные качества продукта, решаемая продуктом проблема	50
3.2 Защита интеллектуальной собственности	50
3.3 Объем и емкость рынка	52
3.4 Анализ современного состояния и перспектив отрасли	55
3.5 Расчет себестоимости продукта.....	56
3.6 Конкурентные преимущества продукта и обзор технико-экономических	

характеристик аналогов	58
3.7 Описание целевых сегментов потребителей	61
3.8 Бизнес-модель проекта и стратегия продвижения.....	62
3.9 Экономическое обоснование стартап-проекта.....	63
3.9.1 Инвестиционные издержки	63
3.9.2 Ценовая политика	64
3.9.3 Показатели эффективности проекта.....	66
4. Социальная ответственность	72
4.1 Правовые и организационные вопросы обеспечения безопасности.....	73
4.1.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства.....	73
4.1.2 Организационные мероприятия при компоновке рабочей зоны.....	73
4.2 Производственная безопасность	74
4.2.1 Отклонение показателей микроклимата	76
4.2.2 Превышение уровня шума.....	77
4.2.3 Недостаточная освещенность рабочей зоны	77
4.2.4 Повышенное значение напряжения в электрической цепи, которое может пройти через тело человека	78
4.2.5 Умственное перенапряжение	79
4.2.6 Перенапряжение зрительного анализатора	79
4.2.7 Эмоциональные перегрузки	79
4.2.8 Отсутствие естественного света	80
4.3 Экологическая безопасность.....	81
4.3.1 Влияние объекта исследования на гидросферу.....	81
4.3.2 Влияние объекта исследования на литосферу.....	82
4.3.3 Влияние объекта исследования на атмосферу	82
4.4 Безопасность в чрезвычайных ситуациях.....	83
4.4.1 Анализ вероятных ЧС, которые может инициировать объект исследований	83
4.4.2 Обоснование мероприятий по предотвращению ЧС	83
Вывод по разделу.....	84

Заключение.....	86
Conclusion	87
Список публикаций студента.....	88
Список литературы.....	89
ПРИЛОЖЕНИЕ А	97
Листинг А.1 – Сгенерированный код на языке ST	97
ПРИЛОЖЕНИЕ Б.....	99
Пульт управления фрезером наклонных поверхностей	99
ПРИЛОЖЕНИЕ В.....	100
Граф-схема алгоритма управления станком для автомата №1 (часть 1).....	100
ПРИЛОЖЕНИЕ Г	101
Граф-схема алгоритма управления станком для автомата №1 (часть 2).....	101
ПРИЛОЖЕНИЕ Д.....	103
Граф-схема алгоритма управления станком для автомата №1 (часть 3).....	103
ПРИЛОЖЕНИЕ Е.....	105
Граф-схема алгоритма управления станком для автомата №2.....	105
ПРИЛОЖЕНИЕ Ж.....	106
Листинг Ж.1 – Сгенерированный код на языке ST для станка-фрезера.....	106
ПРИЛОЖЕНИЕ И	113
Регистрация программы для ЭВМ (продукт FSMLogGen)	113
ПРИЛОЖЕНИЕ К.....	114
Таблица конкурентных решений	114
ПРИЛОЖЕНИЕ Л.....	120
Модель по А. Остервальдеру и И. Пинье	120

Термины и определения

В данной работе применены следующие термины с соответствующими определениями:

автоматизированная система (АС): Комплекс аппаратных и программных средств, предназначенных для управления различными процессами в рамках технологического процесса. Термин автоматизированная, в отличие от термина автоматическая подчеркивает сохранение за человеком-оператором некоторых функций.

интерфейс: Совокупность средств (программных, технических, лингвистических) и правил взаимодействия (управления, контроля и т.д.) между элементами системы.

конечный автомат: Математическая модель, позволяющая описывать пути изменения состояния объекта в зависимости от его текущего состояния и входных данных, при условии, что общее количество состояний конечно.

автоматное программирование: Парадигма программирования, при использовании которой программа или её фрагмент осмысливается как модель какого-либо формального автомата.

карта Карно: Графический способ минимизации переключательных (булевых) функций, обеспечивающий относительную простоту работы с большими выражениями.

детерминированный конечный автомат: конечный автомат, который исключает самопроизвольные переходы из состояния в состояние без внешних воздействий.

граф: математическая абстракция реальной системы любой природы, объекты которой обладают парными связями.

стартовое состояние: состояние, откуда конечный автомат начинает свою работу.

список ребер – список, в каждой строке которого записаны две смежные вершины и вес, соединяющего их ребра.

Обозначения и сокращения

В данной работе применены следующие обозначения и сокращения:

FBD – Functional Block Diagram – диаграммы функциональных блоков;

ST – Structured Text – Структурированный Текст;

IL – Instructional lists – Список инструкций;

SFC – Sequential Functional Chart – Последовательность Функциональных Диаграмм;

LD – Ladder Diagram – Лестничная Диаграмма;

АС – автоматизированная система;

АСУ ТП – автоматизированная система управления технологическими процессами;

ДКА – детерминированный конечный автомат;

КА – конечный автомат;

ПК – персональный компьютер;

ПО – программное обеспечение;

ПР – программируемое реле;

ФБ – функциональный блок;

ЭВМ – электронно-вычислительная машина.

ПЛК – программируемый логический контроллер.

МЭК – Международная электротехническая комиссия.

ЯП – язык программирования.

ГСА – граф-схема алгоритма.

Введение

Выпускная квалификационная работа посвящена разработке графического инструмента для автоматизации разработки программного обеспечения с использованием формальных моделей. На сегодняшний день разрабатывается большое количество программных средств для различных типов задач, в частности, существующих в области автоматизации. Разрабатываемый инструмент позволит упростить процессы написания программного кода для АСУ ТП.

Основным этапом разработки ПО является написание исходного кода. Один из возможных подходов к автоматизации данного процесса – использование автоматного программирования. Данная парадигма подразумевает формализацию поведения будущего продукта в виде конечного автомата. Такая формализация позволяет автоматизировать как процесс написания исходного программного кода, так и этап тестирования.

Автоматное программирование [1], иначе называемое «программирование от состояний» или «программирование с явным выделением состояний» – это парадигма программирования, при использовании которой программа или ее фрагмент осмысливается как модель какого-либо формального автомата. Является методом создания программ, поведение которых можно описать через автомат. В автоматном программировании используются такие понятия, как: входное воздействие, состояние и выходная реакция. Связь этих понятий осуществляется при построении графов-переходов автоматов.

Определяющими для автоматного программирования являются следующие особенности:

1. временной период выполнения программы разбивается на шаги автомата, каждый из которых представляет собой выполнение определенной (одной и той же для каждого шага) секции кода с единственной точкой входа; такая секция может быть оформлена, например, в виде отдельной функции и может быть разделена на подсекции, соответствующие отдельным состояниям

или категориям состояний

2. передача информации между шагами автомата осуществляется только через явно обозначенное множество переменных, называемых состоянием автомата; между шагами автомата программа, не может содержать неявных элементов состояния. Полностью выполнение кода в автоматном стиле представляет собой цикл (возможно, неявный) шагов автомата.

Одним из терминов автоматного программирования является «Switch-технология» [1, 2], предназначенная для разработки систем логического управления на базе конечных автоматов. Она охватывает процесс спецификации, проектирования, реализации, отладки, верификации, документирования и сопровождения. В программной реализации switch-технология реализуется с использованием case-операторов (например, в C, C++).

Автоматное программирование находит свое применение и в области автоматизации при написании программного кода для программируемых логических контроллеров.

Одними из основных языков программирования таких устройств, как ПЛК, ПР, предназначенных для управления технологическими процессами, являются языки стандарта МЭК 61131-3-2016. Каждый из языков обладает недостатками при написании кода. Такой язык, как FBD, используемый для построения блок-диаграмм, характеризуется сложностью внесения исправлений в последовательную логику работы алгоритма и громоздкостью схем. Язык LD является наглядным средством программирования последовательных операций (срабатывания реле, ключей), но с возрастанием сложности алгоритма понимание его принципов затрудняется. Разработка кода на языке ST отличается монотонностью процесса и не обладает достаточной наглядностью для человека с минимальными навыками программирования. Язык IL считается устаревшим и не рекомендуется к использованию. Более того, он исключен в третьей редакции ИЕС 61131-3:2013 [3]. И последним языком современного стандарта ГОСТ Р МЭК 61131-3-2016

является SFC. Данный язык обладает достаточной наглядностью, но при этом отображает лишь структуру организации кода алгоритма, промежуточные шаги и переходы которого пишутся на других языках программирования стандарта МЭК 61131-3-2016. Таким образом, при использовании данных языков не исключаются возможности допущения ошибок из-за отсутствия наглядности или человеческого фактора.

Описание систем со сложным поведением – довольно непростая задача и без использования диаграмм приводит к трудностям на всех этапах жизненного цикла [4]. Достоинство автоматного программирования по сравнению с языками стандарта МЭК 61131-3-2016 заключается в том, что с использованием данной парадигмы существует возможность создания программного обеспечения, позволяющего пользователю задавать диаграммы (графы переходов), что делает логику того или иного алгоритма наглядной и прозрачной. Более того, с использованием автоматных программ упрощается внесение изменений в них специалистами в предметной области, не являющихся профессиональными программистами. Ещё одним достоинством автоматного программирования с точки зрения графического отображения алгоритма, является возможность для программиста на время забыть особенности программного обеспечения, графически отобразив их с тем, чтобы спустя определенное время, вспомнить принципы построения кода.

После создания графической схемы процесса производится её преобразование в программный код по заранее заданному шаблону, что позволяет значительно ускорить разработку ПО для автоматизированных систем управления.

Актуальность выбранной темы обусловлена тем, что в настоящее время необходимо внедрять более эффективные методы разработки программного обеспечения, направленные на увеличение производительности создания программ для АСУ и минимизации человеческого фактора при написании кода. Стоит также отметить, что в настоящее время существует нехватка специалистов в области ИТ. С применением кодогенерации в перспективе

можно существенно сократить потребность компаний в IT-специалистах. Предлагаемое решение можно считать эффективной современной технологией разработки программного обеспечения.

1 Техническое задание

1.1. Назначение программного обеспечения

Разрабатываемое программное обеспечение предназначено для графического представления детерминированного КА с последующей генерацией кода на текстовом языке программирования ST для дальнейшей загрузки в устройства управления технологическими процессами (ПЛК, ПР и др.).

1.2. Основные цели и задачи разработки ПО

Целью разработки ПО является упрощение процессов разработки программного обеспечения для автоматизированных систем управления.

Основные задачи по созданию интерфейса и функционала программного обеспечения, а также тестированию его работоспособности:

- создание возможности отрисовки таких графических примитивов, как прямоугольник с закругленными углами и стрелка (состояние и переход соответственно);
- разработка пунктов меню для генерации кода на текстовых языках программирования;
- создание таблиц, отражающих концепцию построенного в программе ДКА;
- осуществление вывода логических выражений из графа;
- реализация меню для создания нового проекта, сохранения и открытия проектов, а также изменения цвета заголовков состояний;
- предоставление пользователю возможности использовать горячие клавиши при просмотре таблиц на основе графов (ctrl+T), для выхода из проекта (ctrl+X), переименования проекта (ctrl+R), сохранения проекта в виде картинки

(ctrl+F), а также сохранение самого проекта (ctrl+S), с последующей возможностью его открытия (ctrl+O) и создания нового проекта (ctrl+N);

- осуществление возможности использования горячей клавиши ctrl+G при создании параллельных автоматов с последующим выбором цвета всех заголовков состояний нового параллельного графа.

- создание возможности для пользователя обозначить стартовое состояния (изображается обособленной стрелочкой с небольшой закрашенной окружностью);

- предоставление возможности для пользователя перемещать состояния посредством зажатия ЛКМ и переноса в нужную точку пространства на форме;

- создание возможности пользователю задавать приоритетность переходов из одного состояния в другое (обозначается окружностью с номером приоритета в том месте, где пользователь начал проводить переход);

- реализация редактирования состояний и переходов (предоставление возможности изменения имени и описания состояния, а также наименование перехода);

- осуществление возможности переноса состояний в режиме их редактирования;

- при входе в режим редактирования предоставить возможность удаления состояния и всех связанных с ним переходов. При этом реализовать возможность удаления обособленных переходов между состояниями;

- создание возможности в режиме редактирования перехода перемещать наименования переходов в удобное место на форме;

- при перемещении курсора по границе состояния предоставить пользователю возможность наблюдать его в виде окружности малых размеров, что помогает понять в каком месте существует потенциальная возможность по созданию перехода;

- при клике левой кнопкой мыши на форме по умолчанию пользователю дать возможность создания состояния;

- реализовать удаление через красный крестик в режиме

редактирования, нажатие на который удаляет либо переход, либо состояние и все связанные с ним стрелки;

– разработать программу, реализующую управление фрезером наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона. Применить сгенерированный код для создания макроса на языке ST в среде программирования Owen Logic. Протестировать созданное ПО на реальном ПР200 и убедиться в его работоспособности.

1.3. Требования к программному обеспечению

Используемое ПО должно соответствовать техническому заданию.

Основной функционал должен включать:

1. Возможности сохранения проекта и его открытия;
2. Возможности редактирования состояний и переходов;
3. Возможности удаления графических примитивов;
4. Генерацию кода на различных текстовых языках программирования.
5. Возможность поддержки параллельных автоматов.

Программные средства должны быть достаточными для обеспечения заданного функционала станка при загрузке кода в ПЛК или ПР. При этом правки в генерируемый код должны быть минимальными. Генерация с использованием автоматной парадигмы должна обеспечить снижение требований к погруженности программиста в производственные процессы (сферу технолога) при создании программ для ПЛК [6], уменьшить трудозатраты программиста и увеличить эффективность его труда при написании программного обеспечения АСУ ТП.

Хотя бы один из языков программирования для генерации должен соответствовать стандарту ГОСТ Р МЭК 61131-3.

2. Основная часть

2.1. Описание принципов построения программного обеспечения на основе автоматного программирования

Программирование ПЛК осуществляется на языках стандарта МЭК 61131-3-2016. Одним из языков данного стандарта является ST. В отличие от графических языков программирования (FBD, LD), язык ST характеризуется простотой переноса с одной среды программирования в другую, осуществляемого посредством копирования. При этом промежуточный результат для переноса на разные системы возможно сохранить в обычном текстовом документе. В свою очередь программу в графическом виде (на языках FBD, LD, SFC) невозможно перемещать через копирование между разными средами программирования. Более того, язык SFC и вовсе служит для организации кода, а не для его написания. При этом язык IL, ранее входивший в стандарт, хоть и является текстовым языком программирования, но он исключен в третьей редакции МЭК 61131-3:2013. Исходя из рассмотренных преимуществ языка ST и недостатков других языков, необходимо сделать упор на генерацию кода из графов именно на языке ST.

Более того, компания Овен, производящая программируемые реле, внедряет язык ST в таких моделях ПР, как ПР100, ПР102, ПР200 в среде разработки Owen Logic. Программируемые реле являются более дешевыми устройствами по сравнению с ПЛК. Поэтому ПР выгоднее использовать в локальных задачах автоматизации. Таким образом, внедрение языка ST для ПР в Owen Logic существенно упростит программирование ПР, и данный шаг позволит применить кодогенератор на практике для управления различными станками, в частности, фрезером наклонных поверхностей адресной балки железнодорожной тележки грузового вагона.

Разработка программного обеспечения включает в себя этап построения алгоритмов. В настоящее время существуют такие инструменты создания алгоритмов, как блок-схемы и автоматные модели. Блок-схемы придуманы на этапе становления вычислительной техники. Преимущество автоматной

модели в том, что графы-переходов автоматной модели более наглядно и компактно отображают алгоритм, чем блок-схема. При этом автоматная модель является не просто рисунком, отображающим алгоритм, но и является математической моделью логики системы, а также спецификацией для последующей эволюции ПО. На основе автоматной модели можно в автоматическом режиме получить готовый программный код, который может быть загружен в программируемый логический контроллер, что значительно упрощает процессы разработки ПО. При этом уменьшается человеческий фактор при разработке ПО. На основе автоматных моделей разрабатывается программное обеспечение для проектирования алгоритмов с последующей конвертацией в текстовые языки программирования.

Создаваемое программное обеспечение основывается на принципах автоматного программирования. Построение графических диаграмм реализуется через направленный граф: автомат Мура. Автомат Мура – это такой автомат, у которого выходное воздействие зависит лишь от состояния, в которое автомат перешел и не зависит от входного воздействия. Поэтому выходные сигналы приписаны прямо к состояниям [5].

Состоянием в программе называют множество текущих значений всех используемых в ней переменных (Рис. 1).

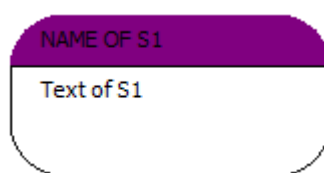


Рисунок 1 – изображение состояния в программе

На рисунке 1 представлен образец состояния, задаваемого в программе. В заголовке указывается имя состояния, а в нижней части состояния – действия в нём (присвоение переменных или символьное описание процессов).

При разработке ПО также берется за основу концепция детерминированного конечного автомата, изображаемого в виде графа, вершины которого являются состояниями, а ребра (стрелочки) – переходы

между ними.

Пример построения диаграмм показан на рисунке 2.

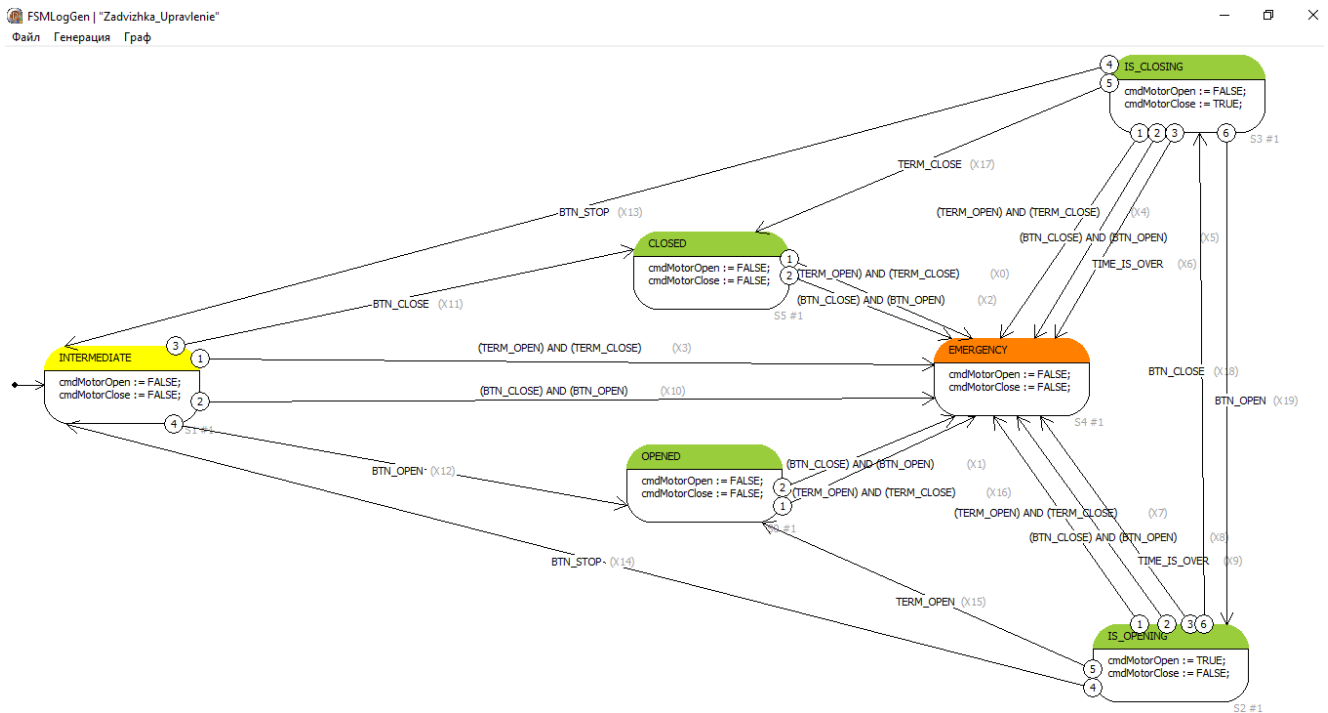


Рисунок 2 – Пример построения ДКА алгоритма управления задвижкой в ПО

На рисунке 2 отображен алгоритм управления задвижкой, построенный с помощью графа на основе детерминированного конечного автомата. В данном случае работа алгоритма начинается со стартового состояния, на которое указывает одиночная стрелка слева от состояния INTERMEDIATE, обозначенное желтым цветом. Вверху пишется название состояния, внизу – действия в данном состоянии. В крайнем правом нижнем углу с внешней стороны состояния пишется обозначение его порядкового номера, например, S0, а рядом с помощью символа # отображается номер графа. На стрелочках задается название условия перехода. Данный граф состоит из шести состояний:

- INTERMEDIATE (промежуточное состояние);
- IS_OPENING (задвижка открывается);
- IS_CLOSING (задвижка закрывается);
- OPENED (задвижка открыта);
- CLOSED (задвижка закрыта);
- EMERGENCY (аварийное состояние, обозначено оранжевым цветом).

Также на графе отображены следующие условия:

-BTN_STOP (нажата кнопка остановки задвижки);

-BTN_CLOSE (нажата кнопка закрытия задвижки);

-(BTN_CLOSE) AND (BTN_OPEN) (условие одновременного нажатия кнопок закрытия и открытия задвижки);

-(TERM_OPEN) AND (TERM_CLOSE) (сработали концевики на закрытие и открытие задвижки одновременно);

-TERM_OPEN (сработал концевик на открытие задвижки);

-TERM_CLOSE (сработал концевик на закрытие задвижки);

-TIME_IS_OVER (закончилось время, необходимое для открытия/закрытия задвижки).

Внутри каждого из состояний возможно описание действий в виде присваивания переменных. В данном случае задействуются переменные cmdMotorOpen и cmdMotorClose булевого типа. При переходе автомата из одного состояния в другое внутри каждого состояния будут меняться значения переменных, приведенных ранее. При попадании в состояние IS_CLOSING будет происходить присваивание переменной cmdMotorClose значения TRUE, а значению cmdMotorOpen значения FALSE. В свою очередь, при нахождении в состоянии IS_OPENING, наоборот, переменной cmdMotorOpen присвоится значение TRUE, а переменной cmdMotorClose присвоится значение FALSE. В остальных состояниях двум вышеуказанным переменным будут присвоены значения FALSE. Также в соответствии с тем, что условия переходов в аварийное состояния являются приоритетными, чем условия переходов между другими состояниями, расставлены приоритеты.

При работе с таблицами вместо имени состояния, заданного пользователем, используется имя SM (M – порядковый номер состояния), поскольку это позволяет более компактно отображать создаваемые на основе графов таблицы. При этом не усложняется работа программы при использовании произвольного сложного имени состояния. Номер графа реализован, чтобы в случае задания параллельного графа (с выбором нового

цвета заголовков состояния), можно было отличить два параллельных конечных автомата не только по цвету, но и по символьному обозначению. Например, для состояния STOP номер графа #1.

Также справа от названий переходов, задаваемых пользователем, автоматически отображено символьное обозначение условия перехода, например, X0. Это сделано также для того, чтобы при работе с таблицами использовать более короткие и компактные символьные последовательности в виде XN (N – порядковый номер перехода).

2.2. Пользовательский интерфейс программного обеспечения

В программном обеспечении рабочее поле выглядит так, как приведено на рисунке 3. В данном случае в качестве примера приведен алгоритм работы кофейного автомата.

FSMLogGen | "New project (31052022, 10-11-21)" *
Файл Генерация Граф

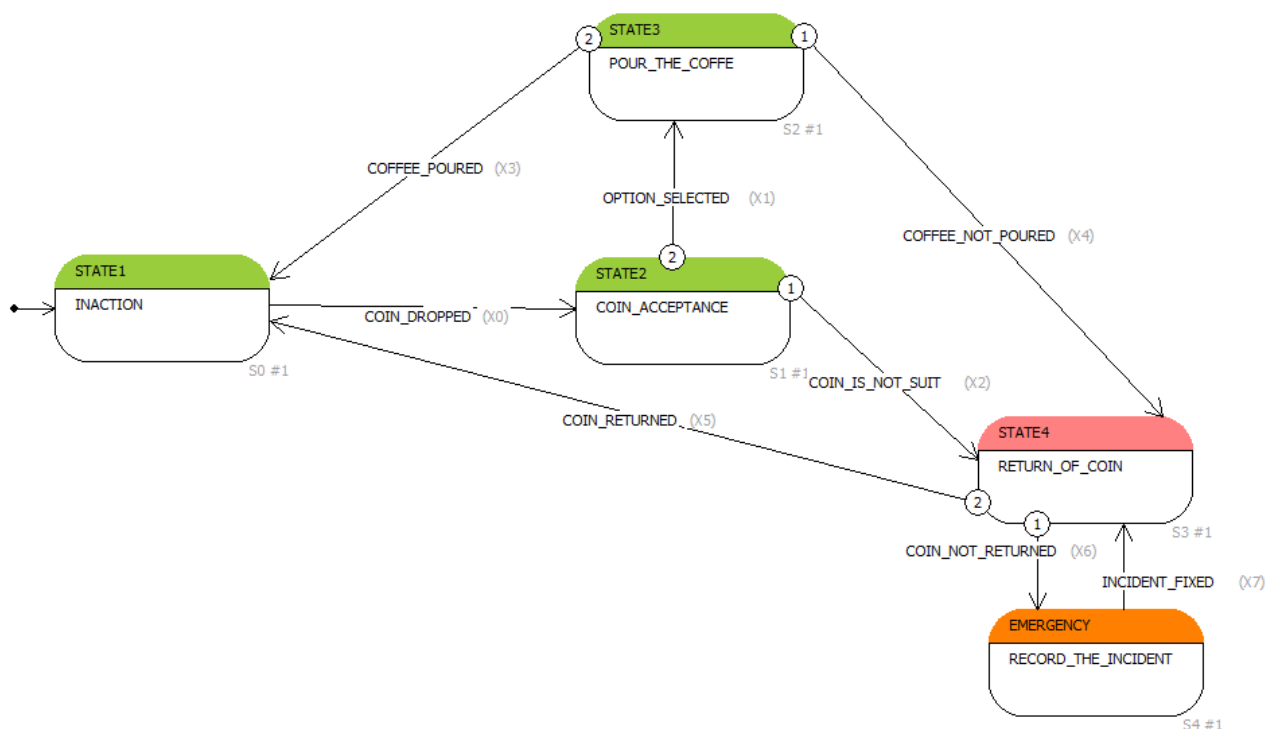


Рисунок 3 – Рабочее поле программного обеспечения

На рисунке 3 видно, что вверху отображается названия самого

программного обеспечения, рядом с ним по умолчанию название проекта «New project», в скобках дата создания текущего проекта и в конце строки находится символ «*», используемый как признак сохранения проекта. Если «*» присутствует, то проект не сохранён. Также видно меню с названиями «Файл», «Генерация», «Граф». При наведении мыши на один из этих пунктов появляется выпадающее меню. На рисунке 4, 5 и 6 представлены выпадающие меню из «Файл», «Генерация» и «Граф» соответственно.

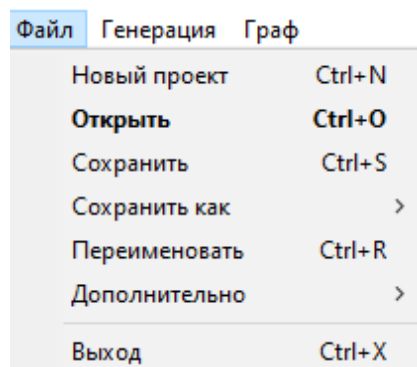


Рисунок 4 – Выпадающее меню из пункта «Файл»

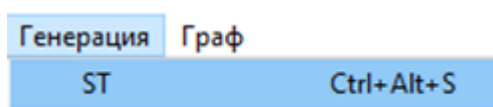


Рисунок 5 – Выпадающее меню из пункта «Генерация»

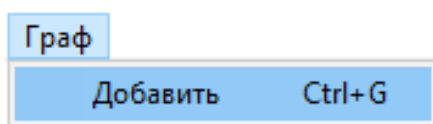


Рисунок 6 – Выпадающее меню из пункта «Граф»

Теперь существует необходимость описать каждый из пунктов выпадающих меню. Из рисунка 4 видно, что в «Файл» предусмотрено создание нового проекта, подгрузка или открытие уже существующего, сохранение текущего проекта, сохранение проекта с изменением директории и названия, переименование проекта, а также меню «Дополнительно», в котором можно подключить функцию «Автосохранение» (Рис. 7). В таком случае после каждого действия пользователя в программе будет происходить автоматическое

сохранение проекта. Этот режим можно отменить повторным нажатием на «Автосохранение». Также реализована функция «Выход», позволяющая выйти из программы.

При нажатии на «Новый проект» создается новый проект с очищенным полем формы и обновленным тайм кодом, который соответствует текущей дате и времени суток.

В пункте меню «Генерация» (Рисунок 5) нажатие на подпункты дает возможность сгенерировать код на языке ST. Пункт «Граф» (Рисунок 6) состоит из единственного подпункта «Добавить». С помощью него можно создавать параллельные графы, увеличивая при каждом нажатии нумерацию нового создаваемого параллельного конечного автомата.

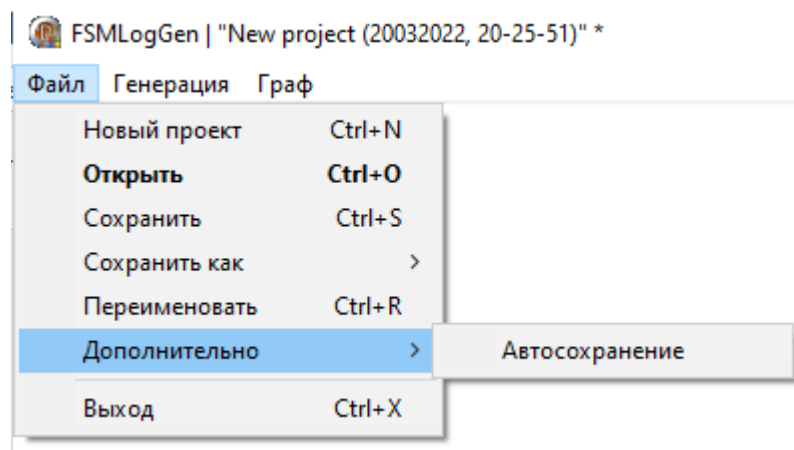


Рисунок 7 – Выпадающее меню из пункта «Дополнительно»

Помимо этого, как видно из рисунков 4-6, существует возможность использования горячих клавиш для подпунктов меню с тем, чтобы каждый раз не нажимать на выпадающее меню, имея возможность более быстрой реализации созданных в программе функций.

В пункте меню «Сохранить как» (Рисунок 8) можно выбрать не только сохранение текущего проекта для возможности его открытия в будущем, но и создание рисунка формы «Скриншот». При этом доступен также подпункт меню «Итоговой таблицей».

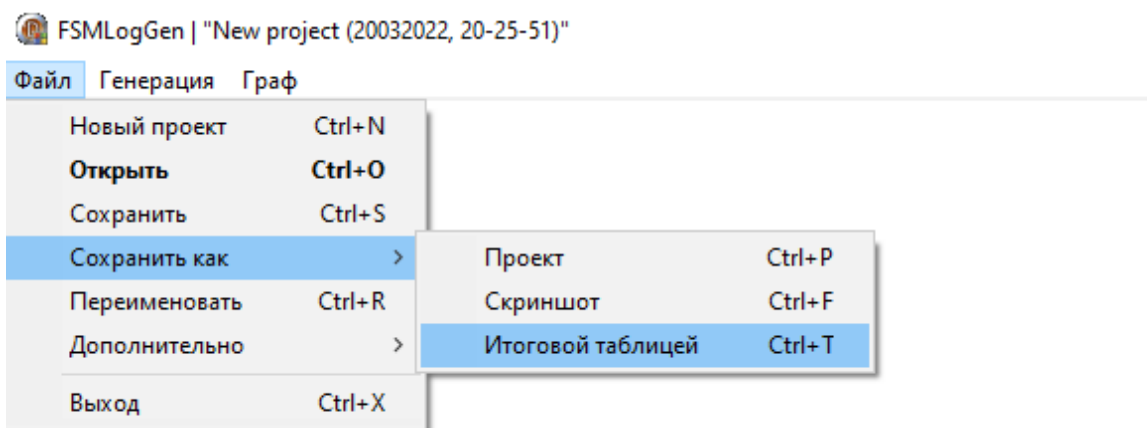


Рисунок 8 – Выпадающее меню из пункта «Сохранить как»

Если нажать на «Итоговой таблицей», но при этом на рабочем поле отсутствуют хотя бы два состояния и один переход между ними, то произойдет вывод сообщения (Рисунок 9).

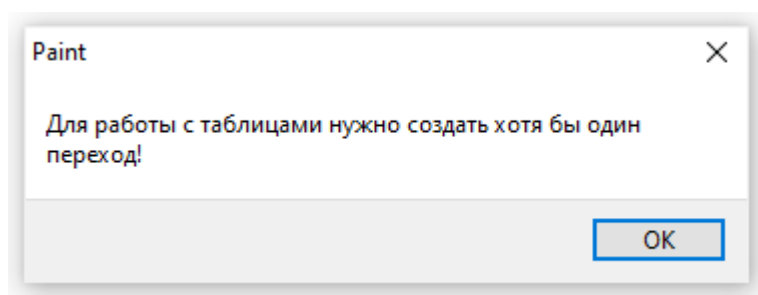


Рисунок 9 – Выпадающее меню из пункта «Сохранить как»

В ином случае, когда хотя бы один переход существует между двумя состояниями, то происходит переход на форму с таблицами. Для демонстрации данной формы создан граф абстрактного конечного автомата, приведенный на рисунке 10.

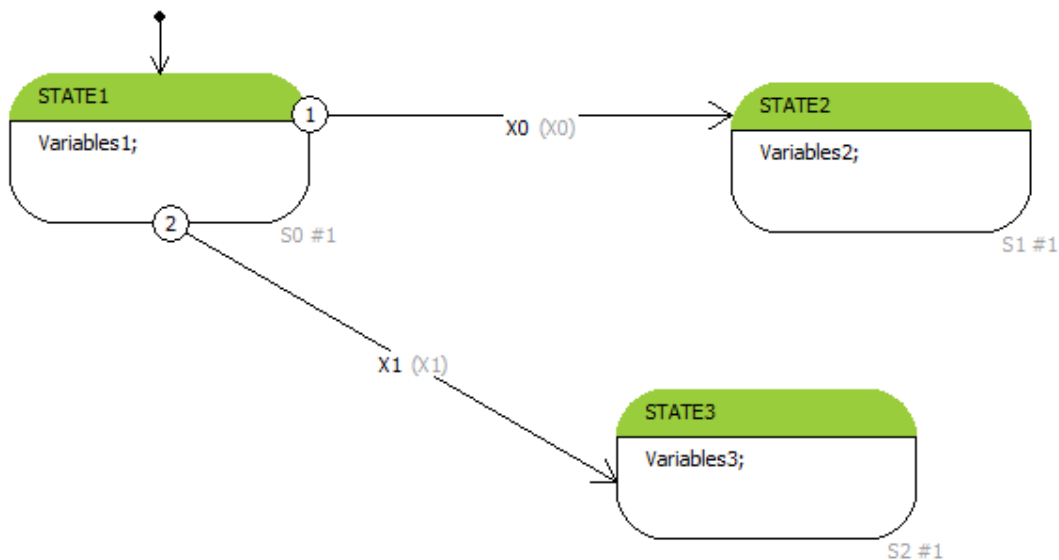


Рисунок 10 – Граф абстрактного конечного автомата

Интерфейс для работы с таблицами изображен на рисунке 11.

Таблица состояний и переходов

Файл

Таблица Состояний и переходов

Начальное	Условие	Конечное
STATE1	X0	STATE2
STATE1	X1	STATE3

Генерация таблиц:

- Состояний и переходов
- Конечного автомата
- Истинности (КНФ)
- Истинности (ДНФ)
- Разложение по единицам
- Разложение по нулям
- Карты Карно

Разложение по нулям

$$f1 = (S1 + S0 + X1 + X0) * (S1 + S0 + \neg X1 + X0) * (\neg S1 + S0)$$

$$f2 = (S1 + S0 + X1 + X0) * (S1 + S0 + X1 + \neg X0) * (S1 + S0 + \neg X1 + \neg X0) * (S1 + \neg S0)$$

Формирование отчета:

Отчет (04062022,23-51-06).xls

Сохранить отчет

Рисунок 11 – Пользовательский интерфейс для работы с таблицами

Если в программном обеспечении пользователь задаст условие входа в

состояние таким же именем, что и условие на выходе из этого состояния (или наоборот), то программа предупредит о некорректно заданном имени условия перехода и произойдет переход в режим редактирования условия перехода. Таким образом, происходит детектирование действий пользователя при вводе названий условий переходов и предупреждение о том, что условия на входе в состояние и на выходе из него не должны иметь одинакового имени (Рисунок 12).

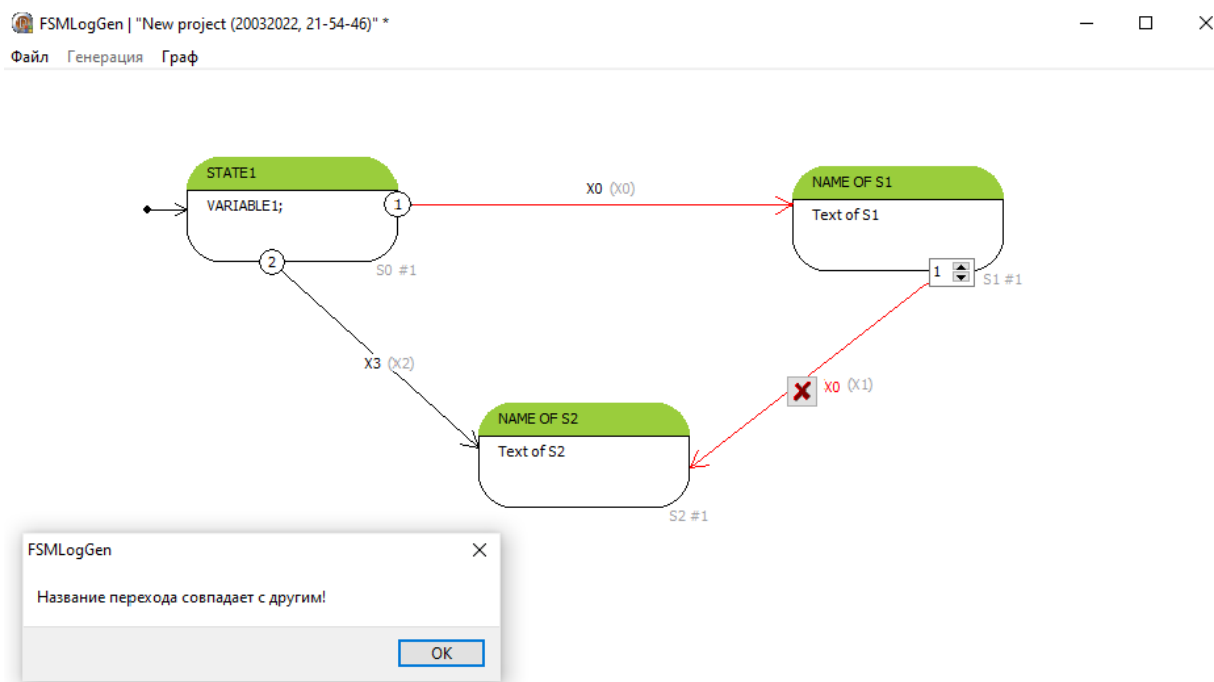


Рисунок 12 – Выдача сообщения пользователю о некорректно названном условии перехода

При нажатии «ОК» происходит возвращение в режим редактирования названия условия перехода (Рисунок 13).

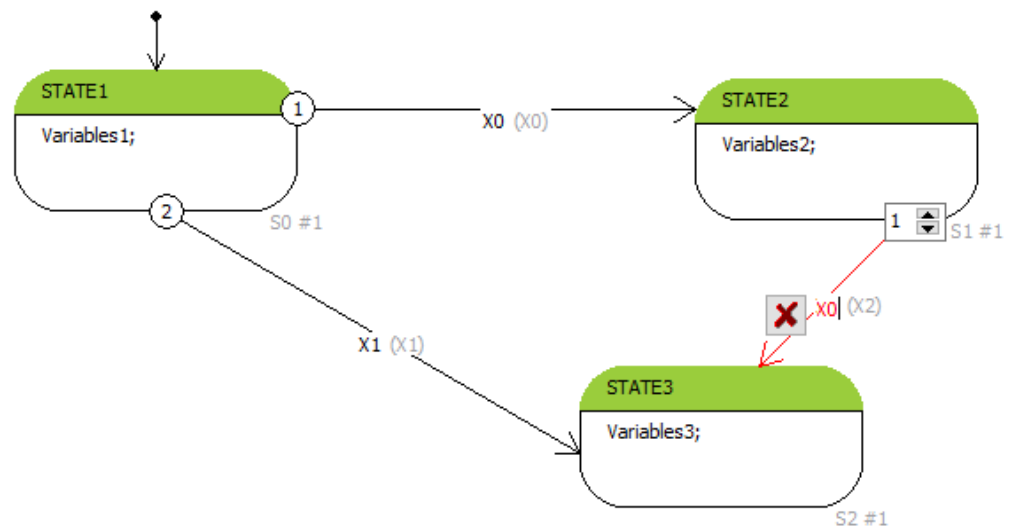


Рисунок 13 – Переход в режим редактирования названия условия перехода

Если попробовать открыть форму с таблицами в случае, когда количество переходов между состояниями равно нулю, то программа не даст это сделать и пользователю будет выдано сообщение, представленное на рисунке 14.

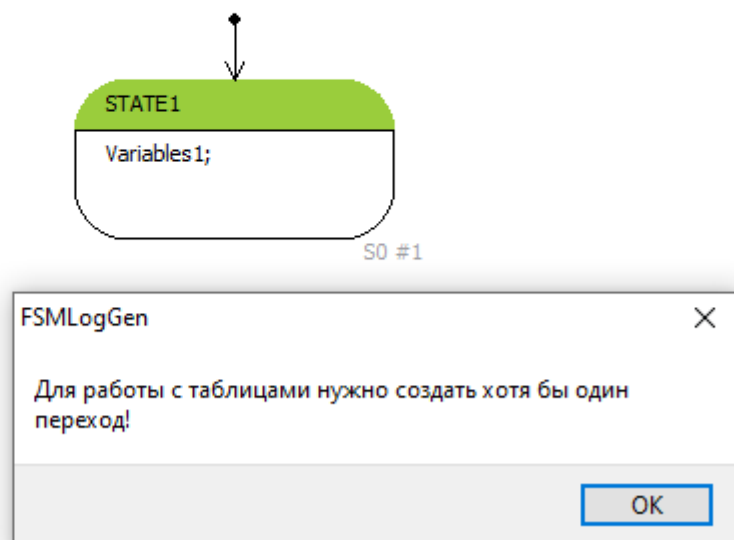


Рисунок 14 – Сообщение о запрете перехода на форму с таблицами

Если попробовать задать одинаковый приоритет, то это также будет некорректно, о чем пользователь будет уведомлен. При этом он попадет в режим редактирования перехода для изменения номера приоритета на правильный (Рисунок 15).

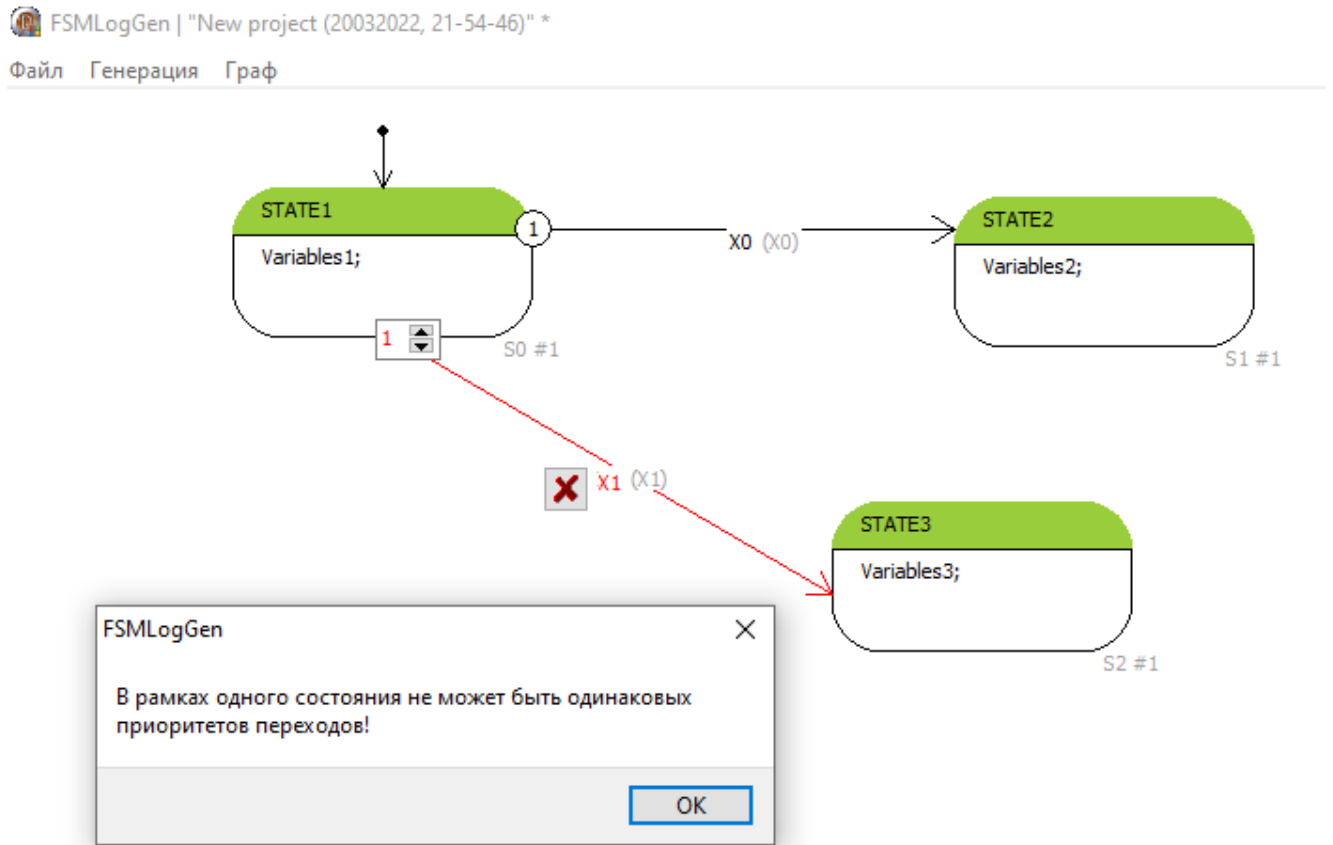


Рисунок 15 – Сообщение о запрете ввода одинаковых приоритетов переходов из одного состояния

2.3. Обеспечение вывода таблицы списка ребер на основе графов

Использование списка ребер [7] позволяет отразить созданный граф, изображенный на рисунке 16, в наглядном табличном виде (Рис. 17). В первом столбце отображается начальное состояние, во втором – условие перехода, в третьем – конечное состояние. Начальное состояние – это прямоугольник, выбранный пользователем для отрисовки перехода нажатием ЛКМ на его границе. Конечное состояние – это прямоугольник, к которому проводится стрелка (переход). Условие перехода – это переменная либо текст, по которым происходит переход из одного состояния в другое. Для получения списка ребер

в качестве примера задан граф алгоритма двухпозиционного регулирования [8]. В алгоритме задействуются такие переменные, как SET_POINT – уставка значения температуры, CURRENT_TEMP – текущее значение температуры и Heater – переменная, принимающая логическую единицу, обозначающую включение нагревателя и логический ноль при его выключении. Включение и выключение происходит в зависимости от выполнения условия превышения текущей температуры значения уставки. При температуре, превышающей установленное значение, происходит отключение нагревателя, а если уставка не достигнута, то нагреватель работает и включается, когда температура опускается ниже уставки. Весь алгоритм приведен на рисунке 16.

Файл Генерация Граф

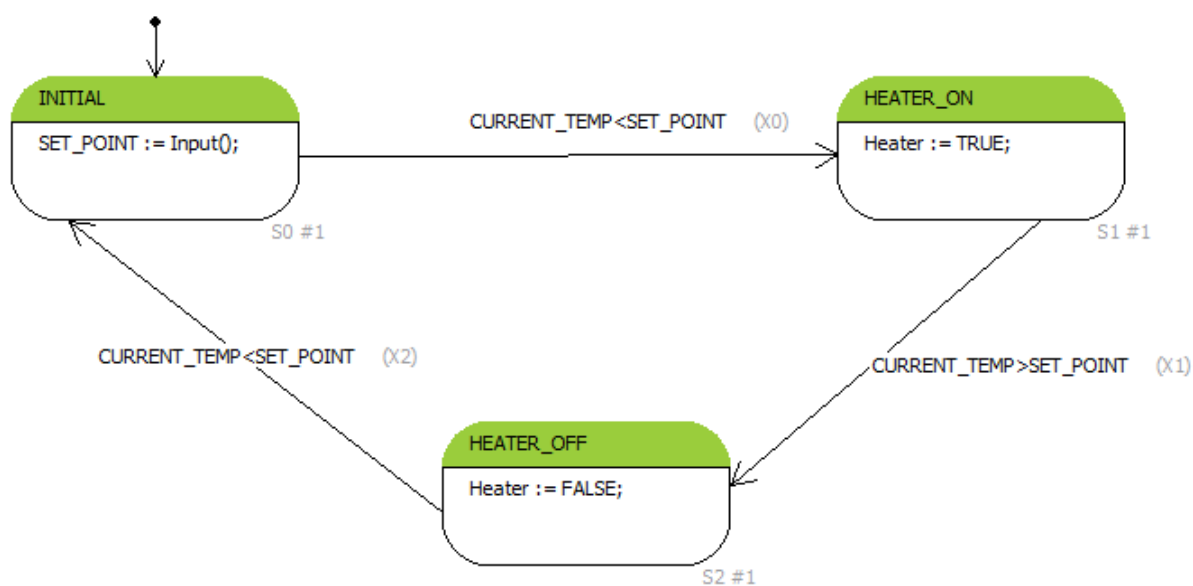


Рисунок 16 – Алгоритм двухпозиционного регулирования температуры

На основе графа из рисунка 16 в ПО формируется список ребер, отображающий алгоритм в табличном виде (Рисунок 17).

Начальное	Условие	Конечное
INITIAL	CURRENT_TEMP<SET_POINT	HEATER_ON
HEATER_ON	CURRENT_TEMP>SET_POINT	HEATER_OFF
HEATER_OFF	CURRENT_TEMP<SET_POINT	INITIAL

Рисунок 17 – Список ребер для графа двухпозиционного регулирования

Для того, чтобы разнообразить приведенный на рисунке 17 пример, добавлено ещё одно состояние, обозначающее аварию при включении нагревателя (EMERGENCY) (Рис. 18). Приоритеты перехода расставлены таким образом, чтобы условие HEATER_BROKEN было важнее, чем условие CURRENT_TEMP>SET_POINT, поскольку в состоянии аварии работа конечного автомата не должна продолжаться, процесс должен быть остановлен. Стоит отметить, что приоритеты переходов из того или иного состояния появляются в случае, если количество стрелок больше единицы.

FSMLogGen | "New project (20032022, 16-11-23)" *
 Файл Генерация Граф

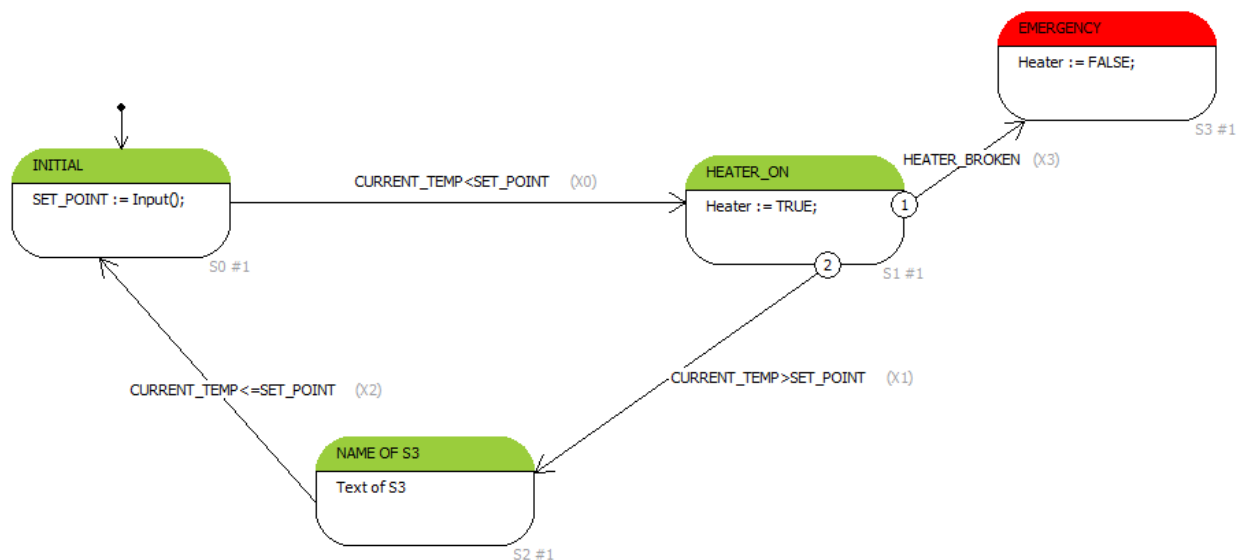


Рисунок 18 – Добавление нового состояния и появление приоритетов

2.3.1. Генерация кода на языке ST

Одним из языков программирования стандарта МЭК 61131-3-2016 является ST. Данный ЯП выбран для реализации генерации исходя из его преимуществ. Он имеет возможность переноса кода из одной среды программирования ПЛК в другую, а также возможность создания программ в любом текстовом редакторе [9]. Данный язык широко применяется в программировании ПЛК. Более того, он находит применение и в программировании ПР. Например, компания Овен начала внедрять язык ST для некоторых моделей программируемых реле в среде Owen Logic в виде функций, встроенных в отдельные макросы [10]. Это значительно упрощает процессы написания программ в совокупности с использованием классического языка FBD для программирования ПР. Предоставления языка ST в макросах Owen Logic способствует возможности перехода с проектов на языке ST, реализованных под ПЛК, на проекты с ПР, что значительно сократит затраты на оборудование.

Для языка ST на основе рисунка 18 также произведена генерация кода и результат приведен в приложении А в Листинге А.1.

2.4. Применение разработанного программного обеспечения

2.4.1. Реализация алгоритма работы станка

Применена возможность программного обеспечения по созданию параллельных графов. Это существенно упрощает понимание алгоритма работы автоматизируемого станка [11]. Данный станок показан на рисунках 19 и 20. Приведенный станок имеет название «Станок СФ-1» и предназначен для механической обработки наклонных плоскостей надрессорной балки двухосных тележек модели 18-100, 18-578, 18-194-1 грузовых вагонов. Пульт управления станком приведен в приложении Б.



Рисунок 19 – Внешний вид станка с пультом управления

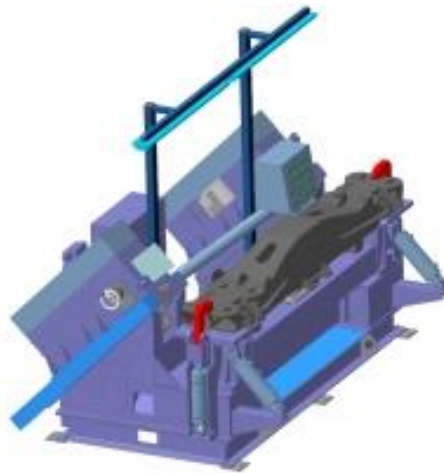


Рисунок 20 – Внешний вид станка

Состояния конечного автомата станка созданы исходя из его назначения, изображения пульта управления (Приложение Б), внешнего вида, показанного на рисунках 19 и 20. Более того, алгоритм работы станка разрабатывался при наличии изображений его крайних (верхнего и нижнего) положений (Рисунки 21 и 22).

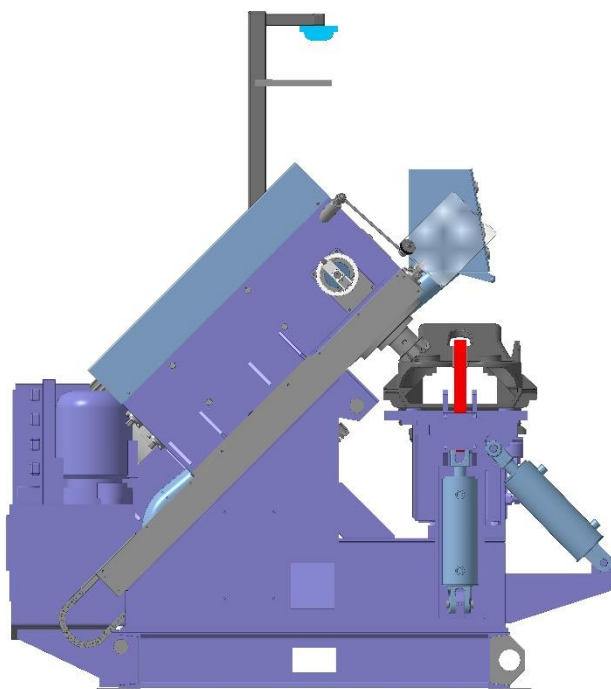


Рисунок 21 – Изображение фрезера в крайнем верхнем положении (вид сбоку)

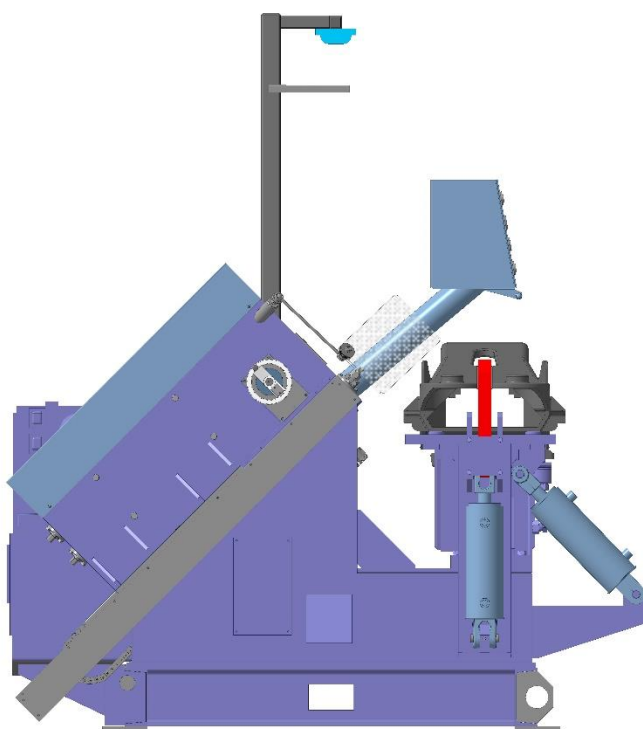


Рисунок 22 – Изображение фрезера в крайнем нижнем положении (вид сбоку)

При достижении крайнего верхнего положения (Рисунок 21) срабатывает концевик на движение вперед, выполняется условие (TERM_FORWARD) и фреза, в случае, если она включена, начинает двигаться

в крайнее нижнее положение. Аналогично, если фреза включена и достигает крайнего нижнего положения (Рисунок 22), срабатывает условие TERM_BACKWARD, фреза начинает движение в сторону крайнего верхнего положения. При достижении среднего положения, в случае, если фреза включена, срабатывает датчик среднего положения балки (TERM_MIDDLE), что приводит к меньшему ходу фрезы, фреза начинает раньше двигаться в обратную сторону. Благодаря этому станок значительно ускоряет свою работу. Если же фреза выключилась и сработал концевой выключатель на движение вперед (если автомат находится в состоянии MOVE_FORWARD, достигнув крайнего верхнего положения) или концевой выключатель на движение назад (если автомат находится в состоянии MOVE_BACKWARD, достигнув крайнего нижнего положения), то система принудительно переходит в состояние STOP для остановки работы.

При проектировании автоматной модели для автомата №1 применены следующие состояния и переходы между ними:

-IDLE1 (исходное состояние ожидания включения фрезы). Данное состояние имеет следующее условие перехода из него: ENABLE (запуск системы в работу) для перехода в состояние;

-STOP (состояние остановки работы фрезы). Данное состояние имеет следующее условия перехода из него:

-(NOT ENABLE) (условие остановки работы системы);

-(BTN_FORWARD) (условие нажатия кнопки без фиксации «ВПЕРЁД» движения фрезы);

-(BTN_BACKWARD) (условие нажатия кнопки без фиксации «НАЗАД» движения фрезы);

-MOVE_FORWARD (состояние движения фрезы вперёд). Данное состояние имеет следующее условия перехода из него:

-(NOT ENABLE) (условие остановки работы системы);

-(BTN_STOP) (условие нажатия кнопки без фиксации «СТОП» движения фрезы);

-(NOT CUTTER_ON) AND (TERM_FORWARD) (условие, выполняемое если фреза отключена и сработал концевик на движение вперёд);

-(CUTTER_ON) AND (TERM_MIDDLE) (условие, выполняемое если фреза включена и сработал датчик среднего положения балки);

-(CUTTER_ON) AND (TERM_FORWARD) (условие, выполняемое если фреза включена и сработал концевик на движение вперёд);

-MOVE_BACKWARD (состояние движения фрезы назад). Данное состояние имеет следующие условия перехода из него:

-(NOT ENABLE) (условие остановки работы системы);

-(BTN_STOP) (условие нажатия кнопки без фиксации «СТОП» движения фрезы);

-(NOT CUTTER_ON) AND (TERM_BACKWARD) (условие, выполняемое если фреза отключена и сработал концевик на движение назад);

-(CUTTER_ON) AND (TERM_MIDDLE) (условие, выполняемое если фреза включена и сработал датчик среднего положения балки);

-(CUTTER_ON) AND (TERM_BACKWARD) (условие, выполняемое если фреза включена и сработал концевик на движение назад);

-INTERMEDIATE_F (промежуточное состояние после движения вперёд и срабатывания датчика среднего положения балки). Данное состояние имеет следующее условие перехода из него:

-(BTN_STOP) (условие нажатия кнопки без фиксации «СТОП» движения фрезы);

-(TRUE) (безусловный переход);

-INTERMEDIATE_V (промежуточное состояние после движения назад и срабатывания датчика среднего положения балки). Данное состояние имеет следующие условия перехода:

-(BTN_STOP) (условие нажатия кнопки без фиксации «СТОП» движения фрезы);

-(TRUE) (безусловный переход).

В автоматной модели для автомата №2 применены следующие состояния и переходы между ними:

-IDLE2 (исходное состояние ожидания пуска системы). Данное состояние имеет следующее условие перехода из него: (ENABLE) (запуск системы в работу);

-OFF_CTR (состояние выключения фрезы). Данное состояние имеет следующее условие перехода из него: (BTN_CUTTER_ON) (условие нажатия кнопки без фиксации «ПУСК» фрезы);

-ON_CTR (состояние включение фрезы). Данное состояние имеет следующие условия перехода из него:

- (NOT ENABLE) (условие остановки работы системы);

- (BTN_CUTTER_OFF) (условие нажатия кнопки без фиксации «СТОП» фрезы).

Внутри состояний изменяются значения булевых переменных `cmdforward` и `cmdbackward`.

Автоматная модель, отображающая описанный выше алгоритм и состояния с переходами, создана в рабочем поле программы и расположена на рисунке 23.

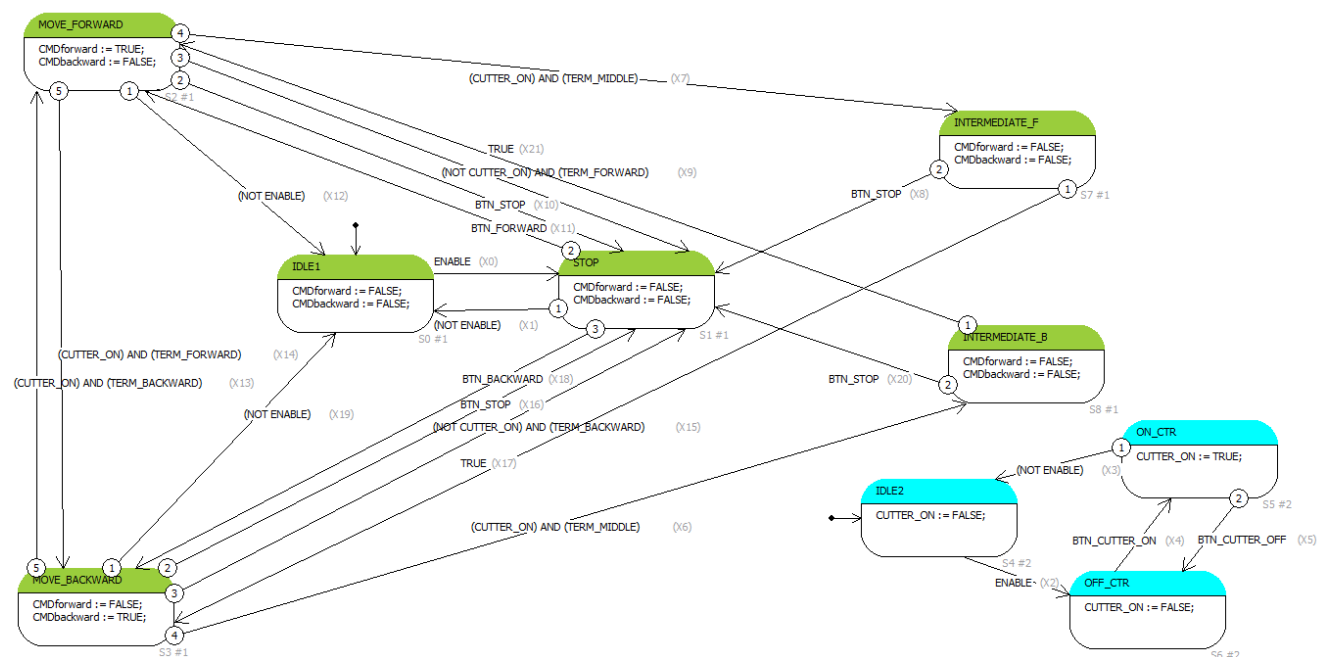


Рисунок 23 – Спроектированный алгоритм для станка-фрезера

В результате на рисунке 23 заданы два параллельных конечных автомата. Применение параллельных конечных автоматов значительно упрощает понимание алгоритма и его проектирование, так как позволяет сократить количество задействованных состояний. Без применения параллельных автоматов граф становится более сложным, большое количество переходов пересекаются между собой, затрудняется понимание его работы. Более того, граф на рисунке 23 отображает лишь одну половину станка. Но для второй половины граф идентичен.

Таким образом, задан алгоритм работы фрезера наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона [11].

На основе графов-переходов, приведенных на рисунке 23, осуществлена генерация программного кода на языке ST для последующего использования при создании программы для ПР (Приложение Ж).

С целью сравнения с автоматной моделью алгоритма, приведенной на рисунке 23, построен этот же алгоритм, но в традиционном виде ГСА. Для первого конечного автомата в приложении В приведена первая часть

алгоритма, в приложениях Г и Д – вторая и третья части соответственно. Для второго конечного автомата, работающего параллельно первому, в приложении Е также приведен алгоритм в виде ГСА. В данном случае можно сделать вывод, что автоматная модель более наглядно отображает алгоритм, чем ГСА, поскольку автоматная модель занимает меньше места, обладает компактностью, а также изоморфна (отображает алгоритм лишь в единственном виде, в то же время ГСА может быть нарисована и разработана в различных вариациях). В сложных системах автоматная модель воспринимается более наглядно.

2.4.2. Генерация кода из алгоритма работы станка

На основе заданной на рисунке 21 автоматной модели автоматически получен готовый программный код.

Сгенерированный код предназначен для реализации управления фрезером наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона [11].

Программа на языке ST, реализующая управление фрезером наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона, представлена в приложении Ж.

Для программирования ПР200 использована программная среда OWEN Logic [12]. Для составления программы используется язык функциональных блоков FBD.

2.4.3. Применение сгенерированного кода на языке ST в программном обеспечении Owen Logic

С целью управления станком выбрана среда программирования Owen Logic и устройство ПР200 на 8 дискретных входов, 8 дискретных выходов и 4 аналоговых входов, 4 аналоговых выхода.

Сгенерированный кодогенератором программный код на языке ST был перенесен в среду программирования Owen Logic. Создана схема на языке FBD, использующая макрос на языке ST (Рис. 24). В результате программа

управления станком была успешно скомпилирована и загружена в устройство ПР200.

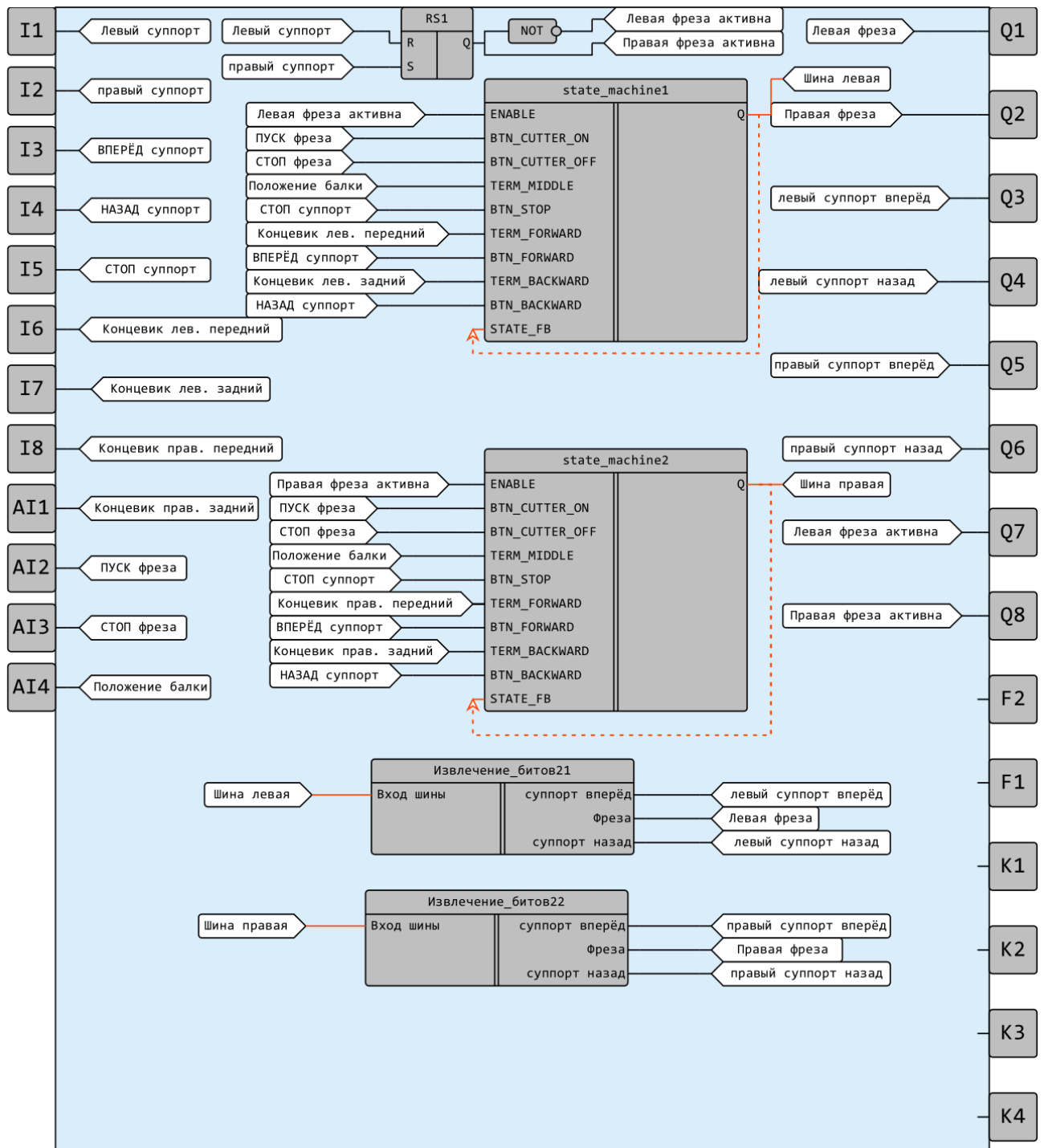


Рисунок 24 – Схема на языке FBD для управления станком

Изначально среда Owen Logic не имела доступа к программированию и созданию макросов на языке ST, но в версиях встроенного программного обеспечения 2.29 написание кода на данном языке стало доступно в ПР200,

ПР102, ПР100[МО2] и ИПП120 [13].

Исходя из того, что станок имеет две стороны (левую и правую), в данной схеме приведены два макроса (`state_machine1` и `state_machine2`) на языке ST для управления фрезером наклонных поверхностей (Рисунок 24). Первый предназначен для управления левой стороной станка, второй макрос необходим для управления правой стороной. Внутри обоих макросов находится одинаковый для каждой из сторон станка код (Листинг Е.1) на языке ST, сгенерированный ранее из графа на рисунке 23.

Исходя из возможностей программного обеспечения Owen Logic (отсутствует возможность добавления нескольких выходов к макросу, макрос работает как функция, которая не запоминает состояния), организована подача выхода макроса, с задержкой в один такт, на входную переменную `STATE_FB`, используемую для запоминания состояний конечного автомата. Также реализован разбор и сбор битов, характеризующих определенные сигналы. Это осуществлено, так как в Owen Logic функция на языке ST пока имеет только один выход макроса. Поэтому для подачи разных выходных сигналов на выходы программируемого реле используются побитовые операции.

Таким образом, реализовано управление фрезером наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона.

2.4.4. Перспективы дальнейшего развития программного обеспечения

В будущем программа может подвергнуться модернизации. То есть, доработан интерфейс ПО и его функционал. Интерфейс программного обеспечения может быть улучшен по следующим пунктам:

– внедрение кривых Безье, как способа улучшения наглядной составляющей рабочего поля программы при отрисовке переходов между

состояниями;

- реализовать возможность создания вложенных автоматов;
- осуществление возможности переноса стрелок в нужную точку окна;
- добавление алгоритмов упрощения логических функций (Карты Карно, метод Квайна – Мак-Класки, метод неопределённых коэффициентов или аналитический метод);

- добавление новых текстовых языков программирования для генерации кода: Assembler, C, C#, Python, Verilog и другие;

- оптимизация алгоритмов по вычислению значений логических функций;

- оптимизация времени создания таблицы истинности;
- создать возможность автоматического огибания состояний кривыми Безье;

- создать корректный прогноз времени выполнения вывода данных в таблицы программы и в Excel;

- внедрить возможность отмены действий пользователя;
- создать раздел Help, где будет инструкция по использованию программы.

- при необходимости осуществить добавление возможности масштабирования объектов;

- по возможности сделать интерфейс более дружелюбным пользователю;

- создать обучающий интерфейс для того, чтобы при необходимости пользователь мог ознакомиться с возможностями программы.

3. Концепция стартап-проекта

Сутью проекта является вывод на рынок продукта, предназначенного для разработки программ для АСУ ТП. Цель – разработать маркетинговый план продвижения данного ПО среди потенциальных клиентов на российском рынке.

3.1 Основные качества продукта, решаемая продуктом проблема

Разрабатываемая программа является инструментом упрощения разработки ПО на основе визуального программирования. Применение данного инструмента позволяет создавать алгоритмы в виде автоматных моделей на основе автомата Мура и осуществлять генерацию кода из графически заданных диаграмм на языке программирования ST, используемого в области автоматизации при создании программ для программируемых логических контроллеров.

3.2 Защита интеллектуальной собственности

В ВКР производилась разработка ПО для систем автоматизации с возможностью автоматической генерации кода для ПЛК. Оно было успешно применено для автоматизации станка – фрезера наклонных поверхностей надрессорной балки железнодорожной тележки грузового вагона. Защита интеллектуальной собственности может производиться как для ПО, так и для приведенного выше станка.

Для защиты интеллектуальной собственности в соответствии с законом необходимо направить заявку в Федеральную службу по интеллектуальной собственности (Роспатент), дождаться получения патента, после чего программное обеспечение или способ автоматизации будут регулироваться сразу несколькими актами - частью четвертой Гражданского кодекса Российской Федерации, а также федеральными законами № 98-ФЗ от 29 июля 2004 года «О коммерческой тайне», № 149-ФЗ от 27 июля 2006 года «Об информации, информационных технологиях и о защите информации», № 135-ФЗ от 26 июля 2006 года «О защите конкуренции», а также Кодексом об административных

правонарушениях РФ, Уголовным кодексом РФ и другими. На данном этапе получение патента не планируется. Принято решение не распространять данные, имеющие коммерческую ценность, сохранив их как коммерческую тайну.

В рамках защиты интеллектуальной собственности осуществлена государственная регистрация программы для ЭВМ под наименованием FSMLogGen (Приложение И).

С целью выявления существующих на данный момент инновационных решений, проведем патентный обзор.

Патент RU2676405C2 [1] защищает способ автоматизированного проектирования производства и эксплуатации прикладного программного обеспечения. В результате практического применения изобретения автоматизируется процесс разработки программного обеспечения. В данном патенте описано, что графическое отображение элементов и процессов объекта управления и мониторинга преобразуется в исполняемый модуль прикладного программного обеспечения. В результате применения технологии значительно сокращаются сроки разработки и создания прикладного программного обеспечения.

Патент RU2738460C1 [15] защищает изобретение способа выявления аномалий в работе сети автоматизированной системы. Технический результат использования данного способа позволяет повысить точность при выявлении аномалий в работе сети автоматизированной системы. В соответствии с патентом, для моделирования поведения АСУ отдельные программно-технические средства могут быть отражены в многоагентной модели в качестве агентов, представленных конечными автоматами.

Патент RU2769961C2 [16] защищает изобретение, относящееся к программируемой логической схеме управления электрической установкой, в частности, ядерной установкой. Описана проблема перепрограммирования программируемой логической интегральной схемы (ПЛИС) в случае модификации приложения команды управления. В таком случае перепрограммирование необходимо осуществлять на языках VHDL, Verilog. Но

этот процесс требует знания оператором данных языков, переквалификация при этом потребует много времени и усилий для проверки. В данном патенте обозначено решение – применение функциональных блоков. Исполнительный модуль для ПЛИС создан в качестве конечного автомата. В результате из графического представления функциональных блоков автоматическим генератором генерируется прикладная программа, что не требует специальных знаний на языке описания VHDL или Verilog.

Патент RU2447487C2 [17] защищает изобретение, относящееся к вычислительным системам, информационным технологиям. Техническим результатом является автоматизация тестирования программно-аппаратных комплексов. Целями изобретения является уменьшение времени разработки последовательностей машинных команд вычислительных устройств, улучшение восприятия и повышение наглядности программных интерфейсов и графических программ, снижение вероятности возникновения ошибок при разработке программного обеспечения и др. В данном патенте предложен способ разработки машинных команд, согласно которому часть последовательности машинных команд получают в результате компиляции диаграммы автоматов с конечным числом состояний, которую интерактивно строят в редакторе диаграмм состояний конечных автоматов.

Приведенные патенты позволяют:

- получить представление о современном состоянии, существующих направлениях решения технологических задач;
- констатировать, что предлагаемые проектные решения в данном проекте не запатентованы.

3.3 Объем и емкость рынка

В РФ на конец 2020 года в сферу разработки, проектирования, внедрения и оказания сопутствующих услуг в области автоматизации было вовлечено более 200 крупных компаний и предприятий малого и среднего бизнеса. При этом в своей деятельности автоматизацию используют не менее 1500

предприятий и организаций в России [18].

Согласно J'son & Partners Consulting [19] за 2021 год доля аппаратно-независимого программного обеспечения в «традиционных» и облачных АСУ ТП на российском рынке составила 29 млн долл. США, что эквивалентно 2 млрд 60 млн рублей РФ (Рисунок 25).

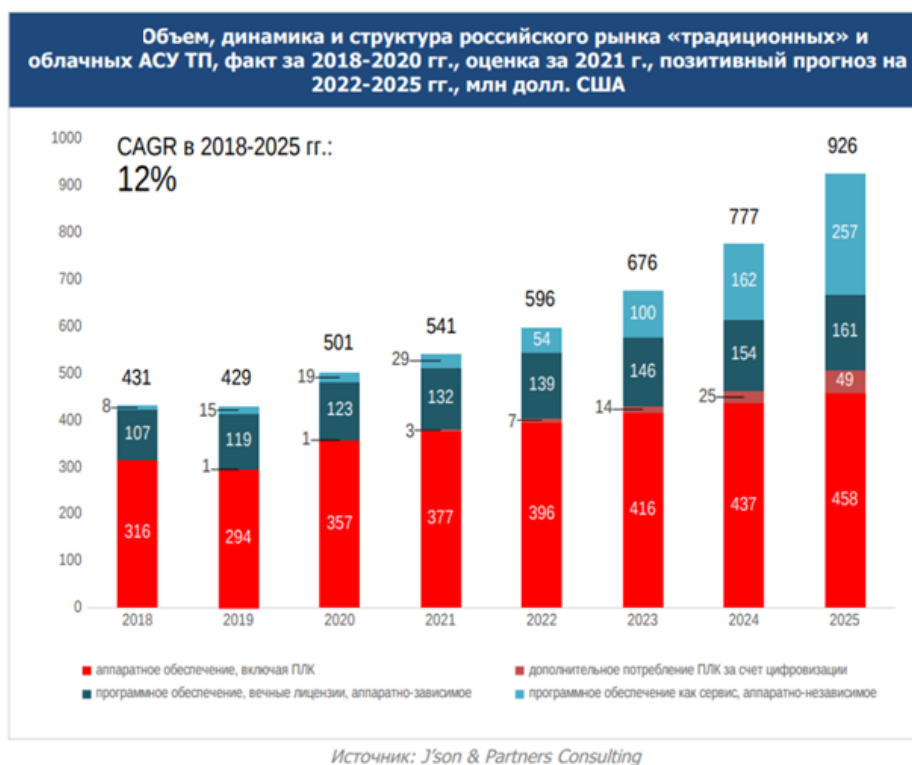


Рисунок 25 – Аналитика от J'son & Partners Consulting относительно рынка АСУ ТП России [19]

В сумме рынок АСУ ТП России составляет 541 млн долл. США за 2021 год. Также известно, что мировой объём рынка АСУ ТП насчитывает 92 млрд долларов или 92000 млн долларов в 2021 году [19]. Таким образом, через соотношения между рынками можно рассчитать приблизительную долю рынка аппаратно-независимого ПО на глобальном рынке АСУ ТП:

$$29 \text{ млн долл.} - 541 \text{ млн долл.}$$

$$x \text{ млн долл.} - 92000 \text{ млн долл.}$$

x – доля рынка аппаратно-независимого ПО в глобальном рынке АСУ ТП.

Величина x равна:

$$x = 92000 \times \frac{29}{541} = 4931,608 \text{ млн долл.}$$

Далее осуществлен перевод долларов США в рубли РФ по актуальному курсу на 02.05.2022: 1 долл. США = 71,08 Российских рублей. Итак, в результате расчётов объём рынка аппаратно-независимого ПО на глобальном рынке АСУ ТП составляет 350 млрд 548 млн рублей, а на рынке АСУ ТП России – 2 млрд 60 млн рублей.

В России, исходя из аналитики, приведенной ранее, около 1700 компаний задействованы в сфере автоматизации. Таким образом, количество потенциальных компаний = 1700. При этом среднее количество заказов на лицензии ПО одной компании – 10. Средняя сумма заказа на одну лицензию, исходя из стоимости лицензий конкурентов, составляет 1 180 000 руб.

Объём целевого рынка в руб. = Количество потенциальных компаний * Среднее количество заказов в год * Средняя сумма заказа в руб.

$$\text{Объём целевого рынка (руб.)} = 1700 \times 10 \times 1180000 = 20,06 \text{ млрд. руб.}$$

$$\text{Объём целевого рынка (долл.)} = 286,572 \text{ млн. долл.}$$

Таким образом, емкость рынка составляет 286,572 млн. долл., округлим до 290 млн. долл.

На сегодняшний день в условиях глобальных изменений в связи с санкциями в отношении России остро стоит вопрос импортозамещения в области АСУ ТП внутри страны. Полагаю, что динамика рынка аппаратно-независимого программного обеспечения АСУ ТП в последующие годы будет двигаться в стороны увеличения по следующим причинам:

- прогноз аналитики от J'son & Partners Consulting, созданный до введения беспрецедентных санкций в технологической сфере и области АСУ ТП (Рис. 25). Согласно прогнозу с каждым годом рынок будет расти вплоть до 2025;

- с учетом изменения предложения происходит возрастание спроса на ПО АСУ ТП внутри России в условиях санкций.

Таким образом, автоматизация технологических процессов будет осуществляться силами российских производителей, поскольку крупнейшие

поставщики решений и производителей оборудования в области автоматизации (Schneider Electric, Siemens) ушли с рынка России в связи с наложенными на них санкциями. Тем самым, возрастет спрос на отечественные решения в области автоматизации. В результате перестраивания экономики спрос на АСУ ТП возрастет и предложение отечественных производителей внутри рынка России в результате импортозамещения также увеличится.

3.4 Анализ современного состояния и перспектив отрасли

В настоящее время рынок программного обеспечения АСУ ТП является одним из динамично растущих рынков. По данным экспертов из компании J'son & Partners Consulting – международной консалтинговой компании, специализирующейся на рынках телекоммуникаций, медиа, ИТ в России, СНГ и Центральной Азии [20] – объем российского рынка АСУ ТП растет со скоростью 8% ежегодно и к концу 2022 года достигнет 596 млн. долларов. При этом объём рынка аппаратно-независимого программного обеспечения в 2022 возрастет вдвое по сравнению 2021 годом (29 млн долл. США) и составит 54 млн долл. США. Данный прогноз был осуществлен до введения санкций в отношении РФ, поэтому цифры могут отличаться от ранее спрогнозированных. Но при этом стоит отметить, что санкции предоставляют дополнительные возможности ИТ-компаниям внутри России, поскольку в результате их введения расширяется рынок отечественного ПО АСУ ТП, уменьшается конкуренция в данной области внутри страны, производственные процессы отечественных компаний замыкаются в России для решения проблем автоматизации внутри страны, разворачиваются импортозамещающие производства. В результате увеличивается спрос на программные продукты отечественного производства для АСУ ТП.

Конечный продукт, согласно классификатору ОКВЭД, принадлежит [21]:

- разделу J;
- классу ОКВЭД 62: Разработка компьютерного программного обеспечения, консультационные услуги в данной области и другие

сопутствующие услуги;

- подклассу 62.0: Разработка компьютерного программного обеспечения, консультационные услуги в данной области и другие сопутствующие услуги;

- коду ОКВЭД 62.01: Разработка компьютерного программного обеспечения.

В дополнении можно отметить, что правительство в конце декабря 2018 года подготовило проект обновленной стратегии развития отрасли информационных технологий в РФ, охватывающей период 2019-2025 гг. с перспективой до 2030 года [22]. В связи с этим ожидается достижения показателя объёма экспорта оказываемых услуг в разных отраслях экономики в размере 100 млрд. долларов к 2024 году. Мероприятия стратегии позволят добиться роста отрасли ИТ на уровне, превышающем средний темп роста ВВП – не менее чем в 3 раза за весь период. Таким образом, сфера ИТ будет динамично развиваться, что позволит рынку программного обеспечения АСУ ТП продолжать стабильно расти.

3.5 Расчет себестоимости продукта

Расчет себестоимости программного обеспечения проведем, исходя из расходов на разрабатываемый продукт:

- регистрация ПО в Роспатенте (уплата госпошлины, приобретение усиленной квалифицированной электронной подписи, покупка лицензии программы КриптоПро на год, установка и настройка КриптоПро) – 6 тыс. руб.;

- расходы на модернизацию ноутбука (два ОЗУ на 8 Гб, твердотельный накопитель SSD) – 14 тыс. руб.;

- расходы на использование интернета НИ ТПУ в течение двух лет – 7000 руб.;

- расходы на приобретение оборудования для тестирования работоспособности программного обеспечения ПР200-220.25.0.0 (10 шт.) – 160200 рублей и программируемых логических контроллеров (ПЛК) разных фирм (200000 руб.).

В результате общие материальные затраты на разработку программного продукта составляют: 387200 рублей.

Основную часть себестоимости продукта составляют не материальные затраты, а фонд оплаты труда (высокоинтеллектуальный труд программиста). Разработка программного обеспечения проводилась 6 часов в неделю в течение 2 лет. Пусть оплата труда программиста составляет 1000 рублей за 1 час. Недельная оплата труда составляет 6000 рублей. В результате в месяц оплата труда составит 24000 рублей. Таким образом, за 2 года моя оплата труда составит:

$$24000 \times 12 = 576000 \text{ рублей.}$$

То есть, фонд оплаты высокоинтеллектуального труда составляет 576000 рублей.

Общая себестоимость рассчитывается как сумма фонда оплаты труда и материальных затрат на разработку программного продукта:

$$576000 + 387200 = 963200 \text{ рублей.}$$

Доходы от продукта представлены доходами от лицензионной модели продажи ПО предприятиям и подписки на обновления. Планируется продажа продукта исходя из следующей модели:

- приобретение лицензии с полными возможностями программы, включающей генерацию кода на ST, C++ и Assembler и доступом к табличному интерфейсу – 100 тыс. руб. и подписки, обеспечивающей постоянное обновление программного продукта от компании (1% ежемесячно от стоимости лицензии на ПО);

- приобретение базовой лицензии с возможностями программы, включающей генерацию кода на ST, и доступом к табличному интерфейсу – 70 тыс. руб. и подписки, обеспечивающей постоянное обновление программного продукта от компании (1% от стоимости ПО ежемесячно);

- докупка более полной версии лицензии (добавление к базовому варианту лицензии возможности использования дополнительных языков программирования для генерации кода, стоимость добавления одного языка – 10

тыс. рублей);

– бесплатная версия с ограниченными возможностями программы (отсутствие возможности генерации кода, доступ предоставляется лишь к одной таблице, автоматически создаваемой на основе построенного в ПО графа. Есть лишь возможность пользовательской отрисовки алгоритма на основе автоматных моделей).

3.6 Конкурентные преимущества продукта и обзор технико-экономических характеристик аналогов

Программное обеспечение FSMLogGen (под данным наименованием была произведена регистрация моего ПО в ФИПС) разрабатывается на базе RAD Studio 2010 и реализует возможность проектирования алгоритмов в автоматных моделях. Это позволяет сгенерировать готовый для загрузки в ПЛК программный код на языке программирования контроллеров ST стандарта МЭК 61131-3-2016. Также существуют дополнительные возможности: вывод пользователю данных, отображающих алгоритм в табличном виде (таблица истинности, список рёбер). Это нужно для реализации упрощения логических выражений с использованием метода карт Карно, что позволит создавать схемы на логических элементах, тем самым проектировать и создавать с их использованием микросхемы. Данная возможность является дополнительным направлением развития программного продукта (первое направление – генерация кода).

Преимущества в использовании программного обеспечения заключаются в упрощении процессов разработки программного обеспечения АСУ ТП и согласования технического задания между программистом и технологом.

Вместо классического подхода по написанию текста программы в разрабатываемом ПО реализован подход автоматного программирования, автоматически сгенерировать готовый программный код из алгоритма. Приведем ряд преимуществ применения данного подхода:

- значительное сокращение времени на разработку программного

обеспечения для систем автоматики;

- дополнительная документация проекта (алгоритм автоматной модели);
- поддержка кода на всех этапах жизненного цикла;
- менее громоздкая структура алгоритмов в виде автоматных моделей по сравнению с классическими алгоритмами;
- упрощается этап согласования технического задания между технологом и программистом, разделяются сферы их деятельности.

Примеры ПО-аналогов:

- MATLAB (пакеты Simulink и Stateflow) [23];
 - MATLAB Coder (генерация кода C/C++ из MATLAB);
 - Simulink Coder (генерация кода C/C++);
 - Embedded Coder (генерация производственного кода C/C++);
 - HDL Coder (генерация кода на языках VHDL/Verilog);
 - Simulink PLC Coder (генерация кода на языке ST , Structured Text для ПЛК)
- Simintech (включает создание математических моделей, в том числе автоматных моделей, на основе которых можно сгенерировать программный код на языке Си, Structured Text для программируемых контроллеров) [24];
 - Unimod (включает проектирование моделей конечных автоматов с использованием диаграмм UML, но генерация кода из них доступна лишь на языке Symbian C++, предназначенном для программирования мобильных телефонов, создания приложений для них [25]. Также данный продукт предназначен для разработки консольных приложений, клиент-серверных Java-приложений, веб-приложений) [26].
 - YAKINDU Statechart Tools (программный продукт, использующий конечные автоматы и позволяющий генерировать исходный код на языках Java, C, C++, Python для использования в веб-приложениях) [27];
 - DRAKON Editor (Программный продукт для рисования алгоритмов на языке дракон. Поддерживается языки программирования: C, C++, Java, C#, Python, Tcl, Javascript, Erlang, Lua, Verilog и др. Используется в космической

области для разработки программного обеспечения, управляющего космическим кораблём. Получил распространение и за пределами аэрокосмической отрасли. На данном инструменте возможно описание программных систем различных типов) [28];

- Veremiz (среда разработки программ для программируемых логических контроллеров) [29];

- Codesys (аппаратно независимый комплекс для прикладного программирования ПЛК и встраиваемых контроллеров) [30];

- Visio2ST и Visio2Switich (Конвертеры, предназначенные для автоматической генерации ST и C/C++ кода соответственно по автоматным графам, нарисованным в Visio. Первый конвертер применяется для разработки прикладного ПО для контроллеров, второй – для микроконтроллеров) [31][32];

- IsaGraph от компании «Rockwell Automation» (инструментальная графическая среда разработки прикладных программ для программируемых логических контроллеров (ПЛК) на языках стандарта IEC 61131-3 и IEC 61499, позволяющая создавать локальные или распределенные системы управления) [33];

- Simatic Step 7 (программное обеспечение от фирмы Siemens для разработки систем автоматизации с использованием таких устройств, как программируемые логических контроллеры типа Simatic S7-300/S7-400/M7/C7 и WinAC).

- Рефлекс (язык программирования Рефлекс ориентирован на создание управляющих алгоритмов в промышленной автоматизации, в частности, для ПЛК).

В результате исследования аналогичной продукции выбраны наиболее сильные решения и внесены в сравнительную таблицу К.1 (Приложение К).

Проанализировав возможности разных программных продуктов, можно заключить, что преимуществом разрабатываемого продукта является более низкая стоимость, аппаратно-независимый программный код, генерируемый не только на ST, но и на C++ и Assembler. В перспективе это позволит расширяться

не только на рынки автоматизации, работающие с ПЛК, но и на рынки, работающие с программированием ПЛИС, микроконтроллеров, цифровых сигнальных процессоров, программируемых контроллеров автоматизации.

Ещё одним преимуществом является поддержка стандарта МЭК 61131-3-2016. Благодаря этому ПО будет востребовано у компаний, которые используют среды разработки программирования ПЛК, следующие концепции данного стандарта. Также ПО FSMLogGen является отечественным, что позволяет заменить ряд иностранных аналогов своими возможностями. Необходимость такой замены обусловлена санкциями, введёнными рядом компаний, в отношении российских импортеров оборудования и программного обеспечения АСУ ТП (например, Siemens, Scheider Electric, MATLAB).

При этом стоит отметить, что разрабатываемый продукт обладает довольно низкими системными требованиями, что позволяет конкурировать с другими программными продуктами. При этом ПО обладает простым, интуитивно-понятным интерфейсом. Также стоимость лицензии программного обеспечения FSMLogGen в зависимости от дополнительных возможностей находится в пределах от 70 тыс. рублей до 100 тыс. рублей, что на порядок ниже, чем у аналогов. Таким образом, разрабатываемый продукт является конкурентоспособным и может быть востребован на российском рынке.

3.7 Описание целевых сегментов потребителей

Как было ранее написано, продукт упрощает процессы разработки ПО для АСУ ТП и повышает эффективность взаимодействия технолога и программиста. Ориентируясь на предназначение продукта, а также его относительно небольшую стоимость по сравнению с конкурентами, можно заключить, что основным целевым сегментом являются компании, поставляющие решения в области автоматизации для определенной отрасли. Заказчик (компания, нуждающаяся в автоматизации собственного производства) согласовывает техническое задание с поставщиком таких решений. Таким образом, поставщик разрабатывает систему автоматизации, подбирает

аппаратные и программные средства для её реализации, осуществляет пуск, наладку и тестирование системы, а затем отдает монтажникам или сам осуществляет монтаж системы.

На этапе выбора программных средств программный продукт FSMLogGen может стать достойной альтернативой приведенным ранее продуктам для программирования ПЛК. Также целевым сегментом могут являться не только компании, разрабатывающие системы автоматизации и поставляющие их на рынок, но также и те компании, которые активно используют автоматизацию на собственном производстве без использования возможностей посторонних компаний. В таком случае в каждой компании, занимающейся производством товаров, есть отдел АСУ ТП, который самостоятельно в рамках компании занимается разработкой систем автоматического управления для собственного производства. В частности, таким отделом ведется разработка программного обеспечения для АСУ ТП. На этапе создания ПО для АСУ ТП может быть применен разрабатываемый программный продукт FSMLogGen в рамках программирования ПЛК.

3.8 Бизнес-модель проекта и стратегия продвижения

Применение матрицы Остервальдера довольно эффективно, поскольку отражает как производственный план, так и план продаж, и саму бизнес-модель процесса в целом. Матрица Остервальдера приведена в приложении Л.

Среди способов, которые будут использоваться для продвижения на рынок, можно выделить:

- email-маркетинг (Direct mail с выходом на прямые продажи. Предоставление коммерческого предложения компаниям через отправку писем директорам или ответственным за разработку ПО АСУ ТП в компании);
- работа в социальных сетях и соцмедиа (группа во Вконтакте);
- реклама в поисковых системах (SEM — Search Engine Marketing, SEO + контекстная реклама);

– собственный сайт, Landing Page для установки ПО и предоставления информации.

Методы, приведенные выше, будут способствовать внедрению разработки на рынок, увеличению объема продаж.

3.9 Экономическое обоснование стартап-проекта

3.9.1 Инвестиционные издержки

Далее произведем расчет инвестиций, требуемых для стартап проекта (Таблица 1).

Таблица 1 – Инвестиционные издержки

Наименование	Сумма, руб.
Государственная пошлина за регистрацию ИП	800
Государственная регистрация программы для ЭВМ	6000
Услуги юриста для составления Пользовательского соглашения и договора Оферты	7 000
Первоначальная реклама	40 000
Разработка ПО (без учета регистрации программы для ЭВМ)	957 200
Прочие расходы	20 000
Итого	1 031 000

Инвестиционные издержки проекта составили 1031000 рублей, согласно таблице. Самой большой позицией в инвестиционных издержках является разработка программного обеспечения. В таблице указана стоимость и примерный срок каждого этапа разработки ПО. Так как уже произведена существенная часть написания программного обеспечения, значительная часть инвестиционного цикла уже пройдена. Степень готовности продукта составляет 40% от общих затрат на разработку программного обеспечения. Таким образом, вложено собственных первоначальных инвестиций в размере 382880 рублей. Основные этапы разработки и их стоимость со сроками приведены в таблице 2.

Таблица 2 – Этапы, стоимость и сроки разработки программного обеспечения

Этап	Стоимость, руб.	Примерный срок, мес.
Формирование и анализ требований к ПО	50 000	1
Проектирование	200 000	1,5
Написание программного кода	400 000	10
Отладка и тестирование	400 000	8,5
Внедрение и поддержка программного продукта	157 200	5,5
Итого	957 200	24

Источником финансирования может являться Фонд микрофинансирования Томской области, который занимается предоставлением займов для бизнеса. Также планируется осуществлять привлечение инвестиционного капитала с помощью частных инвесторов. И ещё одним фондом, который способен оказать поддержку бизнесу, является фонд «ТомскИнвестСервис», средства планируется привлекать с его помощью.

3.9.2 Ценовая политика

Для грамотного определения стоимости продукта и выбора метода ценообразования необходимо проанализировать внешние и внутренние факторы, влияющие на цену. К внешним факторам ценообразования относят состояние рынка и спрос, а также конкурентную среду.

К внутренним факторам ценообразования относят маркетинговые цели компании и ее издержки. Внутренние факторы:

- Получение прибыли и её максимизация;
- Покрытие расходов.

Исходя из определенных выше факторов, был выбран метод расчета цены, исходя из издержек, как наиболее оптимальный. С использованием

данного метода учитывается высокая цена продукции конкурентов. Радикальное снижение цены на разрабатываемое ПО позволяет привлечь клиентов, которые не обладают возможностями купить для своей компании такую дорогую программу. Значительно меньшая цена по сравнению с конкурентами предоставляет возможность работать в другой ценовой категории. И тем не менее, с учетом этого первоначальные вложения и текущие затраты довольно быстро окупаются.

В первую очередь необходимо спрогнозировать и рассчитать ежемесячные издержки (Таблица 3).

Таблица 3 – Ежемесячные издержки

Наименование затрат	Сумма, руб.
Поддержка программы от компании разработчика	22 000
Поддержание расчетного счета в банке	500
Маркетинговая кампания	20 000
Страховые взносы	3 500
Хостинг сайта + интернет	500
Итого	46500

Далее рассмотрим вероятный план продаж и рассчитаем приблизительную стоимость одной единицы продажи. Предположительно в среднем в месяц на первоначальном этапе проекта осуществляется 4 продажи лицензии на программное обеспечение.

$$\text{Себестоимость} = \frac{46500}{4} = 11625 \text{ рублей.}$$

Следовательно, минимальный процент от средневзвешенной цены (при её расчетах используется стоимость лицензий MATLAB, SimInTech, Simatic Step 7, средневзвешенная цена равна 1 180 000) по фактору количества продаж лицензии на программное обеспечение составит:

$$\text{Процент (комиссия)} = \frac{11625}{1180000} = 0,985\%.$$

Принято решение об установке комиссия в 7% с учетом надбавки, покрывающей инвестиционные издержки. Это позволит в случае продажи

лицензий менее 4 в месяц получить запас финансовой прочности на покрытие ежемесячных издержек.

Средняя цена продажи одной лицензии составит:

$$\text{Цена} = 1180000 \times 0,07 = 82600 \text{ рублей.}$$

Таким образом, составлен расчет с минимальной ценой и минимальной версией ежемесячной подписки, но все равно затраты быстро окупались.

3.9.3 Показатели эффективности проекта

Определение эффективности проекта позволяет оценить уровень его привлекательности для потенциальных участников и привлечения внешних инвестиций.

Ставка дисконтирования

По прогнозу ЦБ РФ инфляция в 2022 году составит 18-23%. Усредненное значение инфляции за 2022 – 20,5%.

Взята довольно существенная ставка дисконтирования, значение которой находится на уровне инфляции и составляет 20%.

Период окупаемости

Исходя из таблицы инвестиционных издержек и средней стоимости одной лицензии, а также среднего количества продаж на первоначальном этапе в месяц (в среднем, 4 продажи в месяц), период окупаемости составит:

$$PP = \frac{1031000}{4 \times 82600} = 3,12$$

Таким образом, точный период окупаемости проекта – 3,12 месяца.

Дисконтированный период окупаемости

В зависимости от набора языков программирования стоимость лицензии составляет 80 (базовая версия, включает язык ST), 90 (версия лицензии с одним добавленным языком программирования C++ или Assembler) 100 тыс. рублей (версия лицензии с тремя языками программирования ST, C++, Assembler).

Пусть в первый месяц произведено две закупки базовой лицензии, стоимость которой 80 тыс. рублей, одна закупка лицензии за 90 тыс. рублей и

одна закупка полной версии за 100 тыс. рублей. Составим ежемесячную таблицу прогноза продаж лицензий (Таблица 4). При этом стоит включить в таблицу доход с учетом подписок на обновления ПО (1% от стоимости лицензии).

Таблица 4 – Денежные потоки от лицензий

Период	Версии лицензии						Итого, тыс. руб. (Лицензии)
	Базовая, шт.	Улучшенная, шт.	Полная, шт.	Базовая + подписка, шт.	Улучшенная + подписка, шт.	Полная + подписка, шт.	
1	1	0	1	1	1	0	350
2	2	0	0	0	1	1	350
3	0	1	0	1	1	1	360
4	1	0	1	0	1	1	370
5	0	2	2	0	0	0	380
6	0	1	2	1	0	0	370
7	1	1	0	1	1	0	340
8	2	0	1	1	0	0	340
9	1	2	0	0	1	0	350
10	0	1	1	0	1	1	380
11	1	1	0	1	0	1	350
12	2	0	0	2	0	0	320

Также стоит отметить, что у пользователей есть возможность осуществлять покупку лицензий с подписками, значит, в последующие месяцы пользователям будет необходимо производить платежи в размере стоимости подписки на обновления. Отдельно составлена таблица для того, чтобы учесть поток денежных средств исходя из лицензий с подписками (Таблица 5). Денежный поток определяется как 1% от стоимости того или иного вида лицензии. Для базовой лицензии стоимость подписки каждый месяц – 800 рублей, улучшенная – 900 рублей, полная – 1000 рублей. Осуществив покупку лицензии с подпиской на обновления, пользователь в первый месяц не платит за неё, плата от подписки взимается в последующие месяцы после покупки.

Таблица 5 – Денежные потоки от подписок на каждый месяц

Месяцы	Версия подписки			Сумма на текущий месяц, тыс. рублей	Сумма на следующий месяц, тыс. рублей
	Базовая подписка, шт.	Улучшенная подписка, шт.	Полная подписка, шт.		
1	1	1	0	0	1,7
2	0	1	1	1,7	3,6
3	1	1	1	3,6	6,3
4	0	1	1	6,3	8,2
5	0	0	0	8,2	8,2
6	1	0	0	8,2	9
7	1	1	0	9	10,7
8	1	0	0	10,7	11,5
9	0	1	0	11,5	12,4
10	0	1	1	12,4	14,3
11	1	0	1	14,3	16,1
12	2	0	0	16,1	17,7

В результате составлена таблица поступления денежных потоков по каждому месяцу исходя из продаж лицензии и с учетом подписок (Таблица 6).

Таблица 6 – Суммарный денежный поток

Месяцы	Сумма, тыс. рублей
1	350
2	351,7
3	363,6
4	376,3
5	388,2
6	378,2
7	349
8	350,7
9	361,5
10	392,4
11	364,3
12	336,1

Дисконтированный период окупаемости в обобщенном виде рассчитывается по формуле:

$$DPP = \sum_{t=1}^n \frac{CF_t}{\left(1 + \frac{r}{100}\right)^t} \geq 0,$$

где CF – денежные поступления в период t;

r – ставка дисконтирования;

I – инвестиции в проект.

Исходя из расчетов в программе Mathcad по приведенной формуле дисконтированный период окупаемости составляет:

$$DPP = 4 + \frac{1031000 - 927800}{156000} = 4,7 \text{ мес.}$$

Чистый дисконтированный доход

Величина чистого дисконтированного дохода определяется как сумма дисконтированных денежных потоков в каждом месяце за минусом первоначальных инвестиций.

$$NPV = \sum_1^{12} PV - I = 1610000 - 1031000 = 579000 \text{ руб.}$$

Таким образом, положительный чистый дисконтированный доход за 1 год показывает эффективность проекта.

Индекс прибыльности

Индекс рентабельности инвестиций, который также называют индексом доходности и индексом прибыльности – это относительный показатель эффективности инвестиций. Он показывает уровень дохода, получаемый на один рубль инвестиций, с учетом временной стоимости денег

$$PI = \frac{1610000}{1031000} = 1,561.$$

Индекс прибыльности PI за 1 год больше 1, что означает эффективность проекта.

Внутренняя норма доходности

Внутренняя норма доходности (рентабельности) (IRR) – представляет собой ту ставку дисконтирования, при которой чистый дисконтированный доход равняется нулю.

Внутренняя норма рентабельности рассчитана с помощью инструмента Mathcad 15 (Рисунок 26).

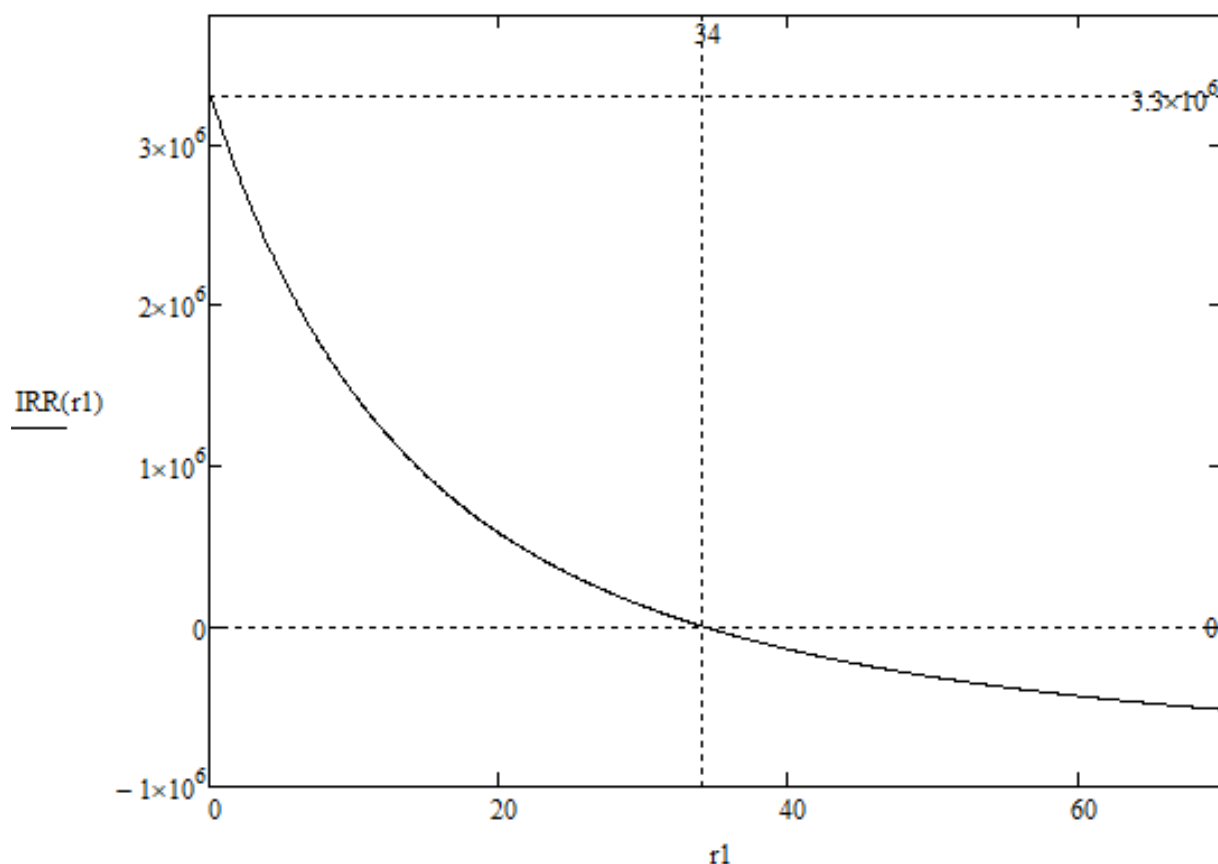


Рисунок 26 – Расчет внутренней нормы рентабельности в Mathcad 15

Из рисунка видно, что график пересекает ось абсцисс примерно на 34%. Именно это значение и является внутренней нормой доходности. Таким образом, разница между ставкой дисконтирования и внутренней нормой рентабельности (IRR) достаточно большая. Поэтому риски проекта с точки зрения удорожания заемных средств являются низкими.

Основные показатели эффективности данного проекта, полученные в ходе расчетов отражены в таблице (Таблица 7).

Таблица 7 – Основные показатели эффективности

Показатель	Значение
Ставка дисконтирования, %	20
Период окупаемости – РВ, мес.	3,12
Дисконтированный период окупаемости – DPВ, мес.	4,7
Чистый индексированный доход – NPV, руб.	579000
Индекс прибыльности – PI	1,561
Внутренняя норма доходности – IRR, %	34

Таким образом, в результате проведения экономических расчетов подтверждается эффективность проекта. При успешном старте продаж и после формирования узнаваемого бренда появится возможность повысить цену лицензии и подписки, что дополнительно способствует улучшению финансовых показателей проекта.

Вывод по разделу

Итак, в результате написания раздела «Концепция стартап-проекта», создан план продвижения программного обеспечения FSMLogGen на рынке АСУ ТП России. Произведен анализ конкурентов. Можно сделать вывод, что программный продукт будет успешно конкурировать на рынке, поскольку обладает рядом преимуществ перед аналогами. Сформирована модель денежных потоков от лицензий, составлена матрица Остервальдера. Эффективность проекта доказана экономическими расчетами.

4. Социальная ответственность

В данной работе производится создание программного обеспечения, позволяющего упростить процессы разработки алгоритмов и программ для АСУ. Потенциальные пользователи разрабатываемого решения – программисты и технологи на предприятиях, разрабатывающих системы автоматизации или использующих автоматизацию в своей деятельности. Разработка ПО осуществляется в лаборатории ОАР НИ ТПУ, размеры рабочей зоны - 7,5×5,5 м. Актуальность проекта обусловлена проблемой взаимодействия технолога и программиста на этапе составления технического задания на разработку программного обеспечения для систем автоматизации предприятия. Разрабатываемый программный продукт позволит упростить данный процесс, ускорить разработку ПО, уменьшить человеческий фактор.

В работе используется следующее оборудование рабочей зоны: ноутбук, лабораторный стол, Ethernet-кабель, лампа настольная, блок питания. Рабочие процессы, связанные объектом исследования, осуществляющиеся в рабочей зоне, включают: разработку программного обеспечения для генерации программного кода, написание программы на языке FBD в среде программирования Owen Logic с использованием сгенерированного кода, проверку работоспособности программного обеспечения.

В разделе освещаются проблемы производственной и пожарной безопасности, эргономики, а также охраны окружающей среды. Охрана труда в Российской Федерации представляет собой важный комплекс экологических, социально-экономических и иных мероприятий, нацеленных на обеспечение безопасности трудовой деятельности [50].

В международном стандарте [51] определено, что социальная ответственность является ответственностью организации за воздействие её решений и деятельности на общество и окружающую среду через прозрачное и этическое поведение.

4.1 Правовые и организационные вопросы обеспечения безопасности

4.1.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства.

Правовые нормы, регулирующие взаимоотношения между сотрудником и организацией, включают в себя положения по оплате труда, режиму рабочего времени. Основные пункты отношений между сотрудником и организацией описаны в трудовом кодексе РФ [52].

Количество рабочего времени нормировано и не может быть выше 40 часов в неделю. На протяжении рабочего дня разработчик программного обеспечения должен иметь перерыв для отдыха и принятия пищи продолжительностью не менее 30 минут и не более двух часов.

Работу разработчика можно отнести ко второй категории работ (IIa), поскольку разработка ПО может быть классифицирована как работа, производимая сидя, и сопровождающаяся незначительным физическим напряжением, а тестирование работоспособности ПО производится на ПР200, являющегося небольшим прибором, масса которого 0,6 кг. Известно, что согласно пункту 3.2. «ТОИ Р-45-084-01, Типовая инструкция по охране труда при работе на персональном компьютере», устанавливается, что продолжительность непрерывной работы с компьютером без регламентированного перерыва не должна превышать 2-х часов. Поэтому для второй категории работ предусматриваются 2 перерыва по 15 минут, используемые через 2 часа после начала рабочей смены и через 2 часа после обеденного перерыва.

4.1.2 Организационные мероприятия при компоновке рабочей зоны

При разработке ПО используется ноутбук. Рабочее место обустраивается в соответствии с ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования» [53]. Помещением, где ведётся разработка программного обеспечения, является лаборатория Томского политехнического университета. В

случае недоступности Wi-Fi подключения общедоступной университетской сети, будет необходимо использовать Ethernet-кабель для подключения ноутбука к интернету. Также разработчик ПО при проведении тестов кодогенератора использует программируемое реле ПР200. Осуществление данных работ должно производиться в соответствии с ГОСТ 12.2.033-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ стоя. Общие эргономические требования» [54].

4.2 Производственная безопасность

Список вредных и опасных факторов, характерных для производственной среды, приведены в таблице 8. В ней указаны факторы, которые возникают на стадиях разработки и тестирования объекта.

Таблица 8 – Возможные опасные и вредные производственные факторы на рабочем месте разработчика программного обеспечения

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Отклонение показателей микроклимата	СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания
Превышение уровня шума	СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания
Умственное перенапряжение	МР 2.2.9.2311-07. Состояние здоровья работающих в связи с состоянием производственной среды. Профилактика стрессового состояния работников при различных видах профессиональной деятельности.

Продолжение таблицы 8

Перенапряжение зрительного анализатора	МР 2.2.9.2311-07. Состояние здоровья работающих в связи с состоянием производственной среды. Профилактика стрессового состояния работников при различных видах профессиональной деятельности.
Недостаточная освещенность рабочей зоны	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания
Эмоциональные перегрузки	МР 2.2.9.2311-07. Состояние здоровья работающих в связи с состоянием производственной среды. Профилактика стрессового состояния работников при различных видах профессиональной деятельности.
Повышенное значение напряжения в электрической цепи, которое может пройти через тело человека	ГОСТ 12.1.019-2017. ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты (с Поправкой)
Отсутствие естественного света	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания

4.2.1 Отклонение показателей микроклимата

Причиной отклонения показателей микроклимата является интенсивная умственная активность человека, перемещение предметов, масса которых менее 1 кг, в положении сидя или стоя.

В случае продолжительного перенапряжения, связанного с перемещением предметов до 1 кг, может произойти развитие профессиональных заболеваний опорно-двигательного аппарата (ОДА) и периферической нервной системы (ПНС). Интенсивная умственная активность человека может стать причиной возникновения эмоционального напряжения, которое способно привести к стрессу, заболеваниям сердечно-сосудистой системы (атеросклероз, ишемическая болезнь сердца), к невротическим расстройствам.

Энергозатраты данной категории работ составляют 151-200 ккал/ч. На рабочих местах должны обеспечиваться оптимальные параметры микроклимата. Обеспечение оптимальных параметров микроклимата регламентируется исходя из требований СанПиН 1.2.3685-21 [55]. Они приведены в таблице 9.

Таблица 9 – Оптимальные и допустимые величины показателей микроклимата на рабочих местах

Допустимые значения						
Период года	Температура воздуха, °С		Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
	Диапазон ниже оптимальных величин	Диапазон выше оптимальных величин			Диапазон ниже оптимальных величин	Диапазон выше оптимальных величин
Холодный	17,0-18,9	21,1-23,0	16-24	15-75	0,1	0,3
Теплый	18,0-19,9	22,1-27,0	17-28	15-75	0,1	0,4

В зимнее время средством защиты является отопление. Поддержание показателя относительной влажности предлагается осуществлять с использованием увлажнителей воздуха. В летнее время должны быть предусмотрены системы вентилирования и кондиционирования.

4.2.2 Превышение уровня шума

Источником возникновения фактора являются студенты, работающие в лабораториях ОАР, стендовое оборудование.

Наиболее типичными заболеваниями у людей, работающих в условиях воздействия интенсивного шума, являются заболевания сердечно-сосудистой системы (гипертоническая болезнь сердца, коронардиосклероз, стенокардия, инфаркт миокарда), шумовая болезнь.

Допустимые значения звукового давления в соответствии с санитарными нормами [55] приведены в таблице 10.

Таблица 10 – Допустимые уровни звукового давления

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука, дБА
31,5	63	125	250	500	1000	2000	4000	8000	
99	95	87	82	78	75	73	71	69	80

В качестве индивидуальных средств защиты от негативного воздействия звукового давления можно применять беруши и наушники. В качестве средств коллективной защиты могут быть использованы оградительные устройства, глушители шума, звукопоглощающие устройства и материалы.

4.2.3 Недостаточная освещенность рабочей зоны

Данный фактор возникает из-за отсутствия организации естественного освещения в аудиториях. Недостаток освещенности рабочей зоны приводит к снижению работоспособности. В таблице 11 представлены допустимые значения показателей освещенности рабочих мест в соответствии с СанПиН [56]. Средствами нормализации освещенности рабочих мест служат осветительные приборы, защитные очки, светофильтры.

Таблица 11 – Допустимые значения освещенности в учебных помещениях, кабинетах, аудиториях, комнатах самоподготовки

Искусственное освещение				
Освещенность, лк			Показатель дискомфорта М, не более	Коэффициент пульсации освещенности, Кп, %, не более
При комбинированном освещении		При общем освещении		
всего	от общего			
–	–	300	21	10

4.2.4 Повышенное значение напряжения в электрической цепи, которое может пройти через тело человека

Источники возникновения данного фактора: электричество для питания ПЭВМ, а также электричество для питания ПР200. Причиной прохождения тока по телу человека является прикосновение к проводам, оказавшимся под напряжением. Предельно допустимые значения силы токов и напряжений, расположены в таблице 12.

Таблица 12 – Предельно допустимые значения силы тока и напряжения

	Переменный ток при частоте, Гц:		Постоянный ток
	50	400	
Напряжение, В	2	2	8
Сила тока, мА	0,3	0,4	1

Средствами защиты от предельно допустимых значений напряжения являются изолирующие устройства, устройства автоматического отключения и др [57].

Помещение, согласно ПУЭ [58] относится к помещению с повышенной опасностью, характеризуется наличием возможности одновременного прикосновения человека к металлоконструкциям зданий, имеющим соединение с землей, технологическим аппаратам, механизмам и т.п., с одной стороны, и к металлическим корпусам электрооборудования(открытым проводящим частям), с другой. Согласно приказу от 15 декабря 2020 года [59], группа

электробезопасности разработчика ПО – I. В соответствии с этой группой должен быть заведён журнал, где будут отображены ФИО работника, должность, дата присвоения группы I по электробезопасности, подпись проверяемого и проверяющего. Должен быть проведен инструктаж работника и проверка приобретенных навыков безопасных способов работы.

4.2.5 Умственное перенапряжение

При разработке ПО ресурсы человека задействуются в большом объёме. В результате может возникнуть умственное перенапряжение, стресс.

Для профилактики и снижения умственного перенапряжения, согласно [60], работникам умственного нервно-напряженного труда рекомендуется проводить регламентированные перерывы длительностью 7-10 минут, комплексы физических упражнений.

4.2.6 Перенапряжение зрительного анализатора

Работа разработчика происходит за монитором ПЭВМ. В результате продолжительной работы у человека возникает перенапряжение зрительного анализатора. В соответствии с [6061] снятие зрительного перенапряжения может производиться с использованием профилактических упражнений зрительной гимнастики.

4.2.7 Эмоциональные перегрузки

Основными факторами развития стресса, приводящего к эмоциональным перегрузкам, являются [60]:

- при умственной нагрузке – длительный и ненормированный рабочий день с работой в сменном режиме, служебные командировки и др;
- при зрительной нагрузке – высокая точность работы.

При разработке ПО возникают оба вида нагрузок. Мерой снижения эмоционального перенапряжения служат сеансы «психологической разгрузки» продолжительностью 10-20 минут.

4.2.8 Отсутствие естественного света

Естественным освещением [61] называется освещение помещений светом неба, проникающим через световые проёмы в наружных ограждающих конструкциях. Отсутствие естественного света возникает при реализации проекта в лабораториях ОАР на цокольном этаже. На данных этажах может не хватать естественного света. Отсутствие естественного освещения приводит к усталости глаз, головной боли. Процесс проведения тестирования разработанного ПО на реальном оборудовании относится к зрительной работе очень высокой точности, размер объекта варьируется от 0,15 до 0,3 мм. Данная работа относится ко второму разряду зрительной работы. В таблице 13 представлены требования к проведению работ данного типа.

Таблица 13 – Требования к освещению рабочей зоны

Подразряд зрительной работы	Контраст объекта с фоном	Фон	КЕО е, %	
			Верхнее или комбинированное освещение	Боковое освещение
А	Малый	Темный	4,2	1,5
Б	Малый Средний	Средний Темный		
В	Малый Средний Большой	Светлый Средний Темный		
Г	Средний Большой	Светлый Средний		

Таким образом, в соответствии с СП 52.13330.2016 КЕО при верхнем или комбинированном освещении – 4,2%, при боковом – 1,5%. Недостаток или отсутствие естественного света определяется через коэффициент естественной освещенности. Если его величина ниже требуемой, то для увеличения КЕО плоскость остекления лабораторных окон может быть расположена под углом 15-20 градусов к вертикали. В такое окно будет попадать большее количество естественного света, чем в вертикальное. Также для повышения КЕО может применяться световая отделка внутренних поверхностей с большими

коэффициентами отражения, что приводит к увеличению доли естественной освещенности. Также возможно применение световодов, отражающих потолков.

В редких случаях допускается отсутствие естественного света или его недостаток, при условии, что это будет компенсировано за счет искусственного освещения.

4.3 Экологическая безопасность

В этом подразделе рассматриваются особенности воздействия разработки и тестирования ПО на окружающую среду, определяются источники её загрязнения.

При разработке кодогенератора и тестировании его работоспособности задействуется ПЭВМ. Для расширения возможностей ноутбука осуществляется замена его деталей при выходе из строя или при необходимости улучшения технических характеристик устройства. Возникает необходимость утилизации неисправных (устаревших) деталей. При этом известно, что комплектующие создаются с использованием драгоценных металлов. Радиоэлектронный лом комплектующих ПЭВМ необходимо сдавать на аффинажные заводы – предприятия, получающие высокочистые драгоценные металлы. В результате аффинажные заводы из микросхем получают банковские слитки [62].

4.3.1 Влияние объекта исследования на гидросферу

Если осуществлять утилизацию отходов классическими способами, например, захоронениями отходов на полигонах, то в случае близости таких полигонов к водоемам, отходы от утилизации деталей могут оказаться в речных водах. При этом электронное оборудование состоит из опасных химических соединений, в результате разложения которых может причиняться вред природе. Мощности сооружений для очистки бытовых сточных вод должно хватить для сведения к нулю негативного эффекта от попадания отходов. Но попадание твердых бытовых отходов в пресные и сточные воды удорожает их очистку. Необходимо сдавать комплектующие компьютера на аффинажные заводы с

целью переработки, а также в компании-производители оборудования для утилизации. После процесса утилизации на захоронение отправляется не 96%, а всего около 7% отходов. Все остальное – на вторичное использование [62].

4.3.2 Влияние объекта исследования на литосферу

Захоронение твердых бытовых отходов производится на полигонах. Для минимизации ущерба комплектующие сдаются на переработку. Для переработки отходов может использоваться высокотемпературный пиролиз. В результате процесса высокотемпературного пиролиза образуются твердые продукты в виде шлака и золы, т.е. непиролизуемые остатки. Технологическая схема этого метода утилизации включает в себя четыре последовательных этапа, описанных в [62]. Высокотемпературный пиролиз позволяет экологически чисто, экономически выгодно и технически довольно просто перерабатывать твердые бытовые отходы без их предварительной подготовки, т.е. сортировки, сушки и т.д.

4.3.3 Влияние объекта исследования на атмосферу

В результате производства электронных компонентов для ПЭВМ происходит загрязнение атмосферы. Вредные выбросы образуются при производстве электроэнергии, питающей ноутбук, поскольку для её получения применяется ископаемое топливо (например, уголь). Для регулирования содержания вредных веществ в атмосфере устанавливаются нормативы на допустимые выбросы [63]. Таким образом, регулируется оптимальное содержание вредных веществ в атмосфере. Предприятия для снижения выбросов могут использовать электролизеры или фильтры.

Также определена категория объекта, оказывающего значительное негативное воздействие на окружающую среду, в соответствии с Постановлением Правительства РФ от 31 декабря 2020 г. N 2398 [64]. Данная категория является второй (II), поскольку электрическая энергия для питания ноутбука вырабатывается на ТЭЦ-1, мощность энергоблока 15 МВт, что меньше

250 МВт.

4.4 Безопасность в чрезвычайных ситуациях

4.4.1 Анализ вероятных ЧС, которые может инициировать объект исследований

При реализации работ по разработке ПО, а также его тестировании, могут возникать различные ЧС: разрушение зданий в результате разрядов атмосферного электричества, ураган, землетрясение, а также пожар. Наиболее вероятной ЧС является пожар. При тестировании ПО на ПР200 причиной пожара могут стать токи короткого замыкания, искрение в местах неплотных контактов проводов и др.

4.4.2 Обоснование мероприятий по предотвращению ЧС

В случае возникновения пожара тушение горящего электрооборудования должно производиться огнетушителями ОУ-5 или ОУ-10 (углекислотные) и огнетушителями типа ОП-10 (порошковые).

Во избежание пожара необходимо соблюдать ряд правил:

- соблюдать чистоту в помещении, вовремя выносить мусор;
- план эвакуации из здания следует располагать на видном месте;
- работать только при наличии исправного оборудования;
- по окончании работы на оборудовании проверить выключены ли электроприборы.

При этом необходимо также обеспечить:

- присутствие наглядных пособий, для предотвращения пожара;
- содержание помещения в чистоте;
- противопожарный инструктаж сотрудников

Категория помещений по взрывопожарной и пожарной опасности, в соответствии с [65], является категорией Г (умеренная пожароопасность), поскольку в лаборатории ОАР при работе оборудования материалы (пластик, металлы) могут находиться в горячем состоянии, может выделяться лучистое

тепло, не исключено возникновение искр.

Для предотвращения травматизма следует не производить работы по тестированию ПО на оборудовании без индивидуальных средств защиты (рукавицы, защитные очки, спецодежда), не производить такие работы, как регулировка, наладка при работающем оборудовании. Если пожар не удастся ликвидировать самостоятельно, необходимо позвонить по номеру 101 в пожарную охрану, сообщив о пожаре и месте его возникновения, а затем покинуть помещение в соответствии с планом пожарной эвакуации.

Вывод по разделу

В данном разделе были рассмотрены вопросы социальной ответственности при разработке программного обеспечения и его тестировании на реальном оборудовании. К таким вопросам относятся социальная и экологическая безопасность, а также безопасность в чрезвычайных ситуациях, правовые аспекты обеспечения безопасности.

Были проанализированы вредные и опасные факторы, возникающие в ходе разработки и тестирования ПО. К ним относятся отклонение микроклимата в помещении, недостаточная освещённость рабочей зоны, пожароопасность, умственное перенапряжение, превышение уровня шума, эмоциональные перегрузки, отсутствие естественного света, перенапряжение зрительного анализатора. Перечисленные факторы при превышении норм оказывают отрицательное воздействие на здоровье человека.

Как было описано ранее, категория помещения, где производится разработка ПО и проверка его работоспособности, согласно ПУЭ [58], относится к помещению с повышенной опасностью. Группа электробезопасности, в соответствии с Приказом от 15 декабря 2020 [59] – первая (I). Категория тяжести труда по СанПиН 1.2.3685 [55] является второй (IIa). Категория помещений по взрывопожарной и пожарной опасности, в соответствии с [65], является категорией «Г» (умеренная пожароопасность). Также определена категория объекта, оказывающего значительное негативное воздействие на окружающую среду, в соответствии с Постановлением Правительства РФ от 31 декабря 2020

г. N 2398 [64]. Данная категория является второй (II).

В подразделе «Экологическая безопасность» были рассмотрены негативные аспекты, способные повлиять на окружающую среду в ходе проведения работ.

Установлено, что наиболее вероятно чрезвычайной ситуацией является пожарная опасность. Для этой ЧС были разработаны организационные мероприятия по её предотвращению и ликвидации.

В результате использования программного обеспечения с целью его тестирования на оборудовании могут возникать вредные и опасные производственные и экологические факторы. Их конкретное описание требует применения программного обеспечения вне рамок тестирования, на реальном станке. При соблюдении регламентов и норм, станок, для которого генерируется программный код, не представляет опасностей для обслуживающего персонала и работников.

Заключение

Результатом выполнения ВКР стало разработанное программное обеспечение для проектирования автоматных моделей и их преобразования в программный код.

В ходе работы были реализованы: отрисовка состояний и переходов между ними, редактирование графических примитивов, сохранение и загрузка проектов, конвертер автоматных моделей на язык ST для программирования ПЛК и ПР. Помимо этого, осуществлена возможность создания параллельных конечных автоматов.

С использованием разработанного функционала программы был спроектирован алгоритм на основе автоматных моделей для управления станком: фрезером наклонных поверхностей надressорной балки железнодорожной тележки грузового вагона. В результате из модели работы станка сгенерирован код на языке ST и применен к макросу на языке ST для модели устройства ПР200 программного обеспечения Owen Logic.

В результате формального описания алгоритма конечными автоматами синтезирована программа на языке FBD с применением макросов на языке ST в среде разработки Owen Logic. Возможность использования макросов на языке ST позволила осуществить применение программного кода, полученного из автоматных моделей, на относительно дешевом программируемом реле взамен более дорогого промышленного контроллера ПЛК160.

Таким образом, было предложено более эффективное и экономичное решение в виде программного обеспечения для проектирования автоматных моделей и генерации кода.

Conclusion

The result of the work was the developed software for designing automata models and their conversion to program code.

In the course of the work the following was implemented: drawing of states and transitions between them, editing of graphic primitives, saving and loading of projects, converter of automata models into ST language for PLC and programmable relay programming. Besides, possibility to create parallel finite state machines are implemented.

Using the developed program functional, an algorithm was designed on the basis of automata models to control a machine tool: milling machine for inclined surfaces of a bolster of a freight car bogie. As a result, the code in ST language was generated from the model of machine work and applied to the macro in ST language for PR200 device model of Owen Logic software.

As a result of formal description of the algorithm by finite automata, a program is synthesized in FBD language with macros in ST language applied in Owen Logic development environment. The possibility of using macros in ST language made it possible to apply program code, derived from automata models, on a relatively cheap programmable relay instead of more expensive industrial controller PLC160.

Thus, it was proposed a more efficient and cost-effective solution in the form of software for the design of automata models and code generation.

Список публикаций студента

1. Тутов, И. А. Кодогенерация посредством графов-переходов / И. А. Тутов, В. С. Гительман // Молодежь и современные информационные технологии : Сборник трудов XVIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 22–26 марта 2021 года. – Томск: Национальный исследовательский Томский политехнический университет, 2021. – С. 451-453.

2. Тутов, И. А. Кодогенератор графического представления детерминированного конечного автомата для систем автоматизации / И. А. Тутов, В. С. Гительман, О. Б. Воскобойникова // Известия Тульского государственного университета. Технические науки. – 2021. – № 12. – С. 474-480.

Список литературы

1. Поликарпова Н. И., Шалыто А. А.. Автоматное программирование — СПб.: Питер, 2011. – 176 с.
2. Шалыто А. А. Switch-технология. Алгоритмизация и программирование задач логического управления – СПб.: Наука, 1998 – 628с.
3. IEC 61131-3: 2013 // Webstore International Electrotechnical Commission – URL: <https://webstore.iec.ch/publication/4552> – Режим доступа: открытый. – Дата обращения: 18.05.2022.
4. Шалыто А.А. Парадигма автоматного программирования // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2008. №53. С. 6-7.
5. Конечный автомат - Finite-state machine // wikiboard – URL: https://wikiboard.ru/wiki/Finite-state_machine – Режим доступа: открытый. – Дата обращения: 19.05.2022
6. Шалыто А.А. Использование граф-схем и графов переходов при программной реализации алгоритмов логического управления. II // Автомат. и телемех, 1996. Вып. 7. С. 144 – 169.
7. Граф. Структуры данных // Граф. – URL: <https://progcpp.ru/data-graph> – Режим доступа: открытый. – Дата обращения: 19.05.2021.
8. Описание регулирования температуры по двухпозиционному закону // studwood.net – URL: https://studwood.net/1715315/tehnika/opisanie_regulirovaniya_temperatury_dvuhpozitsionnomu_zakonu – Режим доступа: открытый. – Дата обращения: 19.05.2022.
9. Понимание языков программирования IEC61131-3 // Про АСУ ТП – URL: https://www.proasutp.com/articles/plc/understanding_the_iec61131_3_programming_languages.html?page=4 – Режим доступа: открытый. – Дата обращения: 19.05.2022.
10. Руководство пользователя // ОВЕН – URL: https://docs.owen.ru/product/programmnoe_obespechenie_owen_logic/986/53812#to

[pic-53812](#) – Режим доступа: открытый. – Дата обращения: 19.05.2022.

11. СФ-1/ Станок фрезерный для механической обработки наклонных поверхностей надрессорной балки // ТЕХНАД – URL: https://tehnad.ru/view_151.html – Режим доступа: открытый. – Дата обращения: 19.05.2022.

12. Программное обеспечение OwenLogic // ОВЕН – URL: https://owen.ru/product/programmnoe_obespechenie_owen_logic – Режим доступа: открытый. – Дата обращения: 19.05.2022.

13. Список основных изменений среды программирования OWEN Logic // ОВЕН – URL: http://ftp-ow.owen.ru/softupdate/OWEN%20Logic/infoOf1_22.html – Режим доступа: открытый. – Дата обращения: 19.05.2022.

14. Патент РФ № 2016129653, 19.07.2016. Заозерский С.А., Березов И.В., Коромысличенко В.Н., Николаев Д.А., Ничипорович О.П., Охтилев М.Ю., Пикулёв П.А., Россиев А.Ю., Черников А.Д., Чуприков А.Ю. Способ автоматизированного проектирования производства и эксплуатации прикладного программного обеспечения и система для его осуществления. Патент России №2 676 405. 2016 [Электронный ресурс]. – URL: <https://new.fips.ru/iiss/document.xhtml?faces-redirect=true&id=24a3ec8c969599c4944c2e0596d7500a> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

15. Патент РФ № 2020108174, 26.02.2020. Антипинский А.С., Домоуховский Н.А., Комаров Д.Е., Синадский А.Н. Способ выявления аномалий в работе сети автоматизированной системы. Патент России №2 738 460. 2020 [Электронный ресурс]. – URL: <https://new.fips.ru/iiss/document.xhtml?faces-redirect=true&id=26b531f4ebf52c84498cd3a779371c40> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

16. Патент РФ № 2019128044, 08.03.2018. Аллори Матьё, Дююни Никола. Программируемая логическая схема для управления электрической установкой, в частности ядерной установкой, ассоциированная с устройством и

способом управления. Патент России №2 769 961. 2018 [Электронный ресурс]. – URL: <https://new.fips.ru/iiss/document.xhtml?faces-redirect=true&id=abc21be781aded97d710ecd45a9e94c8> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

17. Патент РФ № 2010100992/08, 14.01.2010. Бюл. № 10 / Окладников С. Способ разработки последовательности машинных команд вычислительной машины. Патент России №2 447 487. 2010 [Электронный ресурс]. – URL: <https://new.fips.ru/iiss/document.xhtml?faces-redirect=true&id=7330cab87f21493a89c6202d85f6e94f> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

18. АСУ ТП (рынок России) // TADVISER – URL: [https://www.tadviser.ru/index.php/Статья:АСУ_ТП_\(рынок_России\)](https://www.tadviser.ru/index.php/Статья:АСУ_ТП_(рынок_России)) – Режим доступа: открытый. – Дата обращения: 12.05.2022.

19. Облачная трансформация глобального и российского рынков ПЛК, АСУТП И MES // ГОРТ – URL: <https://gort.su/posts/rinokasu2021> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

20. J'son & Partners Consulting. // TADVISER – URL: https://www.tadviser.ru/index.php/Компания:J'son_&_Partners_Consulting – Режим доступа: открытый. – Дата обращения: 12.05.2022.

21. Класс ОКВЭД 62 - Разработка компьютерного программного обеспечения, консультационные услуги в данной области и другие сопутствующие услуги // БУХпрофи – URL: <https://www.buxprofi.ru/spravochnik/okved-2/razdel-J/klass-okved-62> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

22. Стратегия развития отрасли информационных технологий в России // TADVISER – URL: https://www.tadviser.ru/index.php/Статья:Стратегия_развития_отрасли_информационных_технологий_в_России. – Режим доступа: открытый. – Дата обращения: 12.05.2022.

23. Production Code Generation with Simulink and Embedded Coder // MathWorks – URL: <https://www.mathworks.com/videos/production-code-generation-with-simulink-and-embedded-coder-99132.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
24. Программное обеспечение динамического моделирования simintech // МАДИ – URL: <https://www.madi.ru/3420-kafedra-avtomatizaciya-proizvodstvennyh-processov-materia.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
25. Symbian C++. Программирование для мобильных телефонов // КО ИТ ДЛЯ БИЗНЕСА – URL: https://ko.com.ua/symbian_c_programmirovanie_dlya_mobilnyh_telefonov_53387 – Режим доступа: открытый. – Дата обращения: 12.05.2022.
26. Introduction to UniMod. // EXECUTABLE UML UNIMOD – URL: <https://unimod.sourceforge.io/intro.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
27. Yalindu Statechart Tools use the power of state machine. // Yalindu Statechart Tools – URL: <https://www.itemis.com/en/yalindu/state-machine> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
28. Язык ДРАКОН // DrakonHub – URL: <https://drakonhub.com/ru/drakon> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
29. Beremiz - свободная среда программирования ПЛК. Часть 1 // Сообщество EasyElectronics – URL: <http://we.easyelectronics.ru/plc/beremiz-svobodnaya-sreda-programmirovaniya-plk-chast-1.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
30. CoDeSys – повседневный инструмент программиста ПЛК // Автоматизация в промышленности – URL: <https://avtprom.ru/article/codesys-%E2%80%93-povsednevnyi-instrumen> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

31. Преобразование графов переходов, представленных в формате MS Visio, в исходные коды программ для различных языков программирования (инструментальное средство MetaAuto) // УНИВЕРСИТЕТ ИТМО Кафедра «Технологии программирования» – URL: <http://is.ifmo.ru/projects/metaauto/> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
32. Switch-технология. Программирование задач логического управления. // ОБЕН – URL: <https://owen.ru/forum/showthread.php?t=11839> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
33. Технология ISaGRAF // ISaGRAF Rockwell Automation – URL: <https://isagraf.ru/tehnologiya-isagraf/isagraf-segodnya/isagraf-glavnaya/isagraf> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
34. Вячеслав Петухов, ООО «3В Сервис» «Использование технологии SimInTech для проектирования и создания АСУ ТП» // DOCPLAYER – URL: <https://docplayer.com/32425533-Vyacheslav-petuhov-ooo-3v-servis-ispolzovanie-tehnologii-simintech-dlya-proektirovaniya-i-sozdaniya-asu-tp.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
35. Массовый уход иностранных компаний: проблемы и решения по импортозамещению в ИТ // CLOUD4Y – URL: <https://www.cloud4y.ru/blog/business-out-problems-and-solutions> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
36. Microsoft прекратила продажи товаров и услуг в России // ВЕДОМОСТИ – URL: <https://www.vedomosti.ru/business/news/2022/03/04/912205-microsoft-prekratila-prodazhi-v-rossii> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
37. Siemens остановила поставки в Россию // NR NEW RETAIL – URL: <https://www.vedomosti.ru/business/news/2022/03/04/912205-microsoft-prekratila-prodazhi-v-rossii> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
38. В Минцифре рассказали об обновленных данных цифровой блокады россии. Ушли Salesforce, Hexagon, FANUC и другие // borg.expert –

URL: <https://borgexpert.com/ru/news/v-myntsyfre-rasskazaly-ob-obnovlennykh-dannykh-tsyfrovoj-blokady-rossyy-ushly-salesforce-hexagon-fanuc-y-druhye> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

39. Beremiz // altlinux – URL: <https://www.altlinux.org/Beremiz> (Режим – Режим доступа: открытый. – Дата обращения: 12.05.2022.

40. System Requirements for MATLAB R2022a // MathWorks – URL: <https://www.mathworks.com/support/requirements/matlab-system-requirements.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

41. Visio Professional 2019 // MSSOFT.RU – URL: https://www.mssoft.ru/Makers/Microsoft/Visio_Professional_2019/SysReq – Режим доступа: открытый. – Дата обращения: 12.05.2022.

42. Среда динамического моделирования технических систем SimInTech // DOCPLAYER – URL: <https://docplayer.com/57122181-Sreda-dinamicheskogo-modelirovaniya-tehnicheskikh-sistem-simintech.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

43. SIMATIC STEP 7 Professional/Basic V16 для Totally Integrated Automation Portal – URL: <https://assets.new.siemens.com/siemens/assets/api/uuid:2223b6f4-6ba1-4074-b395-9d3ec5711efa/03-TIA-STEP7-V16.pdf> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

44. CODESYS V3 // ОБЕИ – URL: https://owen.ru/product/codesys_v3 – Режим доступа: открытый. – Дата обращения: 12.05.2022.

45. Release Notes - ISaGRAF 6.5 ACP – URL: <https://shop.fiord.com/phocadownload/software/ISaGRAF%206.5%20ACP%20Release%20Notes-rus.pdf> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

46. Контроллер программируемый логический МКLogic200. Программирование в среде Beremiz МКLogic200. Руководство пользователя // Нефтеавтоматика – URL: <https://www.nefteavtomatika.ru/upload/iblock/24f/24fef07865d566f863df7dbbb5248aa4.pdf> – Режим доступа: открытый. – Дата обращения: 12.05.2022.

47. DRAKON Editor 1.9 // Best-Soft.ru – URL: <http://best-soft.ru/programs/8658.html> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
48. Pricing and Licensing // MathWorks – URL: <https://www.mathworks.com/pricing-licensing.html?prodcode=ME&intendeduse=comm>. – Режим доступа: открытый. – Дата обращения: 12.05.2022.
49. SIMATIC STEP 7 Prof. V16. ПО разработки в рамках T // ЭТМ – URL: <https://www.etm.ru/cat/nn/8026550> – Режим доступа: открытый. – Дата обращения: 12.05.2022.
50. ГОСТ 12.0.002-2014 ССБТ. Термины и определения.
51. Международный стандарт IC CSR-08260008000 «Социальная ответственность организации. Требования» М.: ВОК, – 2011, – 36 с.
52. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 05.02.2018) // Собрание законодательства РФ. - 07.01.2002.
53. ГОСТ 12.2.033-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ стоя. Общие эргономические требования. – URL: <http://docs.cntd.ru/document/1200005187> – Режим доступа: открытый. – Дата обращения: 27.04.2022.
54. ГОСТ 12.2.033-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ стоя. Общие эргономические требования. – URL: <http://docs.cntd.ru/document/1200005187> – Режим доступа: открытый. – Дата обращения: 27.04.2022.
55. СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания. – М.: ИПК Изд-во стандартов, 2021. – 496 с.
56. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*. – М.: ИПК Изд-во стандартов, 2017. – 122 с.
57. ГОСТ 12.1.019-2017 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты (с Поправкой). – М.: ИПК Изд-во

стандартов, 2017. – 20 с.

58. ПРАВИЛА УСТРОЙСТВА ЭЛЕКТРОУСТАНОВОК Седьмое издание Раздел 1 ОБЩИЕ ПРАВИЛА Глава 1.1 ОБЩАЯ ЧАСТЬ.

59. ПРИКАЗ от 15 декабря 2020 года N 903н Об утверждении Правил по охране труда при эксплуатации электроустановок.

60. МР 2.2.9.2311-07. Состояние здоровья работающих в связи с состоянием производственной среды. Профилактика стрессового состояния работников при различных видах профессиональной деятельности: Методические рекомендации. – М.: Федеральный центр гигиены и эпидемиологии Роспотребнадзора, 2008. – 52 с.

61. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*. – М.: ИПК Изд-во стандартов, 2017. – 122 с.

62. Утилизация и переработка твёрдых бытовых отходов : учебное пособие / А. С. Клинков, П. С. Беляев, В. Г. Однолько, М. В. Соколов, П. В. Макеев, И. В. Шашков. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 100 экз. – 188 с.

63. ГОСТ Р 58577–2019 Правила установления нормативов допустимых выбросов загрязняющих веществ проектируемыми и действующими хозяйствующими субъектами и методы определения этих нормативов – URL: <https://docs.cntd.ru/document/1200168569> – Режим доступа: открытый. – Дата обращения: 27.04.2022.

64. Постановление Правительства РФ от 31 декабря 2020 г. N 2398 «Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III и IV категорий».

65. СП 12.13130.2009 ОПРЕДЕЛЕНИЕ КАТЕГОРИЙ ПОМЕЩЕНИЙ, ЗДАНИЙ И НАРУЖНЫХ УСТАНОВОК ПО ВЗРЫВОПОЖАРНОЙ И ПОЖАРНОЙ ОПАСНОСТИ.

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг А.1 – Сгенерированный код на языке ST

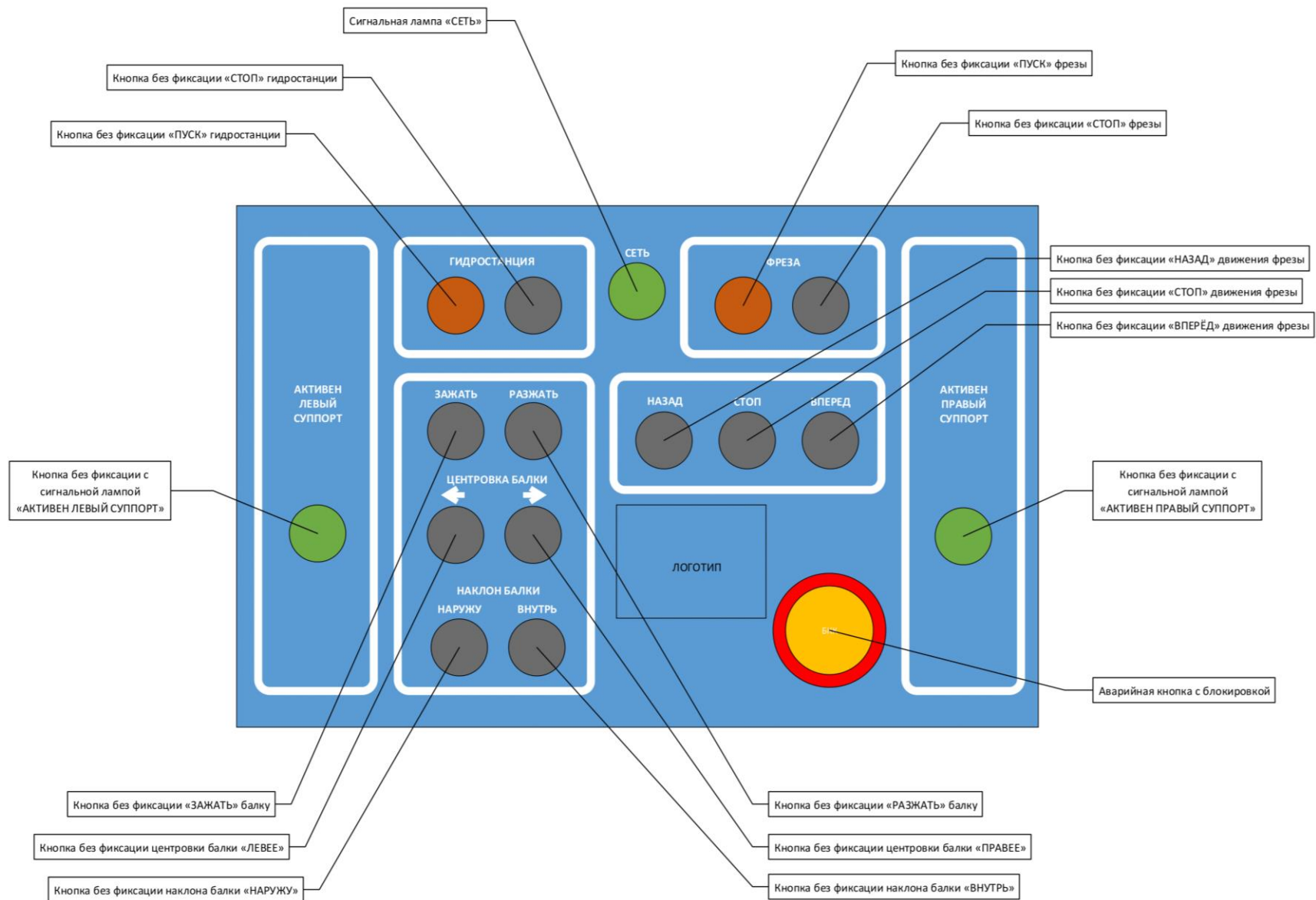
```
1 function state_machine: udint; //имя функции и тип данных выхода);
2 var //объявление локальных переменных
3 STATE1: udint := 1;
4 INITIAL: udint := 1
5 HEATER_ON: udint := 2
6 HEATER_OFF: udint := 3
7 EMERGENCY: udint := 4
8
9 SET_POINT: BOOL
10 HEATER: BOOL
11 end_var
12
13 var_input
14 CURRENT_TEMP: BOOL;
15 HEATER_BROKEN : BOOL;
16 end_var
17
18 CASE STATE1 OF
19     1: // INITIAL
20         SET_POINT := INPUT();
21         IF CURRENT_TEMP<SET_POINT
22             THEN
23                 STATE := 2; // HEATER_ON
24             END_IF;
25
26     2: // HEATER_ON
27         HEATER := TRUE;
28         IF CURRENT_TEMP>SET_POINT
29             THEN
30                 STATE := 3; // HEATER_OFF
31             END_IF;
```

Продолжение Листинга А.1

```
32 IF HEATER_BROKEN
33 THEN
34     STATE := 4; // EMERGENCY
35 END_IF;
36 3: // HEATER_OFF
37     HEATER := FALSE;
38 IF CURRENT_TEMP < SET_POINT
39 THEN
40     STATE := 1; // INITIAL
41 END_IF;
42
43 4: // EMERGENCY
44     HEATER := FALSE;
45 END_CASE
46
47 state_machine := shl(bool_to_udint(SET_POINT), 1)
48                 + shl(bool_to_udint(HEATER), 2);
49 end_function
```

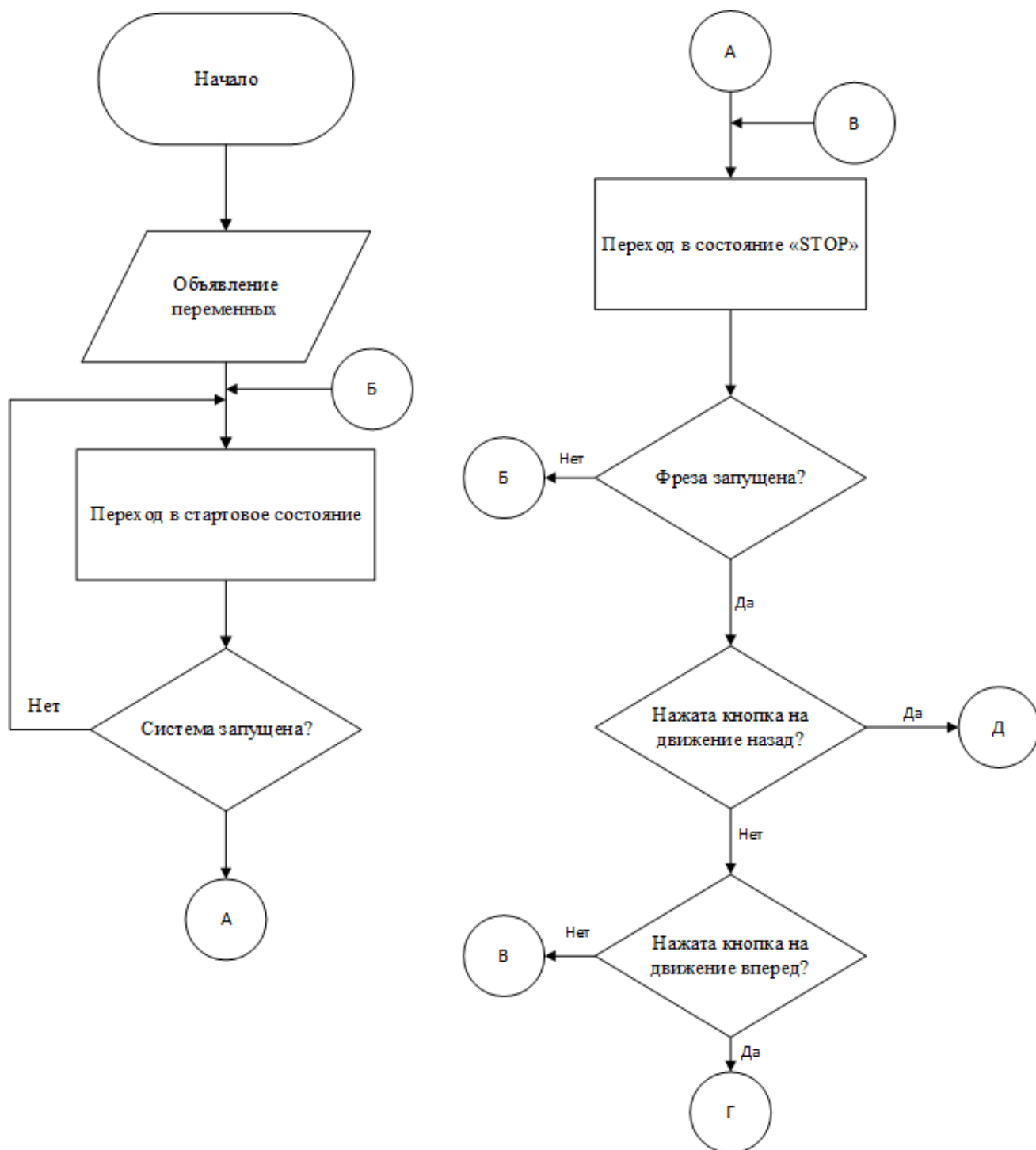
ПРИЛОЖЕНИЕ Б

Пульт управления фрезером наклонных поверхностей



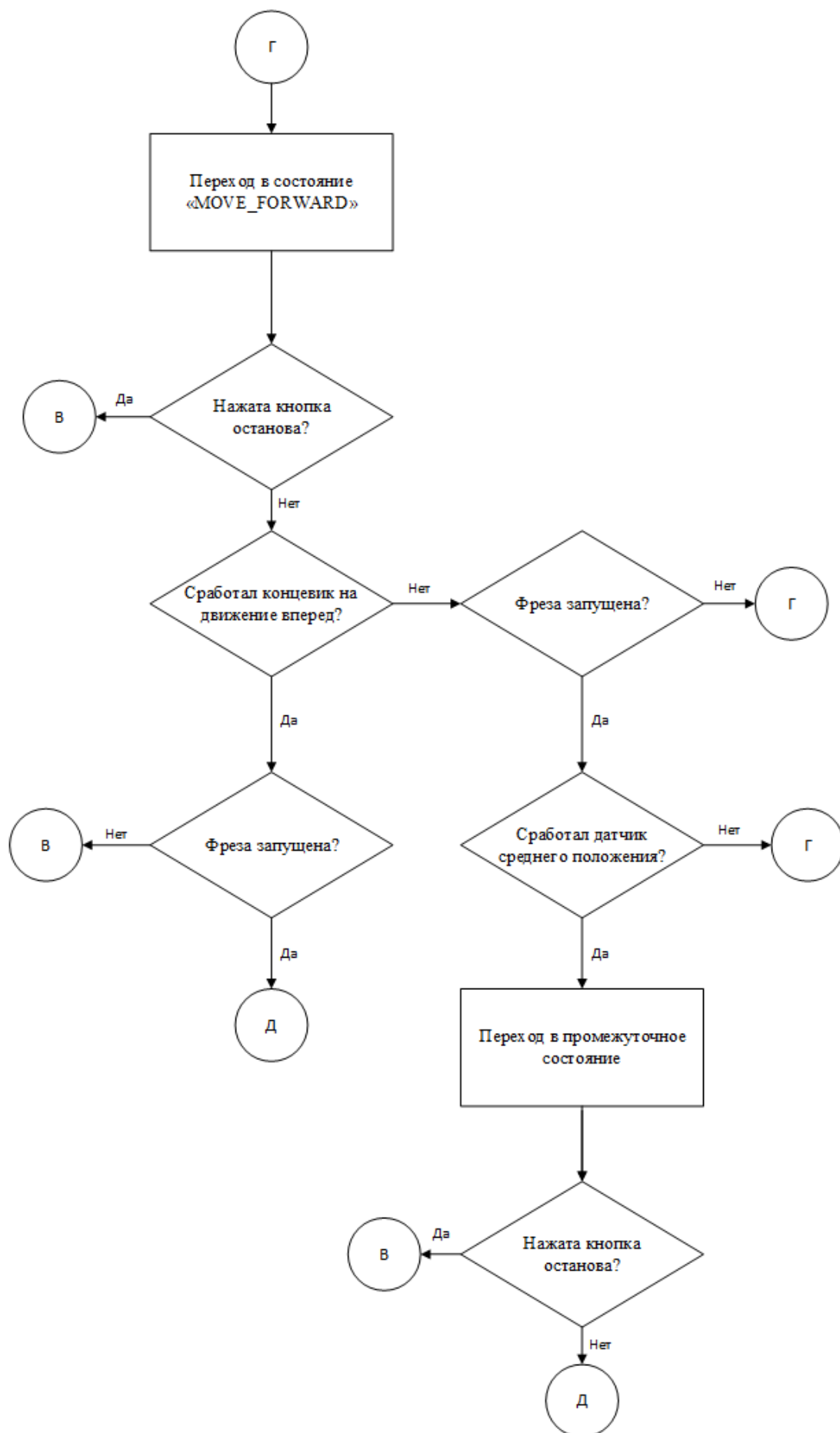
ПРИЛОЖЕНИЕ В

Граф-схема алгоритма управления станком для автомата №1 (часть 1)



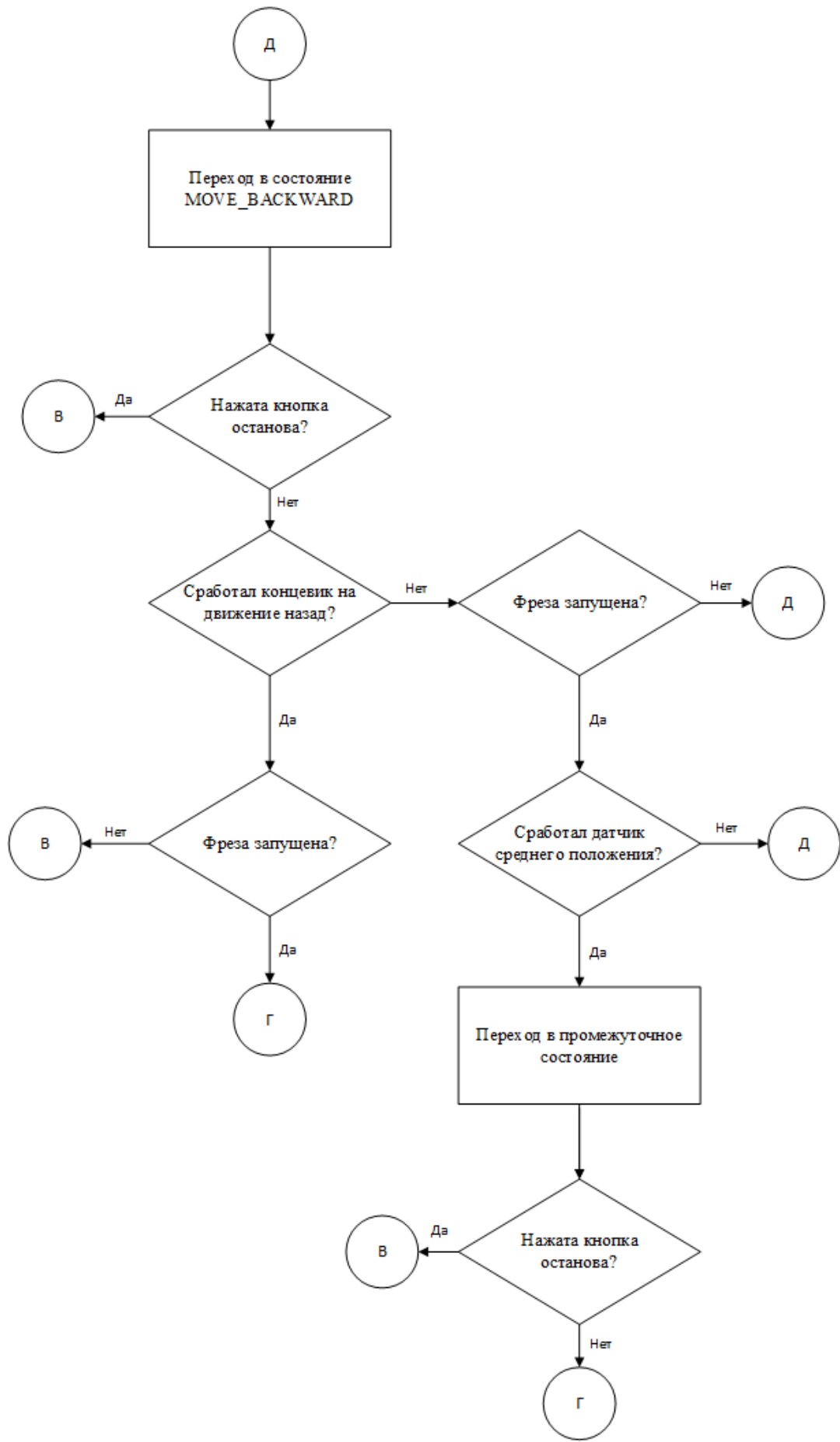
ПРИЛОЖЕНИЕ Г

Граф-схема алгоритма управления станком для автомата №1 (часть 2)



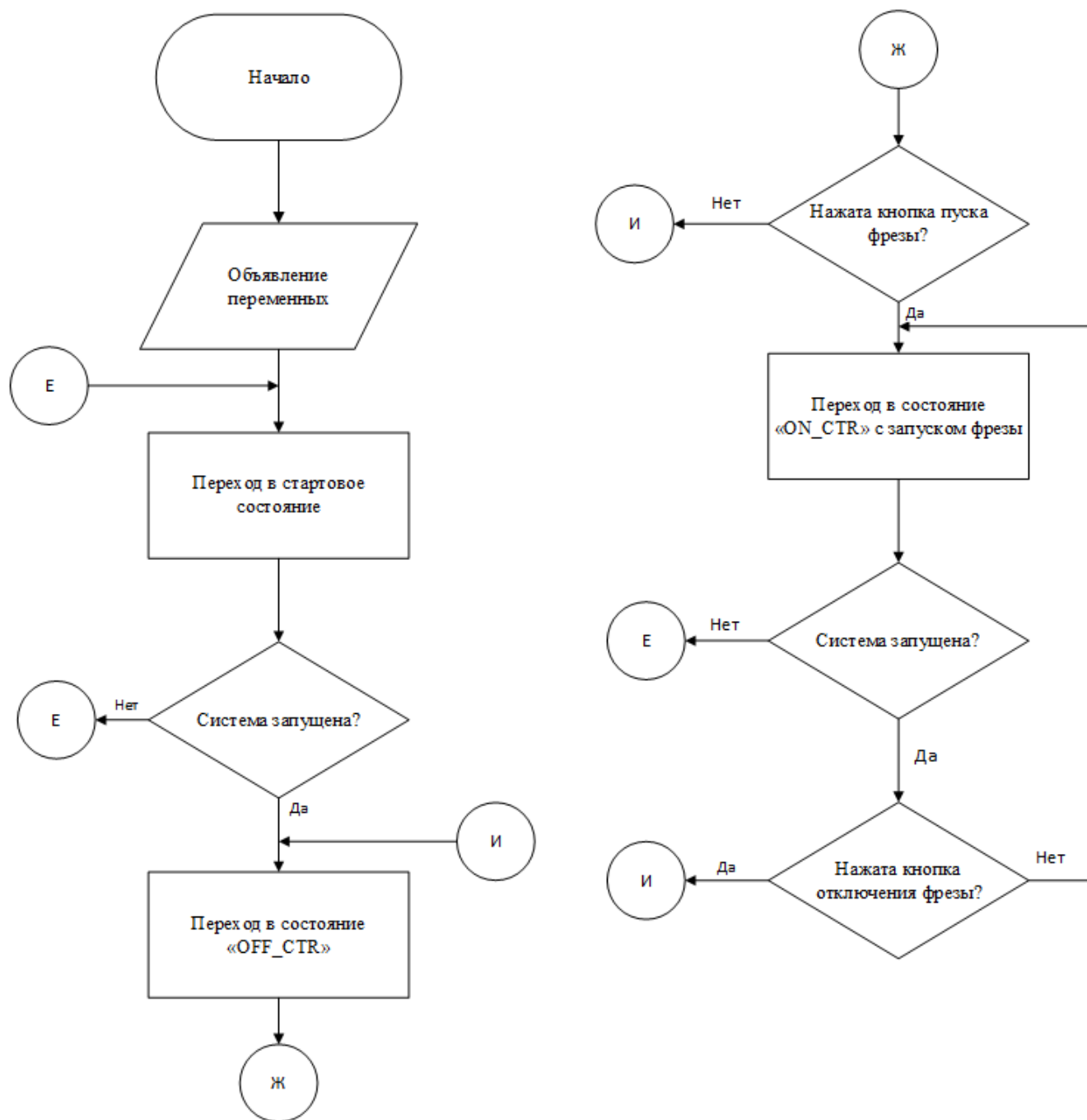
ПРИЛОЖЕНИЕ Д

Граф-схема алгоритма управления станком для автомата №1 (часть 3)



ПРИЛОЖЕНИЕ Е

Граф-схема алгоритма управления станком для автомата №2



ПРИЛОЖЕНИЕ Ж

(обязательное)

Листинг Ж.1 – Сгенерированный код на языке ST для станка-фрезера

```
1 function state_machine: udint; //имя функции и тип данных выхода;
2 var //объявление локальных переменных
3 STATE1: udint := 0;
4 STATE2: udint := 0;
5 IDLE1: udint := 0;
6 STOP: udint := 1;
7 MOVE_FORWARD: udint := 2;
8 MOVE_BACKWARD: udint := 3;
9 INTERMEDIATE_F: udint := 4;
10 INTERMEDIATE_B: udint := 5;
11
12 IDLE2: udint := 0;
13 ON_CTR: udint := 1;
14 OFF_CTR: udint := 2;
15
16 CMDFORWARD: BOOL;
17 CMDBACKWARD: BOOL;
18 CUTTER_ON: BOOL;
19 qqffxxx: udint;
20 qqff: udint;
21 x000qqq: udint;
22 ff: udint;
23 qq: udint;
24
25 qqsb: udint;
26 qqffsb: udint;
27 qqffxxxsb: udint;
28 qqffxxxsb: udint;
29 qqffxs: udint;
30
31 end_var
```

Продолжение Листинга Ж.1

```
32 var_input
33 ENABLE: BOOL;
34 BTN_CUTTER_ON: BOOL;
35 BTN_CUTTER_OFF: BOOL;
36 TERM_MIDDLE: BOOL;
37 BTN_STOP: BOOL;
38 TERM_FORWARD: BOOL;
39 BTN_FORWARD: BOOL;
40 TERM_BACKWARD: BOOL;
41 BTN_BACKWARD: BOOL;
42 STATE_FB: uint;
43 end_var
44
45 qqfff:=shr(STATE_FB, 3);
46 qqf := shr(qqfff,2);
47 ff := qqfff-shl(qqf,2);
48 STATE1 := qqf;
49 STATE2 := ff;
50
51 CASE STATE1 OF
52   0: // IDLE1
53     CMDFORWARD := FALSE;
54     CMDBACKWARD := FALSE;
55
56   IF ENABLE
57   THEN
58     STATE1 := 1; // STOP
59   END_IF;
60
61   1: // STOP
62     CMDFORWARD := FALSE;
63     CMDBACKWARD := FALSE;
64
65   IF BTN_BACKWARD
66   THEN
```

Продолжение Листинга Ж.1

```
67     STATE1 := 3; // MOVE_BACKWARD
68     END_IF;
69
70     IF BTN_FORWARD
71     THEN
72         STATE1 := 2; // MOVE_FORWARD
73     END_IF;
74
75     IF (NOT ENABLE)
76     THEN
77         STATE1 := 0; // IDLE1
78     END_IF;
79
80     2: // MOVE_FORWARD
81         CMDFORWARD := TRUE;
82         CMDBACKWARD := FALSE;
83
84     IF (CUTTER_ON) AND (TERM_FORWARD)
85     THEN
86         STATE1 := 3; // MOVE_BACKWARD
87     END_IF;
88
89     IF (CUTTER_ON) AND (TERM_MIDDLE)
90     THEN
91         STATE1 := 4; // INTERMEDIATE_F
92     END_IF;
93
94     IF (NOT CUTTER_ON) AND (TERM_FORWARD)
95     THEN
96         STATE1 := 1; // STOP
97     END_IF;
98
99     IF BTN_STOP
100    THEN
```

Продолжение Листинга Ж.1

```
101     STATE1 := 1; // STOP
102 END_IF;
103
104 IF (NOT ENABLE)
105 THEN
106     STATE1 := 0; // IDLE1
107 END_IF;
108
109 3: // MOVE_BACKWARD
110     CMDFORWARD := FALSE;
111     CMDBACKWARD := TRUE;
112
113 IF (CUTTER_ON) AND (TERM_BACKWARD)
114 THEN
115     STATE1 := 2; // MOVE_FORWARD
116 END_IF;
117
118 IF (CUTTER_ON) AND (TERM_MIDDLE)
119 THEN
120     STATE1 := 5; // INTERMEDIATE_B
121 END_IF;
122
123 IF (NOT CUTTER_ON) AND (TERM_BACKWARD)
124 THEN
125     STATE1 := 1; // STOP
126 END_IF;
127
128 IF BTN_STOP
129 THEN
130     STATE1 := 1; // STOP
131 END_IF;
132
133 IF (NOT ENABLE)
134 THEN
135     STATE1 := 0; // IDLE1
```

Продолжение Листинга Ж.1

```
136     END_IF;
137
138     4: // INTERMEDIATE_F
139         CMDFORWARD := FALSE;
140         CMDBACKWARD := FALSE;
141
142     IF BTN_STOP
143     THEN
144         STATE1 := 1; // STOP
145     END_IF;
146
147     IF TRUE
148     THEN
149         STATE1 := 3; // MOVE_BACKWARD
150     END_IF;
151
152     5: // INTERMEDIATE_B
153         CMDFORWARD := FALSE;
154         CMDBACKWARD := FALSE;
155
156     IF BTN_STOP
157     THEN
158         STATE1 := 1; // STOP
159     END_IF;
160
161     IF TRUE
162     THEN
163         STATE1 := 2; // MOVE_FORWARD
164     END_IF;
165
166 END_CASE;
167
168 CASE STATE2 OF
169     0: // IDLE2
```

Продолжение Листинга Ж.1

```
170     CUTTER_ON := FALSE;
171
172     IF ENABLE
173     THEN
174         STATE2 := 2; // OFF_CTR
175     END_IF;
176
177     1: // ON_CTR
178         CUTTER_ON := TRUE;
179
180     IF BTN_CUTTER_OFF
181     THEN
182         STATE2 := 2; // OFF_CTR
183     END_IF;
184
185     IF (NOT ENABLE)
186     THEN
187         STATE2 := 0; // IDLE2
188     END_IF;
189
190     2: // OFF_CTR
191         CUTTER_ON := FALSE;
192
193     IF BTN_CUTTER_ON
194     THEN
195         STATE2 := 1; // ON_CTR
196     END_IF;
197 END_CASE;
198
199 qqqsbor := shl(qqq,2);
200 qqffsbor := qqqsbor + ff;
201 qqffxsbor := shl(qqqffsbor,1) + bool_to_udint(CMDforward);
202 qqffxxsbor:= shl(qqqffxsbor,1) + bool_to_udint(CMDbackward);
203 qqffxxxsbor:= shl(qqqffxxsbor,1) + bool_to_udint(CUTTER_ON);
```

Продолжение Листинга Ж.1

```
204 state_machine := qqffxxsbor;  
205 end_function
```


ПРИЛОЖЕНИЕ И

Регистрация программы для ЭВМ (продукт FSMLogGen)

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022611279

FSMLogGen

Правообладатель: *Гительман Владислав Сергеевич (RU)*

Автор(ы): *Гительман Владислав Сергеевич (RU)*



Заявка № 2022610545

Дата поступления 19 января 2022 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 24 января 2022 г.

*Руководитель Федеральной службы
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 0x75699C0009AE878543BD83DF819A6CD1
Владелец **Ивлиев Григорий Петрович**
Действителен с 24.12.2021 по 24.12.2022

Г.П. Ивлиев

ПРИЛОЖЕНИЕ К

Таблица конкурентных решений

Таблица К.1 – Сравнение FSMLogGen с аналогичными продуктами

	Simulink PLC Coder + StateFlow (от MATLAB)	Visio2ST	FSMLogGen	Simintech	Simatic Step 7	Codesys	IsaGraf	Beremiz	DRAKON
Языки генерации кода	ST, но используя другие «Coder» C, C++, Verilog	ST	ST, C++, Assembler	ST, C	-	-	-	-	C, C++, Java и др. ST отсутствует
Языки ручного написания программы	-	-	-	-	GRAPH 7, HiGraph 7, LAD, FBD, STL, SCL, SFC	ST, LD, FBD, SFC	IL, ST, LD, FBD, SFC	IL, ST, LD, FBD, SFC	-
Удобство интерфейса	Непростой интерфейс, необходимо тщательное изучение при использовании ПО	Интерфейс Visio не сложный, но для генерации необходимо осуществить ряд действий, привлечь доп. исполняемые файлы и утилиты	Простой интуитивно-понятный интерфейс, ПО содержит инструмент построения графов-переходов и их конвертер на языки программирования	Средний по сложности интерфейс, необходимо его изучение при создании программ для АСУ ТП	Интерфейс со своей спецификой, требует тщательного изучения разработчикам и программ для АСУ ТП	Интерфейс со своей спецификой, требует тщательного изучения разработчикам программ для АСУ ТП	Интерфейс со своей спецификой, требует тщательного изучения разработчикам и программ для АСУ ТП	Интерфейс со своей спецификой, требует тщательного изучения разработчикам и программ для АСУ ТП	Довольно простой и наглядный интерфейс

Продолжение таблицы К.1

	Simulink PLC Coder + StateFlow (от MATLAB)	Visio2ST	FSMLogGen	Simintech	Simatic Step 7	Codesys	IsaGraf	Beremiz	DRAKON
Графическое представление	Графы-переходы	-	Графы-переходы, таблицы	Графы-переходы	В HiGraph – графы-переходы, в остальных языках своя специфика	Графов-переходов нет, в остальных языках своя специфика	Графов-переходов нет, в остальных языках своя специфика	Графов-переходов нет, в остальных языках своя специфика	Графов-переходов нет, представление в виде граф-схемы алгоритма
Возможности отладки	+	-	Отладка через IDE RAD Studio	+	+	+	+	+	+
Применение	В промышл. автоматизации при построении программ и алгоритмов для ПЛК. Есть доп. возможности при использовании других генераторов получать код для микроконтроллеров, цифровых сигнальных процессоров, программируемых логических интегральных схем, интегральных схем специального назначения, программируемых контроллеров автоматизации	В промышл. автоматизации при построении программ и алгоритмов для ПЛК	В промышл. автоматизации при построении программ и алгоритмов для ПЛК	В промышл. автоматизации при построении программ и алгоритмов для ПЛК	В промышл. автоматизации при построении программ для ПЛК	В промышл. автоматизации при построении программ для ПЛК	В промышл. автоматизации при построении программ для ПЛК	В промышл. автоматизации при построении программ для ПЛК	Основное применение при создании программ и алгоритмов в ракетно-космической отрасли. Но охват расширился и на сферу медицины, образования. Возможно применение при разработке программ для ПЛК

Продолжение таблицы К.1

	Simulink PLC Coder + StateFlow (от MATLAB)	Visio2ST	FSMLogGen	Simintech	Simatic Step 7	Codesys	IsaGraf	Beremiz	DRAKON
Аппаратные ограничения	С помощью PLC Coder генерируются как платформонезависимый и платформозависимый код (код под контроллеры конкретного производителя Siemens, Rockwell Automation и др.)	Платформонезависимый код	Платформонезависимый код	Платформозависимый код на ST (для ПО Schneider Unity, на котором программируются контроллеры ПЛК Modicon M340, Premium и Quantum, а также Atrium) При необходимости можно изменить шаблон генерации [34] и для других компиляторов и ПО	Платформозависимый код для программируемых логических контроллеров Simatic S7-300/S7-400/M7/C7 и WinAC	Платформозависимый	Платформозависимый	Платформозависимый	Платформонезависимый
Соответствие генерируемого кода стандарту программирования контроллеров	Соответствует, но есть доп. возможность для генерации платформонезависимого (частично несоответствующего стандарту) кода	Соответствует	Соответствует	Соответствует. Есть доп. возможности получения платформозависимого кода (частично несоответствующего стандарту) при изменении шаблона для генерации	Частично соответствует	Соответствует	Соответствует	Соответствует	Не соответствует

Продолжение таблицы К.1

	Simulink PLC Coder + StateFlow (от MATLAB)	Visio2ST	FSMLogGen	Simintech	Simatic Step 7	Codesys	IsaGraf	Beremiz	DRAKON
Подверженность продукта возможности отзыва лицензий/блокировки использования в России и иные санкции	MATLAB отозвал лицензии у российских студентов и учебных заведений [35]	Отсутствует (свободное в распространении ПО). Но требуется Microsoft Visio. Компанией Microsoft были введены ограничения на продажу своих программных продуктов в России, в том числе, Visio [36]	Отсутствует	Отсутствует (отечественное ПО)	Компания Siemens уходит с рынка России, прекращает поставки оборудования и сотрудничество в рамках новых проектов [37]	Отсутствует (отечественное ПО)	Компания-производитель ПО и оборудования для промышленной автоматизации «Rockwell Automation» (США) остановила продажи в России. Приостановлена её деятельность в России, компания отказалась от продажи своего ПО на территории России [38]	Beremiz является средой разработки для ПЛК с открытым исходным кодом, при этом данное ПО бесплатное. Поэтому продукт не подвержен санкциям и ограничениям, может использоваться в России без опасений [39]	Отсутствует (отечественное ПО)

Продолжение таблицы К.1

	Simulink PLC Coder + StateFlow (от MATLAB)	Visio2ST	FSMLogGen	Simintech	Simatic Step 7	Codesys	IsaGraf	Beremiz	DRAKON
Системные требования к техническим средствам	Последняя версия MATLAB R2022a требует Windows 10 (версия 1909 или выше), Windows 11. ОЗУ – 4 Гб, но рекомендуется 8 Гб. Занимает на диске 3,6 Гб на диске [40]. Установка пакета Simulink и StateFlow потребует ещё около 9 Гб места на диске	Требуется Visio от Microsoft. Для поддержки данного ПО необходимо 4 Гб ОЗУ, 4 Гб свободного места на диске [41] и Windows 10. Конвертер Visio2ST требует места на диске в размере 42,06 Мб	Для использования ПО требуется 1,957 Мб свободного пространства на диске, 1 Гб ОЗУ. Операционная система от Windows XP Home до Windows 11	В настоящее время системные требования программы не находятся в открытом доступе. На 2016 год системные требования включали следующее: 1,6 Гб свободного места на жестком диске, 2 Гб ОЗУ, MS Windows Vista, и MS Windows 7-10 (32 бит) [42]	Step 7-V16 включает следующие рекомендуемые требования к аппаратуре: 16 Гб ОЗУ, 50 Гб свободного пространства на диске, Windows 7 Professional, Windows 10 Professional, а также требуется установка Windows Server 2012 R2StdE И Windows Server 2016 Standard [43]	Windows 7/8/10 (32/64 бит), 4 Гб ОЗУ, 2 Гб ПЗУ [44]	Для ОС на базе x86 размер памяти не менее 1 Гб, для ОС на базе x64 не менее 2 Гб. Для виртуальной машины нужна дополнительная память 512 Мб. Размер дискового пространства 4 Гб. Windows 7-10 (x86 и x64). [45]	ОЗУ 1 Гб, свободное место на жестком диске 1 Гб, Windows XP/Vista/7/8/10 или Linux [46]	Минимальные системные требования: 5 Мб свободного места на диске, 1 Гб ОЗУ, Windows 7, Windows XP, Windows Vista, Windows 9x [47]

Продолжение таблицы К.1

	Simulink PLC Coder + StateFlow (от MATLAB)	Visio2ST	FSMLogGen	Simintech	Simatic Step 7	Codesys	IsaGraf	Beremiz	DRAKON
Стоимость продукции	Сумма лицензии MATLAB и пакетов Simulink, Stateflow, и PLC Coder составляет 14250 долларов США (947625 руб.). С учетом стоимости остальных конвертеров, приблизительная стоимость составляет 30000 долларов (1995000 рублей) за лицензию [48]	Бесплатное ПО	50 – 100 тысяч рублей в зависимости от возможностей ПО и лицензии. Также есть бесплатная пробная версия	Приблизительная стоимость 22000 долларов или 1463000 рублей.	Одиночная лицензия Simatic Step 7-V16 составляет 72080.35 рублей [49]	Бесплатное программное обеспечение	В открытом доступе стоимость отсутствует. Коммерческое ПО	Бесплатное ПО	Бесплатное ПО

ПРИЛОЖЕНИЕ Л

Модель по А. Остервальдеру и И. Пинье

<p>Ключевые партнеры</p> <ul style="list-style-type: none"> Отдел АСУ ТП «ТомскНИПИнефть» Компания «ЭлеСи» Компания «СТК» ООО "Автоматизация Производств" ООО "ТомскАСУпроект" Завод "Латат" / ООО "Монолит-Строй" ООО «ПТ-СЕРВИС» др. компании, занимающиеся разработкой АСУ ТП и их поставкой на рынок автоматизации др. компании, имеющие отделы АСУ ТП и занимающиеся разработкой ПО для собственного производства самостоятельно 	<p>Ключевые виды деятельности</p> <p>Продажа: Программное обеспечение для АСУ ТП</p> <p>Решение проблем:</p> <ul style="list-style-type: none"> упрощение разработки алгоритмов и программ для АСУ ТП; уменьшение времени на согласование ТЗ технологом и программистом; Сокращение трудозатрат и времени программистов при разработке ПО АСУ ТП; Частичное решение проблемы нехватки IT-специалистов на рынке труда 	<p>Ценностные предложения</p> <ul style="list-style-type: none"> Стоимость в 15-20 раз ниже, чем у SimInTech и MATLAB; Возможность выбора языка генерации кода; Низкие системные требования к аппаратным средствам; Простой и интуитивно-понятный интерфейс; Поддержка языка генерации кода ST для промышленных контроллеров стандарта МЭК 61131-3-2016; Отсутствие рисков потери доступа к ПО (отечественная разработка) 	<p>Взаимоотношения с клиентами</p> <ul style="list-style-type: none"> Электронная почта для оформления заказа на лицензию; Помощь при выборе версии ПО, индивидуальная консультация; Участие в конференциях; Освещение вопросов применения разрабатываемого ПО в научных статьях, демонстрация реальных примеров использования в проектах автоматизации; Предоставление исчерпывающей информации о программе на сайте. 	<p>Потребительские сегменты</p> <ul style="list-style-type: none"> Предприятия-поставщики решений в области промышленной автоматизации; Предприятия, использующие автоматизацию в производстве и разрабатывающие системы АСУ самостоятельно.
<p>Структура издержек</p> <ul style="list-style-type: none"> Фиксированные издержки – аренда помещения, реклама, фиксированные взносы в ФНС и ПФР, содержание сайта, заработная плата, амортизация, регистрация ПО. Переменные издержки – повременная заработная плата работнику, закупка оборудования для тестирования ПО, энергия. 		<p>Потоки поступления доходов</p> <ul style="list-style-type: none"> Продажи программного обеспечения Доход с обновления программного обеспечения (1% от изначальной стоимости ПО) Масштабирование продукта (докупка версии с более широкими возможностями) 		