

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

Р.С. Малсугенов
Р.А. Байрамуков
А.О. Байрамуков

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕХАТРОННЫХ УСТРОЙСТВ И РОБОТОВ

Методические рекомендации по выполнению
курсового проекта для обучающихся всех форм обучения,
направления подготовки 15.03.06 Мехатроника и робототехника,
направленность (профиль) «Мехатронные и роботизированные
технологические системы и комплексы»

Черкесск
2025

УДК 621
ББК 34.4
М18

Рецензент: Эльканова Л.М. – к.ф-м. н., доцент

М18 Малсугенов Р.С. Программное обеспечение мехатронных устройств и роботов: методические рекомендации по выполнению курсового проекта для обучающихся всех форм обучения, направления подготовки 15.03.06 Мехатроника и робототехника, направленность (профиль) «Мехатронные и роботизированные технологические системы и комплексы» / Р.С. Малсугенов, Р.А. Байрамуков, А.О. Байрамуков – Черкесск: БИЦ СевКавГА, 2025. – 29 с.

Пособие разработано согласно учебному плану образовательной программы бакалавров по направлению подготовки 15.03.06. Мехатроника и робототехника. Оно предназначено для обеспечения выполнения курсовых проектов по курсу «Программное обеспечение мехатронных устройств и роботов».

**УДК 621
ББК 34.4**

© Малсугенов Р.С. Байрамуков Р.А., Байрамуков А.О., 2025
© ФГБОУ ВО СевКавГА, 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. ХАРАКТЕР ЗАДАНИЯ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ	5
2. СТРУКТУРА И ОБЪЕМ КУРСОВОГО ПРОЕКТА	6
2.1. Объем курсового проекта	6
2.1.1. Пояснительная записка	6
2.1.2. Графическая часть	8
2.1.3. Оформление пояснительной записки	9
2.1.4. Выполнение графической части работы	10
2.1.5. Защита курсовых проектов	11
3. СОДЕРЖАНИЕ ОТДЕЛЬНЫХ ЧАСТЕЙ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ	12
3.1. Вводная часть	12
3.2. Проектирование архитектуры программного обеспечения	13
3.3. Разработка алгоритмов управления	14
3.4. Реализация программного обеспечения	15
3.5. Тестирование и отладка программного обеспечения	16
4. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА	18
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	22
Приложение 1	25
Приложение 2	28

ВВЕДЕНИЕ

Курсовой проект по дисциплине «Программное обеспечение мехатронных устройств и роботов» направлен на закрепление теоретических знаний и формирование практических навыков разработки программного обеспечения для управления мехатронными и робототехническими системами. В процессе выполнения проекта студент осваивает:

- методы алгоритмизации и программирования для систем реального времени;
- принципы проектирования архитектуры программного обеспечения мехатронных систем;
- технологии разработки, отладки и тестирования ПО;
- инструменты компьютерного моделирования и симуляции робототехнических устройств.

Курсовой проект является важным этапом подготовки к выполнению выпускной квалификационной работы.

Цель выполнения проекта: разработка программного обеспечения для управления конкретным мехатронным или робототехническим устройством (системой) с использованием современных языков программирования, инструментов разработки и методологий.

Задачи выполнения проекта:

- анализ предметной области и существующих решений;
- проектирование архитектуры программного обеспечения;
- разработка алгоритмов управления;
- реализация программного кода;
- тестирование и отладка ПО;
- моделирование и верификация работы системы.

1. ХАРАКТЕР ЗАДАНИЯ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Задание на курсовой проект по дисциплине «Программное обеспечение мехатронных устройств и роботов» представляет собой техническое задание на разработку программного обеспечения для конкретной мехатронной или робототехнической системы. Задание формулируется индивидуально для каждого студента и согласовывается с научным руководителем.

В задании на курсовой проект указываются:

- объект проектирования;
- технические требования к системе;
- требования к программному обеспечению;
- перечень представляемых материалов;
- этапы выполнения и сроки.

Образец задания на курсовой проект приведен в Приложении 2.

Примерная тематика курсовых проектов

1. Разработка ПО управления однокоординатным мехатронным приводом.
2. Программная реализация системы управления двухосевым манипулятором.
3. Разработка алгоритмов и ПО для мобильного робота с дифференциальным приводом.
4. Программное обеспечение управления сервоприводом в мехатронной системе.
5. Разработка системы обработки сигналов датчиков в робототехническом комплексе.
6. Программная реализация системы управления шаговым электроприводом.
7. Разработка ПО для программируемого логического контроллера (ПЛК) мехатронной ячейки.
8. Программное обеспечение компьютерного моделирования динамики мехатронного привода.
9. Разработка многопоточной системы управления робототехническим комплексом.
10. Программная реализация интерфейса оператора для системы управления роботом.

2. СТРУКТУРА И ОБЪЕМ КУРСОВОГО ПРОЕКТА

2.1. Объем курсового проекта

Курсовой проект (далее по тексту КП) содержит пояснительную записку и графическую часть.

Объем курсового проекта должен составлять 30-35 страниц машинописного текста на листах формата А4 пятым шрифтом “GOST type A” либо “Times New Roman” с высотой букв и цифр в 14 пунктов и междустрочным 1,5-ым интервалом (без приложения).

2.1.1. Пояснительная записка

Записка должна включать следующие разделы:

- Титульный лист
- Задание КП (1 стр.);
- Оглавление (1-2 стр) ;
- Введение (1-2 стр);
- Анализ предметной области и существующих решений (4-5 стр);
- Проектирование архитектуры программного обеспечения (6-8 стр);
- Разработка алгоритмов управления (4-5 стр);
- Реализация программного обеспечения (6-8 стр);
- Тестирование и отладка программного обеспечения (3-4 стр);
- Заключение (1 стр.);
- Список использованных источников;
- Приложения.

ПРИМЕЧАНИЯ.

1. Объем отдельных разделов пояснительной записки, по согласованию с преподавателем, может быть несколько изменен.

2. Записка пишется на листах писчей бумаги нормальных размеров с одной стороны (без оборотной стороны) с соблюдением правил “Единой системы конструкторской документации”.

3. Записка должна сопровождаться заглавным (титульным) листом.

4. После заглавного листа должно быть дано оглавление для информирования расположений в ней достаточно обособленных частей с указанием наименования и номера страницы, с которой они начинаются.

Содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование), заключение, список использованных источников и приложений.

5. В конце записки должен быть дан список литературы.

Обращаем Ваше внимание, что в 2000г. были внесены изменения в ГОСТ 7.1. - 84 (библиографическое описание документа). Суть изменения состоит в том, что в заголовке приводится имя только одного автора, а в сведениях об ответственности (за косой чертой) обязательно приведение одного, двух или трех авторов.

Ниже мы приводим примеры библиографического описания документов.

Описание книги одного автора

Ефимова О. В. Финансовые анализы / О. В. Ефимова. - М : Бухгалтерский учет, 1999. -351 с.

Хачатуров К.А. Латиноамериканские уроки для России - Latinoamericanas para Rusia / К.А. Хачатуров ; Дипломат, акад. МИД Рос. Федерации. - М. : Междунар. отношения, 1999. - 394 с.

Эриашвчи Н.Д. Экологическое право : учебник для вузов / Н.Д. Эриашвчи. - М. : ЮНИТИ, 2000. -415 с.

Описание книги 2. 3-х авторов

В заголовке описания книги двух или трёх авторов приводят фамилию одного автора, как правило, первого из указанных на титульном листе:

Донцова Л.В. Анализы бухгалтерской отчетности / Л.В. Донцова, Н.А. Никифорова. - М. : Дело и Сервис, 1999. - 298 с.

2.1.2. Графическая часть

Графическая часть курсового проекта представляет собой визуализацию ключевых аспектов разработанного программного обеспечения и должна:

- Наглядно демонстрировать архитектуру и структуру ПО;
- Иллюстрировать алгоритмы и логику работы системы;
- Показывать интерфейсы и взаимодействие компонентов;
- Представлять результаты тестирования и моделирования;
- Обеспечивать понимание проекта без изучения полного текста пояснительной записки.

Результаты работы над КП должны быть оформлены в виде таблиц, схем, диаграмм, графиков, чертежей и представлены в виде иллюстративных материалов, выполненных на белой бумаге с использованием оргтехники (плоттеров), или (возможный вариант) в виде слайдов, демонстрируемых на большом экране с помощью мультимедийного проектора, что обеспечивает наилучшее качество представления информации. В обоих случаях исходная информация должна быть создана на компьютере.

Для разработки иллюстративных материалов, на которых будут помещены таблицы, диаграммы и графики следует использовать компьютерные программы Microsoft Word, Microsoft Excel или Microsoft PowerPoint, КОМПАС - 3D, AutoCAD, Solid Works и др. Все надписи в таблицах, на диаграммах и графиках должны быть выполнены шрифтом (тип, размер и толщина обводки), который хорошо читается на расстоянии. На графиках линии, отражающие зависимости, должны выполняться достаточно толстыми линиями, имеющими разный цвет или стиль в случае, если представлены несколько зависимостей. На диаграммах поля, отражающие величину различных составляющих, также должны выделяться либо разными цветами, либо разным стилем штриховки.

Рекомендуется уменьшенные до формата А4 и отпечатанные на принтере копии всех демонстрируемых плакатов (слайдов), в том числе и чертежей, подшить к ПЗ в разделе «Приложения».

Титульный лист оформляется в соответствии с требованиями ЕСКД. Все надписи на титульном листе располагаются в строго определенных местах. Образец титульного листа дан в приложении №1

Рекомендуется следующие номера шрифта для конкретных надписей:

«МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ» –

шрифт Times New Roman 14 пт;

«СЕВЕРОКАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ» - шрифт
Times New Roman 14 пт заглавным буквами;

«Инженерный институт» и «Кафедра «Мехатронные и робототехнические
системы» - шрифт Times New Roman 14 пт жирным шрифтом;

название темы курсового проекта – шрифт Times New Roman 14 пт
заглавным буквами;

«Выполнил ...», «Проверил ...» - Times New Roman 14пт;

год выполнения работы – шрифт Times New Roman 14пт.

2.1.3. Оформление пояснительной записки

Пояснительная записка ВКР является основным текстовым документом, в котором последовательно излагается материал, раскрывающий вопросы соответствующих разделов.

Общие требования к ПЗ - четкость построения, логичность изложения материала, убедительность аргументации, краткость и точность

формулировок и расчетов, конкретность изложения результатов работы, доказательность выводов и обоснованность рекомендаций.

Текст располагается на одной стороне листа писчей бумаги формата А4 (210x297мм) и с соблюдением ниже перечисленных требований. Ширина полей: сверху и снизу - 20 мм; слева - 30 мм; справа - 10 мм.

При её оформлении следует руководствоваться указаниями ГОСТ 7.32.-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления», ГОСТ 2.105-95 «Общие требования к текстовым документам». ГОСТ 7.1-2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления», ГОСТ 7.12-93 «Библиографическая запись. Сокращение слов на русском языке», ГОСТ 8.417-81 «Единицы физических величин».

В тексте РПЗ используется шрифт Times New Roman, 14 пт(черный).

Междустрочным интервал- полуторный (1,5 строки).

Выравнивание текста – по ширине.

Абзацный отступ одинаковый по всему тексту документа-15 мм.

Расстановка переносов – нет.

Нумерация страниц РПЗ должна быть сквозной, начиная с титульного листа и до последней страницы, проставляется в центре нижней части листа арабскими цифрами.

2.1.4. Выполнение графической части работы

При выполнении курсового проекта по дисциплине «Программное обеспечение мехатронных устройств и роботов» все графические материалы (схемы, диаграммы, чертежи, графики результатов моделирования) должны оформляться на стандартных листах согласно ГОСТ 2.301-68. Указанный стандарт устанавливает следующие основные форматы:

А0- 841x1189 мм; А3- 297 х 420 мм;

А1 -594 х 841 мм; А4 - 210 х 297 мм;

A2 -420 x 594 мм; A5 - 149 x210 мм.

A3 — 297 × 420 мм

A4 — 210 × 297 мм (*для текстовых документов и уменьшенных копий*)

A5 — 149 × 210 мм

Размер чертежного формата определяется сложностью и размерами изображаемой схемы/диаграммы, допускается горизонтальное или вертикальное расположение, для схем алгоритмов и UML-диаграмм рекомендуется **A2** или **A3**, для презентационных плакатов с комплексной информацией — **A1**

2.1.5. Защита курсовых проектов

Каждым студентом все курсовые проекты должны выполняться и сдаваться на проверку преподавателю в сроки, предусмотренные графиком работы студентов в текущем семестре. После исправления студентом всех ошибок, отмеченных преподавателем при проверке, каждая курсовая работа должна быть защищена. При исправлении ошибок из проверенной работы ни в коем случае ничего не выбрасывается. Исправления аккуратно записываются студентом на чистых страницах. На защиту обучающиеся приносят исправленные работы, сдают их преподавателю, получают индивидуальные вопросы по разделам курсовой работы. На подготовку отводится 30-40 мин. Если обучающийся успешно ответил на вопросы и у преподавателя нет никаких дополнительных замечаний по расчетно-графической работе, то защита считается законченной. После защиты курсовой проект остается у преподавателя. Если студентом курсовой проект защищен успешно и в срок, то в конце семестра он может автоматически получить дифференцированный зачет по дисциплине. В случае, когда обучающийся при защите не справляется с вопросами, то преподавателем назначается дополнительная защита (не более двух раз!). Если обучающимся курсовой проект не защищен в течение семестра, то защита и сдача ее производится в экзаменационную сессию.

3. СОДЕРЖАНИЕ ОТДЕЛЬНЫХ ЧАСТЕЙ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

3.1. Вводная часть

Краткий исторический обзор, освещающий развитие выбранной темы курсового проекта, по данным, опубликованным в литературе. Роль отечественных и зарубежных ученых, инженеров, рационализаторов, изобретателей в развитии темы исследования. Задачи, стоящие перед данным производством в настоящее время.

3.2. Анализ предметной области и существующих решений

Анализ предметной области и существующих решений представляет собой фундаментальный этап работы, на котором происходит глубокое исследование контекста и границ решаемой задачи. Здесь необходимо не просто перечислить аналоги, а провести критическое исследование современных подходов к разработке программного обеспечения для мехатронных и робототехнических систем. Анализ начинается с изучения предметной области: рассматриваются типичные архитектуры управления, распространенные протоколы обмена данными, особенности работы с датчиками и исполнительными механизмами в реальном времени. Далее следует обзор существующих программных решений - как коммерческих, так и открытых. Важно рассмотреть не только готовые продукты, но и библиотеки, фреймворки и инструменты, которые могут быть использованы в разработке. Критический анализ проводится по нескольким критериям: функциональная полнота, производительность, надежность, масштабируемость, сложность внедрения и сопровождения, лицензионные ограничения и стоимость. Особое внимание уделяется решениям, реализующим сходную функциональность или работающим в аналогичных условиях эксплуатации. В результате этого анализа формируется четкое понимание конкурентных преимуществ и недостатков существующих аналогов, что позволяет обоснованно определить нишу для разрабатываемого программного обеспечения. На основе проведенного

исследования формулируются выводы, которые лягут в основу последующих проектных решений: выбор технологического стека, определение ключевых требований и ограничений, а также выделение аспектов, в которых разрабатываемое решение должно обеспечить улучшение или инновацию по сравнению с существующими практиками.

3.2. Проектирование архитектуры программного обеспечения

Проектирование архитектуры программного обеспечения является центральным и наиболее ответственным этапом, на котором абстрактные требования преобразуются в конкретную структурную схему будущей системы. Этот процесс начинается с формализации и детализации технических требований, полученных на предыдущем этапе. Затем определяется общий стиль архитектуры: будет ли это монолитное приложение, микросервисная архитектура или, например, событийно-ориентированная система. Для мехатронных систем критически важным часто является выбор между централизованным и распределенным управлением, а также решение о том, какие компоненты будут работать в реальном времени. Следующим шагом проводится декомпозиция системы на логические модули или компоненты. Каждый компонент получает четко определенную ответственность, например, модуль управления двигателями, модуль обработки данных с датчиков, модуль связи с верхним уровнем или модуль ведения журнала событий. Проектируется взаимодействие между этими компонентами: определяются интерфейсы (API), форматы передаваемых сообщений или данных, а также протоколы обмена. Для систем, работающих в жестких временных рамках, отдельно проектируется модель параллельного выполнения, определяются потоки, процессы, их приоритеты и синхронизация. Архитектура также должна учитывать аспекты надежности и безопасности: проектируются механизмы обработки ошибок, восстановления после сбоев, резервирования критических функций. Визуальным итогом этого этапа становится набор диаграмм, выполненных в нотации UML или аналогичной: диаграммы

компонентов, диаграммы последовательностей для ключевых сценариев, диаграммы состояний для критических модулей. Проектирование завершается разработкой модели данных, описывающей ключевые структуры, их отношения и способы персистентного хранения.

3.3. Разработка алгоритмов управления

Данный раздел переводит архитектурные решения на уровень конкретных вычислительных процедур, которые обеспечат выполнение системой своих целевых функций. Этот раздел фокусируется на алгоритмах, которые на основе входных данных (заданий оператора, показаний датчиков) формируют управляющие воздействия для исполнительных механизмов. Разработка начинается с создания или выбора математической модели объекта управления, будь то кинематическая модель манипулятора, динамическая модель привода или модель среды для мобильного робота. На основе этой модели и требований к качеству управления (быстродействие, точность, устойчивость) выбираются или синтезируются управляющие алгоритмы. Это могут быть классические ПИД-регуляторы, методы модального управления, адаптивные или нечеткие алгоритмы. Каждый выбранный алгоритм детально описывается: приводятся его математические формулировки, блок-схемы, поясняется физический смысл параметров. Особое внимание уделяется алгоритмам обработки сигналов с датчиков: фильтрации шумов, компенсации систематических ошибок, преобразованию сырых данных в значимую информацию. Также разрабатываются логические алгоритмы более высокого уровня, определяющие поведение системы в различных сценариях: алгоритмы инициализации и калибровки, алгоритмы переключения режимов работы, алгоритмы диагностики неисправностей и реакции на аварийные ситуации. Для каждого алгоритма оценивается его вычислительная сложность и требования к ресурсам, что особенно важно для встраиваемых систем. Результатом этапа является полный набор

формализованных процедур, готовых к программной реализации, с четким пониманием их взаимосвязи и вклада в общее функционирование системы.

3.4. Реализация программного обеспечения

Реализация программного обеспечения - это этап воплощения проектных решений и алгоритмов в работающий программный код. Здесь абстрактные диаграммы и математические формулы преобразуются в конкретные классы, функции и конфигурационные файлы. Работа начинается с организации структуры проекта: создания репозитория кода, настройки системы управления версиями, разбиения кодовой базы на логические модули в соответствии с разработанной архитектурой. Затем следует непосредственное кодирование. Каждый модуль реализуется на выбранном языке программирования с соблюдением принятых стандартов кодирования и принципов написания чистого, поддерживаемого кода. Критически важные модули, такие как драйверы оборудования или алгоритмы реального времени, пишутся с особым вниманием к эффективности и детерминированности выполнения. Параллельно с написанием бизнес-логики реализуются служебные механизмы: логирование событий, загрузка конфигураций, обработка исключений. Отдельная сложная задача - реализация взаимодействия с аппаратной частью: написание или интеграция драйверов для датчиков и приводов, реализация протоколов обмена данными. Если система включает пользовательский интерфейс, он создается с использованием выбранных библиотек, при этом уделяется внимание удобству использования и ясности отображаемой информации. Весь процесс реализации сопровождается комментированием кода и начальным документированием API модулей. Важным аспектом является обеспечение переносимости и конфигурируемости кода, чтобы систему можно было адаптировать к изменяющимся требованиям или разному аппаратному обеспечению без глубокого вмешательства в код. Результатом этапа является полный

исходный код системы, который может быть собран, развернут и запущен на целевой платформе.

3.5. Тестирование и отладка программного обеспечения

Тестирование и отладка программного обеспечения - это комплексная деятельность, направленная на верификацию корректности реализации и выявление дефектов. Начинается она с планирования, где определяется стратегия тестирования: какие уровни тестирования будут применяться (модульное, интеграционное, системное), какие методы (статический анализ, динамическое тестирование, моделирование) и какие инструменты. Первым обычно выполняется модульное тестирование, при котором каждый отдельный компонент, класс или функция проверяется на соответствие техническому заданию в изоляции от остальных частей системы. Для этого создаются тестовые стенды и пишутся модульные тесты, которые автоматически проверяют различные сценарии работы, включая граничные случаи и ошибочные ситуации. После успешного тестирования модулей начинается интеграционное тестирование, в ходе которого проверяется взаимодействие между компонентами. Здесь выявляются проблемы с интерфейсами, несоответствия в форматах данных, ошибки синхронизации. Системное тестирование проверяет работу всего программного комплекса как единого целого на соответствие всем исходным требованиям. Для мехатронных систем особую важность имеет тестирование в условиях, близких к реальным, которое часто проводится на стендах или с использованием симуляторов. В процессе тестирования неизбежно обнаруживаются ошибки, для устранения которых проводится отладка — локализация источника ошибки в коде и его исправление. Используются различные инструменты отладки: отладчики, профайлеры, анализаторы покрытия кода и статические анализаторы. Помимо функционального тестирования, часто проводится нагрузочное тестирование для оценки производительности и проверки работы в реальном времени, а также тестирование на надежность и

устойчивость к сбоям. Все обнаруженные инциденты и процесс их устранения документируются. Итогом этого этапа является не только исправленный и стабильный код, но и комплект тестов, который может использоваться для регрессионного тестирования в будущем, а также отчет, содержащий оценку качества ПО, описание выявленных и устраненных проблем и подтверждение того, что система готова к эксплуатации или следующему этапу внедрения..

4. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

При выполнении курсового проекта по дисциплине «Программное обеспечение мехатронных устройств и роботов» основной акцент делается на проектировании и реализации программных решений для управления мехатронными и робототехническими системами. Это может быть разработка нового программного обеспечения с нуля, модернизация существующей программной системы, интеграция различных программных модулей в единый комплекс, или создание системы управления для конкретного технологического процесса. Студентам рекомендуется рассматривать проект как комплексную задачу, включающую анализ требований, проектирование архитектуры, реализацию, тестирование и документирование программного продукта, предназначенного для работы в реальном времени и взаимодействия с аппаратными компонентами.

Графическая часть проекта, отражающая архитектуру, алгоритмы и результаты работы ПО, должна выполняться в соответствующих масштабах, обеспечивающих наглядность и читаемость всех элементов. Для схем, диаграмм и графиков следует выбирать масштабы из стандартизированных рядов согласно общепринятой практике визуализации технической информации. Рекомендуется использовать целочисленные масштабы, обеспечивающие компактное и понятное расположение элементов на листе: для общих архитектурных схем может применяться уменьшение, а для детальных диаграмм последовательностей или блок-схем алгоритмов - увеличение. Главным критерием выбора масштаба является удобство восприятия информации как при рассмотрении плаката вблизи, так и с расстояния во время защиты.

Каждый графический документ должен содержать основную надпись (штамп), оформленную в соответствии с принципами технической документации. На графических материалах (плакатах, схемах) основная надпись размещается в правом нижнем углу и включает необходимые

атрибуты: наименование проекта, данные исполнителя, руководителя, дату и номер листа. Для текстовых документов, к которым относится пояснительная записка, титульный лист и разделы приложений, также предусматривается заглавный лист с аналогичными реквизитами. Система обозначений документов в рамках курсового проекта должна быть единой и логичной. Например, для проекта «Разработка системы управления манипулятором с обратной связью» могут быть приняты следующие обозначения: пояснительная записка - КП-СевКавГА-15.03.06-01-25 ПЗ; диаграмма компонентов программной архитектуры - КП-СевКавГА-15.03.06-01-25-00.00.001 ДК; схема базы данных - КП-СевКавГА-15.03.06-01-25-00.00.002 СБД; а для схемы развертывания программных модулей - КП-СевКавГА-15.03.06-01-25-00.00.003 СР. Подобная система кодирования обеспечивает четкую идентификацию всех материалов проекта.

Спецификация, как документ, описывающий состав программного продукта, относится к важным текстовым артефактам. Её рекомендуется оформлять в виде отдельного документа, структурированного в виде таблицы. Спецификация составляется на всю систему или её ключевые подсистемы и включает перечень программных модулей, библиотек, внешних зависимостей, конфигурационных файлов, а также описание аппаратных компонентов, с которыми взаимодействует ПО. Документ может быть выполнен на листах формата А4 и приложен к пояснительной записке. К структуре и оформлению спецификации предъявляются требования четкости и единообразия: разделы и подразделы, такие как «Исполняемые модули», «Библиотеки», «Конфигурации», «Внешние интерфейсы», оформляются в виде заголовков в соответствующей графе таблицы. Каждый элемент спецификации должен иметь однозначное наименование и, при необходимости, версию или идентификатор.

Проектирование программного обеспечения включает этап моделирования и расчета ключевых параметров системы. В рамках технологического и технического расчета определяются характеристики, критически важные для работы ПО: быстродействие системы (частота цикла управления), объемы используемой памяти, пропускная способность каналов связи, время отклика на внешние события. Рассчитываются параметры, связанные с обработкой данных: требуемая частота дискретизации сигналов с датчиков, сложность алгоритмов фильтрации и управления, оценка нагрузки на процессор. Эти данные являются основой для выбора аппаратной платформы, проектирования архитектуры ПО и разработки алгоритмов. Полученные расчетные значения должны найти отражение в проектной документации они указываются на соответствующих схемах (например, на диаграммах развертывания или в описании интерфейсов) и служат исходными данными для последующей реализации и тестирования.

Особое внимание уделяется разработке схем, наглядно отображающих структуру и логику программного обеспечения. Архитектурная схема системы (выполняемая, например, в нотации UML) представляет собой условное изображение всех основных программных компонентов, модулей и способов их взаимодействия. Она должна давать ясное представление об иерархии системы, распределении функций, потоках данных и управления между модулями, а также об интерфейсах с внешними системами и аппаратурой. Такая схема может выполняться на формате A1 или A2. На схемах непосредственно указываются ключевые параметры компонентов: идентификаторы модулей, версии интерфейсов (API), используемые протоколы обмена. Все элементы схемы нумеруются и сопровождаются пояснениями в виде легенды или выносных текстовых блоков. При описании принципов работы программного обеспечения в пояснительной записке обязательны ссылки на номера элементов этих схем. Использование условных обозначений, соответствующих

общепринятым стандартам (таким как UML), обязательно. На схемах, отображающих потоки данных или последовательности вызовов, стрелками указывается направление взаимодействия. При этом внутренние детали реализации алгоритмов, не влияющие на архитектурный уровень, на общей схеме не отображаются.

Технический расчет также включает оценку ресурсоемкости и производительности программного обеспечения. Определяются такие параметры, как требуемое быстродействие центрального процессора для реализации алгоритмов в реальном времени, объем оперативной и постоянной памяти для хранения кода и данных, характеристики каналов ввода-вывода. Проводится оценка времени выполнения критических участков кода, задержек в передаче данных, что особенно важно для систем с жесткими временными ограничениями. Результаты этих расчетов используются для обоснования выбора аппаратной платформы, оптимизации алгоритмов и структуры данных, а также являются критериями при последующем тестировании и приемке программного продукта.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ ЛИТЕРАТУРА

1. Шустров, А. С. Основы прикладного программирования на языке Python : учебное пособие / А. С. Шустров. — Москва : Ай Пи Ар Медиа, 2026. — 91 с. — ISBN 978-5-4497-4770-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/154932.html> (дата обращения: 19.12.2025). — Режим доступа: для авторизир. пользователей
2. Гегечкори, Е. Т. Программирование на языке Python : учебное пособие / Е. Т. Гегечкори. — Омск : Омский государственный технический университет, 2023. — 172 с. — ISBN 978-5-8149-3665-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/140856.html> (дата обращения: 19.12.2025). — Режим доступа: для авторизир. пользователей.
3. Горбаченко, В. И. Машинное обучение: настраиваем ПО, готовим данные, анализируем : практическое пособие / В. И. Горбаченко, К. Е. Савенков, М. А. Малахов. — Москва, Алматы : Ай Пи Ар Медиа, EDP Hub, 2024. — 248 с. — ISBN 978-5-4497-2314-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/133452.html> (дата обращения: 19.12.2025). — Режим доступа: для авторизир. пользователей.
4. Кузьмин, А. К. Распознавание образов с использованием искусственных нейронных сетей : учебное пособие / А. К. Кузьмин, А. В. Бровко, А. В. Ермаков. — Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2024. — 148 с. — ISBN 978-5-7433-3637-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/150089.html> (дата обращения: 19.12.2025). — Режим доступа: для авторизир. пользователей
5. Буйначев, С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг ; под редакцией Ю. В.

Песин. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2014. — 92 с. — ISBN 978-5-7996-1198-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/66183.html> (дата обращения: 19.12.2025). — Режим доступа: для авторизир. пользователей

6. Боровская, Е. В. Основы искусственного интеллекта : учебное пособие / Е. В. Боровская, Н. А. Давыдова. — 6-е изд. — Москва : Лаборатория знаний, 2024. — 128 с. — ISBN 978-5-93208-797-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/144313.html> (дата обращения: 19.12.2025). — Режим доступа: для авторизир. пользователей

7. Запечников, С. В. Основы интеллектуального анализа данных и машинного обучения: конспект лекций : учебное пособие / С. В. Запечников. — Москва : Национальный исследовательский ядерный университет «МИФИ», 2022. — 136 с. — ISBN 978-5-7262-2856-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/132687.html> (дата обращения: 19.12.2025). — Режим доступа: для авторизир. пользователей

8. Макаренко, С. И. Интероперабельность человеко-машинных интерфейсов : монография / С. И. Макаренко. — Санкт-Петербург : Научное издание, 2023. — 186 с. — ISBN 978-5-907618-37-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/130087.html> (дата обращения: 23.12.2025). — Режим доступа: для авторизир. Пользователей

9. Акчурин, Э. А. Человеко-машинное взаимодействие : учебное пособие / Э. А. Акчурин. — Москва : СОЛОН-Пресс, 2021. — 94 с. — ISBN 978-5-91359-022-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/141893.html> (дата обращения: 23.12.2025). — Режим доступа: для авторизир. пользователей

10. Березкина, Л. В. Эргономика : учебное пособие / Л. В. Березкина, В. П. Кляуззе. — Минск : Вышэйшая школа, 2013. — 432 с. — ISBN 978-985-06-2309-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/24090.html> (дата обращения: 23.12.2025). — Режим доступа: для авторизир. пользователей

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ
Кафедра «Мехатронные и робототехнические системы»

ЗАДАНИЕ

на курсовой проект по дисциплине «Программное обеспечение
мехатронных устройств и роботов»

Тема проекта: Разработка программного обеспечения управления
двухосевым манипулятором с обратной связью

Студент: Иванов Иван Иванович, группа МР-41

Руководитель: Петров Пётр Петрович, к.т.н., доцент

Срок выполнения: с 01.09.2025 по 25.12.2025

Дата выдачи задания: 01.09.2025

1. Объект проектирования

Двухосевой антропоморфный манипулятор с электроприводами постоянного тока, энкодерами обратной связи и концевыми датчиками. Предназначен для выполнения операций захвата и перемещения деталей в рабочей зоне 500×500 мм.

2. Технические требования

2.1. Функциональные требования:

Позиционное управление по двум координатам (X, Y); Режимы работы: Ручное управление (джойстик или клавиши); Автоматическое движение по заданной траектории; Запись и воспроизведение перемещений; Контроль предельных положений (концевые датчики); Аварийная остановка при срабатывании защиты; Самодиагностика системы

2.2. Требования к производительности:

Частота цикла управления: не менее 100 Гц; Время отклика на аварийный сигнал: не более 10 мс; Точность позиционирования: $\pm 0,5$ мм; Скорость перемещения: 0,1-1,0 м/с

2.3. Требования к интерфейсам:

Интерфейс с драйверами двигателей: ШИМ + направление; Интерфейс с энкодерами: квадратурный сигнал; Цифровые входы для концевых датчиков; Последовательный интерфейс для связи с ПК (UART/USB); Графический интерфейс оператора на ПК

3. Требования к программному обеспечению

3.1. Для микроконтроллера управления: язык программирования: C; Среда разработки: STM32CubeIDE; Микроконтроллер: STM32F407; Операционная система: FreeRTOS

3.2. Для программного обеспечения ПК: язык программирования: Python 3.x; Графическая библиотека: PyQt5; Протокол связи: Modbus RTU поверх UART.

4. Задачи проектирования

4.1. Аналитическая часть (к 15.09.2025): Исследование методов управления манипуляторами; Анализ существующих решений и библиотек; Выбор алгоритмов управления и фильтрации

4.2. Проектная часть (к 30.09.2025):

Разработка архитектуры ПО (модульная структура); Проектирование алгоритма ПИД-регулятора; Разработка протокола обмена данными; Проектирование графического интерфейса

4.3. Реализация (к 15.11.2025):

Написание кода для микроконтроллера: Драйверы двигателей и энкодеров; Алгоритм управления; Обработка прерываний; Написание кода для ПК: Графический интерфейс; Модуль связи; Логика управления

4.4. Тестирование (к 05.12.2025): Модульное тестирование компонентов; Интеграционное тестирование системы; Тестирование на макете/симуляторе; Испытания в реальных условиях

4.5. Документирование (к 20.12.2025):

Пояснительная записка (35-40 страниц); Документация на код (Doxugen); Руководство оператора; Презентация для защиты

5. Перечень представляемых материалов

- Пояснительная записка в печатном и электронном виде
- Исходный код программы (архив Git-репозитория)
- Файлы проектов для STM32CubeIDE и Python
- Файлы конфигурации и настройки
- Отчёт по тестированию с результатами
- Графические материалы (схемы, диаграммы, скриншоты)
- Презентация для защиты (15-20 слайдов)

6. Особые указания

- Код должен соответствовать стандарту MISRA-C (для части на C)
- Обязательно использование системы контроля версий Git
- Должна быть обеспечена возможность конфигурации системы без перекомпиляции
- Предусмотреть режим симуляции работы без реального оборудования
- Обеспечить журналирование событий и ошибок

7. Этапы выполнения

- | | | | |
|---|----------------------------|------------|------------------|
| 1 | Анализ и исследование | 15.09.2025 | Отчёт по анализу |
| 2 | Проектирование архитектуры | 30.09.2025 | UML-диаграммы |
| 3 | Реализация ядра системы | 20.10.2025 | Код модулей |
| 4 | Разработка интерфейсов | 10.11.2025 | Рабочий прототип |
| 5 | Тестирование и отладка | 05.12.2025 | Отчёт по тестам |
| 6 | Оформление документации | 20.12.2025 | Черновик ПЗ |
| 7 | Подготовка к защите | 25.12.2025 | Презентация |

С заданием ознакомлен: _____ /Иванов И.И./ (дата)

Задание выдал: _____ /Петров П.П./ (дата)

Приложение 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ»

Инженерный институт
Кафедра «Мехатронные и робототехнические системы»

КУРСОВОЙ ПРОЕКТ

По дисциплине
«Программное обеспечение мехатронных устройств и роботов»

На тему:
«Разработка программного обеспечения управления двухосевым
манипулятором с обратной связью»

Разработал:
Обучающийся 4 курса, гр.

Иванов И.И.
Руководитель:

Черкесск 2025 г

МАЛСУГЕНОВ Роман Сергеевич
БАЙРАМУКОВ Рашид Альбертович
БАЙРАМУКОВ Ахмат Османович

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕХАТРОННЫХ УСТРОЙСТВ И РОБОТОВ

Методические рекомендации по выполнению
курсового проекта для обучающихся всех форм обучения,
направления подготовки 15.03.06 Мехатроника и робототехника,
направленность (профиль) «Мехатронные и роботизированные
технологические системы и комплексы»

Печатается в редакции автора

Корректор Темирлиева Р.М.
Редактор Темирлиева Р.М.

Формат 60x84/16
Бумага офсетная.
Печать офсетная.
Усл. печ. л. 1,8
Заказ № 3971
Тираж 100 экз.

Оригинал-макет подготовлен в Библиотечно-издательском
центре СевКавГА
369000, г. Черкесск, ул. Ставропольская, 36