

охране труда - подпись руководителя по промышленной экологии - презентация - доклад																				
Устранение замечаний по всей ВКР	08.06- 15.06																			
Рецензирование Сдача работ на кафедру	20.06																			
Диск с материалами ВКР	21.06																			
Защита ВКР	22.06 – 23.06																			

Руководитель ВКР _____ / _____ /

« ____ » _____ 2022 г.

Студент _____ / _____ /

« ____ » _____ 2022 г.

Согласовано:

Председатель ПЦК «Естественнонаучных дисциплин» _____ / М.Н. Апталаев /
« ____ » _____ 2022 г.

Министерство науки и высшего образования Российской Федерации
Лысьвенский филиал федерального государственного автономного образовательного
учреждения высшего образования
«Пермский национальный исследовательский политехнический университет»
ПЦК «Естественнонаучных дисциплин»

Утверждаю:
Председатель ПЦК
_____ М.Н.Апталаев
«__» _____ 2022 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

студенту РОДИОНОВУ Александру Михайловичу курса 4

группы КСК9-18-1спо
специальность 09.02.01 Компьютерные системы и комплексы

Тема задания «Разработка проекта робота-шагохода с микроконтроллерным управлением»

Структура работы такова:

а) Введение. Аргументировать актуальность выбранной темы, ее теоретическое значение и практическую значимость для исследуемого предприятия, сформулировать цель и конкретные задачи исследований. Конкретизировать объект и предмет исследований. Увязать решение темы ВКР с общими научно-техническими задачами цифровизации экономики страны.

б) Исследовательский раздел. Понятие «Мехатронная система». Понятие «Гексапод». Особенности кинематических схем движения роботов на шагающих опорных конструкциях. Обзор существующих на рынке роботов-гексаподов. Формирование требований к проектируемой системе.

в) Конструкторский раздел. Обоснование и выбор инструментального обеспечения проекта. Разработка структурной и функциональной схем проекта. Разработка принципиальной схему контроллера робота. Выбор элементной базы. Разработка кинематической схемы движения конечностей робота-гексапода с учетом требований ТЗ. Разработка управляющей программы робота. Разработка мобильного приложения

г) Охрана труда и промышленная экология. Анализ вредных и опасных производственных факторов на рабочем месте инженера-электроника. Разработка средств защиты от воздействия выбранного ВиОПФ. Экологические требования к утилизации вычислительной и оргтехники, а также их расходных материалов.

д) Организационно-экономический раздел. Выполнение технико-экономического оценки разработанного устройства.

е) Заключение. Краткое изложение решенных задач, актуальность работы, соответствие полученных результатов теме и заданию ВКР.

ж) Список использованных источников.

з) Приложения.

ВВЕДЕНИЕ

1. Исследовательский раздел
 - 1.1. Микроконтроллер как средство автоматизации
 - 1.2. Сущность понятия "умный дом"
 - 1.3. Анализ рынка систем автоматизации типа "умный дом"
2. Конструкторский раздел

- 2.1. Формирование требований к проектируемой системе
- 2.2. Выбор инструментального обеспечения проекта
- 2.3. Разработка аппаратной части проектируемой системы
- 2.4. Разработка программной части проектируемой системы
3. Организационно-экономический раздел
 - 3.1. Расчет себестоимости проекта
 - 3.2. Расчет экономической эффективности проекта
4. Охрана труда и промышленная экология
 - 4.1. Анализ вредных и опасных производственных факторов при пайке деталей, узлов и наладке электронных устройств
 - 4.2. Расчет технических средств обеспечения безопасности труда на рабочем месте инженера-электроника
 - 4.3. Утилизация компьютерной и оргтехники техники

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЯ

Дата выдачи _____

Срок окончания _____

Руководитель ВКР

_____ /М.Н. Апталаев/

«___» _____ 2022 г.

Задание утверждено на заседании ПЦК «Естественнонаучных дисциплин» протокол №____
от _____ 2022 г.

Председатель ПЦК _____ /М.Н. Апталаев/

«___» _____ 2022 г.

РЕФЕРАТ

Родионов А.М. РАЗРАБОТКА РОБОТА-ШАГОХОДА ПОД МИКРОКОНТРОЛЛЕРНЫМ УПРАВЛЕНИЕМ, выпускная квалификационная работа: стр.82, рис. 28, табл. 16. Форм. 60.

Объект исследования - роботы.

Цель работы – выполнить разработку робот-шагохода под микроконтроллерным управлением.

За время выполнения работы был проведён анализ предметной области, и на основании полученных знаний проведена разработка робота-шагохода.

В результате разработки было получено устройство робота, имеющего низкую стоимость относительно конкурентов на рынке, а также выполняющего все требуемые функции в соответствии с техническим заданием.

Основные конструктивные и экономические показатели: низкая себестоимость и простота.

Устройство может использоваться для обучения студентов предметам микроконтроллерных систем, механики и мехатроники.

Министерство науки и высшего образования Российской Федерации
Лысьвенский филиал федерального государственного автономного образовательного
учреждения высшего образования
«Пермский национальный исследовательский политехнический университет»
Факультет профессионального образования

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему «Разработка робота-шагохода под микроконтроллерным
управлением»

студента группы КСК9-18-1спо по специальности 09.02.01 Компьютерные
системы и комплексы

Родионова Александра Михайловича _____

Руководитель: _____ (М.Н.Апталаев)

Консультант по экономической части:
_____ (К.В.Кондратьева)

Консультант по промышленной экологии и охране труда:
_____ (А.К.Торощин)

Рецензент: _____ (_____)

Допуск к защите: _____ (М.Н.Апталаев)

Лысьва, 2022 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	9
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ.....	11
1.1 Понятие мехатронная система.....	11
1.2 Особенности кинематических схем движения роботов на шагающих опорных конструкциях.....	13
1.3 Обзор существующих на рынке роботов.....	23
1.4 Анализ требований.....	27
1.5 Выводы по разделу.....	28
2 КОНСТРУКТОРСКИЙ РАЗДЕЛ.....	29
2.1 Выбор инструментальной базы.....	29
2.2 Выбор материальной базы проекта.....	35
2.3 Разработка структурной и функциональной схем проекта.....	38
2.4 Разработка принципиальной схемы контроллера робота.....	39
2.5 Разработка корпуса робота-шагохода.....	43
2.6 Разработка кинематической схемы движения конечностей робота-гексапода с учетом требований ТЗ.....	47
2.7 Разработка управляющей программы робота.....	53
2.8 Разработка мобильного приложения.....	56
2.9 Вывод по разделу.....	59
3 ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ.....	61
3.1 Утилизация средств вычислительной техники и оргтехники.....	61
3.2 Анализ вредных и опасных производственных факторов на рабочем месте.....	62
3.3 Мероприятия по оптимизации освещенности на рабочем месте.....	66
3.4 Вывод по разделу.....	67
4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ.....	69
4.1 Расчет затрат на разработку программы для микроконтроллера.....	69
4.2 Расчет затрат на внедрение программы для микроконтроллера.....	73
4.3 Расчет эксплуатационных текущих затрат по программе для микроконтроллера.....	75
4.4 Расчет экономической целесообразности разработки и внедрения информационных технологий.....	78

ЗАКЛЮЧЕНИЕ	84
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	85
Приложение А – Техническое задание	88
Приложение Б – Программный код	96
Приложение В – блок-схема алгоритма обработки данных с локатора.....	112
Приложение Г – Чертежи корпуса	113
Приложение Д – принципиальная схема устройства	118

ВВЕДЕНИЕ

В настоящее время большое количество исследований и проектов направленно на разработку универсального и многоэффективного устройства способного выполнять какие-либо действия или собирать информацию, имея возможность удалённого управления. Эти проекты актуальны тем, что они способны заменить рабочую силу, а значит сэкономить средства, затрачиваемые на проведение некоторых работ. Кроме того, такие устройства могут использоваться в зоне, представляющей потенциальную опасность для человека.

Для таких устройств выдвигаются требования по хорошей проходимости, в результате которой устройство должно уметь проходить препятствия, расположенные на поверхности, а также возможность безопасного передвижения по лестницам. С этим плохо справляются устройства на колёсной и гусеничной базе, по причине того, что они способны лишь двигаться по поверхности, не имея возможности передвижения по лестницам и через большие препятствия.

Роботы на дистанционном управлении, передвигающиеся посредством перестановки ног, являются наиболее подходящими для этих целей, так как имеют возможность передвижения по прямым поверхностям, преодолевая препятствия и к тому же имеют возможность передвижения по лестницам.

Сегодня роботы-шагоходы мало где используются, но уже сейчас можно сказать, что по причине их дешевизны и универсальности скоро эти устройства будут иметь наибольшую популярность в отличии от настоящей.

Робот, проектируемый в выпускной квалификационной работе, будет предназначен для обучения студентов робототехнике, механике и микроконтроллерным системам, на примере готового устройства.

Объектом исследования являются роботизированные шагающие платформы.

Предметом исследования выступает проектирование роботов-шагоходов.

Целью выпускной квалификационной работы является разработка робота-шагохода с микроконтроллерным управлением.

Задачи выпускной квалификационной работы:

- провести анализ предметной области исследования,
- провести подбор инструментальной базы проекта,
- выполнить разработку контроллера робота-шагохода,
- провести разработку деталей корпуса робота,
- провести анализ требований охраны труда и промышленной экологии,
- выполнить расчёт экономической эффективности проекта.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Понятие мехатронная система

Понятие мехатроника было выдвинуто в 1969 году в Японии. Мехатроника - это область науки, которая является пересечением таких наук как компьютерное управление, механика и электроника.

Мехатроника представляет собой некоторую систему способную выполнять какие-либо механические движения, управление которых происходит с помощью электрических сигналов, приходящих от компьютера или микроконтроллера [3].

В настоящее время мехатроника представляет собой наиболее востребованную и сильно развивающуюся науку. Из наиболее частых и знакомых всем примерам использования мехатроники можно выделить систему, работающую в лазерном принтере. Большое количество валов и шестерёнок двигается и продвигают бумагу, нанося на неё тонер. Всё это количество валов и шестерёнок было бы сложно соединить в единую систему без использования микроконтроллерного управления, которое представляет собой электронную цифровую систему.

Так же к примерам мехатроники можно отнести и жёсткий диск, предназначенный для хранения данных не зависимо от питания. В этом случае механическими движениями будут являться движения магнитного диска, на котором хранятся данные, а также движения магнитной головки. Для всей этой системы так же необходимо управление, в качестве которого используется плата управления.

В прошлом веке мехатроника использовалась для разработки некоторых вспомогательных периферийных устройств, предназначенных для хранения информации, отпечатки текста и других полезных делах для развивающейся техники. В настоящее время мехатроника имеет наибольшее развитие по причине того, что всё чаще начинают разрабатываться роботизированные системы, в задачи которой входит процесс движения на подобию живого существа.

Роботы - это новое движение в технике. Эти устройства имеют возможность перемещения и выполнения каких-либо действий, посредством, механических движений.

Роботы могут быть исполнены на основе передвижения ног, колёсной или гусеничной базы. В любом из этих случаев, будет использоваться мехатронная система, в состав которой будет входить устройства выполнения механического движения (механические приводы или двигатели для колёс), а также система управления этими устройствами, чаще в качестве которых используются микроконтроллеры.

Роботы так же могут использоваться и на предприятиях, связанных с однотипными процессами сборки корпусов. Так, например, на предприятии сборки автомобилей, используется большое количество роботом манипуляторов, предназначенных для автоматизации процесса сборки автомобилей, и экономии на рабочем труде.

На рисунке 1 приведена схема простой мехатронной системы [4].

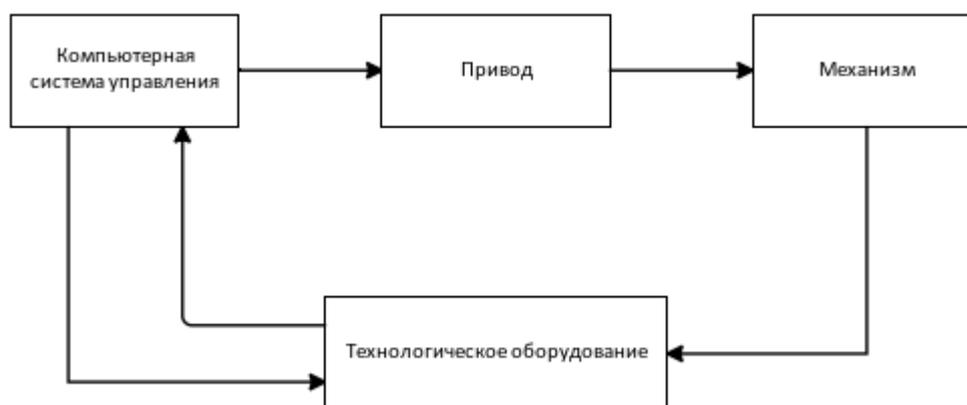


Рисунок 1 - Схема мехатронной системы

Как можно видеть на рисунке 1, компьютерная система управления воздействует на приводы, которые являются исполнительными элементами системы, и в свою очередь приводят в движение механизм, на котором установлены приводы. В результате такая система может влиять на техническое оборудование и технологический процесс.

В качестве приводов могут использоваться как электронные, так и гидравлические, пневматические приводы.

Так можно считать, что мехатронной системой можно считать любое устройство способное выполнять механические движения под управлением компьютерной или иной другой электронной системой [4].

1.2 Особенности кинематических схем движения роботов на шагающих опорных конструкциях

Походка роботов шагоходов чаще всего списана с походок живых существ, по этой причине тип походки большинства шагоходов был назван биологическим.

Роботы, построенные на биологическом типе походок, является наиболее эффективными, по причине того, что биологические походки используются живыми существами, которые их получили в процессе эволюции, что говорит о их наибольшей эффективности.

Основные виды походок и конструкции ног, которые чаще используются в проектировании роботов – шагоходов показаны на рисунке 2 [2].

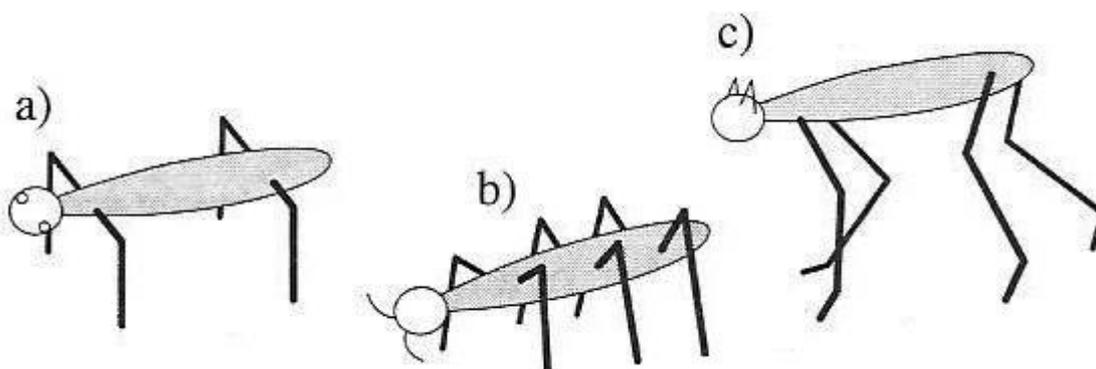


Рисунок 2 – Походки и конструкция ног шагоходов

Из примеров, приведённых на рисунке 2 наиболее интересна конструкция ног робота под буквой 'b'. При этой конструкции обеспечивается универсальность походки, а также возможность плавности и стабильности.

Походка, указанная под буквой с является наименее подходящей для проектирования робота, так как в этом случае используется большое количество сгибов ног, а также точка центра тяжести робота расположена высоко, что является не самым положительным вариантом для проектирования робота шагохода, по причине повышения риска опрокидывания корпуса.

Расположение ног, показанное на рисунке 2 под буквой 'а', также является не самым хорошим вариантом расположения ног на роботе, по причине, не обеспеченной стабильной и плавной походки, что приведёт к раскачке робота, а в определённых случаях может и опрокинуть его.

Среди различий роботов шагоходов, помимо расположения и конструкции ног, важным фактором является количество ног робота. Этот фактор влияет прямым образом на стабильность и плавность походки, а также на его устойчивость [2].

Гексапод – это робот, выполняющий передвижения посредством перестановки шести ног, выполненный в форме паука, конечности и походка которого также напоминают походку паука [5].

При проектировании робота шагохода используется такой термин как статистически стабильная и динамическая стабильная походка.

Статистически стабильная походка представляет собой походку, при которой робот может быть остановлен в любой момент времени, без возможности опрокидывания. К этой походке относятся роботы, имеющие большое количество ног, и не использующие при походке динамических движений.

В случае динамически стабильной походки, сам процесс передвижения, от части, выполняется с помощью динамических движений, которые чаще встречаются у роботов имеющих количество ног меньше четырёх. При динамической походке робот может быть остановлен только в определённые моменты времени, в противном случае появляется большой риск опрокидывания корпуса робота.

При разных типах походок, робот выполняет разные движения и перестановки ног, что прямым образом влияет на качество походки, стабильность и плавность движения. Для разработки робота – шагохода существует большое количество различных видов походок. Чем больше количество ног робота, тем больше всевозможных походок можно придумать [2].

В понятие походка можно отнести совокупность времени перестановки и последовательности перестановки ног, а также расположение ног при перестановке. В совокупности все эти факторы позволяют получить некоторую походку, позволяющую передвигаться в заданном направлении и с определённой скоростью.

Для изучения походок можно использовать диаграмму, на которой показано в какое время, и какая нога робота касается поверхности, на которой стоит и передвигается робот.

Для обеспечения наиболее стабильной походки, перед перестановкой ног, все остальные ноги, участвующие до этого в походке, должны быть установлены на поверхности.

На рисунке 3 показан пример походки робота, имеющего четыре ноги [2].

По горизонтали диаграммы расположена ось времени, а по вертикали расположены все ноги робота. Для большей понятности на рисунке 3 также показан пример исследуемого робота шагохода с подписанными названиями каждой ноги.

На самой схеме можно видеть интервалы время, в течение которого нога, находится в перестановке, и не касается поверхности, на которой расположен робот.

Все действия в походке являются циклическими, и повторяются бесконечно, до конца походки робота.

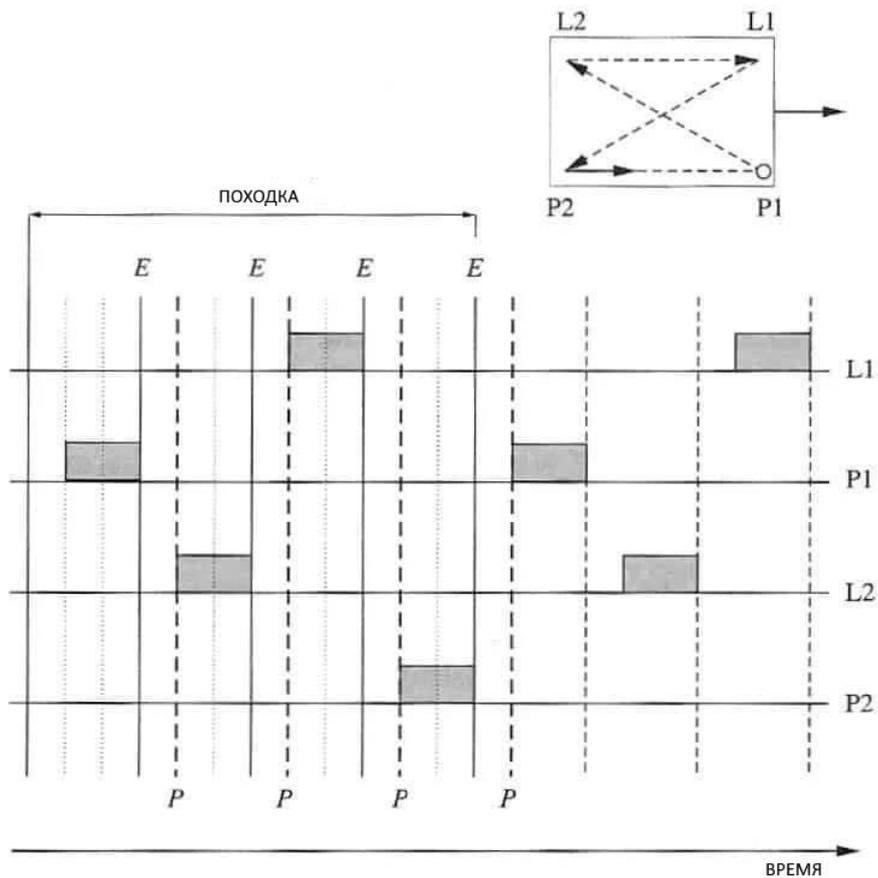


Рисунок 3 – Диаграмма походки робота

В результате рассмотренной походки на рисунке 3, робот паук будет передвигаться, переставляя ноги по диагонали, начиная с правой передней ноги, после которой переставляется полностью обратная ей левая задняя нога, после чего переставляется левая передняя и правая задняя ноги.

Такая диаграмма даёт понятие о времени и последовательности перестановки ног, но в ней не содержится никакой информации о том, в какую точку и на какое расстояние относительно корпуса будет поставлена передвигаемая нога.

Для этих целей используется опорный многоугольник, который на себе содержит информацию о точках соприкосновения с поверхностью. Эти точки в совокупности представляют собой некоторый многоугольник, где точки соприкосновения с поверхностью выступают вершинами.

Основной задачей опорного многоугольника является просчёт стабильности состояния корпуса. Для этого на опорном многоугольнике

добавляется точка центра тяжести, которая должна находиться всегда в центре многоугольника. При выходе за границы многоугольника, считается что расположение ног робота шагохода является не правильным, что грозит опрокидыванием корпуса, по причине не правильного распределения массы корпуса.

На опорном многоугольнике располагаются несколько вариантов положения ног, при которых робот выполняет движения во всех направления, и проводится проверка на правильность расположения точек соприкосновения.

Данная процедура является максимально простой, но при проектировании робота, имеющего мало ног, от трёх до шести, данная операция является необходимой, так как в случае не правильного расположения ног на этапе проектирования походки, в дальнейшем опрокидывания будут частой проблемой.

При создании робота с большим количеством ног, данная операция является не столь важной, так как большое количество опорных точек всегда создаст условия, при которых центр тяжести не покинет границы опорного многоугольника [2].

Пример опорного многоугольника для робота шагохода с четырьмя ногами представлен на рисунке 4 [3].

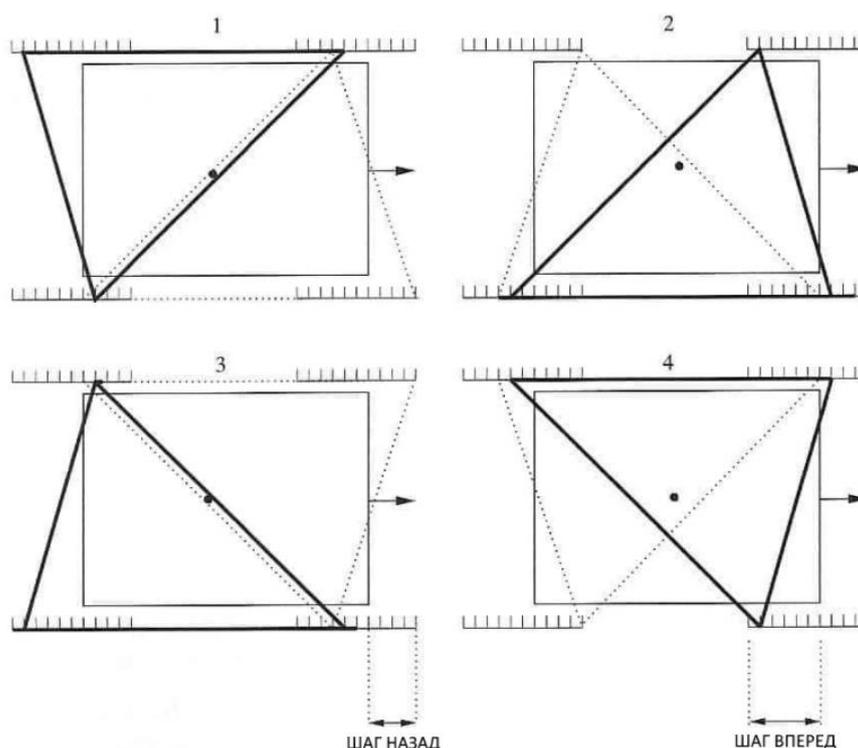


Рисунок 4 - Опорный многоугольник

После грамотной расстановки точек соприкосновения ног с поверхностью, следует рассчитать углы поворота каждого привода, установленного на сгибе ног робота.

1.4 Расчёт параметров ног робота-шагохода при выполнении движения

Для расчёта углов поворота приводов, необходимо поставить задачу для расчёта, которая будет представлять вид координат точки, в которую необходимо поставить конечность ноги робота. Для выполнения этой задачи необходимо найти углы поворота каждой составляющей ноги шагохода, что в совокупности и даст точку координат, в которой будет установлена нога робота.

Следует так же разобрать понятие степени свободы ноги робота, под которым понимается количество приводов на одной ноге, способные менять положение ноги и давая ей возможности находиться в разных положениях.

На рисунке 5 можно видеть основные части, из которых состоит нога робота-шагохода имеющая три степени свободы [1].

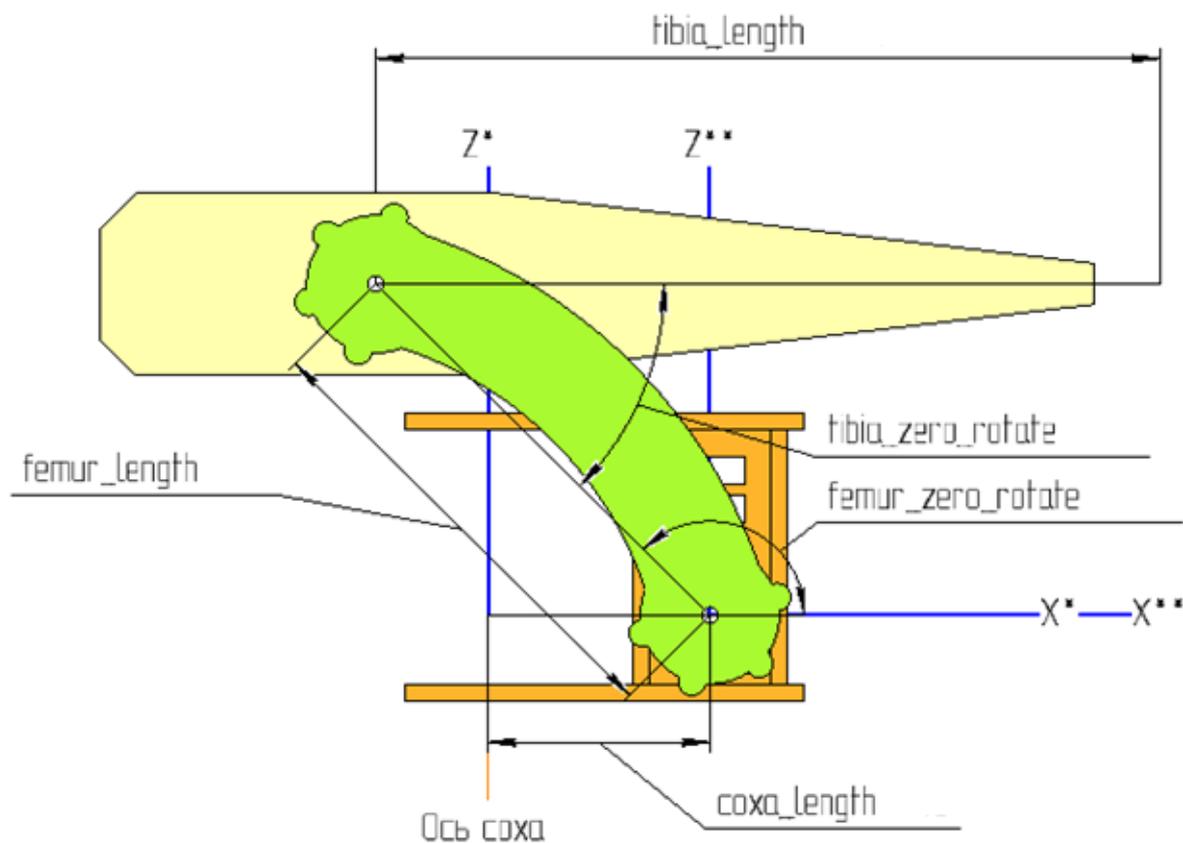


Рисунок 5 - Составные части ноги робота паука

Как можно видеть на рисунке 5 нога робота состоит из трёх основных частей: Соха, Femur и Tibia. Под этими терминами понимаются детали ноги, закрепляющиеся между собой приводами.

Соха является самым маленьким звеном ноги, и предназначена лишь для изменения положения ноги по горизонтали, что позволит роботу передвигать ноги вперёд и назад.

Оставшиеся две детали имеют роль передвижения ноги в даль, и установки ноги в конкретную точку.

От длины этих составляющих ноги, зависит максимальное расстояние, на которое робот может переставлять ноги, но не стоит забывать, что чем длиннее часть ноги, тем сильнее должен быть привод, управляющей этой частью ноги. [1]

Процесс поиска угла поворота приводов будет сдвигаться к нахождению необходимого угла поворота части Соха, после чего останется лишь найти угол поворота двух оставшихся приводов, путём нахождения углов треугольника, образованного частями Femur и Tibia.

Для начала будет разобран процесс поиска угла Соха. На рисунке 6 [1] показан пример, где необходимо найти угол системы координат привода и прямой проведённой к необходимой точке.

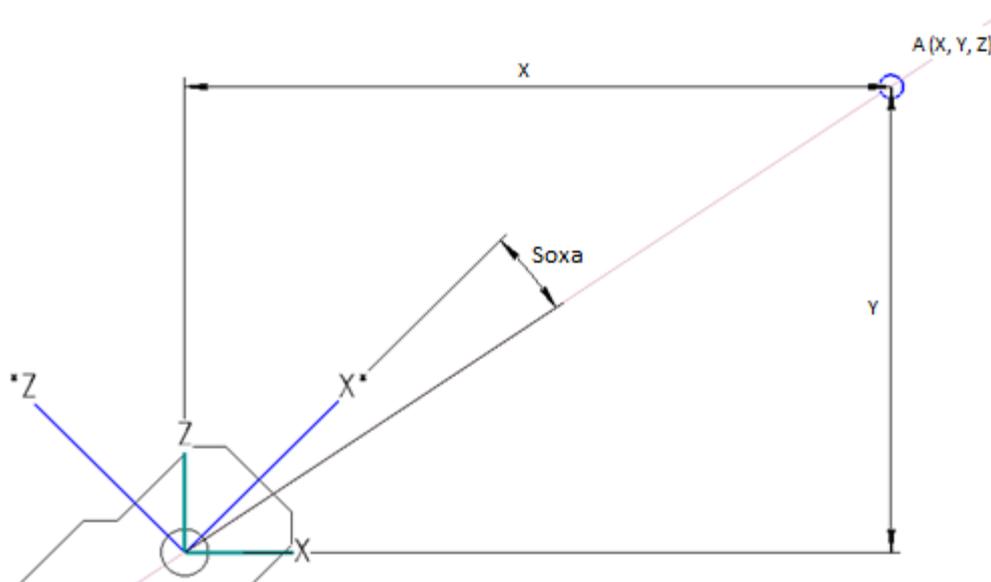


Рисунок 6 – Расчёт угла Соха

На рисунке 6 указаны две системы координат XZ и X'Z'. Система координат XZ представляет собой систему координат относительно которой удобно задавать точку, в которую необходимо поставить конечность ноги. Система координат X'Z' представляет систему в которой находится привод, управляемый частью Соха.

Для начала необходимо перевести точку, заданную в системе координат XZ, в систему координат X'Z'. Для этого рекомендуется использовать формулу 1, в которой ищется координата X, и формулу 2, в которой ищется координата Z. В двух этих формулах необходимо использовать угол поворота X'Z' относительно XZ [1].

$$x' = x \cdot \cos(a) + z \cdot \sin(a) \quad (1)$$

$$z' = -x \cdot \sin(a) + z \cdot \cos(a) \quad (2)$$

После того как были получены координаты искомой точки в системе координат $X'Z'$. Остаётся лишь высчитать арктангенс треугольника с катетами полученных значений. Пример формулы поиска угла показан в формуле 3 [1].

$$\text{Cоха} = \text{arctg2}(z, y) \quad (3)$$

Для поиска углов Femur и Tibia необходимо перейти к системе координат $X'Y'$, а после этого повернуть плоскость на только что найденный угол. Для этого необходимо лишь изменить значение координаты X , что можно сделать с помощью формулы 4 [1].

$$x' = x' \cdot \cos(a) + y' \cdot \sin(a) \quad (4)$$

После расчётов точки X' , так же необходимо вычесть из неё длину части Соха, для того, что б в координате $(0,0)$ находилась вершина образованного треугольника.

Таким образом из частей ноги Femur и Tibia образуется некоторый треугольник. На рисунке 7 можно видеть пример этого треугольника [1].

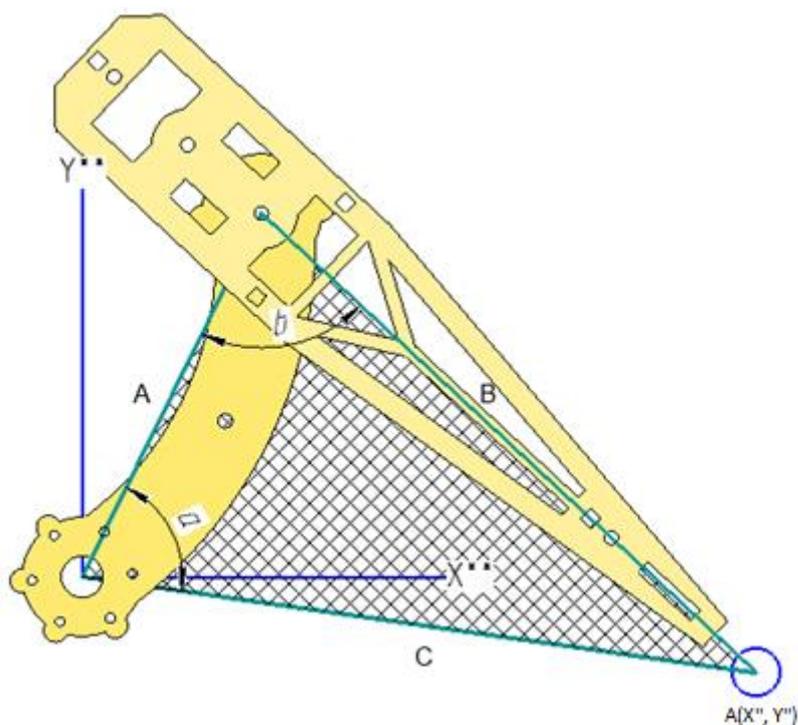


Рисунок 7 – Треугольник, образованный частями ноги Femur и Tibia

В результате перехода с плоскости $X'Y'$ на угол полученный ранее, была получена новая координатная плоскость

Стороны, образованного треугольника, A и B уже известны, и имеют значение длин частей ноги робота. Задача поиска углов поворота здесь сводится к поиску третьей стороны треугольника, её сравнению на досягаемость, и поиску углов треугольника.

Для нахождения стороны C , показанной на рисунке 7, необходимо представить эту сторону вектором, для нахождения которого необходимо представить прямоугольный треугольник с катетами, имеющими длину с значением координаты X'' и координаты Y'' . При этом нахождение данного вектора будет склонено к поиску гипотенузы в прямоугольном треугольнике по теореме Пифагора.

Формула для нахождения стороны C показана в формуле 5 [1].

$$C = \sqrt{X''^2 + Y''^2} \quad (5)$$

В результате будет найдена сторона C , которая также является и расстоянием от части ноги под названием *Coxa*, до точки куда необходимо поставить конец ноги. Поэтому прежде чем продолжить поиск, необходимо проверить, хватает ли длины *Femur* и *Tibia* для преодоления расстояния до необходимой точки.

Зная все стороны треугольника по формуле 6 можно найти угол a , а по формуле 7 будет находится угол b [1].

$$a = \arccos\left(\frac{A^2 + C^2 - B^2}{2AC}\right) \quad (6)$$

$$b = \arccos\left(\frac{B^2 + A^2 - C^2}{2BA}\right) \quad (7)$$

Теперь зная углы треугольника необходимо выполнить расчёт углов поворота приводов, зная начальный угол положения ноги робота, а также для этого необходимо знать угол наклона прямой C относительно оси X . Формула для этого показана в формуле 8 [1].

$$\varphi = \arctg2(X'', Y'') \quad (8)$$

Теперь производится расчёт углов поворотов приводов, по формуле 9 для части Femur, а по формуле 10 для Tibia [1].

$$FEMUR = femurZeroRotate - a - \varphi \quad (9)$$

$$TIBIA = b - tibiaZeroRotate \quad (10)$$

Таким образом при проектировании походки будет выполняться расчёт углов поворота приводов. Данный расчёт будет необходим при разработке программной части проекта [1].

1.3 Обзор существующих на рынке роботов

В этом пункте будут разобраны существующие решения по проектированию роботов-шагоходов под микроконтроллерным управлением.

Первым рассмотренным роботом шагоходом на базе микроконтроллера будет робот гексапод CR-6. Этот паук имеет 6 ног, в которых в общей сложности установлено 18 высокоточных сервоприводов, которые приводят робота в движение.

Управление роботом производится дистанционно, через мобильное приложение. Связь между роботом и мобильным устройством поддерживается с помощью беспроводного интерфейса Bluetooth. Само управление роботом производится через управление его направлением, подобно джойстику, только расположенному на пользовательском интерфейсе мобильного приложения.

Главной особенностью робота является возможность пользователя программирования микроконтроллера, создавая дополнительные программы взаимодействия с установленными на робота электронными модулями. Из установленных периферийных устройств на роботе можно выделить звуковой датчик, OLED-экран, датчик света, ультразвуковой дальномер.

Управление роботом гексаподом производится от контроллера Arduino nano, на плате которого установлен микроконтроллер Atmega 328p. Аккумуляторов, установленных на роботе, хватает в среднем на 90 минут непрерывной работы.

Ещё одна особенность рассматриваемого робота заключается в прочном корпусе гексапода, который выполнен твёрдого алюминиевого сплава и стекловолокна, который при увеличении прочности снижает вес корпуса робота. На рисунке 8 можно видеть внешний вид рассматриваемого робота [9].

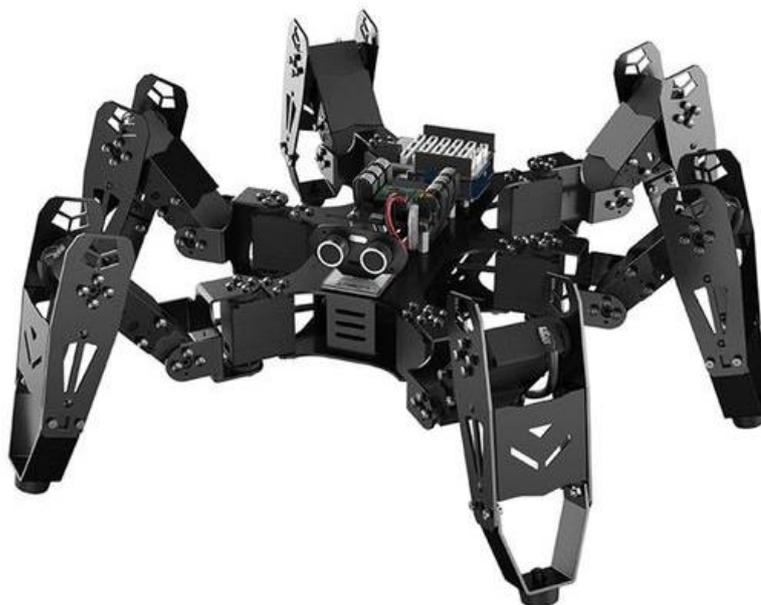


Рисунок 8 – Робот гексапод CR-6

Цена такого робота составляет примерно 20 тысяч рублей. На рисунке 7 можно видеть фото рассмотренного робота [9].

Далее будет рассмотрен бионический робот-шагоход SpiderPi, который представляет собой гексапод под управлением микропроцессора Raspberry Pi и имеет в своём составе видеокамеру и ультразвуковой дальномер. Основная особенность этого робота заключается в большой вычислительной мощности платы Raspberry Pi, которая даёт возможность роботу принимать и обрабатывать информацию с видеокамеры установленной на корпусе робота.

Гексапод имеет 6 ног, в каждой из которых установлено по 3 высокомоощных сервопривода, позволяющих роботу выполнять точные передвижения ногами.

Управление робота производится также через мобильное приложение, которое требует подключение по беспроводному интерфейсу Bluetooth.

Стоимость такого робота на данный момент составляет примерно 30 тысяч рублей. На рисунке 8 можно видеть фото этого робота [8]



Рисунок 9 – Фото бионического робота SpiderPi

На роботе установлена аккумуляторная батарея, способная поддерживать работу робота на протяжении 3 часов бесперебойной работы. Кроме того, для отображения заряда аккумуляторов на корпусе расположен индикатор заряда батарей.

Размеры робота шагохода при полном размахе ног составляет примерно 70 на 60 сантиметров, а в состоянии ходьбы высота корпуса будет составлять порядка 15 сантиметров.

Видеокамера, расположенная на корпусе и микроконтроллер, позволяют роботу автономно передвигаться, камера покажет, что находится вокруг робота [8].

Ещё один робота-шагоход под названием PhantomX AX-12 указан на рисунке 10, данный робот-шагоход стоит около 20 тысяч рублей, и построен на базе контроллера ArbotiX [7].

Данный робот построен на четырёх ногах, в отличие от ранее рассмотренных роботов. Каждая его нога имеет три степени свободы, что даёт ему большую свободу и красоту в движении.



Рисунок 10 – Робот-шагоход PhantomX AX-12

Управление данным роботом производится дистанционно через телефон, с помощью беспроводного интерфейса Bluetooth. Имеется режим автоматической походки. В этом режиме робот сам считывает окружающую местность и составляет свой маршрут.

Питается робот от трёх параллельно подключённых литий-ионных аккумуляторных батареи, с током 2100 mAh, чего ему хватит на 3 часа непрерывной работы.

Данный робот предназначен для изучения робототехники. На его основании можно строить свои системы, добавлять различные датчики и исполнительные устройства, для получения в итоге готовой системы. Необходимые материалы для этого идут в комплекте с роботом.

Размеры корпуса PhantomX AX-12 составляют примерно 150 x 150 мм без учёта размаха ног робота. Максимальная длина, на которую робот может поставить ногу составляет примерно 250 мм, при этом средний размер робота 400x400 мм. Корпус выполнен из чёрного пластика [7].

В таблице 1 представлено сравнение рассмотренных выше роботов.

Таблица 1 – Сравнение существующих решений

	CR-6	SpiderPi	PhantomX AX-12
Цена, руб	20 000	30 000	20 000
Размер корпуса, мм	250 x 250	250 x 250	150 x 150
Время работы на одной зарядке, ч	1.5	1.5	3
Управление	Дистанционно, Bluetooth	Дистанционно, Bluetooth	Дистанционно, Bluetooth
Размах ног, мм	200	200	250
Автоматический режим передвижения	-	+	+
Возможность доработки	+	-	+
Количество степеней свободы	3	3	3
Количество ног	6	6	4

Обзор рынка роботов-гексаподов позволил сформулировать основные требования к проектируемой системе.

1.4 Анализ требований

В разрабатываемом устройстве необходимо реализовать передвижения во всех направлениях, а так же выполнение некоторых движений на месте. Для управления направления движения робота было выбранно использовать беспроводной интерфейс Bluetooth, коммуницируемый устройство с мобильным приложением. Мобильное приложение выполняет передачу данных в зависимости от выбранного пользователем направления движения.

Питание робота должно осуществляться от аккумуляторов. Зарядка аккумуляторов производится от встроенной подсхемы питания, с использованием внешнего источника питания.

Требования, выделяемые к эксплуатации и хранению разрабатываемого устройства, заключаются в поддержании необходимой температуры, лежащей в диапазоне от -5 до +30 градусов, в случае хранения температура может достигать +40 градусов. Эксплуатация и хранение устройства в помещении с относительно влажностью воздуха больше 80% не

рекомендуется, так же значения влажности воздуха ниже 30% так же является нежелательным.

1.5 Выводы по разделу

В исследовательском разделе были изучены теоретические материалы по разработке роботов-шагоходов и принципу их работы. Рассмотрена классификация существующих роботов по количеству ног и типу походки, а также различные диаграммы, показывающие алгоритм и механику работы робота, такие как диаграмма походки и опорный многоугольник.

Рассмотрен алгоритм расчёта передвижения ног робота, в которых на основании необходимой точки и плоскости, в которой установлен сервопривод, рассчитываются углы поворота сервоприводов. Данный алгоритм расчёта будет полезен при написании алгоритма управляющей программы.

Рассмотрены существующие решения в области роботов-шагоходов со всеми их параметрами, и на их основании выдвинуты требования к разрабатываемому устройству.

2 КОНСТРУКТОРСКИЙ РАЗДЕЛ

2.1 Выбор инструментальной базы

На начальном этапе проектирования необходимо выбрать программные инструменты, являющиеся наиболее подходящими по функционалу, эффективности и удобству работы. Необходимо выбрать следующие инструменты:

- среда разработки программного кода для управляющего микроконтроллера;
- САПР для разработки деталей корпуса устройства;
- САПР для разработки принципиальной схемы устройства и печатной платы;
- программное обеспечение для симулирования работы микроконтроллера в схеме;
- среда разработки пользовательского приложения для ОС Android.

Для начала будет выполнен выбор среды разработки для написания управляющей программы. Для этого будет выполнен выбор среди сред: Micro chip, MPLAB, WinAVR.

К рассматриваемым средам разработки выдвигаются следующие требования: встроенный компилятор для C++, встроенный программатор, возможность отладки, возможность симулирования работы микроконтроллера, работа на Windows 10 и способ распространения.

Micro chip – среда разработки от компании AVR, которая раньше носила название Atmel Studio. Среда разработки позволяет писать код на языке программирования C++ или на Assembler. Имеется встроенный симулятор работы микроконтроллера, который может показать, что записано в регистры и память микроконтроллера, с учётом разных моделей.

MPLAB – среда разработки являющаяся набором программ для работы с кодом микроконтроллера. MPLAB позволяет писать и отлаживать код на C++ или Assembler, для этого имеется встроенный компилятор кода.

WinAVR – среда разработки для написания кода микроконтроллеров различных семейств. В данном программном продукте имеется возможность написания и отладки кода, без возможности отображения состояния регистров. Интерфейс программы наиболее прост в освоении и работе [18].

Сравнение рассмотренных сред разработки показано в таблице 2.

Таблица 2 Сравнение параметров сред разработки

	Micro chip	MPLAB	WinAVR
Встроенный компилятор для C++	+	+	+
Возможность отладки	+	+	+
Возможность симуляции	+	+	-
Свободно распространяемое ПО	+	+	+
Встроенный программатор	+	+	+
Поддержка работы на Windows 10	+	+	-

После анализа таблицы 2 была выбрана среда разработки micro chip, по причине того, что она наиболее подходит под требуемые задачи. MPLAB является более серьёзным и сложным в освоении инструментом. WinAVR не имеет симуляции, а также имеет проблемы при работе на Windows 10.

Следующей программой будет выбрана САПР для создания принципиальной схемы проектируемого устройства. В качестве рассматриваемых программ будут взяты EasyEDA, DipTrace и Kicad. К данным САПР выдвигаются следующие требования: Возможность разработки печатной платы по принципиальной схеме, автотрассировка, возможность добавления элементов пользователем, отображение печатной платы в формате 3D и возможность экспорта полученных схем в формате png.

EasyEDA – САПР для разработки принципиальной схемы, с возможностью перевода схемы в режим разработки печатной платы. Имеется возможность автотрассировки дорожек платы, готовый вариант можно посмотреть в 3D варианте плате. Имеется online версия программы, для которой требуется лишь авторизация, без необходимости скачивать приложение.

DipTrace – программный продукт позволяющий выполнять разработку принципиальных схем и печатных плат. Предоставляет возможность просмотра разработанной печатной платы в формате 3D модели, с установленными элементами. При работе имеется возможность использовать автотрассировщик.

Kicad – это САПР для разработки как принципиальной схемы, так и печатной платы различных устройств. В программе имеется возможность работы с различными существующими элементами схемы, а также добавления своих. САПР не поддерживает возможности автотрассировки, а также не может выводить полученную печатную плату в 3D форме.

В таблице 3 можно видеть сравнение рассматриваемых программ.

Таблица 3 – Сравнение программ для разработки принципиальной схемы

	EasyEDA	DipTrace	Kicad
Возможность добавления пользователями элементов схемы	+	+	+
Разработка печатной платы	+	+	+
Отображение 3D модели разработанной печатной платы	+	+	-
Возможность экспорта принципиальной схемы и схемы печатной платы в png	+	+	+
Возможность автотрассировки	+	+	-

В результате анализа таблицы 3, для разработки принципиальной схемы и печатной платы был выбран САПР EasyEDA, по причине того, что Kicad не удовлетворяет требованиям возможности автотрассировки и отображения печатной платы в 3D форме. DipTrace не имеет возможности работы в online режиме, это потребует времени на установку.

Ещё одна необходимая программа, используемая во время работы над проектом, это программа для симуляции электронных схем и проверки работы управляющей программы. Для этих целей будут рассмотрены такие программы как Proteus, Fritzing и SimulIDE.

Программа Proteus является большим программным продуктом для создания схем, их симуляции и конвертации в печатную плату. В программе имеется большое количество электронных элементов, что даёт возможность симулирования различных схем.

Fritzing – представляет собой не такой большой проект, а так же не имеет возможности конвертации схемы в печатную плату. В этом приложении имеется возможность симуляции работы микроконтроллеров, а также некоторых модулей, таких как дисплеи, клавиатуры, датчики и тому подобное.

Последняя рассматриваемая программа SimulIDE имеет простой интерфейс и направлена на работу с различными семействами микроконтроллеров, таких как AVR, PIC, Arduino. В приложении имеется встроенный редактор кода, который позволит быстро и удобно менять программный код микроконтроллера. Как объясняет сам автор проекта на главном сайте программы, проект является не доделанным до конца, а также по причине нацеленности на быстроту работы, может быть не точным в результатах симуляции [19].

В таблице 4 можно видеть сравнение параметров, рассматриваемых в качестве симулятора программ.

Таблица 4 – Сравнение программ для симуляции схемы

	Proteus	Fritzing	SimulIDE
Возможность замера напряжения на отдельных участках цепи при симуляции	+	+	+
Возможность загрузки в микроконтроллер Hex файла	+	+	-
Поддержка микроконтроллеров семейства AVR, PIC	+	+	+
Простой и удобный интерфейс	-	+	+
Возможность добавления новых элементов пользователем	+	+	-

После анализа таблицы 4, в качестве симулятора схемы с микроконтроллером, была выбрана программа Fritzing, по причине простоты интерфейса, а также удовлетворения необходимым требованиям. Proteus является более серьезным симулятором, с лишним функционалом и сложным интерфейсом. SimulIDE является не доработанной программой, что говорит сам автор программы на официальном сайте.

Для выбора САПР для разработки корпуса проектируемого устройства были выбраны следующие варианты: Компас 3D, AutoDesk, FreeCAD.

Компас 3D – САПР от российской компании Аскон, позволяющий выполнять разработку 3D моделей и их чертежей. В программе доступна параметризация и дерево операций, позволяющее отслеживать проделанные операции над проектом. Программа является платной, но имеется бесплатная версия для студентов.

AutoDesk – САПР для разработки 3D моделей и чертежей. Имеет импорт и экспорт большого количества расширений файлов, а также библиотек, расширяющих функционал программы.

FreeCAD – Программный продукт для работы с 3D моделированием. Программа позволяет работать с 3D моделями с удобным и понятным

пользовательским интерфейсом. Программа не является оптимизированной, и может показывать не стабильную работу [17].

Сравнительную характеристику рассматриваемых программ можно увидеть в таблице 5.

Таблица 5 - Сравнительная таблица САПР для разработки корпуса

	Компас 3D	AutoDesk	FreeCAD
Работа с 3D моделями	+	+	+
Автоматическое создание чертежей по 3D модели	+	+	-
Работа на Windows 10	+	+	+
Поддержка экспорта для печати на 3D принтере	+	+	+
Стабильность работы	+	+	-
Свободное ПО	+	-	+

После анализа сравнительной таблицы 5, в качестве САПР для создания 3D модели корпуса был выбран Компас 3D, по причине того, полного удовлетворения требований. AutoCAD также подходит по всем требованиям, но на время разработки, этот САПР не поддерживается в России, а также он является платным.

Для выбора среды разработки мобильного приложения на операционную систему android, будут выполнены сравнения между следующими программами: mit app inventor, intellij Idea, android studio. К ним выдвигаются следующие требования: простота интерфейса, простота работы в программе, встроенный компилятор и свободное распространение.

mit app inventor – это онлайн сервис для разработки мобильных приложений для операционной системы Android. Данный инструмент позволяет выполнять разработку приложений на android, через составление программы блоками, которые представляют собой некоторые действия. Загрузка приложения на мобильное устройство осуществляется через Интернет.

IntelliJ Idea является платным программным обеспечением, предназначенным для разработки мобильных приложений на различных языках программирования. Из плюсов можно отметить то, что приложение хорошо работает на слабых компьютерах и почти не загружают систему. Также приложение имеет очень удобный и понятный пользовательский интерфейс, вместе с удобными автозаполнителями, что очень облегчает работу. В приложении имеется встроенный компилятор.

Android studio является бесплатным приложением разработки от компании Google, которое в первую очередь направленно на разработку мобильного приложения на языке программирования Java. В приложении имеется встроенный компилятор. Имеется возможность просмотра дизайна получившегося приложения на симуляторе экрана мобильного устройства.

Сравнительную таблицу рассмотренных программ можно видеть в таблице 6.

Таблица 6 - Сравнение сред разработки мобильного приложения

	mit app inventor	IntelliJ Idea	Android studio
Простота интерфейса	+	-	-
Простота работы в программе	+	-	-
Встроенный компилятор	+	+	+
Свободное ПО	+	-	+

На основании сравнительной таблице, показанной в таблице 6, была выбрана среда разработки mit app inventor, по причине большей удобства в использовании этой среды разработки.

2.2 Выбор материальной базы проекта

В этом разделе необходимо выполнить выбор следующих элементов схемы устройства:

- микроконтроллер;
- сервоприводы;
- Bluetooth – модуль;

– дальномер.

Для начала будет выбран главный микроконтроллер. Из сравниваемых микроконтроллеров будут выбраны варианты с максимальным показателем объёма флеш-памяти, значение которой должно быть минимум 16 Кб, и тактовой частоты. К тому же микроконтроллер должен иметь минимум 15 линий ввода-вывода, и обладать встроенным интерфейсом UART.

Сравниваться будут микроконтроллеры ATmega328p, ATmega2560, ATmega32u4, ATmega168P, ATmega644. Для сравнения были выбраны эти микроконтроллеры по причине того, что они удовлетворяют требованиям по минимальному количеству линий ввода-вывода и интерфейсу UART. Поэтому далее эти микроконтроллеры будут сравниваться по тактовой частоте, объёму флеш-памяти и цене элемента. Сравнение перечисленных микроконтроллеров представлено в таблице 7 [14].

Таблица 7 – Сравнение микроконтроллеров

	Atmega328p	Atmega2560	Atmega32u4	ATmega168	ATmega644
Цена, р	150	600	300	250	500
Тактовая частота, МГц	16	16	16	20	20
Флеш-память, Кб	32	256	32	16	64
Линии ввода-вывода	20	54	20	20	32
UART	1	4	1	1	1

По приведённым характеристикам микроконтроллеров, указанных в таблице 6, микроконтроллер atmega2560 имеет избыточные характеристики, которые не являются необходимыми в проекте. Микроконтроллер Atmega328p, является наиболее подходящим, как по цене, так и по основным характеристикам.

Следующим элементом будут выбраны сервоприводы, используемые для передвижения ног робота. Если учитывать, что в проекте необходимо двенадцать сервоприводов, то главным критерием к сервоприводам является цена. Так же немаловажен крутящий момент, минимальное значение которого необходимо один килограмм на см². Максимальный угол поворота сервопривода необходим не менее 180 градусов.

Сравниваться будут сервоприводы с маркировкой SG-90, MG-90, DS3218, MG-995. В таблице 8 показаны критерии сравнения и характеристики выбранных моделей сервоприводов по этим критериям [15].

Таблица 8 – Сравнение сервоприводов

	SG-90	MG-90	DS3218	MG-995
Цена, р	130	300	730	335
Крутящий момент, кг на см	1.8	1.8	21	10
Максимальный угол поворота, град.	180	180	360	180

После анализа сравнительной таблицы, приведённой в таблице 7, для проекта был выбран сервопривод sg-90, по причине их дешевизны и удовлетворению всех требований.

От Bluetooth-модуля необходима поддержка работы режима Slave, возможность подключения по интерфейсу UART, а также необходимо выбрать более чувствительную антенну и большую дальность работы. Для сравнения были выбраны модули HC-05, HC-06, JDY-31, CC2541, CC2540. Цена модуля также является важным критерием выбора.

Интерфейс UART наиболее важен, по той причине, что подключение к микроконтроллеру планируется именно по этому интерфейсу, для удобного взаимодействия с модулем и общения с пользователем [16].

Сравнение выбранных Bluetooth модулей представлено в таблице 9.

Таблица 9 – Сравнение Bluetooth модулей

	HC-05	HC-06	JDY-31	CC2541	CC2540
Цена, р	250	230	400	220	250
Чувствительность, dBm	-80	-80	-97	-94	≤-84
UART	+	+	+	+	+
Поддержка Slave	+	+	+	+	+
Дальность работы, м	10	10	30	30	40

После сравнения Bluetooth модулей, был выбран модуль CC2541, по причине меньшей цены, а также большой дальности работы модуля и

высокого показателя чувствительности. У выбранного модуля имеется необходимый интерфейс подключения.

Ещё необходимо выполнить выбор элемента предназначенного для считывания расстояния до ближайшего препятствия. Среди таковых устройств рассматривается ультразвуковой дальномер hcSR04 и дальномер parallax 28015. Из требований к этим элементам робота можно выделить максимальное расстояние, которое должно быть не менее одного метра, напряжение питания, равное 5 вольт, шаг точности измерения и цена датчика. Сравнение выше указанных датчиков расстояния представлено в таблице 10 [20].

Таблица 10 – Сравнение датчиков расстояния

	hcSR04	Parallax 28015
Цена, руб	50	2500
Напряжение питания , В	5	5
Максимальное расстояние, м	4	3
Шаг точности, мм	3	1

После анализа таблицы 10, в качестве используемого датчика расстояния был выбран hcSR04, по причине меньшей стоимости, а также удовлетворению требований.

2.3 Разработка структурной и функциональной схем проекта

Для дальнейшей разработки устройства, необходимо создание функциональной схемы устройства, которая наглядно показывает функциональные блоки, выполняющие определённые функции, и их связь между собой.

На основании разработанной функциональной схеме можно будет выполнять дальнейшую разработку робота-шагохода.

Из функциональных блоков устройства можно отметить:

- блок, отвечающий за питание;

- блок микроконтроллера, который отвечает за обработку полученных данных и управление устройством;
- блок Bluetooth, отвечающий за приём данных из вне системы;
- исполнительный блок, который представляет собой набор сервоприводов, выполняющих механические движения работа;
- блок локатора, который выполняет считывание помещения, с целью получения расстояния до ближайшей преграды в некотором диапазоне.

На рисунке 11 показана полученная структурно-функциональная схема устройства.

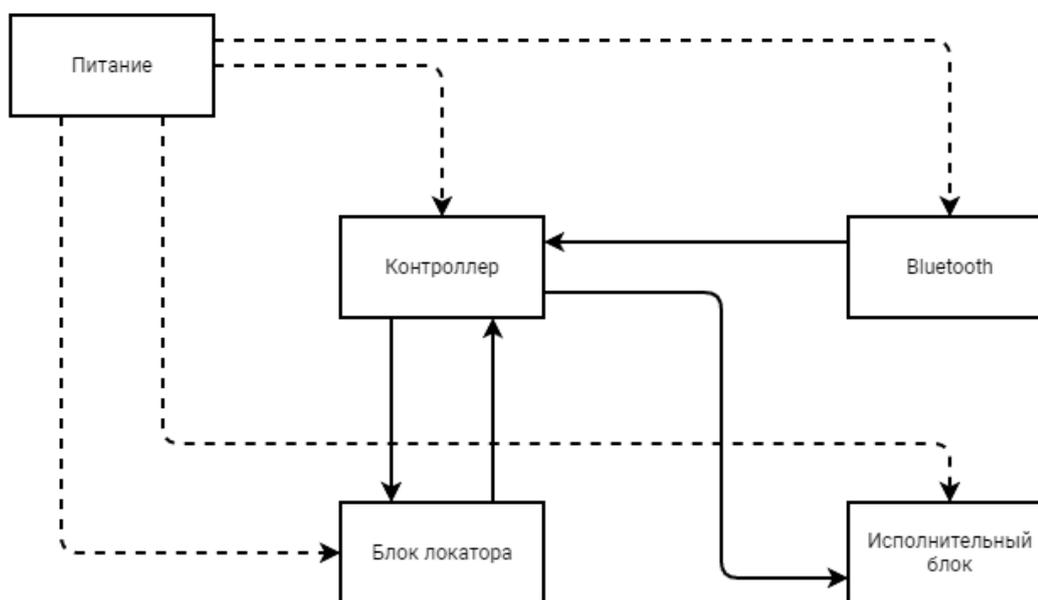


Рисунок 11 – Структурно-функциональная схема устройства

На рисунке 11 можно видеть расположенные функциональные блоки, а также связь между ними. Стрелочками указано направление сигнала.

Так, можно видеть, что контроллер будет получать данные от Bluetooth и блока локатора, а передача данных от контроллера будет осуществляться к блоку локатора, и исполнительный блок.

От блока, отвечающего за питание, исходит питание, для всех остальных функциональных блоков. Линии, отвечающие за питание обозначены на схеме пунктиром.

2.4 Разработка принципиальной схемы контроллера работа

После проработки связей основных элементов схемы между собой, можно приступить к разработке принципиальной схемы устройства.

На принципиальной схеме устройства показаны все элементы схемы, и связи между контактами каждого из них, а также, добавленные элементы схемы, не выполняемые никаких основных задач, но необходимых для обеспечения полноценной и качественной работы других элементов схемы, как например, для обеспечения стабильного питания или ограничения тока.

При разработке принципиальной схемы устройства некоторые элементы будут изображены в виде нескольких контактов, количество которых равно количеству контактов на изображаемом устройстве.

На рисунке 12 можно видеть пример полученной принципиальной схемы устройства. В приложении Д схема представлена в большом масштабе.

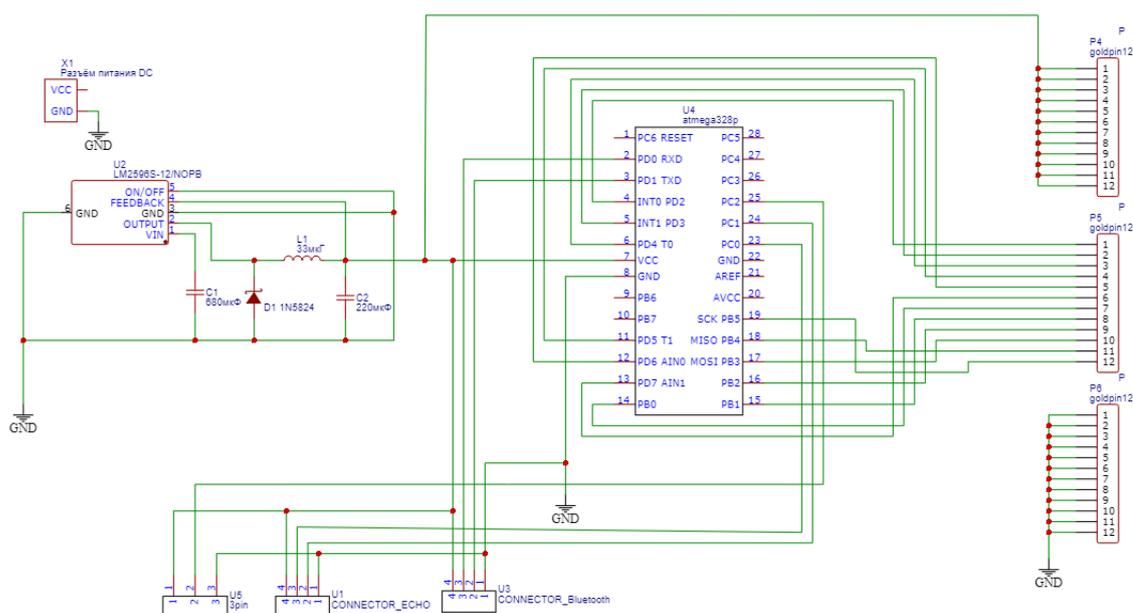


Рисунок 12 – Разработанная принципиальная схема устройства

В центре схемы, показанной на рисунке 12, можно видеть микроконтроллер Atmega328, к которому подключены все остальные элементы схемы.

Справа изображены три массива контактов, которые представляют собой контакты, идущие к сервоприводам, выполняющим движения робота.

Каждый массив контактов представляет собой определённую группу контактов, для питания, земли и передачи информации. Информационные контакты подключены к выводам микроконтроллера PD2-PD7, PB0-PB5.

Слева расположена схема, предназначенная для понижения напряжения питания. На вход данной схемы приходит 8 вольт с аккумуляторов, а после схемы напряжение понижается до 5 вольт.

Зарядка аккумуляторов производится через разъём питания X1, куда подаётся напряжение 12 вольт. К данному разъёму будут припаиваться провода, идущие к плате питания аккумуляторов. Напряжение, снятое с аккумуляторов, будет подаваться на питание схемы понижения напряжения.

Снизу принципиальной схемы можно видеть разъёмы для подключения Bluetooth модуля, ультразвукового дальномера и сервопривода, отвечающего за управление положением дальномера. Bluetooth модуль подключается к питанию 5 вольт и контактам интерфейса UART. Дальномер подключается к контактам PC0, PC1, а сервопривод к контакту PC2.

На полученной принципиальной схеме показано подключение каждого контакта элементов схемы между собой, что пригодится при разработке печатной платы устройства.

На схеме печатной платы устройства также важно показать связи между контактами элементов схемы, но в отличии от принципиальной схемы, расположение элементов на схеме имеют важную роль. При разработке печатной платы устройства, необходимо расположить элементы схемы, затратив как можно меньше места на плате. Кроме того, имеется возможности расположения элементов схемы на двух сторонах платы, но для этого необходимо выполнить соединение токопроводящих дорожек с разных сторон платы.

При создании токопроводящих дорожек на печатной плате необходимо выполнить дорожки, отвечающие за питание элементов устройства, с большей шириной, так как с помощью этого будет уменьшено сопротивление, что не даст упасть проходящему через дорожку току.

На рисунке 13 можно видеть полученную схему верхнего слоя печатной платы, а на рисунке 14 нижний слой печатной платы.

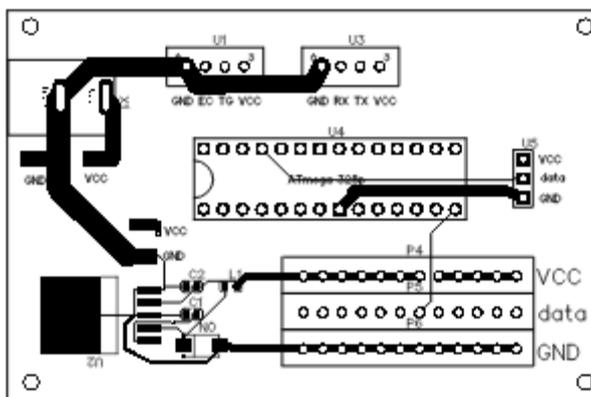


Рисунок 13 – Верхний слой печатной платы

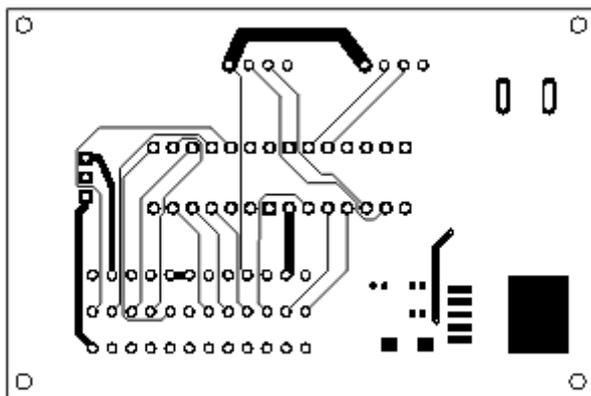


Рисунок 14 – Нижний слой печатной платы

Как можно видеть на рисунках 13 и 14, печатная плата представляет собой некоторую двухстороннюю поверхность, на которой расположены элементы схемы.

Стоит отметить, что на плате элементы, как правило, могут быть установлены двумя видами монтажа, среди которых поверхностный и двухсторонний монтаж элемента. Поверхностный монтаж элемента представляет собой монтаж элемента, не касаясь второй стороны платы и не требует выполнения отверстия на плате.

Некоторые токопроводящие дорожки на верхней стороне получившейся печатной платы, выполнены с большей шириной, для обеспечения питания элементов устройства.

На верхнем слое платы можно видеть микроконтроллер в DIP корпусе, а нижнего гребёнки контактов в 3 ряда, которые предназначены для удобного подключения 12 сервоприводов, для движения ног робота. Сверху находится 2 элемента, имеющих по 4 контакта. Они предназначены для подключения ультразвукового дальномера и Bluetooth модуля. Каждый контакт на плате подписан. Справа находится ещё один разъём, имеющий 3 контакта, и предназначенный для подключения сервопривода, для управления положением дальномера.

2.5 Разработка корпуса робота-шагохода

Корпус робота будет состоять из 5 деталей корпуса, среди которых: основание корпуса, крышка корпуса, часть ноги соха, часть ноги femur, часть ноги tibia. Часть ног будут повторяться 4 раза, так как они будут повторяться для каждой ноги.

Части ног разработаны с учётом посадочного места для сервоприводов sg-90, рядом с которыми имеются отверстия для крепления сервоприводов болтами, а также отверстия для проводки питания и управления сервоприводами.

В результате разработки корпуса устройства, длина части ноги соха составляет 32 мм, длина femur и tibia 48 и 85 мм соответственно. Длина частей ног робота пригодится при разработке походки робота и алгоритма управляющей программы.

На рисунках с 15 по 20 представлены эскизы деталей корпуса.

На рисунке 15 можно видеть эскиз основания корпуса, который представляет собой нижнюю часть корпусной коробки, к которой прикреплены посадочные места для сервоприводов. На основании корпуса предусмотрены отверстия для проводки проводов питания сервоприводов, находящихся на частях ног робота.

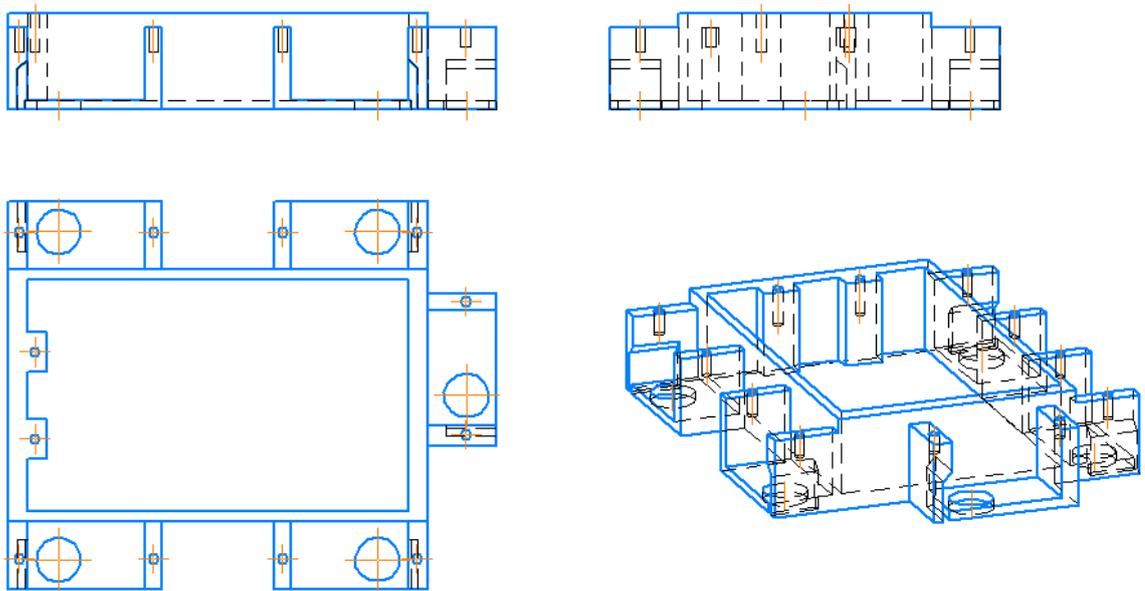


Рисунок 15 - Эскиз основания корпуса робота-шагохода

На рисунке 16 приведён эскиз крышки корпуса, эта деталь будет закрывать корпус, после установки в нём всех необходимых устройств.

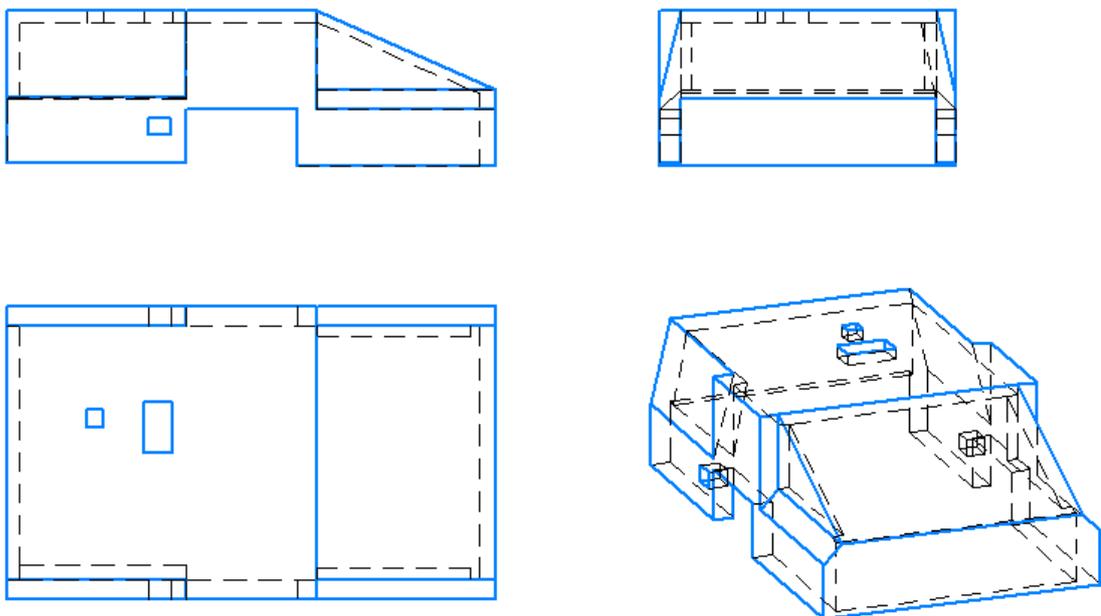


Рисунок 16 - Эскиз крышки корпуса

На рисунке 17 можно видеть эскиз части ноги робота Соха, которая крепится к сервоприводу, расположенному на основании корпуса, одним концом, а на другом конце крепится сервопривод, управляющий частью ноги Femur.

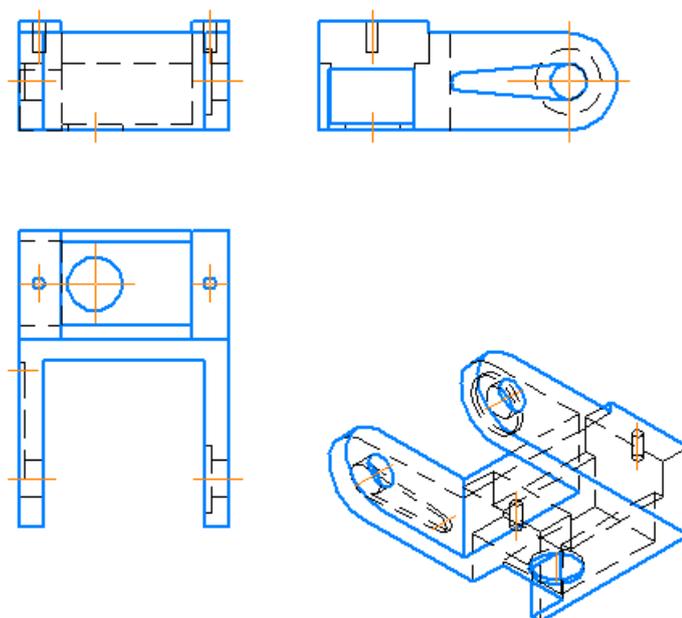


Рисунок 17 - Эскиз части ноги Соха

При креплении сервопривода, установленного на корпусе, его крыло будет крепиться за специальное крепление, расположенное на части ноги Соха, которое можно видеть на рисунке 17, на виде с боку. При этом, крыло сервопривода не будет выпадать, и тем самым терять управление частью ноги.

Следующей степенью свободу, будет являться сервопривод, установленный на части ноги Соха, для которого предусмотрено посадочное место и крепление, осуществляющееся двумя винтами М2.

Крепление второго сервопривода к части ноги Femur, будет осуществляться благодаря тому же способу, что и использовался при креплении первого сервопривода.

На рисунке 18 можно видеть эскиз части ноги Femur.

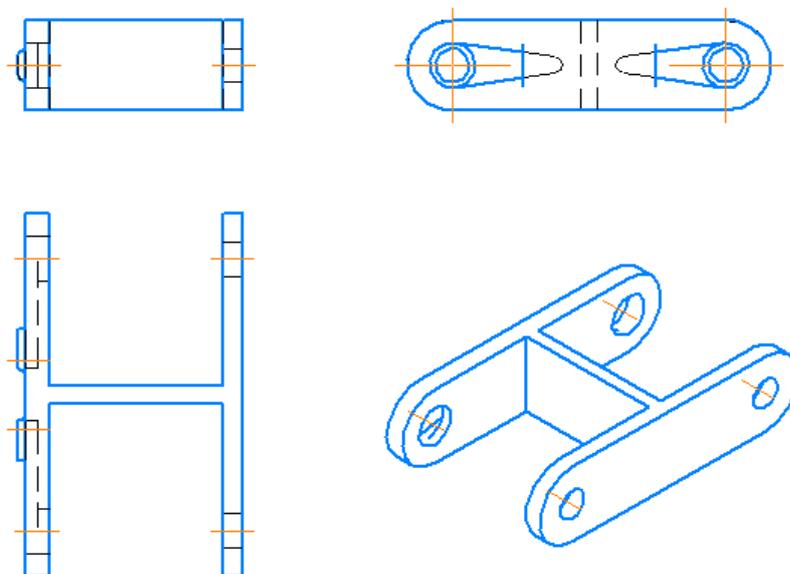


Рисунок 18 - Эскиз части ноги Femur

На части ноги Femur не будут расположены сервоприводы, здесь находятся лишь крепление для крепежа крыльев сервоприводов.

На рисунке 19 можно видеть эскиз части ноги Tibia.

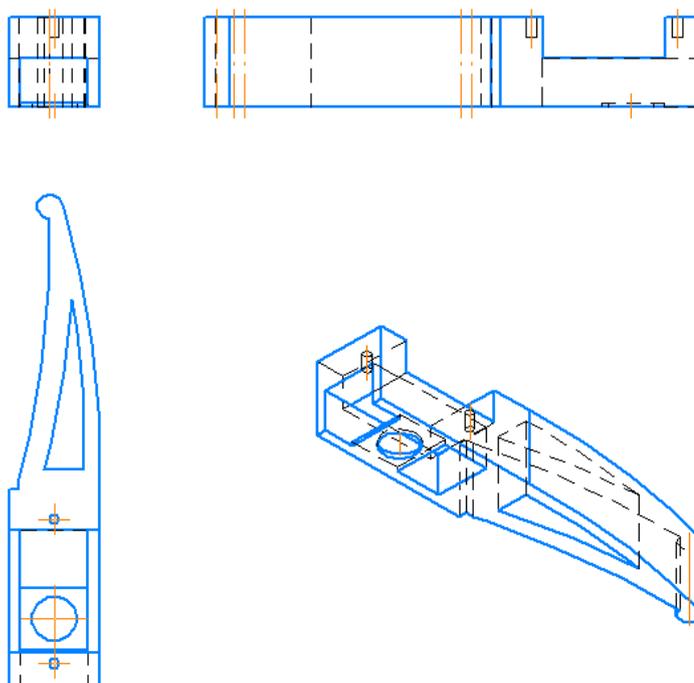


Рисунок 19 - Эскиз части ноги Tibia

Полные чертежи деталей представлены в приложении Г.

2.6 Разработка кинематической схемы движения конечностей робота-гексапода с учетом требований ТЗ

Для написания программного кода и выполнения движения роботом, необходимо выполнить разработку метода движения шагохода, расположение точек, в которые будут ставиться конечности ног, при выполнении движения во все стороны, а также алгоритм движения и перестановки ног, в который будет входить последовательность ног, которые будут переставляться при движении.

Для начала разработки походки робота будет выполнена разработка последовательности переставляемых ног, на которой в виде диаграммы будут изображены периоды, когда определённая нога будет не касаться поверхности, а выполнять передвижение. Такая диаграмма позволит понять временные интервалы, в течении которых робот будет выполнять один цикл передвижения.

На рисунке 20 можно видеть разработанную диаграмму последовательности передвижения ног робота шагохода.

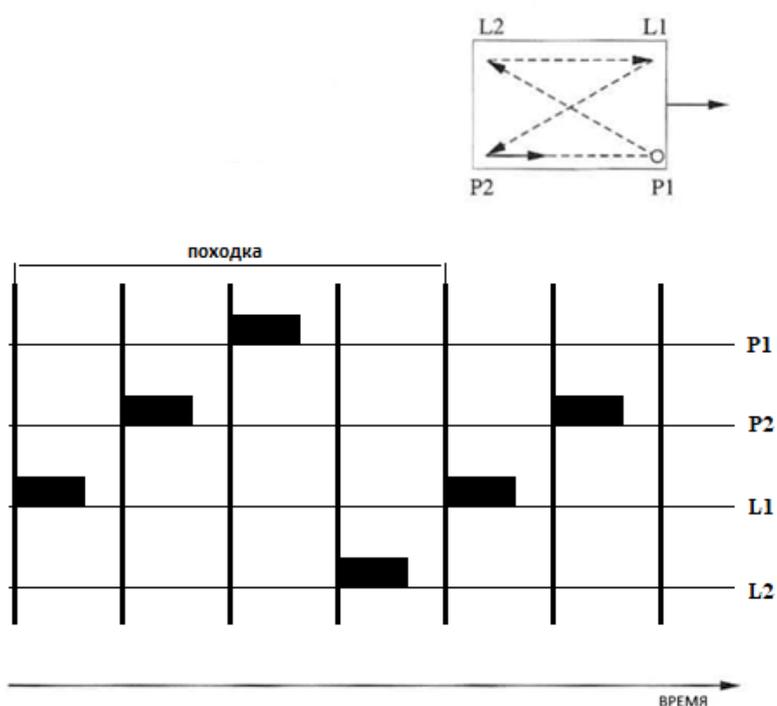


Рисунок 20 - Диаграмма последовательности передвижения ног робота

Как можно видеть на рисунке 20, последовательность передвижения ног проектируемого робота будет выполняться перекрёстно. Сначала будет передвигаться левая передняя нога, после чего будет некоторый перерыв, предназначенный для укрепления стабильности робота, этот перерыв будет длиться порядка 0,5 секунд. Данный перерыв будет осуществляться после каждого шага робота. После левой передней ноги, будет переставляться правая задняя нога, за ней правая передняя, и последней передвигаться будет левая задняя нога.

Данная походка будет цикличной, и будет всегда повторяться при движении робота в любом направлении.

Также при разработке и конструировании робота, будут проводиться работы с 12 сервоприводами, что может запутать при написании программного кода и разработке кинематики.

Для этого необходимо выполнить разработку схемы, показывающей расположение всех сервоприводов на корпусе устройства, а также их условный номер и контакт, к которому они подключены.

Следующим шагом в проектировании походки робота необходимо выполнить расчёт устойчивости походки робота, которая производится на основании опорного многоугольника. Для этого необходимо указать точки относительно корпуса устройства, на которые будет вставать робот конечностями ног.

На рисунке 21 можно видеть пример опорного многоугольника, в состоянии, когда робот стоит на месте, а на рисунке 22 многоугольник в состоянии, когда робота делает первый шаг походки.

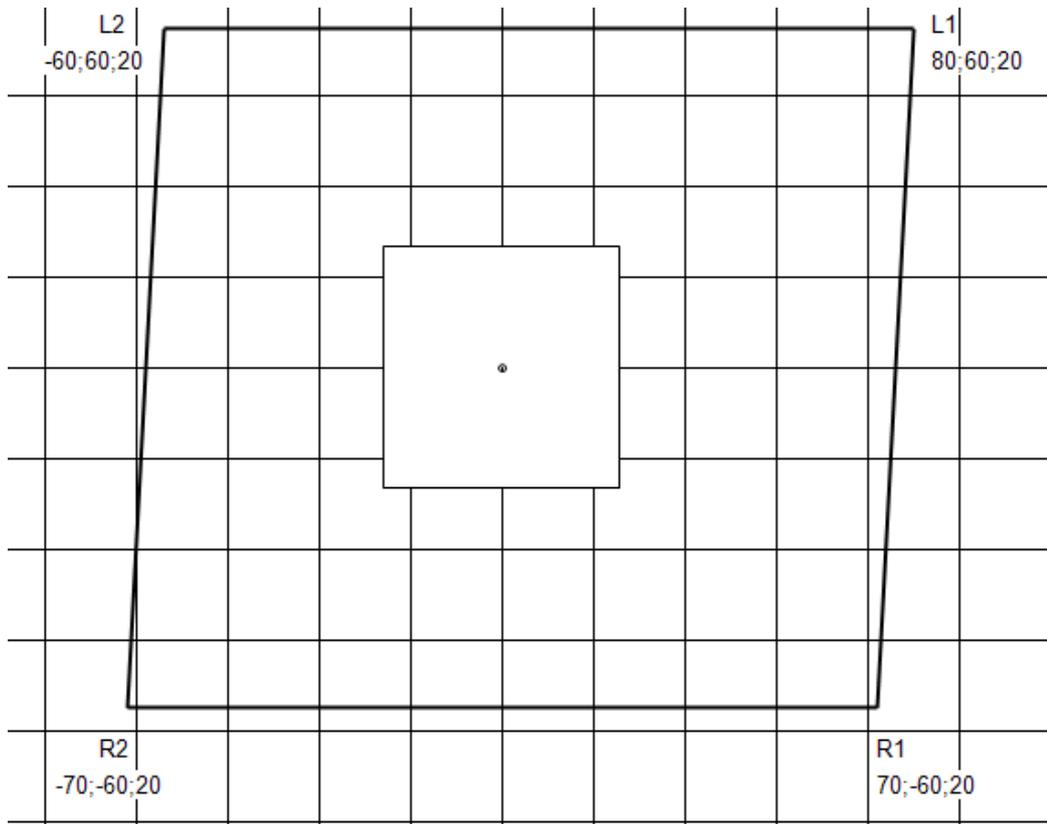


Рисунок 21 - Стойка робота шагохода

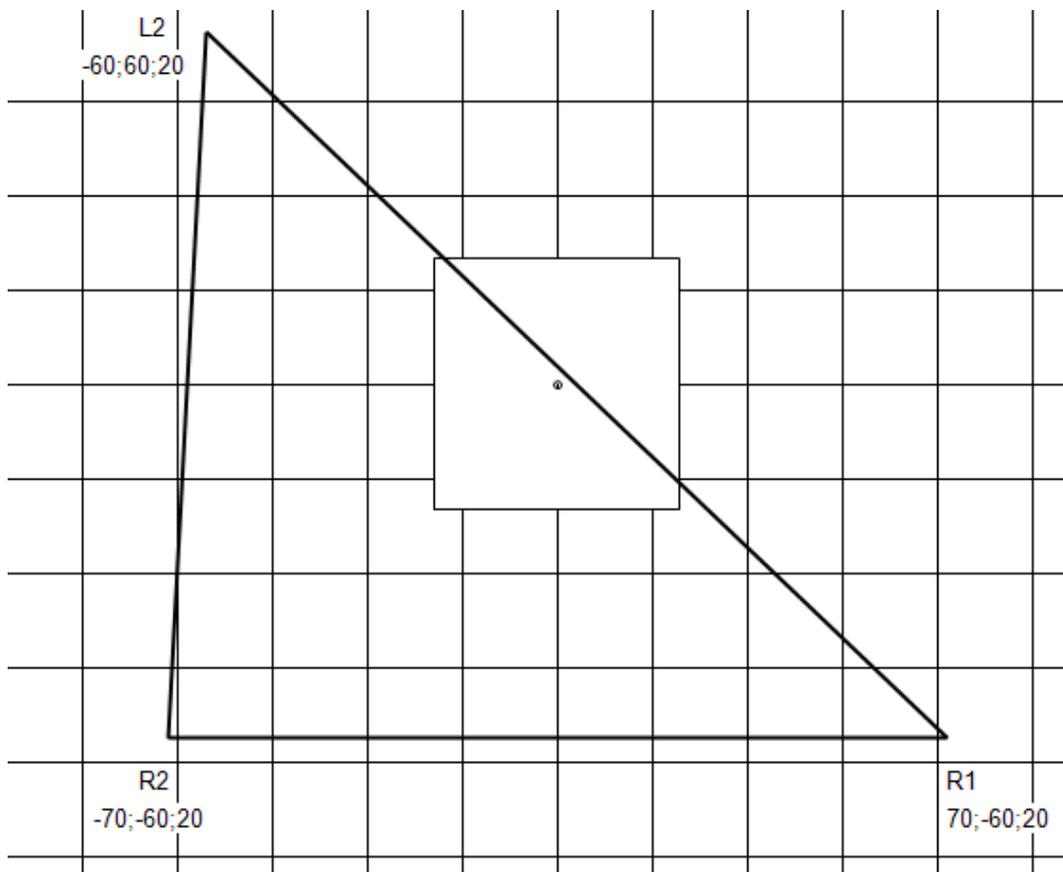


Рисунок 22 - Выполнение первого шага в походке

Как можно видеть из рисунка 21, стойка робота является слегка не ровной, так как это обеспечивает большую устойчивость дальше, во время движения робота. На рисунке 22 можно видеть, что при выполнении шага, центр тяжести находится далеко от границы опорного многоугольника, что обеспечивает устойчивость робота. На представленных рисунках имеется сетка, размеры клеток которой равны 25мм. Координаты каждой точки соприкосновения ноги робота с поверхностью представлены относительно угла, к которому прикреплена нога робота на корпусе.

Далее будут выполнены расчёты углов поворота сервоприводов на ноге робота-шагохода, для этого будут выбраны координаты точки, в которую необходимо поставить ногу робота. Для примера будет взята правая передняя нога робота-шагохода.

Угол, на который повёрнута система координат сервопривода, относительно системы координат, в которой представлена точка, составляет 45 градусов. На рисунке 23 приведена схема, на которой наглядно можно видеть искомый угол, и расположение систем координат.

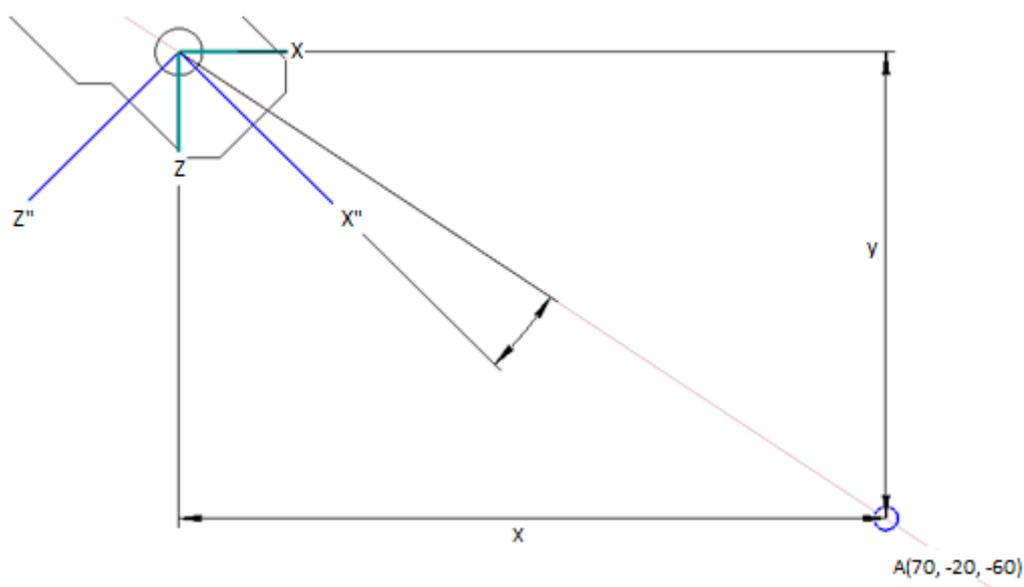


Рисунок 23 - Схема для поиска угла Соха

Для поиска угла Соха используются формулы 11 и 12, которые позволяют перевести точку из одной системы координат в другую.

$$x' = 70 \cdot \cos(45) - 60 \cdot \sin(45) = 91 \quad (11)$$

$$z' = -70 \cdot \sin(45) - 60 \cdot \cos(45) = 7 \quad (12)$$

Для поиска угла Соха необходимо вычислить арктангенс по полученным значениям координат точки в системе координат X''Z'', так как полученная точка, образует прямоугольный треугольник, с катетами равными значениям координат точки. Данные вычисления представлены в формуле 13.

$$\text{Соха} = \arctg2(-91, -7) = 4 \quad (13)$$

После получения угла Соха, известен угол для поворота сервопривода соответствующей части ноги, а также, имеется возможность перехода к соответствующей плоскости.

По формуле 14 можно перевести точку, в которую необходимо поставить ногу робота, из системы координат X''Z'' в систему координат X''Y''

$$x' = -91 \cdot \cos(4) - 20 \cdot \sin(4) = 92 \quad (14)$$

В новой плоскости X''Y'', координата Y не изменяется, поэтому её расчёты не являются нужными.

Для упрощения дальнейших расчётов будет вычтена длина части ноги Соха, так как её угол уже был найден и не будет участвовать в дальнейших расчётах. С учётом этого, координата X в системе координат X''Y'' будет уменьшена на 32. Данный расчёт представлен в формуле 15.

$$x = 92 - 32 = 60 \quad (15)$$

После перехода к плоскости X''Y'', был образован треугольник ABC, показанный на рисунке 24. Где стороны A и B - это стороны Femur и Tibia соответственно, и их длины являются известными.

Теперь задача поиска углов сводится к поиску третьей стороны треугольника, а также поиску всех углов треугольника, с их дальнейшим переводом для необходимых координат.

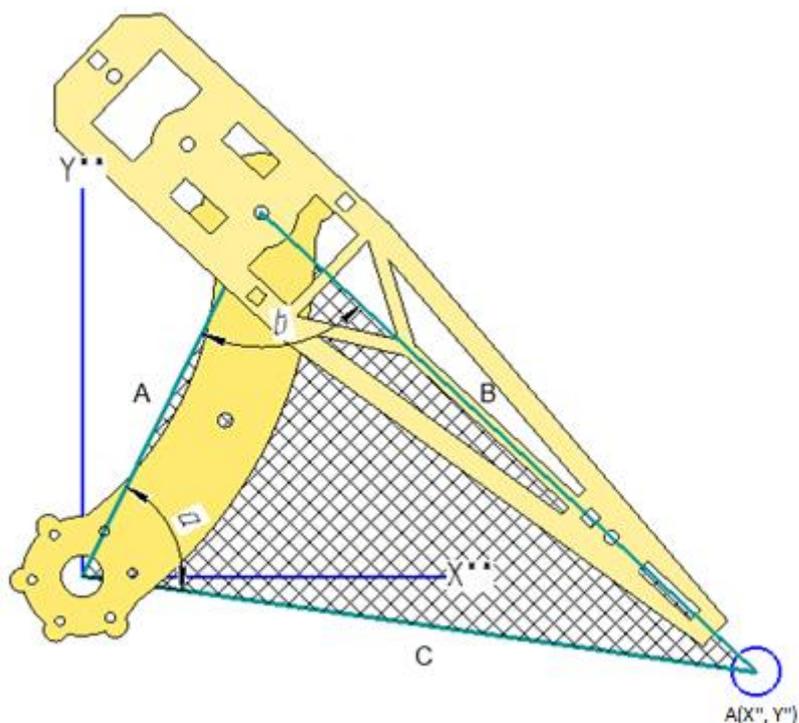


Рисунок 24 - Плоскость X''Y''

Для расчёта стороны C, треугольника, показанного на рисунке 24, можно использовать теорему Пифагора. Для этого можно представить, что искомая сторона является гипотенузой прямоугольного треугольника, образованного от длинного катета с длиной равной значению координаты X, и короткого катета равного значению координаты Y. Расчёт стороны C представлен в формуле 16.

$$C = \sqrt{X''^2 + Y''^2} = \sqrt{60^2 + -20^2} = \sqrt{3600 + 400} = 63 \quad (16)$$

После расчёта стороны C, можно найти углы треугольника a и b . Расчёты для этого указаны в формулах 17 и 18.

$$a = \arccos\left(\frac{48^2 + 63^2 - 85^2}{2 \cdot 48 \cdot 63}\right) = 99 \quad (17)$$

$$b = \arccos\left(\frac{85^2 + 48^2 - 63^2}{2 \cdot 85 \cdot 63}\right) = 58 \quad (18)$$

После расчёте углов треугольника, необходимо вычесть из угла a , угол наклона стороны C, относительно оси x. Необходимые расчёты представлены в формуле 19.

$$\varphi = \text{atan2}(60, 20) = 18 \quad (19)$$

Теперь можно приступить к расчёту непосредственно углов, на которые будут поворачиваться сервоприводы. Для этого, из максимального угла сервопривода необходимо вычесть найденный угол, и дополнительно вычесть угол наклона прямой C . Расчёты представлены в формулах 20 и 21.

$$Femur = 90 - a - \varphi = 90 - (99 - 18) = 9 \quad (20)$$

$$Tebia = 90 - (90 - b) \quad (21)$$

В формулах 20 и 21 значение 90, является углом, на который повернут сервопривод, при наведении на центральную ось. В случае если формулы будут давать отрицательное значение, необходимо будет добавить 90. Данное условие будет добавлено в программном коде.

2.7 Разработка управляющей программы робота

Для написания программного кода был выбран язык программирования Arduino Wiring, который очень похож на C++, имея лишь дополнительные функции, позволяющие сделать написание кода наиболее удобным.

Основной алгоритм программы представляет собой бесконечный цикл, в котором обрабатываются значения, поступающие с Bluetooth модуля по интерфейсу UART. Обрабатывая полученные данные микроконтроллер должен понимать, что необходимо выполнить, и затем выполняет необходимые функции.

Для написания программного кода необходимо выполнить разработку протокола обмена данными между мобильным устройством пользователя и Bluetooth модулем, который затем посылает данные микроконтроллеру.

В разрабатываемом протоколе необходимо указать закреплённые символы, которые будут передаваться между устройствами, а также функции, которые будут выполняться при получении микроконтроллером данной команды.

В таблице 11 можно видеть полученный протокол.

Таблица 11 – Протокол связи устройства с пользователем через Bluetooth

Кодовая комбинация символов	Действия
wa	Автоматическое/ручное управление
w1	Идти вперед
w2	Идти назад
w3	Поворот направо
w4	Поворот налево
w5	Встать/сесть

В бесконечном цикле программы, будут считываться данные с Bluetooth модуля, на основании которых, в переменную mode, отвечающую за режим работы робота-шагохода, будут записаны полученные данные, которые будут обозначать направление движения робота, или работает ли он в автоматическом режиме.

На рисунке 25 можно видеть блок-схему алгоритма работу программы для микроконтроллера.

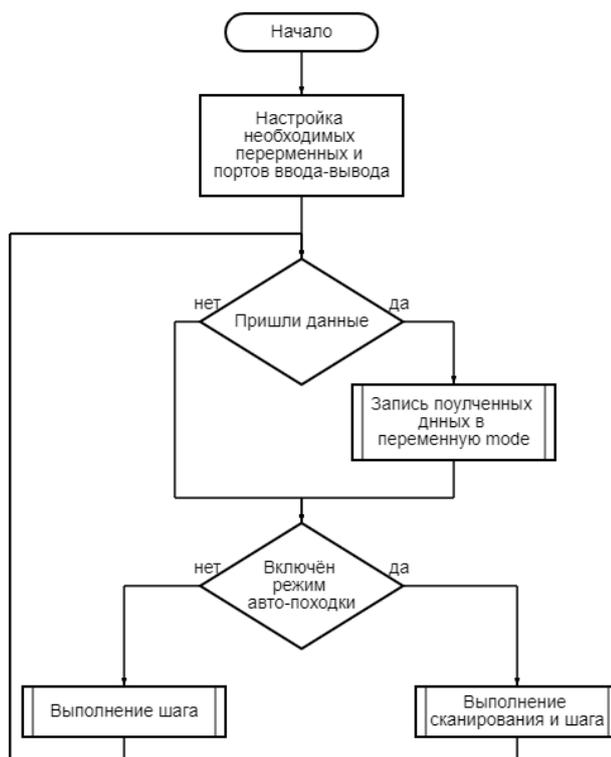


Рисунок 25 - Блок-схема алгоритма программы для микроконтроллера

После получение данных с Bluetooth модуля, выполняется движение робота-шагохода, на основании переменной mode. Переменная необходима

по той причине, что данные с Bluetooth модуля получаются и обрабатываются один раз, а движения, основанные на этих данных, выполняются неопределённое количество раз, по этой причине необходимо сохранить режим работы робота.

При выполнении шага роботом, на основании переменной mode, происходит выбор, из возможных вариантов её значений. При выполнении одного из условий выбора, выполняется соответствующая функция. Блок схема алгоритма выбора показана на рисунке 26.

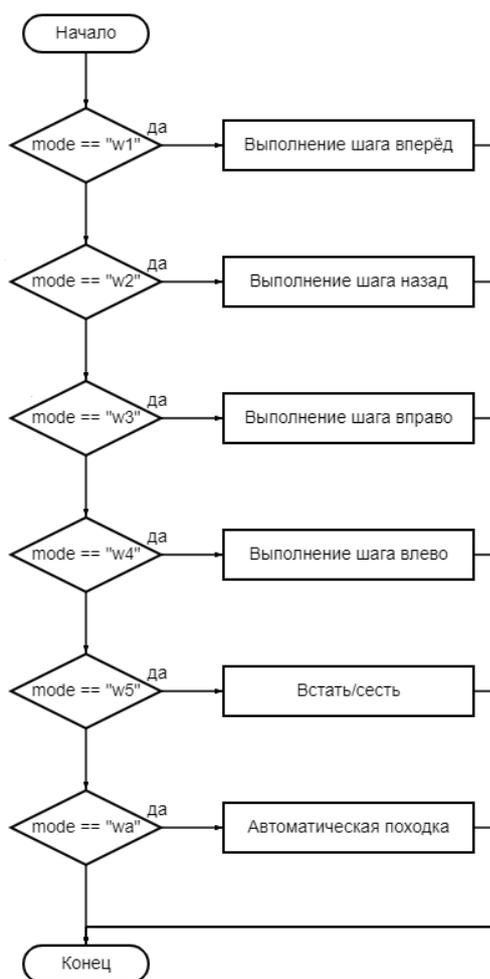


Рисунок 26 - Блок-схема алгоритма выбора движения

При выполнении автоматической походки робота-шагохода выполняется сканирование местности, в результате чего образуется массив с 180 значениями, значения каждого из элементов которого являются

расстоянием от робота до ближайшего препятствия с углом поворота дальногомера на угол номера элемента в массиве.

После получения данных массива, выполняется его обработка, и выполнение движения 3 раза, после чего все выполненные операции повторяются.

При обработке данных этого массива, он делится на 3 группы:

- 1 группа, от 0-го до 45-го элемента, направление налево,
- 2 группа, от 45-го до 135-го элемента, направление вперёд,
- 3 группа, от 135-го до 180-го элемента, направление направо.

Каждая из групп обозначает направление движения робота. С полученным группами выполняется 2 этапа обработки:

- поиск элемента, значение которого меньше 40,
- вычисление среднего арифметического значения группы.

В случае, если в одной из групп имеется хотя бы один элемент, значение которого меньше или равно 40, то данная группа не используется в дальнейшей обработке.

На втором этапе значения элементов в допущенных группах складывается между собой, и делится на количество элементов.

После проведения обработки массива, робот будет двигаться в направлении, которое обозначается группа, которая имеет максимальное среднее арифметическое значение, и не имеет элементов, значение которого меньше 40.

Блок-схема алгоритма обработки данных с локатора помещения представлен в приложении В.

Программный код представлен в приложении Б.

2.8 Разработка мобильного приложения

Разработка мобильного приложения будет осуществляться в среде разработки mit app inventor. Данная среда разработки имеет большое

количество инструментов для разработки мобильных приложений на операционной системе Android.

Для проекта необходимо выполнить разработку мобильного приложения, выполняющего подключения к Bluetooth модулю робота, а затем считывающего нажатия на кнопки приложения, и посылая соответствующие данные на само устройство с помощью беспроводного интерфейса Bluetooth.

Для обмена информацией между мобильным устройством и Bluetooth модулем, будет использоваться протокол, представленный в таблице 11.

При разработке приложения в Mit app inventor используется блочная система, процесс разработки в которой сводится к простановке блоков в определённой последовательности с некоторыми параметрами. Каждый блок представляет собой какое-либо действие.

На рисунке 27 можно видеть готовый интерфейс приложения, на котором можно видеть расположение элементов программы.



Рисунок 27 – Пользовательский интерфейс разрабатываемого мобильного приложения

На рисунке 27 можно видеть 4 кнопки, отображаемых в виде стрелок. Данные кнопки управляют направлением передвижения робота. Красная кнопка по центру отправляет команду на устройство о том, что робот-шагоход должен встать или сесть, в зависимости от того, в каком он сейчас находится положении.

Большая серая кнопка предназначена для выбора найденных Bluetooth точек доступа, и подключения к одному из них.

При постановке галочки в поле Автоматическая походка, робот выполняет передвижения сам, в зависимости от данных с локаатора помещения. При этом режим робот не реагирует на нажатия остальных кнопок, до тех пор, пока режим автоматической походки не будет выключен.

Управляющая программа, разработанная через блоковый интерфейс, представленная на рисунке 28.

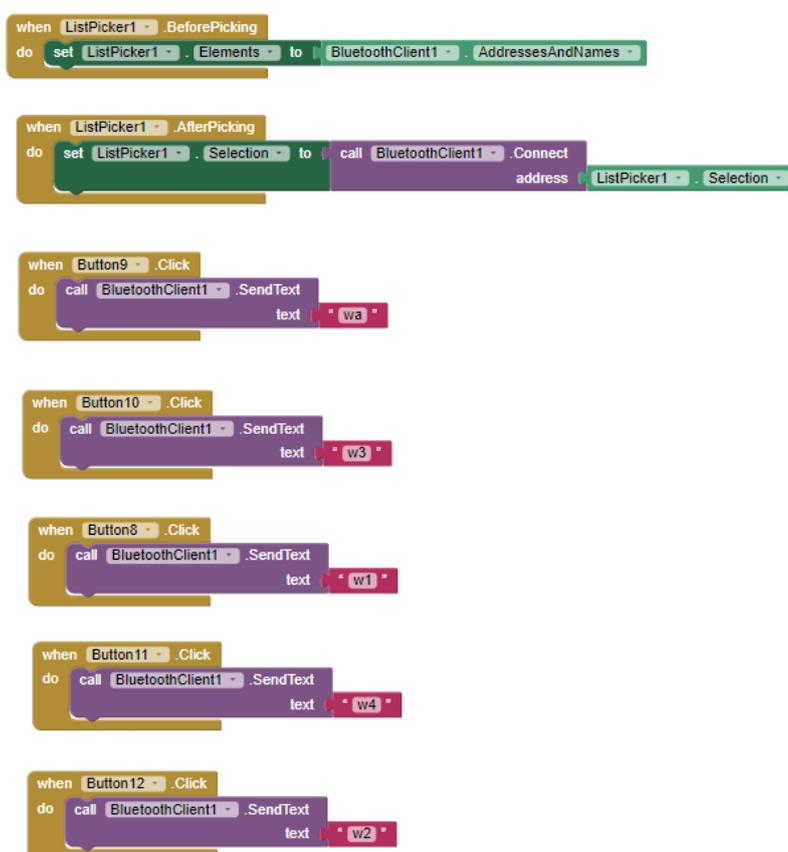


Рисунок 28 – Программа мобильного приложения

Первые два блока, представляют собой управление подключением к Bluetooth устройству, через список найденных устройств. Остальные блоки выполняют отправку строк на мобильное приложение при нажатии на определённую кнопку.

2.9 Вывод по разделу

В конструкторском разделе был выполнен подбор инструментальной базы проекта. Были выбраны программы: среда разработки программы микроконтроллера, среда разработки мобильного приложения, САПР для создания корпуса устройства, САПР для разработки принципиальной схемы устройства и печатной платы, симулятор для проверки работы устройства.

Выполнен выбор элементной базы проекта, в результате которого были выбраны: микроконтроллер atmega328p, Bluetooth модуль hc – 05, сервоприводы sg – 90, датчик расстояния sr – hc04.

Разработана структурно-функциональная схема, показывающая связи между функциональными блоками, их тип и направление. Были описаны задачи каждого из функциональных блоков.

Разработана принципиальная схема устройства, показывающая связи каждого отдельного элемента схемы, с описанием его задачи.

Разработана двухсторонняя печатная плата, на которой расположены элементы схемы устройства с связью, описанной при создании принципиальной схемы. Компоненты устройства расположены в наиболее подходящих для удобства местах платы. Токопроводящие дорожки расположены с размером в зависимости от их назначения.

Разработан корпус устройства и схема походки робота, в результате чего были созданы диаграммы походки, опорные многоугольники, эскизы и чертежи деталей корпуса.

Разработана управляющая программа для микроконтроллера, для чего были описаны задачи, возложенные на управляющую программу, после чего разработаны блок-схему алгоритма и написан программный код.

Разработано мобильное приложение для управления роботом, выполняющее передачу некоторых кодовых символов по беспроводному интерфейсу Bluetooth, на основании которых и будет выбираться действия робота.

3 ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ

3.1 Утилизация средств вычислительной техники и оргтехники

Утилизация не пригодной для работы вычислительной техники осуществляется по выпущенным законам и нормативам. Отработанную вычислительную технику нельзя просто выкинуть, так как это создаст как минимум 2 проблемы:

– в вычислительной технике содержится большое количество вредных и опасных элементов, которые при большом складировании могут представлять опасность как для здоровья человека, так и для экологии в целом. По этой причине не допускается складирование отработанной техники в большом объёме, а требуется переработка устройства, с целью обезвреживания элементов, и дальнейшая переработка, с целью использования материала повторно в производстве;

– в вычислительной технике имеется большое количество драгоценных металлов и камней. Для обеспечения большей эффективности работы вычислительных устройств, в них используются такие металлы, как золото, серебро, палладий и многие другие металлы, которые имеют большую стоимость. В этом случае правильная утилизация и переработка устройства выполняется с целью использования драгоценных металлов и камней повторно в процессе разработки вычислительной техники, или для отправки полученных средств в Госфонд. Данный процесс носит название аффинаж [13].

В большинстве случаев утилизация вычислительных устройств обеспечивается двумя нормативными документами и законами:

- N 89-ФЗ от 24.06.1998 – «Об отходах производства и потребления»;
- N 41-ФЗ от 26.03.1998 – «О драгоценных металлах и драгоценных камнях».

Утилизацию вычислительных устройств могут выполнять лишь организации и предприятия, имеющие лицензию, позволяющую им

выполнять соответствующие работы, выполняя при этом уплату налогов, с учётом того, что организация добывает драгоценные металлы из вычислительной техники.

На основании 8.2 статьи административного правового кодекса, обязательными для утилизации являются:

- компьютеры, мониторы, сканеры, ксероксы, принтеры и картриджи для них;
- медицинская техника;
- различные виды аккумуляторов;
- бытовые приборы;
- люминесцентные лампы.

В процессы утилизации работники утилизирующего предприятия разбирают устройство, извлекая из него компоненты, содержащие драгоценные металлы, для их дальнейшей отдельной переработке и утилизации. После этого из устройства извлекаются элементы содержащие химически опасные вещества для здоровья и экологии. Из таких опасных веществ можно выделить вещества являющиеся:

- токсичными и инфицирующими;
- огнеопасными и легковоспламеняющимися, это вещества, которые при определённых условиях, могут легко воспламениться;
- органическими, это вещества, которые имеют ускоренное разложение, и могут быть потенциально опасными для экологии.

Описанные выше вещества представляют опасность для здоровья и экологии [13].

3.2 Анализ вредных и опасных производственных факторов на рабочем месте.

Для полного анализа вредных факторов, которым подвергается рабочий в производственной среде, необходимо рассмотреть некоторые термины и понятия, имеющиеся в этой сфере.

Главным понятием охраны труда и безопасности является понятие опасности. Опасность – это некоторое свойство рабочей или иной другой среды, которое при воздействии на человека, может грозить ему потерей здоровья. Потеря здоровья может выражаться как слабым заболеванием, так и некоторым продолжительным заболеванием, способным сделать человека инвалидом, а в некоторых случаях это может привести и к летальному исходу работника подверженного опасности на производстве.

В предмете безопасной жизнедеятельности есть аксиома, согласно которой все опасности носят потенциальный характер. Это приводит к тому, что вся деятельность человека является потенциально опасной.

Деятельность рабочего на производстве проходит в сфере, являющейся пересечением сферы обитания человека и сферы повышенной опасности, которая присутствует на производстве, по причине необходимости работы на станках и других подобных тому машинах, носящих потенциальную опасность для рабочего. Взаимодействие человека с сферой повышенной опасности, с целью получения производственного результата, называется трудовой деятельностью.

Для изучения способом улучшения условий труда, минимизации влияния опасных факторов на организм человека и экономии сил рабочего была создана наука под названием эргономика. В этой науке используются некоторые термины, описывающие определённые зоны, в которых работает и находится рабочий [10].

Рабочее место – это закреплённое за работником место, на котором он проводит большую часть работы. При этом он может периодически покидать это место, если на то есть причины.

К постоянному рабочему месту относится рабочее место, на котором работник проводит большую часть рабочего времени. Обычно за такое место принято считать место, где работник проводит 50% рабочего времени, или 2 часа непрерывной работы.

Формирование безопасности и удобства работы на рабочем месте обеспечивается правильной оценкой таких параметров как условия труда, тяжесть и напряжённость труда.

Условия труда – это совокупность всех влияющих факторов на рабочего в процессе производства. Другими словами, это удобство, созданные, для рабочего, правильное освещение, температурный режим и многое другое.

К тяжести труда относится нагрузка на опорно-двигательную и мышечную систему человека, в ходе которых человек получает физическую усталость. В случае неправильной оценки этого параметра рабочий будет сильно и быстро уставать, выполняя много сложной физической нагрузки.

Напряжённость труда оценивает не физическую, а психологическую нагрузку на рабочего. Также, как и в случае с тяжестью, в случае не правильных норм напряжённости труда, рабочий будет сильно уставать психологически. К вредным факторам напряжённости труда можно отнести монотонную и рутинную работу, которая способна уменьшить внимательность и активность рабочего во время производственного процесса [10].

На производстве основными факторами, влияющими на рабочего, являются опасные и вредные факторы.

Опасный фактор – это фактор, который при воздействии на человека может привести к травме или заболеванию. Обычно опасным фактором является несчастный случай на производстве.

Вредным фактором является такой фактор, который при долгом воздействии на человека может привести к заболеванию или ухудшению состояния здоровья рабочего. Этот фактор может быть выражен плохим освещением, что в последствии может ухудшить зрение работника, высоким уровнем шума или низкой температурой в помещении.

При долгой продолжительности воздействия вредных факторов, его воздействие может быть подобным опасному фактору, это зависит от продолжительности и интенсивности воздействия вредного фактора.

Для объединения этих факторов был принят термин негативного фактора, который содержит в себе понятия опасного и вредного фактора.

Все негативные факторы делятся на 4 группы, в зависимости от их способа воздействия на организм человека. Среди этих групп наблюдаются физические, химические, биологические и психологические факторы воздействия на человека. При работе инженер-электронщик подвергается 3 из них, в которые входят все перечисленные, за исключением биологического негативного фактора [10].

Физические негативные факторы – это факторы, взаимодействие на организм человека которых связано с физическим воздействием. От воздействия таких факторов зачастую остаются раны, ожоги или переломы. Кроме того, к этим факторам также относятся и различные производственные загрязнения воздуха, такими веществами, например, как тонер, пыль и тому подобные. Физические опасные факторы являются наиболее распространёнными, и встречаются на всех производствах. Инженер-электронщик на работе подвергается таким физическим негативным факторам как плохое освещение, электромагнитное излучение, повышение температуры, замыкание электрического тока.

Химические негативные факторы – это факторы, чаще встречающиеся на производствах, где проводятся работы с химически опасными веществами. От воздействия этих факторов работник может получить различные химические отравления, отравления химически заражённым воздухом или иные другие опасные последствия для организма человека. На рабочем месте инженер-электронщик подвергается химически негативным факторам в виде опасных паров от расплавленного олова.

Последняя группа психологических негативных факторов содержит в себе факторы, опасно влияющие на психологическое состояние и здоровье

человека. Такую опасность можно встретить на любом предприятии, а опасность создаётся плохими условиями и напряжённостью труда. Наиболее часто инженер-электронщик подвергается этому негативному фактору, по причине монотонности и однообразности работы. Также напряжённость работы может быть увеличена при низком освещении, которое также относится и к физическим негативным факторам. При плохом освещении работнику приходится напрягаться, разглядывая мелкие элементы, что приводит к повышенной напряженности работы, что в следствии даёт усталость работника [10].

3.3 Мероприятия по оптимизации освещенности на рабочем месте

Одним из вредных и опасных факторов на рабочем месте может быть плохое освещение рабочего места. К этому негативному фактору можно отнести как недостаток, так и избыток освещения.

Недостаток освещения может привести как к ухудшению зрения работника, так и к психологически вредному фактору, так как при большой нагрузке на глаза, работник сильно устаёт от нагрузки на глаза.

При избытке освещения также возможно ухудшение зрения, вызванное ослеплением работника, а также напряжённость глаз работника, так как при этом глаза всё также будут напрягаться, что приведёт к лишней нагрузке на глаза.

Из возможных последствий можно видеть, что плохое освещение является как физическим, так и психологическим негативным фактором.

При работе инженера электронщика необходимо что бы на рабочем месте инженера-электронщика, было от 300 до 500 Лк, в области окружающей рабочее место, должно быть от 200 до 300 Лк.

При этом, коэффициент пульсаций освещения на рабочем месте не должно превышать 10%. При превышении этого значения, освещение может напоминать эффект стробоскопа, что будет негативно сказываться на работе и зрении работника [11].

На рабочем месте важно не только какое освещение, но также и то, какой коэффициент отражения света у поверхностей пола, потолка, стен и других окружающих поверхностей помещения. Необходимые коэффициенты отражения:

- для потолков - от 0,7 до 0,9;
- для стен - от 0,5 до 0,8;
- для рабочих поверхностей - от 0,2 до 0,7;
- для пола - от 0,2 до 0,4 [12].

В случае если в помещении недостаточно освещения, необходимо установить источники дополнительного освещения, на расстоянии 0.5 - 0.8 м от рабочего места. Свет должен быть направлен на рабочую поверхность. В случае несоблюдения нормативных коэффициентов отражения света от окружающих поверхностей, возможно уменьшение или увеличение установленного освещения. Также с целью увеличения освещения на рабочем месте на малое количество, возможна смена окружающих поверхностей на более яркие, что также приведёт к увеличению освещения.

В случае если в помещении установлен избыток освещения, необходимо отдалить источники освещения от рабочего места, или заменить их на менее мощные. Допускается и вариант смены окружающих поверхностей на более тёмные, что позволит уменьшить общее освещение в помещении.

3.4 Вывод по разделу

В разделе охрана труда и промышленная экология были рассмотрены темы:

- утилизация отходов вычислительной техники;
- вредные и опасные факторы на производстве;
- мероприятия по оптимизации освещенности на рабочем месте.

В подразделе по утилизации отходов ВТ полученной в результате разработки робота-шагохода. По итогам этого подраздела было выявлено,

что при утилизации отходов ВТ необходимо пользоваться услугами специальных организаций, выполняющих утилизацию, в соответствии с всеми нормами и правилами. Утилизация необходима по причине минимизации ущерба экологии от отходов ВТ, а также по причине извлечения драгоценных металлов из старых компонентов ВТ.

В подразделе анализа вредных и опасных факторов были рассмотрены сами понятия вредных и опасных факторов на производстве, а также понятия, близко связанные с безопасностью труда. Рассмотрена классификация негативных факторов, с подробным рассмотрением каждого пункта и приведением примером негативных факторов.

В подразделе оптимизации освещенности на рабочем месте описаны способы и инструменты оптимизации вредного и опасного физического фактора, влияющего на зрение.

4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ

4.1 Расчет затрат на разработку программы для микроконтроллера

Затраты на разработку программы микроконтроллера зависят от: затрат на программные инструменты, начисления заработной платы для работника, начисления во внебюджетный фонд оплаты труда работника, и некоторые другие составляющие затрат. По формуле 22 можно рассчитать общую сумму затрат на разработку программы для микроконтроллера.

$$Z_{РПР} = Z_{ФОТР} + Z_{ОВФ} + Z_{ЭВМ} + Z_{СПП} + Z_{ХОН} + P_{Н}, \quad (22)$$

В формуле 22 используются следующие обозначения:

$Z_{ФОТР}$ – общий фонд оплаты труда разработчиков программы,

$Z_{ОВФ}$ – начисления на заработную плату разработчиков программы во внебюджетные фонды,

$Z_{ЭВМ}$ – затраты, связанные с эксплуатацией техники,

$Z_{СПП}$ – затраты на специальные программные продукты, необходимые для разработки программы для микроконтроллера,

$Z_{ХОН}$ – затраты на хозяйственно-операционные нужды,

$P_{Н}$ – накладные расходы ($P_{Н} = 30\%$ от $Z_{ФОТР}$).

Элемент формулы, названный $Z_{ФОТР}$ представляет собой зарплату программистов, работающих над программой. Для расчёта их зарплаты, необходимо учесть количество работников, их ставку, время их работы, а также коэффициент дополнительной зарплаты и районный коэффициент. Формула для расчёта общего фонда оплаты труда представлена в формуле 23.

$$Z_{ФОТР} = \sum_{j=1}^m O_{Рj} \cdot T_{РПРj} \cdot (1 + k_{Д})(1 + k_{У}) \quad (23)$$

$O_{Рj}$ – оклад j-го разработчика,

$T_{РПРj}$ – общее время работы над программой в месяцах,

$k_{Д}$ – коэффициент дополнительной зарплаты,

$k_{У}$ – районный коэффициент.

Оклад работника, работающего над программой микроконтроллера, составляет 22000р. Над программой работал один программист, в течении одного месяца.

Под коэффициентом дополнительной зарплаты понимается коэффициент, поднимающий заработную плату, в зависимости от качества выполняемой работы. Данный коэффициент является мотивацией для работника выполнять работу более качественно и быстро. В расчётах коэффициент дополнительной заработной платы будет принят за 0,2, что в процентах имеет значение 20%.

Районный коэффициент является коэффициентом повышения заработной платы, основанный на климатических условиях. Данный коэффициент устанавливается государством и устанавливается для каждой территории страны отдельно. В Пермском крае районный коэффициент составляет 0,15.

Таким образом расчёт общей заработной платы работника будет следующим:

$$Z_{\text{ФОТР}} = 22000 \cdot 1 \cdot (1 + 0,2) \cdot (1 + 0,15) = 30360 \text{ руб}$$

Страховой взнос во внебюджетные фонды складываются из взносов на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний.

Страховые взносы складываются из обязательного пенсионного страхования (ОПС), отчислений в фонд социального страхования и отчислений в фонд обязательного медицинского страхования.

Значения всех используемых ставок приведены в таблице 12.

Таблица 12 – Значения ставок страховых взносов

№	Наименование внебюджетного фонда	Размер ставок, %
1	Пенсионный фонд	22
2	Фонд социального страхования	2,9
3	Федеральный фонд обязательного медицинского страхования	5,1
	Итого:	30

Таким образом, в внебюджетные страховые фонды отчисляется 30% от заработной платы работника, при том, что они отчисляются не от заработной платы работника, а от его работодателя.

Расчёт отчисляемых средств в внебюджетные фонды указан в формуле 24.

$$З_{ОВФ} = 0,3 \cdot З_{ФОТР} \quad (24)$$

$$З_{ОВФ} = 0,3 \cdot 30360 = 9108 \text{ руб.}$$

Затраты, связанные с использованием вычислительной и оргтехники, представляют собой произведение времени, в течении которого будет работать вычислительная техника, её количество и стоимость работы одного часа.

В формуле 25 представлена формула, позволяющая рассчитать затраты вычислительной техники.

$$З_{ЭВМ} = T_{МРПР} \cdot k_{Г} \cdot n \cdot C_{М-ч} \quad (25)$$

$T_{МРПР}$ – машинное время работы над программой.

$k_{Г}$ – коэффициент готовности ЭВМ,

n – количество единиц техники,

$C_{М-ч}$ – себестоимость машино-часа.

Машинное время работы над программой представляет собой более точное время, в течении которого программист работает над программой пользуясь вычислительной техникой, и указывается в часах. Коэффициент готовности ЭВМ, как правило, никогда не бывает равен 1 %, так как в любой момент возможен шанс, что случится какая-либо поломка, в расчётах будет использоваться значение 0,95%. Количество единиц равно одному, а себестоимость машино-часа 5 рублям, основываясь на себестоимости тарифа за электроэнергию.

Для получения точного времени работы ЭВМ, необходимо воспользоваться формулой 26.

$$T_{\text{час}} = T_{\text{мес}} \cdot Ч_{\text{рд}} \cdot T_{\text{см}} \cdot K_{\text{см}}, \quad (26)$$

$T_{\text{час}}$ – рабочее время, в часах,

$T_{\text{мес}}$ – рабочее время, в месяцах, равное 1,

$Ч_{\text{рд}}$ – число рабочих дней, равное 22,

$T_{\text{см}}$ – продолжительность рабочей смены, равное 8 часам,

$K_{\text{см}}$ – количество рабочих смен в день.

Таким образом, расчёт времени на разработку программы для микроконтроллера с использованием ЭВМ и расчёт общих затрат на работу ЭВМ будут следующие:

$$T_{\text{час}} = 1 \cdot 22 \cdot 8 \cdot 1 = 176 \text{ часа,}$$

$$Z_{\text{ЭВМ}} = 176 \cdot 0,95 \cdot 1 \cdot 5 = 836 \text{ руб.}$$

Затраты на специальные программные продукты, необходимые для разработки программы для микроконтроллера рассчитываются по формуле, указанной в формуле 27.

$$Z_{\text{СПП}} = \sum_{p=1}^n Ц_p \quad (27)$$

где $Ц_p$ – цена p -го специального программного продукта.

Перечень программных продуктов специального назначения приведен в таблице 13.

Таблица 13 – Программные продукты специального назначения

№	Название ПП	Цена, руб.
1	Среда разработки Micro chip	0
2	САПР для разработки принципиальной схемы EasyEDA	0
3	Симулятор работы микроконтроллера Fritzing	720
	Итого:	720

Как можно видеть из таблицы 13, лишь одна программа является платной, и её стоимость составляет 720 рублей.

Затраты на хозяйственно–организационные нужды приведены в таблице 14 и вычисляются по формуле, указанной в формуле 28.

$$Z_{\text{ХОИ}} = \sum_{\tau=1}^n C_{\tau} \cdot K_{\tau}, \quad (28)$$

C_{τ} – цена τ -го товара,

K_{τ} – количество τ -го товара.

Таблица 14 – Затраты на хозяйственно–организационные нужды

Наименование	Цена за единицу (руб.)	Кол-во (шт.)	Всего (руб.)
USB–флеш–накопитель	600	1	600
Бумага	0,5	140	70
Итого:			670

Накладные расходы представляют собой некоторые дополнительные расходы, которые не связаны с выполнением разработки программы, но необходимы для их обеспечения косвенно. Формула для расчёта накладных затрат представлена в формуле 29.

$$P_{\text{Н}} = Z_{\text{ФОТР}} \cdot k_{\text{НР}}, \quad (29)$$

Для расчётов необходим коэффициент $k_{\text{НР}}$, который выбирается управляющим предприятием. В расчётах будет использоваться значение 0,30.

Расчёт накладных расходов:

$$P_{\text{Н}} = 30360 \cdot 0,3 = 9108 \text{ руб.}$$

Таким образом, затраты на разработку программы для микроконтроллера рассчитанные по формуле 22 будут следующие:

$$Z_{\text{РПР}} = 30360 + 9108 + 836 + 720 + 670 + 9108 = 50802 \text{ руб}$$

4.2 Расчет затрат на внедрение программы для микроконтроллера

Для расчётов на внедрение разработанной программы для микроконтроллеров, учитываются такие затраты как заработные платы работников, отчисления во внебюджетные фонды, затраты на материальную базу проекта и некоторые другие составляющие. Формула, по которой рассчитываются затраты на внедрение программы для микроконтроллеров, приведена в формуле 30.

$$Z_{\text{ВПР}} = Z_{\text{М}} + Z_{\text{КТС}} + Z_{\text{ПО}} + Z_{\text{ФОТВ}} + Z_{\text{ОФВ}} + Z_{\text{ЭВМ}} + P_{\text{КОМ}} + P_{\text{Н}}, \quad (30)$$

$Z_{\text{М}}$ – затраты на приобретение материалов,

$Z_{\text{КТС}}$ – затраты на приобретение комплекса технических средств,

$Z_{\text{ПО}}$ – затраты на приобретение программного обеспечения (включают стоимость разработанной программы, а также других существующих ПП, необходимых для функционирования системы),

$Z_{\text{ФОТВ}}$ – затраты на оплату труда работников, занятых внедрением проекта,

$Z_{\text{ОФВ}}$ – страховой взнос во внебюджетные фонды с заработной платы работников, занятых внедрением проекта,

$Z_{\text{ЭВМ}}$ – затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения,

$R_{\text{ком}}$ – командировочные расходы,

$R_{\text{н}}$ – накладные расходы,

Затраты на материальную базу проекта ($Z_{\text{М}}$) приведены в таблице 15.

Таблица 15 – Затраты на материальную базу проекта

Наименование	Цена за единицу (руб.)	Кол-во (шт.)	Всего (руб.)
Atmega 328p	150	1	150
Аккумулятор 2100 mAh	350	2	700
Bluetooth модуль CC2541	220	1	220
Сервопривод sg-90	130	13	1690
Пластик чёрный, 1 кг	750	1	750
Винты М2	2	26	52
Текстолит, 100x50	150	1	150
Разъём питания	15	1	15
Ультразвуковой дальномер	50	1	50
Плата зарядки аккумуляторов	60	1	60
Диод 1N5824	5	1	5
Конденсатор 680uF	5	1	5
Конденсатор 220uF	5	1	5
LM2596	11	1	11
Разъём pls, 40 контактов	15	2	30
Итого:			3893

Из дополнительных комплексов технических средств необходимы компьютер, паяльник и программатор. Из этих технических средств, для

покупки необходим лишь паяльник, компьютер и паяльник уже имеется. Стоимость программатора равна 200 рублей., а поэтому $Z_{\text{КТС}} = 200$ рублей.

Расходы на приобретение программного продукта будут равны затратам, на разработку программы для микроконтроллера, расчёты которых были проведены в разделе 4.1 и составляют 50802. Платные программы при внедрении программы для микроконтроллера не используются.

Внедрением занят один системный инженер с окладом 15000 руб. Время внедрения – 0,5 месяца. Отчисления во внебюджетные фонды составляют 30% от заработной платы работника. Ниже можно видеть расчёт заработной платы работника, а также расчёт отчислений в страховые фонды.

$$Z_{\text{ФОТВ}} = 15000 \cdot 0,5 \cdot (1 + 0,15) = 8625 \text{ руб.}$$

$$Z_{\text{ОВФ}} = 7500 \cdot 0,3 = 2587,5 \text{ руб.}$$

Расчёт затрат, связанных с эксплуатацией ЭВМ при внедрении проектного решения, представлен ниже.

$$Z_{\text{ЭВМ}} = 0,5 \cdot 22 \cdot 8 \cdot 5 = 440 \text{ руб.}$$

Командировочные расходы при внедрении программы для микроконтроллера не планируются, следовательно, $P_{\text{ком}} = 0$.

Так как коэффициент накладных расходов по данным организации составляет $k_{\text{НР}} = 0,3$, то величина накладных расходов равна 2587,5 руб.

Суммарные затраты на внедрение составят:

$$Z_{\text{ВПР}} = 3893 + 200 + 50802 + 8625 + 2587,5 + 440 + 0 + 2587,5 = 69135 \text{ руб}$$

4.3 Расчет эксплуатационных текущих затрат по программе для микроконтроллера

Годовые затраты на обработку результатов до внедрения разработанной программы для микроконтроллера рассчитываются по формуле 31.

$$C_1 = 3\Pi_1 + OT_{\text{ВН1}} + Z_{\text{ЭВМ1}} + M_{z1} + НР_1, \quad (31)$$

где $3\Pi_1$ – затраты на оплату труда сотрудника на выполнение функций до внедрения проектного решения,

$OT_{\text{ВН1}}$ – страховой взнос во внебюджетные фонды,

$Z_{ЭВМ1}$ – затраты, связанные с эксплуатацией ЭВМ,

$M_{з1}$ – годовые материальные затраты на сопровождение программы для микроконтроллера составляют 1500 руб.,

$НР_1$ – накладные расходы.

Временные затраты работы сотрудника в месяцах рассчитываются по формуле 32.

$$T_{1мес} = \frac{T_{1час}}{Ч_{рд} \cdot Ч_{рч}}, \quad (32)$$

где $T_{1мес}$, $T_{1час}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{1час} = 462$ часов),

$Ч_{рд}$ – число рабочих дней в месяц,

$Ч_{рч}$ – число рабочих часов в день.

Расчёт временных затрат сотрудника в месяцах представлен ниже.

$$T_{1мес} = \frac{462}{22 \cdot 8} = 2,6 \text{ мес},$$

Затраты на оплату труда сотрудника высчитываются по формуле 33.

$$ЗП_1 = O_c \cdot T_{1мес} \cdot (1 + k_d) \cdot (1 + k_y), \quad (33)$$

где O_c – оклад сотрудника (оклад составляет 20000 руб.), расчёт которого представлен ниже:

$$ЗП_1 = 20000 \cdot 2,6 \cdot (1 + 0,2) \cdot (1 + 0,15) = 71760 \text{ руб.}$$

Страховой взнос до внедрения вычисляют по формуле 34.

$$ОТ_{ВН1} = ЗП_1 \cdot 0,3, \quad (34)$$

$$ОТ_{ВН1} = 71760 \cdot 0,3 = 21528 \text{ руб.}$$

Расчёт затраты, связанных с эксплуатацией ЭВМ до внедрения производится по формуле 46.

$$Z_{ЭВМ1} = T_{1час} \cdot C_{М-ч}, \quad (46)$$

$$Z_{ЭВМ1} = 462 \cdot 5 = 2310 \text{ руб.}$$

Подставив соответствующие значения в формулу 31, получим:

$$C_1 = 71760 + 21528 + 2310 + 1500 = 97098 \text{ руб.}$$

Годовые затраты на эксплуатацию системы после внедрения программы для микроконтроллера рассчитываются аналогично по формуле 47.

$$C_2 = 3П_2 + ОТ_{ВН2} + З_{ЭВМ2} + М_{з2} + НР_2, \quad (47)$$

где $3П_2$ – затраты на оплату труда сотрудника после внедрения,

$ОТ_{ВН2}$ – страховой взнос во внебюджетные фонды,

$З_{ЭВМ2}$ – затраты, связанные с эксплуатацией ЭВМ после внедрения,

$М_{з2}$ – материальные затраты, годовые материальные затраты на сопровождение программы для микроконтроллера составляют 2500 руб.,

$НР_2$ – накладные расходы.

Временные затраты работы сотрудника в месяцах рассчитываются по формуле 48.

$$T_{2мес} = \frac{T_{2час}}{Ч_{рд} \cdot Ч_{рч}}, \quad (48)$$

где $T_{2мес}$, $T_{2час}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{2час} = 224$ часов),

$Ч_{рд}$ – число рабочих дней в месяц,

$Ч_{рч}$ – число рабочих часов в день.

$$T_{2мес} = \frac{224}{22 \cdot 8} = 1,3 \text{ мес.}$$

Тогда затраты на оплату труда сотрудника:

$$3П_2 = O_c \cdot T_{2мес} \cdot (1 + k_D) \cdot (1 + k_Y),$$

где O_c – оклад сотрудника (оклад составляет 20000 руб.)

$$3П_2 = 20000 \cdot 1,3 \cdot (1 + 0,2) \cdot (1 + 0,15) = 35880 \text{ руб.}$$

Страховой взнос после внедрения вычисляются по формуле 49.

$$ОТ_{ВН2} = 3П_2 \cdot 0,3, \quad (49)$$

$$ОТ_{ВН2} = 35880 \cdot 0,3 = 10764 \text{ руб.}$$

Рассчитываем затраты, связанные с эксплуатацией ЭВМ после внедрения по формуле 50.

$$З_{ЭВМ2} = T_{2час} \cdot C_{М-ч}, \quad (50)$$

$$З_{ЭВМ2} = 224 \cdot 5 = 1120 \text{ руб.}$$

Подставив соответствующие значения в формулу, получим:

$$C_2 = 35880 + 10764 + 1120 + 2500 = 50264 \text{ руб.}$$

Таким образом, текущие затраты на содержание системы до внедрения разработанной программы для микроконтроллера составляют 97098 руб., после внедрения 50264 руб.

4.4 Расчет экономической целесообразности разработки и внедрения информационных технологий

Для разрабатываемого проекта расчет экономической эффективности производится исходя из следующих условий:

– годовые текущие затраты до внедрения автоматизированной системы, $C_1 = 97098$ руб.,

– годовые текущие затраты после внедрения системы, $C_2 = 50264$ руб.,

– горизонт расчета принимается исходя из срока использования разработки, $T = T_H = 2$ годам,

– шаг расчета равен одному году, $t = 1$ году,

– капитальные вложения равны затратам на создание системы, $K = 69135$ руб.,

– норма дисконта равна норме дохода на капитал, $E = 12\%$.

Ожидаемая условно-годовая экономия от внедрения системы рассчитывается по формуле 51.

$$\text{Э}_{\text{уг}} = C_1 - C_2 + \sum \text{Э}_i, \quad (51)$$

где $\text{Э}_{\text{уг}}$ – ожидаемая условно-годовая экономия, руб.,

C_1 – годовые текущие затраты до внедрения автоматизированной системы, руб.,

C_2 – годовые текущие затраты после внедрения системы, руб.,

$\sum \text{Э}_i$ – ожидаемый дополнительный эффект от различных факторов, руб.

Так как основным фактором, по которому производится расчет экономической эффективности от внедрения программы для

микроконтроллера, является уменьшение времени обработки результатов тестирования и дополнительный эффект не учитывается, то $\sum \Delta i = 0$.

Подставив вычисленные выше значения в формулу, получим:

$$\Delta_{уг} = 97098 - 50264 = 46834 \text{ руб.}$$

где $\Delta_{уг}$ – ожидаемая условно-годовая экономия, руб.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле 52.

$$\Delta_{г} = \Delta_{уг} - K \cdot E_{н}, \quad (52)$$

где $\Delta_{г}$ – ожидаемый годовой экономический эффект, руб.,

$\Delta_{уг}$ – ожидаемая условно-годовая экономия, руб.,

K – капитальные вложения (равны затратам на создание устройства), руб.,

$E_{н}$ – нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле 53.

$$E_{н} = \frac{1}{T_{н}}, \quad (53)$$

где $T_{н}$ – нормативный срок окупаемости капитальных вложений, которые равны двум годам. $E_{н} = 0,5$.

Подставив вычисленные выше значения в формулу, получим:

$$\Delta_{г} = 46834 - 69135 \cdot 0,5 = 12266,5 \text{ руб.}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет рассчитывается по формуле 54.

$$E_{р} = \frac{\Delta_{уг}}{K}, \quad (54)$$

где $E_{р}$ – расчетный коэффициент экономической эффективности капитальных вложений,

$\Delta_{уг}$ – ожидаемая условно-годовая экономия, руб.,

K – капитальные вложения на создание системы, руб.

Подставив вычисленные выше значения в формулу, получим:

$$E_p = \frac{46834}{69135} = 0,67$$

Расчетный срок окупаемости капитальных вложений рассчитывается по формуле 55.

$$T_p = \frac{1}{E_p}, \quad (55)$$

где E_p – коэффициент экономической эффективности капитальных вложений.

Подставив вычисленные выше значения в формулу, получим:

$$T_p = \frac{1}{0,69} = 1,5 \text{ год.}$$

Срок окупаемости без дисконтирования 1,5 год.

Чистый дисконтированный доход (ЧДД) определяется как сумма текущих эффектов за весь расчетный период, приведенная к начальному шагу, или как превышение интегральных результатов над интегральными затратами.

Если в течение расчетного периода не происходит инфляционного изменения цен или расчет производится в базовых ценах, то величина ЧДД для постоянной нормы дисконта вычисляется по формуле 56.

$$\text{ЧДД} = \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1 + E)^t} - K, \quad (56)$$

где P_t – ожидаемые результаты от внедрения предложенной ИС, руб.,

Z_t – ожидаемые затраты (капитальные и текущие) на создание и эксплуатацию ИС, руб.,

$\Delta t = (P_t - Z_t)$ – эффект, достигаемый на t -м шаге расчета,

K – капитальные вложения,

t – номер шага расчета ($t = 1, 2$),

T – горизонт расчета,

E – постоянная норма дисконта, 12%.

$\Delta t = (P_t - Z_t) = \Delta_{\text{уг}} = 46834$ руб. В том случае, если текущие затраты (Z_t) на весь срок использования разработки равны 0.

$t = 1,2$ год, т.к. предполагается, что результат от внедрения предложенной ИС будет с текущего года внедрения ИС.

Если ЧДД инвестиционного проекта положителен, то проект является эффективным (при данной норме дисконта).

Тогда суммарный чистый дисконтированный доход за весь горизонт расчета вычисляется по формуле 57.

$$\begin{aligned} \text{ЧДД} &= \Delta_1 \cdot \frac{1}{(1 + E)} + \Delta_2 \cdot \frac{1}{(1 + E)^2} - K, \\ \text{ЧДД} &= \frac{46834}{(1 + 0,12)} + \frac{46834}{(1 + 0,12)^2} - 69135 = 10016,8 \text{ руб.} \end{aligned} \quad (57)$$

Положительное значение чистого дисконтированного дохода, $\text{ЧДД} > 0$, свидетельствует о том, что инвестирование целесообразно и данная ИС может приносить прибыль в установленном объеме.

Индекс доходности (ИД) представляет собой отношение суммы приведенных эффектов к величине капитальных вложений и определяется по формуле 58.

$$\text{ИД} = \frac{1}{K} \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1 + E)^t}, \quad (58)$$

где K – величина капиталовложений или стоимость инвестиций.

$$\text{ИД} = \frac{79151,8}{69135} = 1,14$$

Инвестиции считаются эффективными, если индекс доходности выше единицы, $\text{ИД} > 1$, следовательно, инвестиции в данную ИС, эффективны.

Внутренняя норма доходности (ВНД):

при $E_1 \rightarrow \text{ЧДД}_1 > 0$

$E_2 \rightarrow \text{ЧДД}_2 < 0$

Вычисляется по формуле 59.

$$\text{ВНД} = E_1 + \frac{\text{ЧДД}_1}{\text{ЧДД}_1 - \text{ЧДД}_2} \cdot (E_2 - E_1), \quad (59)$$

при $E_1 \rightarrow ЧДД_1 > 0$

$E_2 \rightarrow ЧДД_2 > 0$

Вычисляется по формуле 60.

$$ВНД = E_1 + \frac{ЧДД_1}{ЧДД_1 + ЧДД_2} \cdot (E_2 - E_1) \quad (60)$$

Расчёт $ЧДД_1$ при $E_1 = 0,11$:

$$ЧДД_1 = \frac{46834}{(1 + 0,11)} + \frac{46834}{(1 + 0,11)^2} - 69135 = 11069,3 \text{ руб.}$$

Расчёт $ЧДД_2$ при $E_1 = 0,13$:

$$ЧДД_2 = \frac{46834}{(1 + 0,13)} + \frac{46834}{(1 + 0,13)^2} - 69135 = 8988,9 \text{ руб.}$$

$E_1 \rightarrow ЧДД_1 > 0$

$E_2 \rightarrow ЧДД_2 > 0$

Так как оба полученных значения чистого дисконтного дохода являются положительными, расчёт внутренней нормы доходности будет производиться по формуле 60.

$$ВНД = 0,11 + \frac{12869,3}{12869,3 + 10788,9} \cdot (0,13 - 0,11) = 0,12.$$

Таким образом, норма дисконта должна быть в пределах 11% – 13%.

Показатели экономической целесообразности разработки и внедрения программы для микроконтроллера сведены в таблицу 16.

Таблица 16 – Показатели экономической целесообразности разработки и внедрения программы для микроконтроллера

Наименование показателя	Значения
Затраты на разработку и внедрение ПАП, руб.	69135
Ожидаемая экономия от внедрения ПАП, руб.	46834
Чистый дисконтированный доход, руб.	10016,8
Индекс доходности	1,14
Внутренняя норма доходности	0,12
Дисконтированный срок окупаемости, лет	1,5
Срок морального старения, года	2

Произведенные расчеты свидетельствуют, что внедрение, разработанной в ВКР программы для микроконтроллера, позволит сократить временные затраты на проведения учебных мероприятий по микроконтроллерным

системам и механике, что приведет к сокращению годовых текущих затрат на 46834 рубля.

Опираясь на оценку экономической эффективности можно сделать вывод о том, что разработка и внедрение предлагаемой программы для микроконтроллера является экономически обоснованной и целесообразной.

ЗАКЛЮЧЕНИЕ

В результате проведения выпускной квалификационной работы были выполнены все поставленные задачи:

- Проведён анализ предметной области проекта, было изучено понятие робота-шагохода, рассмотрены их виды и особенности. Проведён анализ рынка роботов-шагоходов, выдвинуты требования для разрабатываемого устройства.

- Проведён подбор инструментальной базы проекта, в результате которого были выбраны САПР для разработки принципиальной схемы и печатной платы, САПР для разработки корпуса устройства, среда разработки программы для микроконтроллера и среда разработки мобильного приложения с обоснованием выбора.

- Выполнена разработка контроллера робота-шагохода, в которую входил подбор элементной базы проекта, разработка принципиальной схемы, печатной платы, схемы походки робота, алгоритм и программа микроконтроллера, мобильное приложение.

- Проведена разработка деталей корпуса устройства, в результате которой были разработаны 2 детали основного корпуса, и три детали ноги робота.

- Проведён анализ требований охраны труда и промышленно экологии, в результате которого были рассмотрены вредные и опасные факторы инженера электронщика, мероприятия по оптимизации освещения на рабочем месте, а также требования при утилизации вычислительной техники.

- Выполнен расчёт экономической эффективности, который показал, что проект является экономически эффективным и реализация его целесообразна.

Разработанное устройство планируется использовать для обучения студентов обучающихся направлениях и специальностях УГС 09.00.00.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Разработка hexarod с нуля (часть 3) — кинематика. [электронный ресурс]. URL: <https://habr.com/ru/post/436748/> - (дата обращения 01.02.2022г.)
2. Шагающие роботы — теория и основы. [электронный ресурс]. URL: <https://monitorbank.ru/shagayushhie-roboty-teoriya-i-osnovy/> - (дата обращения 01.02.2022г.)
3. Что такое мехатроника, мехатронные элементы, модули, машины и системы. [электронный ресурс]. URL: <http://electricalschool.info/automation/2019-chto-takoe-mehatronika.html> - (дата обращения 01.02.2022г.)
4. МЕХАТРОНИКА. [электронный ресурс]. URL: https://bigenc.ru/technology_and_technique/text/4138417 - (дата обращения 01.02.2022г.)
5. ПОСТРОЕНИЕ КИНЕМАТИЧЕСКОЙ МОДЕЛИ ШАГАЮЩЕГО РОБОТА. [электронный ресурс]. URL: <https://izron.ru/articles/aktualnye-problemy-tekhnicheskikh-nauk-v-rossii-i-za-rubezhom-sbornik-nauchnykh-trudov-po-itogam-mezh/sektsiya-4-mashinostroenie-i-mashinovedenie-spetsialnost-05-02-00/postroenie-kinematocheskoj-modeli-shagayushchego-robota/> - (дата обращения 01.02.2022г.)
6. Робот-паук. [электронный ресурс]. URL: <https://robototehnika.ru/content/article/robot-pauk/> - (дата обращения 01.02.2022г.)
7. PhantomX AX-12 четвероногий комплект [электронный ресурс]. URL: <https://www.trossenrobotics.com/p/PhantomX-AX-12-Quadruped-mk1.aspx> - (дата обращения 01.02.2022г.)
8. DIY Raspberry Pi 4B обновленный шестиногий робот паук Hexarod Bionic SpiderPi программируемый [электронный ресурс]. URL: <http://lyvi.ru/p-4001098187152> - (дата обращения 01.02.2022г.)
9. CR-6 шестиногий робот-паук Hexarod. [электронный ресурс]. URL: <https://lazon.ru/p/33013236217> - (дата обращения 01.02.2022г.)

10. Обоснование выбора и разработка методов защиты от воздействующих овпф. [электронный ресурс]. URL: <https://studfile.net/preview/13209556/page:8/> - (дата обращение 01.02.2022г.)

11. НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ ОСВЕЩЕНИЕ РАБОЧИХ МЕСТ ВНУТРИ ЗДАНИЙ. URL: <https://docs.cntd.ru/document/1200105707> (дата обращения: 25.04.2022).

12. КОЭФФИЦИЕНТЫ ОТРАЖЕНИЯ (ρ) НЕКОТОРЫХ НАИБОЛЕЕ РАСПРОСТРАНЕННЫХ МАТЕРИАЛОВ И КРАСОК. URL: <https://sudact.ru/law/mu-224706-98mu-ot-rm-01-98-224-fizicheskie/prilozhenie-7/7.1/tablitsa-p.7.5/> (дата обращения: 25.04.2022).

13. Артеменко С.В., Тихоненко Д.В. Утилизация компьютерной техники. 2020. №8. URL: <https://cyberleninka.ru/article/n/utilizatsiya-kompyuternoj-tehniki> (дата обращения: 25.04.2022).

14. Список микроконтроллеров ATmega. [электронный ресурс]. URL: <https://chipinfo.pro/mcu/avr/atmegalist.shtml> - (дата обращения: 25.04.2022).

15. Сервоприводы Ардуино SG90, MG995, MG996: схема подключения и управление. [электронный ресурс]. URL: <https://arduinomaster.ru/motor-dvigatel-privod/servoprivody-arduino-sg90-mg995-shema-podklyuchenie-upravlenie/> - (дата обращения: 25.04.2022).

16. Bluetooth обозрение. Мини-тест модулей HC-05, HC-06, HM-10, DX-BT18. [электронный ресурс]. URL: <https://zen.yandex.ru/media/id/5f5e5b49d7a3287e1a0163c9/bluetooth-obozrenie-minitest-modulei-hc05-hc06-hm10-dxbt18-611920664a950b74d232a634> - (дата обращения: 25.04.2022).

17. Обзор популярных систем автоматизированного проектирования (CAD) [электронный ресурс]. URL: <https://www.pointcad.ru/novosti/obzor-sistem-avtomatizirovannogo-proektirovaniya> - (дата обращения: 25.04.2022).

18. Программы для микроконтроллеров [электронный ресурс]. URL: https://сhem.net/software/soft_mcu.php - (дата обращения: 25.04.2022).

19. Программы для проектирования печатных плат [электронный ресурс]. URL: https://chem.net/software/soft_PCB.php - (дата обращения: 25.04.2022).

20. Parallax PING))) 28015 User Manual [электронный ресурс]. URL: https://translated.turbopages.org/proxy_u/en-ru.ru.5388ce17-6267fce6-7ea88ce1-74722d776562/https/www.manualslib.com/manual/1552870/Parallax-Ping-28015.html - (дата обращения: 25.04.2022).

Приложение А – Техническое задание

Утверждаю

Должность

_____ Ф.И.О

« ____ » _____ 2022г

Робот-шагоход под микроконтроллерным управлением

Наименование

Робот-шагоход

Сокращенное наименование

Техническое задание

Действует с __71__

Согласовано

Должность

_____ Ф.И.О

« ____ » _____ 2022г

Разработчик

Должность

_____ Родионов А.М _____ Ф.И.О

« ____ » _____ 2022г

Лысьва 2022г.

1 Общие сведения

1.1 Наименование системы

1.1.1 Полное наименование

Полное наименование: Робот-шагоход под управлением микроконтроллера.

1.1.2 Краткое наименование

Краткое наименование: Робот-шагоход.

1.2 Основания для проведения работ

Работа выполняется на основании задания по проведению ВКР.

1.3 Наименование организаций – Заказчика и Разработчика

1.3.1 Заказчик

Заказчик: ЛФ ПНИПУ

1.3.2 Разработчик

Разработчик: Студент ЛФ ПНИПУ Родионов Александр

1.4 Плановые сроки начала и окончания работы

Работы по разработке робота-шагохода будут начаты 01.03.2022г и будут проходить до 15.05.2022г.

1.5 Источники и порядок финансирования

Разработка финансируется разработчиком. Максимальные бюджет на материалы не превышает 3500 рублей.

1.6 Порядок оформления и предъявления заказчику результатов работ

Результаты работ по разработке устройства сдаётся Заказчику с периодичностью в 1 неделю, с целью проверки правильности результатов разработки, конечный результат сдаётся в конце запланированного времени разработки. При сдаче результата Разработчик предъявляет Заказчику разработанное устройство со всеми документами, обговорёнными в договоре.

2 Назначение и цель создания системы

2.1 Назначение системы

Робот-шагоход предназначен для изучения студентами механики, робототехники, а также микроконтроллерного управления. При дальнейшем использовании робота-шагохода возможно проведение обучающих работ по добавлению различных датчиков и модулей, позволяющих студентам изучить способы работы с микроконтроллерами на примере готового устройства.

2.2 Цели создания системы

Робот-шагоход создаётся с целью выступления на выставке.

3 Характеристика объектов автоматизации

Работа робота-шагохода планируется в помещении с температурой от 15 до 28 градусов Цельсия. Поверхность, по которой будет передвигаться робот является ровной, состоящей из древесины и плит. Движение воздуха в помещении не имеется. Примерная площадь, по которой планируется эксплуатировать робота-шагохода, имеет размеры 3х3 метра.

4 Требования к системе

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционирования системы

Робот-шагоход содержит в себе 5 функциональных блоков, среди которых: блок питания, блок управления, блок приёма данных по беспроводному интерфейсу Bluetooth, блок анализа внешнего помещения, блок исполнительных устройств.

Блок питания выполняет работу по распределению питания между всеми остальными элементами системы. К другим функциям блока питания относится понижения питания системы до подходящего и безопасного, а также правильная зарядка и разрядка аккумуляторов.

Блок управления представляет собой управляющий микроконтроллер, в задачи которого входит управление всеми элементами устройства. Функции блока управления:

- принятие и обработка данных с Bluetooth модуля,
- принятие и обработка данных с блока анализа внешнего помещения,
- выполнение управления устройствами, выполняющими передвижение робота-шагохода, на основании полученных данных с Bluetooth модуля и блока анализа внешнего помещения.

Блок приёма данных по беспроводному интерфейсу Bluetooth позволяет выполнить стабильное подключение к устройству с мобильного устройства или ноутбука, с помощью беспроводного интерфейса Bluetooth. При этом выполняет приём и отправку полученных данных с мобильного устройства на управляющий микроконтроллер в функциональный блок управления.

Блок анализа внешнего помещения производит работу по считыванию дистанции до ближайшего препятствия в диапазоне 180 градусов. Полученные данные передаются к управляющему микроконтроллеру в функциональный блок управления.

Блок исполнительных устройств представляет собой набор приводов, выполняющих передвижения ног робота-шагохода. Управление производится в зависимости от полученных от микроконтроллера данных.

4.1.2 Требования к численности и квалификации персонала системы и режиму его работы

4.1.2.1 Требования к численности персонала

Для работы робота-шагохода достаточно одного работника, выполняющего управление им.

4.1.2.2 Требования к квалификации персонала

Для работы с роботом-шагоходом достаточно знаний и умений в области программирования 8-битных микроконтроллеров, а также основах электротехники.

4.1.3 Показатели назначения

4.1.3.1 Параметры, характеризующие степень соответствия системы назначению

Система должна обеспечивать:

прямолинейное передвижение робота-шагохода на дистанцию 1 метра не более чем за 5 секунд,

Занимаемую площадь робота-шагохода при его стойке на месте, не более 10 см²,

Анализ окружающего робота помещения на расстоянии не менее 0,7 м.

4.1.3.2 Требования к приспособляемости системы к изменениям

При резком обнаружении роботом-шагоходом препятствия на дистанции мене 20 см, робот должен резко изменить направление, в направлении, где препятствия находится дальше от робота-шагохода.

Приспособляемость при изменении уровня поверхности не предусмотрена.

4.1.3.3 Требования к сохранению работоспособности системы в различных вероятных условиях

В зависимости от некоторых ошибок, возникших в работе устройства, будут выполняться действия приведённые в таблице 1.

Таблица 1 - вероятные случаи в работе робота-шагохода

Вероятное условие	Действие
Приём некорректных данных от Bluetooth модуля.	Работа по прежнему полученным данным, игнорируя полученные некорректные данные.
Недостаточный уровень напряжения питания	Отключение устройства

4.1.4 Требования к надёжности

4.1.4.1 Состав показателей надёжности для системы в целом

Надёжность в разрабатываемом роботе-шагоходе достигается за счёт:

Качественной и правильной сборке устройства, а также качественной пайке радиоэлементов.

Проверке работоспособности программы для микроконтроллера на стадии её разработки.

Использовании проверенных элементов, показывающих необходимые результаты для применения в проекте.

4.1.4.2 Перечень аварийных ситуаций, по которым регламентируются требования к надёжности

В ходе эксплуатации робота-шагохода могут возникнуть следующие аварийные ситуации:

- сбой связи между роботом и мобильным устройством,
- сбой программного обеспечения,
- сбой работы сервоприводов, установленных на роботе,
- сбой питания робота-шагохода.

4.1.4.3 Требования к надёжности технических средств и программного обеспечения

Для надёжной работы устройства, к аппаратной части устройства предъявляются следующие требования:

- аккумулятор ёмкостью более 4000 mAh,
- максимальное расстояние, которое может сканировать робот не менее 1 метра,
- расстояние, на котором может работать Bluetooth модуль с мобильным устройством не менее 15 метров.

4.1.4.4 Требования к методам оценки и контроля показателей надёжности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами.

Проверка надёжности аппаратных элементов устройства проводится путём опытной проверки работоспособности элементов.

4.1.4 Требования к эргономике и технической эстетике

Интерфейс мобильного приложения, предназначенного для управления роботом, должен иметь приятный графический интерфейс с 6 кнопками, предназначенными для включения и выключения робота, а также передвижения вперёд, назад, влево, вправо и переключения режима передвижения.

Интерфейс приложения должен иметь максимум 3 цвета.

4.1.6 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Хранение и эксплуатация как робота, так и отдельных его частей должны проводиться в температурном режиме от +5 до +35 градусов Цельсия. Недопускается хранение элементов робота под прямыми солнечными лучами. Хранение производится при влажности от 60 до 80 %.

Эксплуатация робота производится при температуре не выше 25 градусов Цельсия, при относительной влажности воздуха 70-85 %.

4.1.7 Требования к защите информации от несанкционированного доступа

4.1.7.1 Требования к информационной безопасности

Для защиты от несанкционированного подключения к Bluetooth с мобильного устройства, необходимо использование пароля.

4.1.7.2 Требования к антивирусной защите

К антивирусной защите требований не предъявляется.

4.1.8 Требования к сохранению информации при авариях

В случае возникновения аварии, нет данных которые необходимо экстренно сохранить, так как при следующем включении робот сам откалибруется и примет необходимое положение для дальнейшей работы.

4.1.9 Требования к защите от влияния внешних воздействий

Для сохранности корпуса устройства необходим локатор помещения, который защитит робота от физических поломок.

Для защиты от повышенного напряжения питания аккумуляторов робота установлена плата для регулируемой зарядки и разрядки робота.

4.1.10 Требования по стандартизации и унификации

При разработке робота-шагохода необходимо использовать стандартный разъём для питания 12 вольт. Для связи с мобильным устройством по Bluetooth, необходимо использовать стандартный Bluetooth версии 4.1. Написанный программный код должен иметь комментарии, объясняющие алгоритм программы. Все схемы и диаграммы должны быть разработаны в соответствии с стандартами UML.

4.2 Требования к функциям, выполняемым системой

Устройство имеет следующие функциональные блоки:

- Блок анализа помещения,
- Блок управления,
- Исполнительный блок,
- Блок питания,
- Блок Bluetooth.

В таблице 2 представлены выполняемые задачи и функции блоком анализа помещения.

Таблица 12 - Функции и задачи блока анализа помещения

Функции	Задачи
Сканирования помещения на расстояние робота-шагохода до ближайшего препятствия.	Сканирование расстояния до препятствий перед началом движения.
	Сканирование расстояния до препятствий по прохождению роботом-шагоходом 0,5 метра.

Сканирования помещения на расстояние до ближайшего препятствия.	Анализ полученных данных с сравнением и заменой их с данными полученными при начале движения.
---	---

В таблице 3 представлены выполняемые задачи и функции блоком управления.

Таблица 3 - Функции и задачи блока управления

Функции	Задачи
Анализ помещения на расстояние до ближайшего препятствия.	Сравнение начального и последнего сканирования помещения.
	Поиск направления с максимальным расстоянием до препятствия.
	Изменение маршрута робота-шагохода в направлении максимального расстояния до препятствия.
Обработка полученных данных с Bluetooth.	Получение данных
	Сравнение данных с командами и последующие выполнение команд

В таблице 4 представлены выполняемые задачи и функции исполнительным блоком.

Таблица 4 - Функции и задачи исполнительного блока

Функции	Задачи
Выполнение шагов ногами робота	Получение команд от блока управления.
	Выполнение передвижения в соответствии с полученными командами.

В таблице 5 представлены выполняемые задачи и функции блоком питания.

Таблица 5 - Функции и задачи блока питания

Функции	Задачи
Обеспечение необходимого уровня напряжения питания.	Зарядка аккумуляторных батарей.
	Понижение напряжения до 5 вольт.
	Защита аккумуляторов от переразрядки и перезарядки.

В таблице 6 представлены выполняемые задачи и функции Bluetooth блоком.

Таблица 6 - Функции и задачи блока Bluetooth

Функции	Задачи
Приём данных с мобильного приложения.	Приём данных и дальнейшая их передача в блок управления по интерфейсу UART.

Кроме описанных функциональных блоков, имеется мобильное приложение, функционал и задачи которого показаны в таблице 7.

Таблица 7 - Функции и задачи мобильного приложения

Функции	Задачи
Управление роботом.	Подключение по интерфейсу Bluetooth к роботу-шагоходу.

Управление роботом.	Вывод элементов управления на экран мобильного приложения.
	Передача данных по Bluetooth в зависимости от нажатой пользователем кнопки.

4.3 Требования к видам обеспечения

4.3.1 Требования к математическому обеспечению

Не предъявляются.

4.3.2 Требования к информационному обеспечению

Необходимо выполнение стабильного обмена данными между устройством и мобильным приложением, с максимальной задержкой в 1 секунду. Связь должна поддерживаться на расстоянии 10 метров.

4.3.3 Требования к лингвистическому обеспечению

Выдвигаются требования к языку программирования используемому в разработке, а также к кодировке, в которой будут передаваться данные между устройством и мобильным приложением.

Программа для микроконтроллеров должна быть написана на языке программирования C++ или ему подобном. Требования к языку программирования, на котором будет написано мобильное приложение не выдвигаются.

4.3.4 Требования к программному обеспечению

Мобильное приложение должно поддерживаться на операционной системе Android выше 10 версии. Для работы с устройством не требуется дополнительных программ, кроме разработанного мобильного приложения для управления роботом.

4.3.5 Требования к техническому обеспечению

Приложение должно работать на мобильном устройстве, с оперативной памятью выше 2 Гб и процессором с выше 4 ядер. Приложение должно расходовать не более 20 Мб памяти телефона.

4.3.6 Требования к метрологическому обеспечению

Для работы робота необходимо выполнение им точных шагов, на равное расстояние для всех ног.

4.3.7 Требования к организационному обеспечению

Не предъявляются.

4.3.8 Требования к методическому обеспечению

По завершению разработки Разработчик должен предоставить Заказчику и диаграммы все схемы разработанные при разработке устройства (функциональная, принципиальная схемы, схемы печатной платы, чертежи корпуса устройства, блок-схемы алгоритма, диаграммы походки робота) а также программный код для микроконтроллера и мобильного приложения.

4.3.9 Требования к патентной чистоте

Разработанное устройство должно быть свободно от авторских прав, и иметь полную возможность на эксплуатацию.

5 Состав и содержание работ по созданию системы

Разработка робота-шагохода должна выполняться в три этапа: проектирование, сборка устройства, тестирование и отладка – 2 месяца, ввод в эксплуатацию – 0,5 месяца.

6 Порядок контроля и приемки системы

На этапе разработки робота-шагохода, Разработчик должен выполнить испытание используемых элементов робота-шагохода, в результате которых, к разработке будут допущены лишь те элементы робота-шагохода, которые полностью выполняют необходимые им функции.

Перед приёмкой устройство будут проводиться испытания по проверки работоспособности робота-шагохода, с проверкой всех описанных функций робота. Проверка проводится на территории Разработчика, в соответствии с ГОСТ Р 60.0.3.1-2016.

7 Требования к составу и содержанию работ по подготовке устройства к вводу в эксплуатацию

Перед вводом устройства в эксплуатацию необходимо выполнить завершение разработки необходимых схем и диаграмм, а также программный код. Необходимо выполнение теста работоспособности устройства, с положительным результатом.

8 Требования к документированию

К документированию не предъявляется особых требований, за исключением разработанных материалов, полученных в процессе разработки устройства.

9 Источники разработки

Техническое задание разработано на основании задания на выпускную квалификационную работу. В процессе разработки устройства необходимо пользоваться:

ГОСТ Р 60.0.0.4-2019 «РОБОТЫ И РОБОТОТЕХНИЧЕСКИЕ УСТРОЙСТВА»

ГОСТ 2.702-2011 «Единая система конструкторской документации»

ГОСТ Р 60.0.3.1-2016 «РОБОТЫ И РОБОТОТЕХНИЧЕСКИЕ УСТРОЙСТВА Виды испытаний»

Приложение Б – Программный код

```
/* Includes -----*/
#include <Servo.h> //to define and control servos
#include <FlexiTimer2.h>//to set a timer to manage all servos
// RegisHsu, remote control
#include <SerialCommand.h>
SerialCommand SCmd; // The demo SerialCommand object

/* Servos -----*/
//define 12 servos for 4 legs
Servo servo[4][3];
//define servos' ports
const int servo_pin[4][3] = { {2, 3, 4}, {5, 6, 7}, {8, 9, 10}, {11, 12, 13} };
/* Size of the robot -----*/
const float length_a = 55;
const float length_b = 77.5;
const float length_c = 27.5;
const float length_side = 71;
const float z_absolute = -28;
/* Constants for movement -----*/
const float z_default = -50, z_up = -30, z_boot = z_absolute;
const float x_default = 62, x_offset = 0;
const float y_start = 0, y_step = 40;
/* variables for movement -----*/
volatile float site_now[4][3]; //координаты конца каждой ноги в реальном времени
volatile float site_expect[4][3]; //ожидаемые координаты конца каждого отрезка пути
float temp_speed[4][3]; //скорость каждой оси, должна быть пересчитана перед
каждым движением
float move_speed; //скорость передвижения
float speed_multiple = 1; //скорость движения несколько
const float spot_turn_speed = 4;
const float leg_move_speed = 8;
const float body_move_speed = 3;
const float stand_seat_speed = 1;
volatile int rest_counter; //+1/0.02s, for automatic rest
//functions' parameter
const float KEEP = 255;
//define PI for calculation
const float pi = 3.1415926;
/* Constants for turn -----*/
//длина темпа
const float temp_a = sqrt(pow(2 * x_default + length_side, 2) + pow(y_step, 2));
const float temp_b = 2 * (y_start + y_step) + length_side;
const float temp_c = sqrt(pow(2 * x_default + length_side, 2) + pow(2 * y_start + y_step
+ length_side, 2));
const float temp_alpha = acos((pow(temp_a, 2) + pow(temp_b, 2) - pow(temp_c, 2)) / 2 /
temp_a / temp_b);
//площадка для разворота
const float turn_x1 = (temp_a - length_side) / 2;
const float turn_y1 = y_start + y_step / 2;
const float turn_x0 = turn_x1 - temp_b * cos(temp_alpha);
```

```

const float turn_y0 = temp_b * sin(temp_alpha) - turn_y1 - length_side;
/* -----*/

/*
- setup function
-----*/
int eyes =A0;

void setup()
{
//start serial for debug
Serial.begin(9600);
Serial.println("Robot starts initialization");

// RegisHsu, remote control
// Настройка обратных вызовов для команд SerialCommand
// action command 0-6,
// w 0 1: встать
// w 0 0: сесть
// w 1 x: идти вперед
// w 2 x: идти назад
// w 3 x: поворот направо
// w 4 x: поворот налево
// w 5 x: помахать
// w 6 x: помахать 2
SCmd.addCommand((char *)"w", action_cmd, action_cmd, action_cmd, action_cmd);

//SCmd.setDefaultHandler(unrecognized);

//initialize default parameter
set_site(0, x_default - x_offset, y_start + y_step, z_boot);
set_site(1, x_default - x_offset, y_start + y_step, z_boot);
set_site(2, x_default + x_offset, y_start, z_boot);
set_site(3, x_default + x_offset, y_start, z_boot);
for (int i = 0; i < 4; i++)
{
for (int j = 0; j < 3; j++)
{
site_now[i][j] = site_expect[i][j];
}
}
//start servo service
FlexiTimer2::set(20, servo_service);
FlexiTimer2::start();
Serial.println("Servo service started");
//initialize servos
servo_attach();
Serial.println("Servos initialized");
Serial.println("Robot initialization Complete");

pinMode(eyes,OUTPUT);
digitalWrite(eyes,HIGH);

```

```

}

void servo_attach(void)//настройка Серво пинов
{
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].attach(servo_pin[i][j]);
      delay(100);
    }
  }
}

void servo_detach(void)//отключение всех переменных Servo
{
  for (int i = 0; i < 4; i++)
  {
    for (int j = 0; j < 3; j++)
    {
      servo[i][j].detach();
      delay(100);
    }
  }
}
/*
- loop function
-----*/

void loop()
{
  SCmd.readString();
}

void do_test(void)//тест работы
{
  Serial.println("Stand");
  stand();
  delay(2000);
  Serial.println("Step forward");
  step_forward(5);
  delay(2000);
  Serial.println("Step back");
  step_back(5);
  delay(2000);
  Serial.println("Turn left");
  turn_left(5);
  delay(2000);
  Serial.println("Turn right");
  turn_right(5);
  delay(2000);
}

```

```

Serial.println("Hand wave");
hand_wave(3);
delay(2000);
Serial.println("Hand wave");
hand_shake(3);
delay(2000);
Serial.println("Sit");
sit();
delay(5000);
}

```

```

// RegisHsu
// w 0 1: stand
// w 0 0: sit
// w 1 x: forward x step
// w 2 x: back x step
// w 3 x: right turn x step
// w 4 x: left turn x step
// w 5 x: hand shake x times
// w 6 x: hand wave x times
#define W_STAND_SIT 0
#define W_FORWARD 1
#define W_BACKWARD 2
#define W_LEFT 3
#define W_RIGHT 4
#define W_SHAKE 5
#define W_WAVE 6
void action_cmd(void)
{
char *arg;
int action_mode, n_step;
Serial.println("Action:");
arg = SCmd.next();
action_mode = atoi(arg);
arg = SCmd.next();
n_step = atoi(arg);

switch (action_mode)
{
case W_FORWARD:
Serial.println("Step forward");
if (!is_stand())
stand();
step_forward(n_step);
break;
case W_BACKWARD:
Serial.println("Step back");
if (!is_stand())
stand();
step_back(n_step);
break;
case W_LEFT:

```

```

    Serial.println("Turn left");
    if (!is_stand())
        stand();
    turn_left(n_step);
    break;
case W_RIGHT:
    Serial.println("Turn right");
    if (!is_stand())
        stand();
    turn_right(n_step);
    break;
case W_STAND_SIT:
    Serial.println("1:up,0:dn");
    if (n_step)
        stand();
    else
        sit();
    break;
case W_SHAKE:
    Serial.println("Hand shake");
    hand_shake(n_step);
    break;
case W_WAVE:
    Serial.println("Hand wave");
    hand_wave(n_step);
    break;
default:
    Serial.println("Error");
    break;
}
}

```

// Он устанавливается в качестве обработчика по умолчанию и вызывается, когда другие команды не совпадают.

```

//void unrecognized(const char *command) {
// Serial.println("What?");
//}

/*
- is_stand
-----*/
bool is_stand(void)
{
    if (site_now[0][2] == z_default)
        return true;
    else
        return false;
}

/*
- sit
- blocking function

```

```

-----*/
void sit(void)
{
  move_speed = stand_seat_speed;
  for (int leg = 0; leg < 4; leg++)
  {
    set_site(leg, KEEP, KEEP, z_boot);
  }
  wait_all_reach();
}

/*
- stand
- blocking function
-----*/

void stand(void)
{
  move_speed = stand_seat_speed;
  for (int leg = 0; leg < 4; leg++)
  {
    set_site(leg, KEEP, KEEP, z_default);
  }
  wait_all_reach();
}

/*
- spot turn to left
- blocking function
- parameter step steps wanted to turn
-----*/

void turn_left(unsigned int step)
{
  move_speed = spot_turn_speed;
  while (step-- > 0)
  {
    if (site_now[3][1] == y_start)
    {
      //leg 3&1 move
      set_site(3, x_default + x_offset, y_start, z_up);
      wait_all_reach();

      set_site(0, turn_x1 - x_offset, turn_y1, z_default);
      set_site(1, turn_x0 - x_offset, turn_y0, z_default);
      set_site(2, turn_x1 + x_offset, turn_y1, z_default);
      set_site(3, turn_x0 + x_offset, turn_y0, z_up);
      wait_all_reach();

      set_site(3, turn_x0 + x_offset, turn_y0, z_default);
      wait_all_reach();

      set_site(0, turn_x1 + x_offset, turn_y1, z_default);

```

```
set_site(1, turn_x0 + x_offset, turn_y0, z_default);
set_site(2, turn_x1 - x_offset, turn_y1, z_default);
set_site(3, turn_x0 - x_offset, turn_y0, z_default);
wait_all_reach();
```

```
set_site(1, turn_x0 + x_offset, turn_y0, z_up);
wait_all_reach();
```

```
set_site(0, x_default + x_offset, y_start, z_default);
set_site(1, x_default + x_offset, y_start, z_up);
set_site(2, x_default - x_offset, y_start + y_step, z_default);
set_site(3, x_default - x_offset, y_start + y_step, z_default);
wait_all_reach();
```

```
set_site(1, x_default + x_offset, y_start, z_default);
wait_all_reach();
```

```
}
```

```
else
```

```
{
```

```
//leg 0&2 move
```

```
set_site(0, x_default + x_offset, y_start, z_up);
wait_all_reach();
```

```
set_site(0, turn_x0 + x_offset, turn_y0, z_up);
set_site(1, turn_x1 + x_offset, turn_y1, z_default);
set_site(2, turn_x0 - x_offset, turn_y0, z_default);
set_site(3, turn_x1 - x_offset, turn_y1, z_default);
wait_all_reach();
```

```
set_site(0, turn_x0 + x_offset, turn_y0, z_default);
wait_all_reach();
```

```
set_site(0, turn_x0 - x_offset, turn_y0, z_default);
set_site(1, turn_x1 - x_offset, turn_y1, z_default);
set_site(2, turn_x0 + x_offset, turn_y0, z_default);
set_site(3, turn_x1 + x_offset, turn_y1, z_default);
wait_all_reach();
```

```
set_site(2, turn_x0 + x_offset, turn_y0, z_up);
wait_all_reach();
```

```
set_site(0, x_default - x_offset, y_start + y_step, z_default);
set_site(1, x_default - x_offset, y_start + y_step, z_default);
set_site(2, x_default + x_offset, y_start, z_up);
set_site(3, x_default + x_offset, y_start, z_default);
wait_all_reach();
```

```
set_site(2, x_default + x_offset, y_start, z_default);
wait_all_reach();
```

```
}
```

```
}
```

```
}
```

```

/*
- spot turn to right
- blocking function
- parameter step steps wanted to turn
-----*/
void turn_right(unsigned int step)
{
move_speed = spot_turn_speed;
while (step-- > 0)
{
if (site_now[2][1] == y_start)
{
//leg 2&0 move
set_site(2, x_default + x_offset, y_start, z_up);
wait_all_reach();

set_site(0, turn_x0 - x_offset, turn_y0, z_default);
set_site(1, turn_x1 - x_offset, turn_y1, z_default);
set_site(2, turn_x0 + x_offset, turn_y0, z_up);
set_site(3, turn_x1 + x_offset, turn_y1, z_default);
wait_all_reach();

set_site(2, turn_x0 + x_offset, turn_y0, z_default);
wait_all_reach();

set_site(0, turn_x0 + x_offset, turn_y0, z_default);
set_site(1, turn_x1 + x_offset, turn_y1, z_default);
set_site(2, turn_x0 - x_offset, turn_y0, z_default);
set_site(3, turn_x1 - x_offset, turn_y1, z_default);
wait_all_reach();

set_site(0, turn_x0 + x_offset, turn_y0, z_up);
wait_all_reach();

set_site(0, x_default + x_offset, y_start, z_up);
set_site(1, x_default + x_offset, y_start, z_default);
set_site(2, x_default - x_offset, y_start + y_step, z_default);
set_site(3, x_default - x_offset, y_start + y_step, z_default);
wait_all_reach();

set_site(0, x_default + x_offset, y_start, z_default);
wait_all_reach();
}
else
{
//leg 1&3 move
set_site(1, x_default + x_offset, y_start, z_up);
wait_all_reach();

set_site(0, turn_x1 + x_offset, turn_y1, z_default);
set_site(1, turn_x0 + x_offset, turn_y0, z_up);

```

```
set_site(2, turn_x1 - x_offset, turn_y1, z_default);
set_site(3, turn_x0 - x_offset, turn_y0, z_default);
wait_all_reach();
```

```
set_site(1, turn_x0 + x_offset, turn_y0, z_default);
wait_all_reach();
```

```
set_site(0, turn_x1 - x_offset, turn_y1, z_default);
set_site(1, turn_x0 - x_offset, turn_y0, z_default);
set_site(2, turn_x1 + x_offset, turn_y1, z_default);
set_site(3, turn_x0 + x_offset, turn_y0, z_default);
wait_all_reach();
```

```
set_site(3, turn_x0 + x_offset, turn_y0, z_up);
wait_all_reach();
```

```
set_site(0, x_default - x_offset, y_start + y_step, z_default);
set_site(1, x_default - x_offset, y_start + y_step, z_default);
set_site(2, x_default + x_offset, y_start, z_default);
set_site(3, x_default + x_offset, y_start, z_up);
wait_all_reach();
```

```
set_site(3, x_default + x_offset, y_start, z_default);
wait_all_reach();
```

```
    }
  }
}
```

```
/*
```

```
- go forward
- blocking function
- parameter step steps wanted to go
```

```
-----*/
```

```
void step_forward(unsigned int step)
```

```
{
  move_speed = leg_move_speed;
  while (step-- > 0)
  {
    if (site_now[2][1] == y_start)
    {
      //leg 2&1 move
      set_site(2, x_default + x_offset, y_start, z_up);
      wait_all_reach();
      set_site(2, x_default + x_offset, y_start + 2 * y_step, z_up);
      wait_all_reach();
      set_site(2, x_default + x_offset, y_start + 2 * y_step, z_default);
      wait_all_reach();

      move_speed = body_move_speed;

      set_site(0, x_default + x_offset, y_start, z_default);
      set_site(1, x_default + x_offset, y_start + 2 * y_step, z_default);
```

```
set_site(2, x_default - x_offset, y_start + y_step, z_default);
set_site(3, x_default - x_offset, y_start + y_step, z_default);
wait_all_reach();
```

```
move_speed = leg_move_speed;
```

```
set_site(1, x_default + x_offset, y_start + 2 * y_step, z_up);
wait_all_reach();
set_site(1, x_default + x_offset, y_start, z_up);
wait_all_reach();
set_site(1, x_default + x_offset, y_start, z_default);
wait_all_reach();
```

```
}
```

```
else
```

```
{
```

```
  //leg 0&3 move
```

```
  set_site(0, x_default + x_offset, y_start, z_up);
```

```
  wait_all_reach();
```

```
  set_site(0, x_default + x_offset, y_start + 2 * y_step, z_up);
```

```
  wait_all_reach();
```

```
  set_site(0, x_default + x_offset, y_start + 2 * y_step, z_default);
```

```
  wait_all_reach();
```

```
move_speed = body_move_speed;
```

```
set_site(0, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(1, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(2, x_default + x_offset, y_start, z_default);
```

```
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_default);
```

```
wait_all_reach();
```

```
move_speed = leg_move_speed;
```

```
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_up);
```

```
wait_all_reach();
```

```
set_site(3, x_default + x_offset, y_start, z_up);
```

```
wait_all_reach();
```

```
set_site(3, x_default + x_offset, y_start, z_default);
```

```
wait_all_reach();
```

```
}
```

```
}
```

```
}
```

```
/*
```

```
- go back
```

```
- blocking function
```

```
- parameter step steps wanted to go
```

```
-----*/
```

```
void step_back(unsigned int step)
```

```
{
```

```
  move_speed = leg_move_speed;
```

```
  while (step-- > 0)
```

```

{
if (site_now[3][1] == y_start)
{
//leg 3&0 move
set_site(3, x_default + x_offset, y_start, z_up);
wait_all_reach();
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_up);
wait_all_reach();
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_default);
wait_all_reach();

move_speed = body_move_speed;

set_site(0, x_default + x_offset, y_start + 2 * y_step, z_default);
set_site(1, x_default + x_offset, y_start, z_default);
set_site(2, x_default - x_offset, y_start + y_step, z_default);
set_site(3, x_default - x_offset, y_start + y_step, z_default);
wait_all_reach();

move_speed = leg_move_speed;

set_site(0, x_default + x_offset, y_start + 2 * y_step, z_up);
wait_all_reach();
set_site(0, x_default + x_offset, y_start, z_up);
wait_all_reach();
set_site(0, x_default + x_offset, y_start, z_default);
wait_all_reach();
}
else
{
//leg 1&2 move
set_site(1, x_default + x_offset, y_start, z_up);
wait_all_reach();
set_site(1, x_default + x_offset, y_start + 2 * y_step, z_up);
wait_all_reach();
set_site(1, x_default + x_offset, y_start + 2 * y_step, z_default);
wait_all_reach();

move_speed = body_move_speed;

set_site(0, x_default - x_offset, y_start + y_step, z_default);
set_site(1, x_default - x_offset, y_start + y_step, z_default);
set_site(2, x_default + x_offset, y_start + 2 * y_step, z_default);
set_site(3, x_default + x_offset, y_start, z_default);
wait_all_reach();

move_speed = leg_move_speed;

set_site(2, x_default + x_offset, y_start + 2 * y_step, z_up);
wait_all_reach();
set_site(2, x_default + x_offset, y_start, z_up);
wait_all_reach();
}
}

```

```

    set_site(2, x_default + x_offset, y_start, z_default);
    wait_all_reach();
}
}
}

```

// add by RegisHsu

```

void body_left(int i)
{
    set_site(0, site_now[0][0] + i, KEEP, KEEP);
    set_site(1, site_now[1][0] + i, KEEP, KEEP);
    set_site(2, site_now[2][0] - i, KEEP, KEEP);
    set_site(3, site_now[3][0] - i, KEEP, KEEP);
    wait_all_reach();
}

```

```

void body_right(int i)
{
    set_site(0, site_now[0][0] - i, KEEP, KEEP);
    set_site(1, site_now[1][0] - i, KEEP, KEEP);
    set_site(2, site_now[2][0] + i, KEEP, KEEP);
    set_site(3, site_now[3][0] + i, KEEP, KEEP);
    wait_all_reach();
}

```

```

void hand_wave(int i)
{
    float x_tmp;
    float y_tmp;
    float z_tmp;
    move_speed = 1;
    if (site_now[3][1] == y_start)
    {
        body_right(15);
        x_tmp = site_now[2][0];
        y_tmp = site_now[2][1];
        z_tmp = site_now[2][2];
        move_speed = body_move_speed;
        for (int j = 0; j < i; j++)
        {
            set_site(2, turn_x1, turn_y1, 50);
            wait_all_reach();
            set_site(2, turn_x0, turn_y0, 50);
            wait_all_reach();
        }
        set_site(2, x_tmp, y_tmp, z_tmp);
        wait_all_reach();
        move_speed = 1;
        body_left(15);
    }
    else

```

```

{
  body_left(15);
  x_tmp = site_now[0][0];
  y_tmp = site_now[0][1];
  z_tmp = site_now[0][2];
  move_speed = body_move_speed;
  for (int j = 0; j < i; j++)
  {
    set_site(0, turn_x1, turn_y1, 50);
    wait_all_reach();
    set_site(0, turn_x0, turn_y0, 50);
    wait_all_reach();
  }
  set_site(0, x_tmp, y_tmp, z_tmp);
  wait_all_reach();
  move_speed = 1;
  body_right(15);
}
}

```

```

void hand_shake(int i)
{
  float x_tmp;
  float y_tmp;
  float z_tmp;
  move_speed = 1;
  if (site_now[3][1] == y_start)
  {
    body_right(15);
    x_tmp = site_now[2][0];
    y_tmp = site_now[2][1];
    z_tmp = site_now[2][2];
    move_speed = body_move_speed;
    for (int j = 0; j < i; j++)
    {
      set_site(2, x_default - 30, y_start + 2 * y_step, 55);
      wait_all_reach();
      set_site(2, x_default - 30, y_start + 2 * y_step, 10);
      wait_all_reach();
    }
    set_site(2, x_tmp, y_tmp, z_tmp);
    wait_all_reach();
    move_speed = 1;
    body_left(15);
  }
  else
  {
    body_left(15);
    x_tmp = site_now[0][0];
    y_tmp = site_now[0][1];
    z_tmp = site_now[0][2];
    move_speed = body_move_speed;

```

```

for (int j = 0; j < i; j++)
{
    set_site(0, x_default - 30, y_start + 2 * y_step, 55);
    wait_all_reach();
    set_site(0, x_default - 30, y_start + 2 * y_step, 10);
    wait_all_reach();
}
set_site(0, x_tmp, y_tmp, z_tmp);
wait_all_reach();
move_speed = 1;
body_right(15);
}
}

/*
- microservos service /timer interrupt function/50Hz
- when set site expected,this function move the end point to it in a straight line
- temp_speed[4][3] should be set before set expect site,it make sure the end point
  move in a straight line,and decide move speed.
-----*/
void servo_service(void)
{
    sei();
    static float alpha, beta, gamma;

    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (abs(site_now[i][j] - site_expect[i][j]) >= abs(temp_speed[i][j]))
                site_now[i][j] += temp_speed[i][j];
            else
                site_now[i][j] = site_expect[i][j];
        }

        cartesian_to_polar(alpha, beta, gamma, site_now[i][0], site_now[i][1], site_now[i][2]);
        polar_to_servo(i, alpha, beta, gamma);
    }

    rest_counter++;
}

/*
- set one of end points' expect site
- this function will set temp_speed[4][3] at same time
- non - blocking function
-----*/
void set_site(int leg, float x, float y, float z)
{
    float length_x = 0, length_y = 0, length_z = 0;

```

```

if (x != KEEP)
    length_x = x - site_now[leg][0];
if (y != KEEP)
    length_y = y - site_now[leg][1];
if (z != KEEP)
    length_z = z - site_now[leg][2];

float length = sqrt(pow(length_x, 2) + pow(length_y, 2) + pow(length_z, 2));

temp_speed[leg][0] = length_x / length * move_speed * speed_multiple;
temp_speed[leg][1] = length_y / length * move_speed * speed_multiple;
temp_speed[leg][2] = length_z / length * move_speed * speed_multiple;

if (x != KEEP)
    site_expect[leg][0] = x;
if (y != KEEP)
    site_expect[leg][1] = y;
if (z != KEEP)
    site_expect[leg][2] = z;
}

/*
- wait one of end points move to expect site
- blocking function
-----*/
void wait_reach(int leg)
{
    while (1)
        if (site_now[leg][0] == site_expect[leg][0])
            if (site_now[leg][1] == site_expect[leg][1])
                if (site_now[leg][2] == site_expect[leg][2])
                    break;
}

/*
- wait all of end points move to expect site
- blocking function
-----*/
void wait_all_reach(void)
{
    for (int i = 0; i < 4; i++)
        wait_reach(i);
}

/*
- trans site from cartesian to polar
- mathematical model 2/2
-----*/
void cartesian_to_polar(volatile float &alpha, volatile float &beta, volatile float
&gamma, volatile float x, volatile float y, volatile float z)
{

```

```

//calculate w-z degree
float v, w;
w = (x >= 0 ? 1 : -1) * (sqrt(pow(x, 2) + pow(y, 2)));
v = w - length_c;
alpha = atan2(z, v) + acos((pow(length_a, 2) - pow(length_b, 2) + pow(v, 2) + pow(z,
2)) / 2 / length_a / sqrt(pow(v, 2) + pow(z, 2)));
beta = acos((pow(length_a, 2) + pow(length_b, 2) - pow(v, 2) - pow(z, 2)) / 2 / length_a
/ length_b);
//calculate x-y-z degree
gamma = (w >= 0) ? atan2(y, x) : atan2(-y, -x);
//trans degree pi->180
alpha = alpha / pi * 180;
beta = beta / pi * 180;
gamma = gamma / pi * 180;
}

/*
- trans site from polar to microservalos
- mathematical model map to fact
- the errors saved in eeprom will be add
-----*/
void polar_to_servo(int leg, float alpha, float beta, float gamma)
{
if (leg == 0)
{
alpha = 90 - alpha;
beta = beta;
gamma += 90;
}
else if (leg == 1)
{
alpha += 90;
beta = 180 - beta;
gamma = 90 - gamma;
}
else if (leg == 2)
{
alpha += 90;
beta = 180 - beta;
gamma = 90 - gamma;
}
else if (leg == 3)
{
alpha = 90 - alpha;
beta = beta;
gamma += 90;
}

servo[leg][0].write(alpha);
servo[leg][1].write(beta);
servo[leg][2].write(gamma);
}

```


Приложение Г – Чертежи корпуса

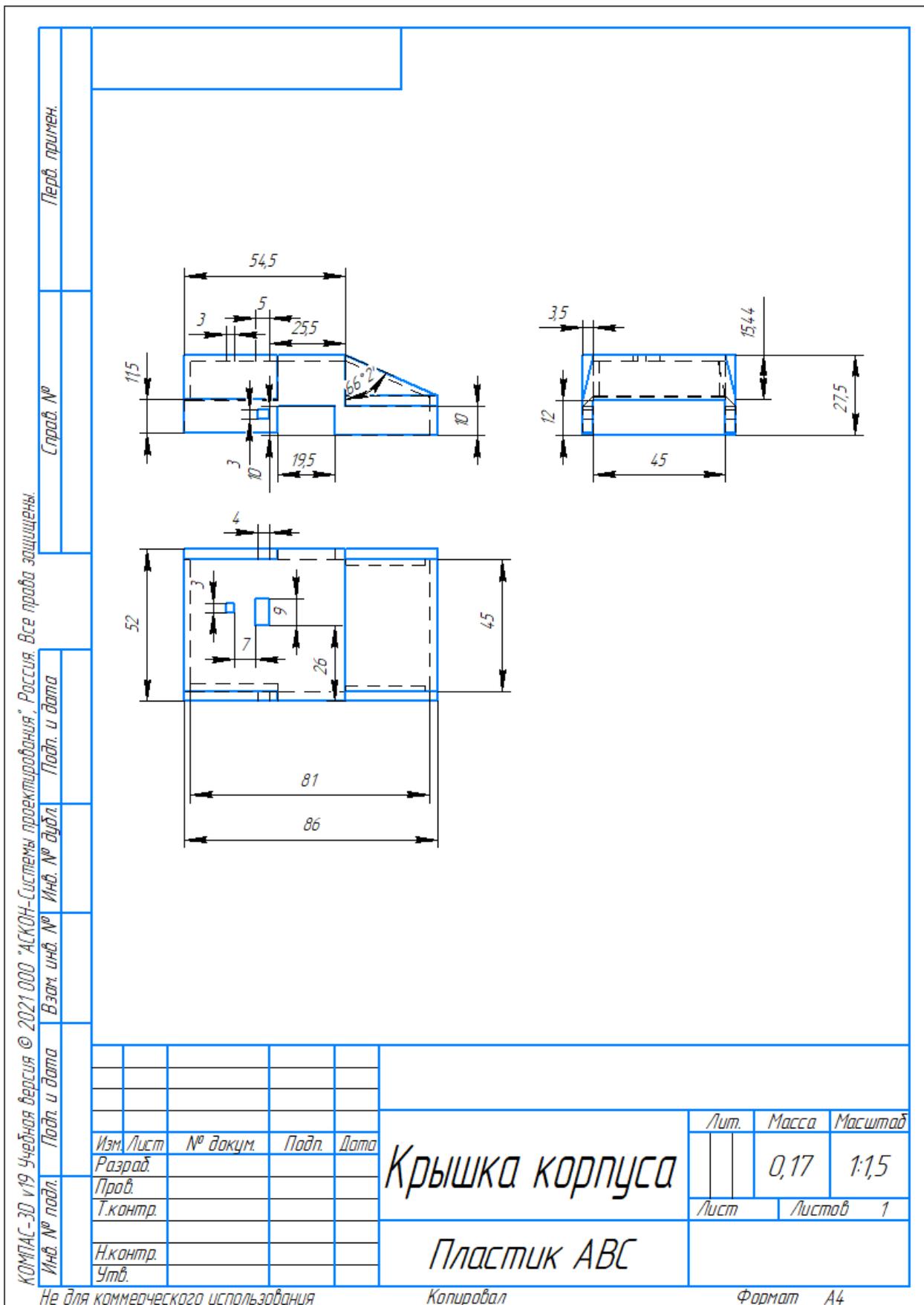


Рисунок 1 - Чертеж детали крышки корпуса

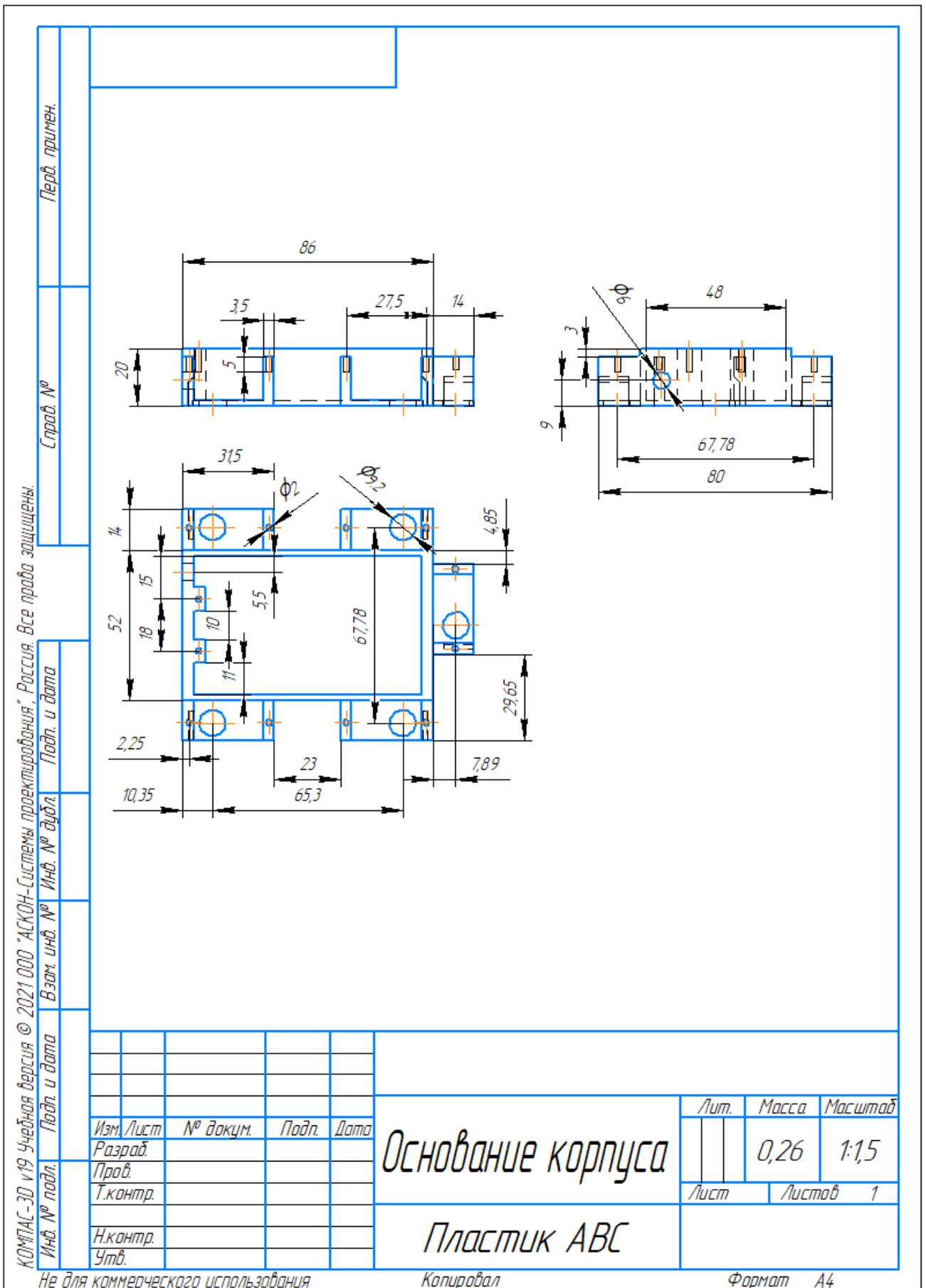


Рисунок 2 - Чертёж детали основания корпуса

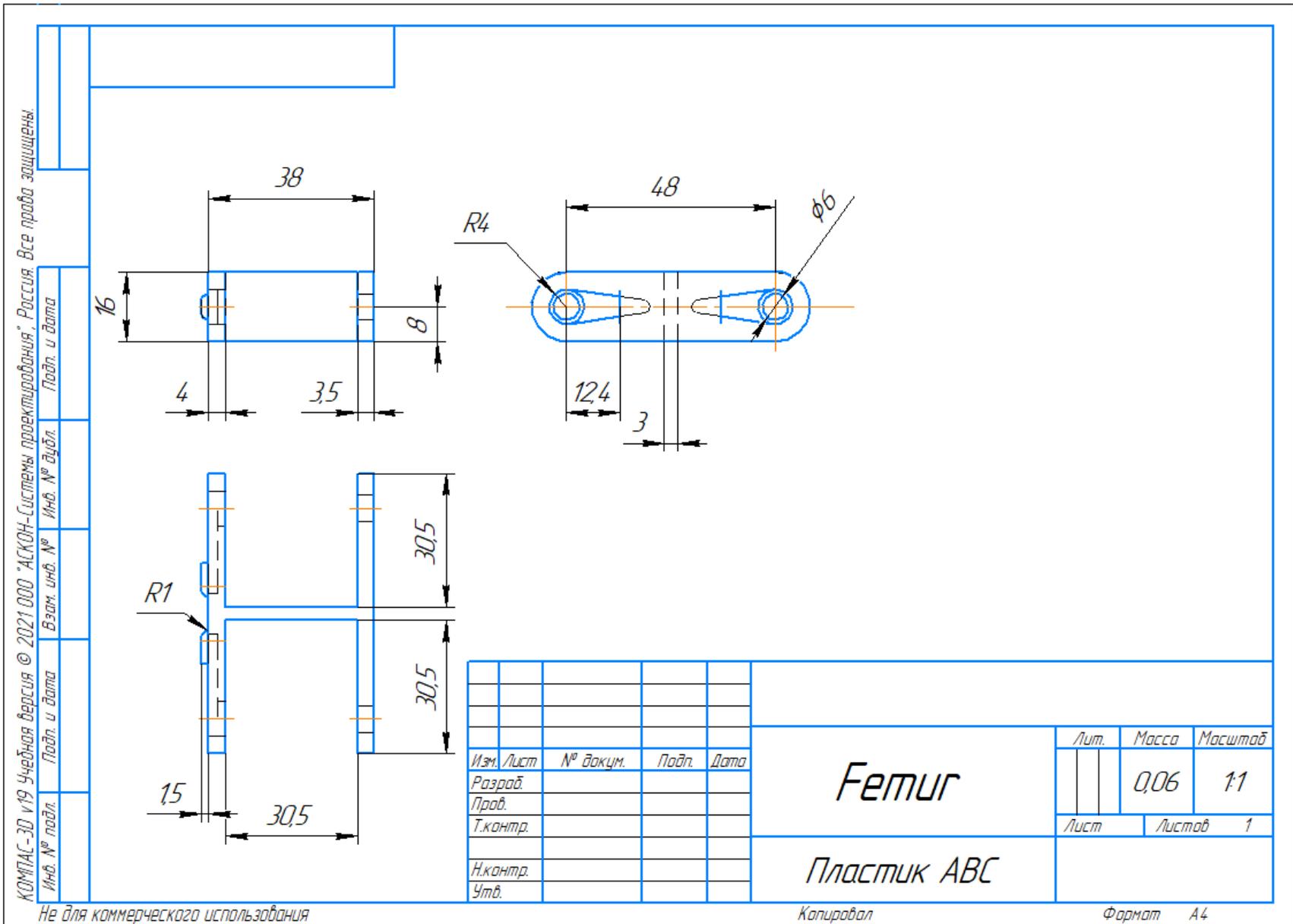


Рисунок 3 - Чертёж детали ноги Femur

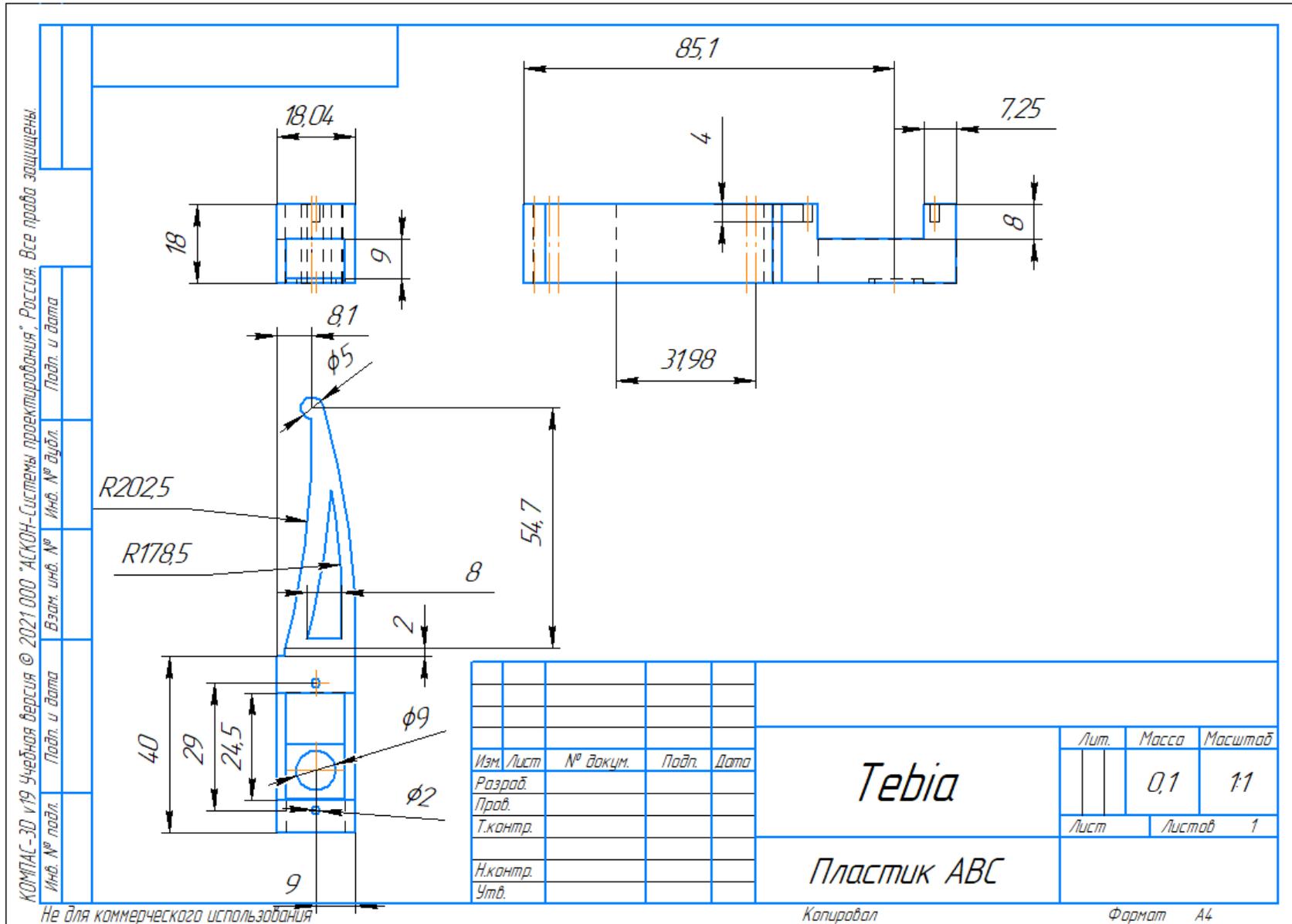


Рисунок 4 - Чертеж детали ноги Tebia

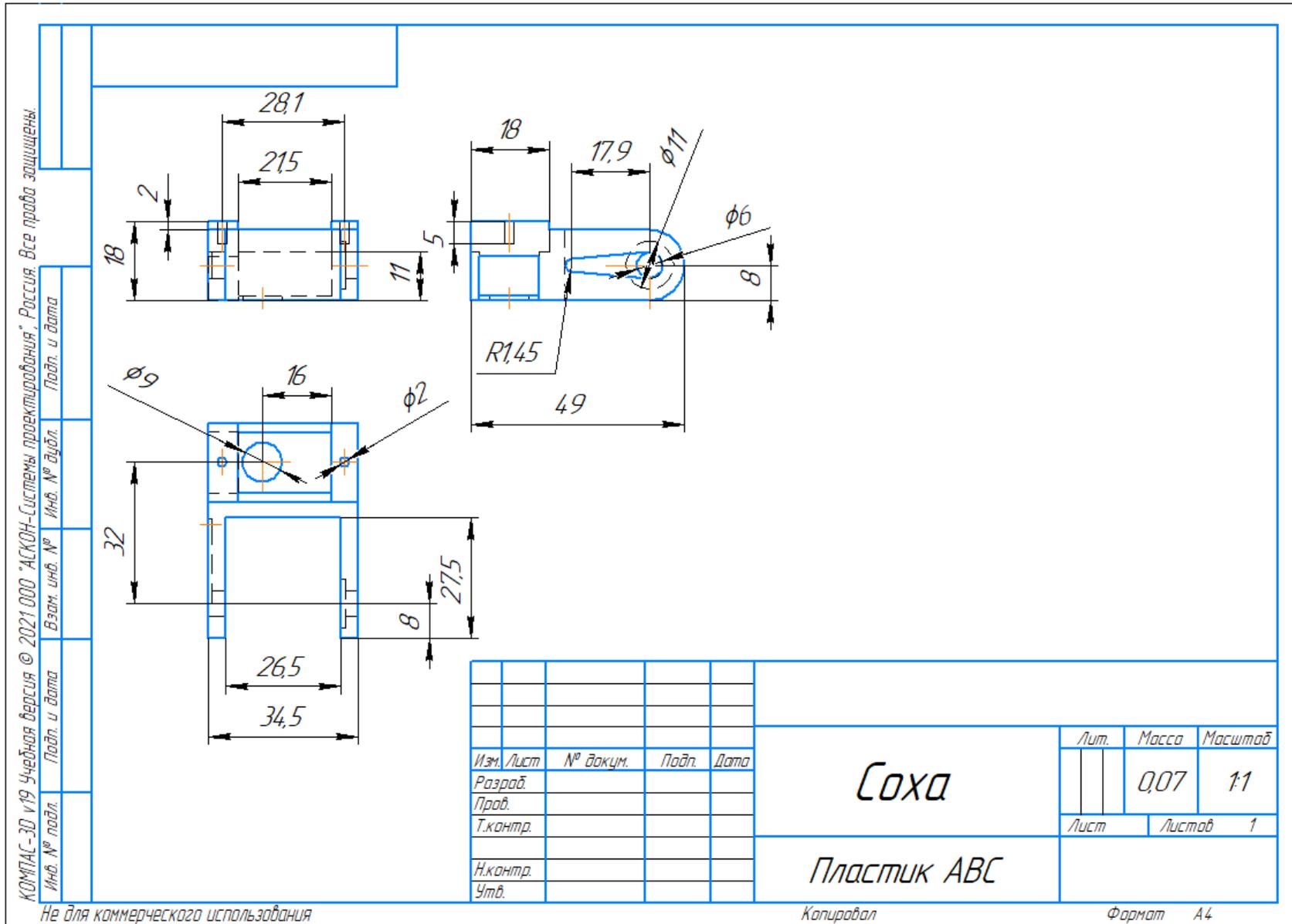


Рисунок 5 - Чертёж детали ноги Соха

Приложение Д – принципиальная схема устройства

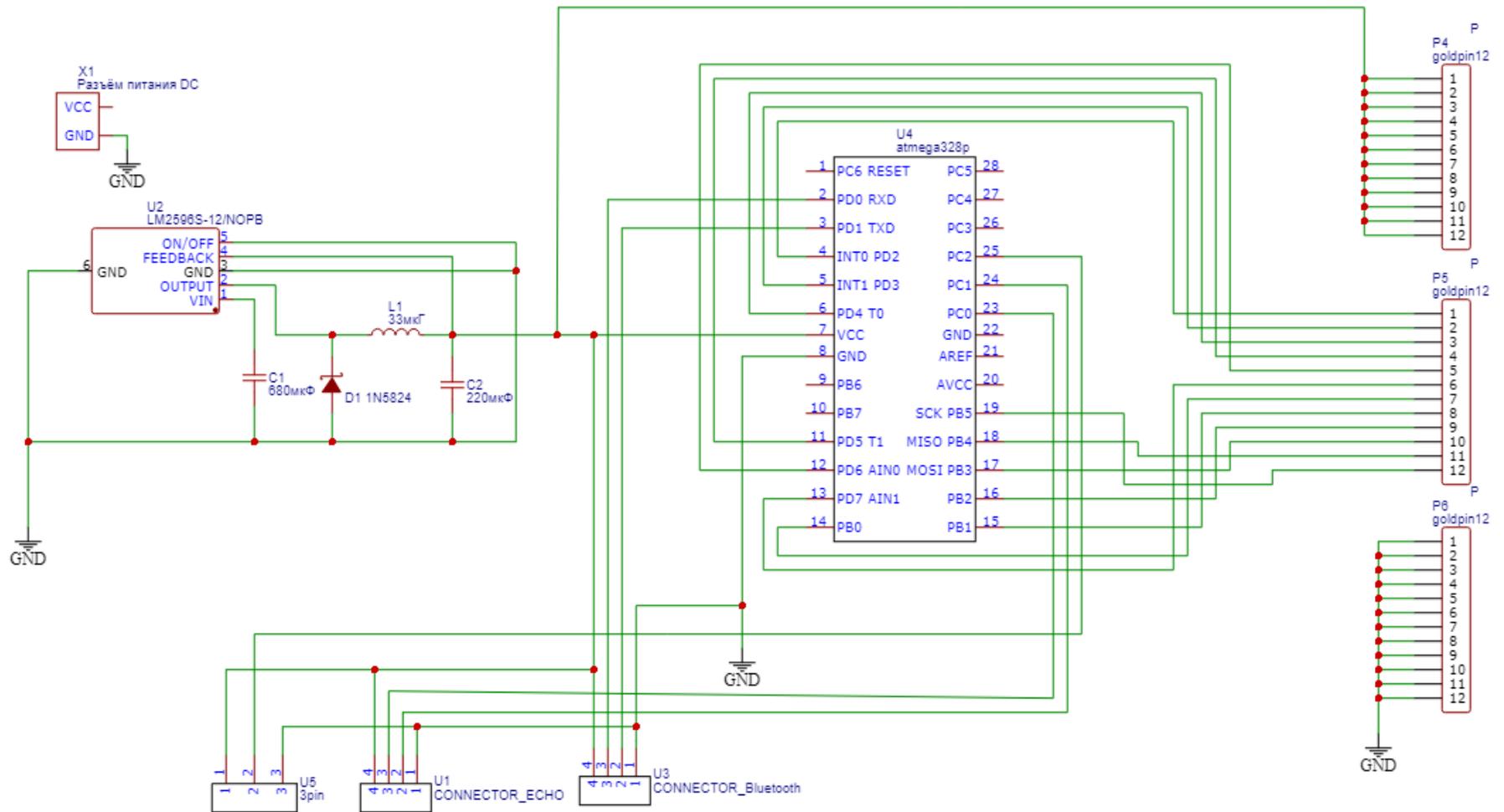


Рисунок 129 - Принципиальная схема устройства