

МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое Акционерное Общество «Казахский Национальный Исследовательский
Технический Университет имени К.И.Сатпаева»

Институт Автоматики и информационных технологий

Кафедра «Робототехники и технических средств автоматики»



Пахнюк Станислав Викторович

Разработка и построение гуманоидного робота с дистанционным управлением

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

Специальность 6В07111 – Робототехника и мехатроника

Алматы 2023

МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое Акционерное Общество «Казахский Национальный Исследовательский
Технический Университет имени К. И. Сатпаева»



Институт автоматизации и информационных технологий

Кафедра «Робототехники и технических средств автоматизации»

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой РТиТСА
кандидат техн.наук
_____ К. А. Ожикенов
« ___ » мая 2023 г.

ДИПЛОМНЫЙ ПРОЕКТ

Тему: «Разработка и построение гуманоидного робота с дистанционным управлением»

по специальности 6В07111 – Робототехника и Мехатроника

Выполнил

Рецензент

Балбаев Г. К.

РНД, ассоц. профессор

подпись

ФИО

« ___ » май 2023 ж.

Пахнюк Станислав

Научный руководитель

Магистр техн.наук,

старший преподаватель

_____ Баянбай Н.А.

« ___ » май 2023 ж.

Алматы 2023

МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое Акционерное Общество «Казахский Национальный Исследовательский
Технический Университет имени К. И. Сатпаева»



Институт автоматизации и информационных технологий

Кафедра «Робототехники и технических средств автоматизации»

УТВЕРЖДАЮ

Заведующий кафедрой РТиТСА

кандидат технических наук

Ожикенов К. А.

«__» _____ 2023 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Пахнюк Станислав Викторович

Тема: Разработка и построение гуманоидного робота с дистанционным управлением

Утверждена приказом Ректора Университета № _____ от «__» декабрь 2022г.

Срок сдачи законченной работы «__» мая 2023 г.

Исходные данные к дипломному проекту: Arduino IDE, Fritzing.

Перечень подлежащих разработке вопросов в дипломном проекте:

- а) Изучение принципа работы робота-гуманоида с дистанционным управлением;
- б) Анализ и сравнение характеристик человекоподобных роботов;
- в) Оценка эффективности существующих аналогов человекоподобных роботов и используемых технологии;
- г) Изучение возможности создания человекоподобных роботов.

Перечень графического материала (с точным указанием обязательных чертежей):

представлены 15 слайдов презентации работы

Рекомендуемая основная литература: 17 из наименований 17

ГРАФИК
подготовки дипломного проекта

| Наименование разделов, перечень разрабатываемых вопросов | Сроки представления научному руководителю | Примечание |
|--|---|------------|
| Исследовательская часть | 24.02.2023 | Выполнено |
| Практическая часть | 29.03.2023 | Выполнено |

Подписи

консультантов и нормоконтролера на законченный проект с указанием относящихся к ним разделов проекта

| Наименования разделов | Консультанты, И.О.Ф. (уч. степень, звание) | Дата подписания | Подпись |
|-----------------------|--|-----------------|---------|
| Нормоконтролер | Ж.С. Бигалиева, магистр технических наук, старший преподаватель | 24.05.2023 | |
| Основная часть | Баянбай Н.А., научный руководитель, магистр техн.наук, старший преподаватель | 24.05.2023 | |
| Расчетная часть | Баянбай Н.А., научный руководитель, магистр техн.наук, старший преподаватель | 24.05.2023 | |

Научный руководитель _____ Баянбай Н.А.

Задание принял к исполнению обучающийся _____ Пахнюк С.В.

Дата «__» май 2023 г

АНДАТПА

Бұл дипломдық жоба гуманоидты роботтың прототипін жасау болып табылады. Жұмыстың мақсаты – адам тәрізді робот пен қашықтан басқару жүйесін жасау. Тақырып адам тәрізді роботтарды зерттеу саласында өзекті болып табылады.

Зерттеу бөлімі ұқсас құрылғыларды, олардың конструкциялық ерекшеліктерін, қолданылуын, артықшылықтары мен кемшіліктерін зерттейді.

Тәжірибелік бөлімде қашықтан басқару жүйесі бар гуманоидты роботты, сондай-ақ микроконтроллер мен радио модульдерді пайдалана отырып, роботты басқару бағдарламасының практикалық орындалуы ұсынылған.

Дипломдық жұмыстың нәтижесінде доңғалақ базасы, басқару пульті, манипуляторлар және радиомодульдер арқылы гуманоид роботты қашықтан басқару мүмкін болды. Мұндай құрылғы үлкен және кіші қашықтықтағы объектілермен өзара әрекеттесу мүмкіндігіне ие және робототехника мен технологиядағы айтарлықтай прогресс болып табылады.

АННОТАЦИЯ

Данный дипломный проект заключается в построении прототипа человекоподобного гуманоидного робота.

В исследовательской части изучаются аналогичные устройства, их конструктивные особенности, сферы применения, достоинства и недостатки.

В практической части представлена реализация гуманоидного робота с дистанционной системой управления, а также программа управления роботом с использованием микроконтроллера и радиомодуля.

В результате дипломной работы появилась возможность дистанционного управления гуманоидного робота с использованием колесной базы, пульта управления, манипуляторов и радиомодулей. Такое устройство имеет потенциал для взаимодействия с объектами на больших и малых расстояниях и является значительным достижением в области робототехники и технологий.

ABSTRACT

This thesis project is to build a prototype of a humanoid robot.

The research part studies similar devices, their design features, applications, advantages and disadvantages.

The practical part presents the implementation of a humanoid robot with a remote control system, as well as a robot control program using a microcontroller and a radio module.

As a result of the thesis, it became possible to remotely control a humanoid robot using a wheelbase, a control panel, manipulators and radio modules. Such a device has the potential to interact with objects at large and small distances and is a significant advance in robotics and technology.

СОДЕРЖАНИЕ

| | |
|---|----|
| Введение | 7 |
| 1 Теоретическая часть | 8 |
| 1.1 Основное понятие гуманоидного робота | 8 |
| 1.2 Исследование человекоподобного робота | 9 |
| 1.2.1 Человекоподобный робот «Atlas» от Boston Dynamics | 9 |
| 1.2.2 Антропоморфный робот Petman | 10 |
| 1.2.3 Антропоморфный робот Pepper | 12 |
| 1.3 Методы управления человекоподобными роботами | 13 |
| 2 Практическая часть | 15 |
| 2.1 Разработка гуманоидного робота | 15 |
| 2.1.1 Разработка технической части гуманоидного робота | 15 |
| 2.1.2 Создание программы гуманоидного робота | 20 |
| 2.2 Кинематика гуманоидного робота | 26 |
| 2.3 Расчет надежности гуманоидного робота | 28 |
| Заключение | 31 |
| Список использованной литературы | 32 |
| Приложение А | |
| Приложение Б | |

ВВЕДЕНИЕ

Целью этой дипломной работы является, построение прототип робота-гуманоида, разработать систему дистанционного управления, подготовка программного алгоритма, проведение необходимых расчетов, исследование возможности реализации и улучшения конструктивных особенностей робота гуманоида.

Современные технологии и научные достижения привели к возникновению интересной и перспективной области исследований – разработке гуманоидного робота. Гуманоидный робот состоит из механических конструкций, созданные с целью имитирования формы и функциональности человеческого тела. Они обладают возможностью выполнять сложные движения, взаимодействовать с окружающей средой и даже воспринимать информацию.

Разработка гуманоидных роботов является междисциплинарной областью, которая объединяет знания и навыки из различных областей, таких как механика, электроника, компьютерное зрение, искусственный интеллект и многие другие.

Основная задача разработки гуманоидного робота заключается в достижении максимально возможного сходства с человеком в анатомической структуре и функциональности. Это требует разработки и интеграции сложных механических систем, электронных устройств и программного обеспечения. Кроме того, для эффективной работы гуманоидного робота необходимо разработать алгоритмы управления, которые обеспечат плавные и точные движения, а также способность адаптироваться к различным ситуациям и взаимодействовать с окружающим миром.

Исследования в области разработки гуманоидных роботов имеют широкий спектр приложений, от использования в робототехнике и промышленности до использования в медицине и развлекательной индустрии. Гуманоидные роботы могут быть использованы в медицинских операциях, помогать людям с ограниченными физическими возможностями, проводить исследования и обучение, а также развлекать и взаимодействовать с людьми.

Имея несколько способов управления (оператором вручную и автоматически), можно с легкостью расширить область применения гуманоидных роботов.

1 Теоретическая часть

1.1 Основные понятия гуманоидного робота

Гуманоидный робот (Рисунок 1.1.1), также известный как человекоподобный робот – тип робота, который сконструирован так, чтобы напоминать человеческое тело, как по внешнему виду, так и по движениям. Целью создания человекоподобных роботов является, разработка машин, которые могут взаимодействовать с людьми естественным и интуитивно понятным способом, что делает их полезными в широком спектре приложений, от развлечений и образования до здравоохранения и поисково-спасательных операций. [1]



Рисунок 1.1.1 – Гуманоидный робот

Гуманоидные роботы обычно имеют сложную структуру, состоящую из множества суставов, двигателей и датчиков, которые позволяют им двигаться и выполнять задачи, похожие на человеческие. Они также часто оснащены усовершенствованными датчиками, такими как камеры и микрофоны, которые позволяют им воспринимать окружающую среду и взаимодействовать с ней.

Одной из ключевых задач при разработке человекоподобных роботов является создание машин, способных сохранять равновесие и устойчивость при выполнении сложных движений. Для этого требуется сложная система управления, которая может управлять сложными взаимодействиями между суставами робота, двигателями и датчиками. [14]

Еще одной проблемой является создание роботов, которые могут двигаться и вести себя естественным и интуитивно понятным для человека образом. Это требует интеграции передовых алгоритмов машинного обучения, таких как глубокие нейронные сети, которые могут учиться на человеческих примерах и имитировать человеческое поведение.

Несмотря на эти проблемы, был достигнут значительный прогресс в разработке человекоподобных роботов. Сегодня человекоподобные роботы используются в различных сферах деятельности: развлечении, образовании, здравоохранении и в поисково-спасательных операциях. [2]

1.2 Исследование человекоподобных роботов

1.2.1 Человекоподобный робот «Atlas» от Boston Dynamics

В сентябре 2021 года Boston Dynamics разработала робота-гуманоида по имени «Atlas» (Рисунок 1.2.1.1).

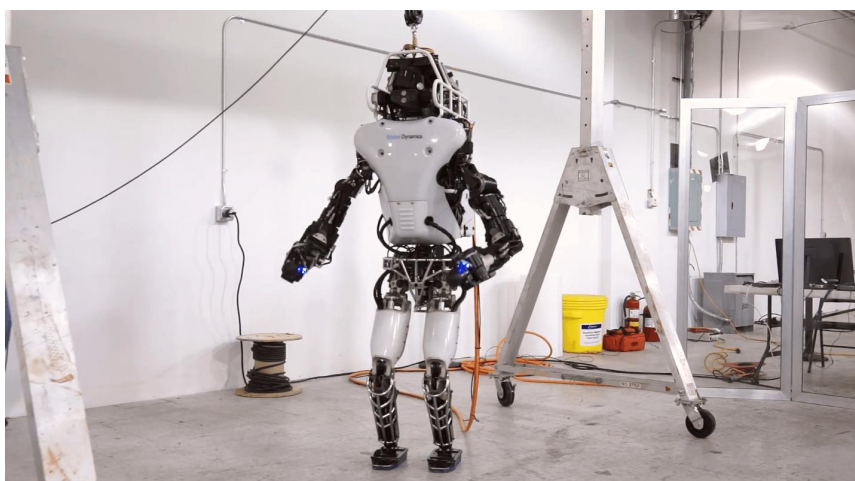


Рисунок 1.2.2.1– Исследование устойчивости робота Atlas

Первым делом в ходе исследовательской работы давайте затронем робота «Atlas». Он разработан, чтобы имитировать общие пропорции и движения человеческого тела. Робот имеет высоту около 1,5 метра и имеет двуногую структуру с двумя руками и ногами, что позволяет ему более естественно ориентироваться и взаимодействовать с окружающей средой. Использует комбинацию электрических, гидравлических и пневматических приводов для обеспечения своих движений. Суставы робота оснащены датчиками и моторами, что позволяет точно контролировать его движения и позволяет выполнять широкий диапазон динамических движений. «Atlas» оснащен различными датчиками, включая камеры, лидары и датчики силы. Эти датчики наделяют робота способностями к восприятию, позволяя ему воспринимать и понимать свое окружение, обнаруживать объекты и сохранять равновесие. Включает в себя встроенный источник питания и предназначен для мобильности. Он может ходить, бегать и выполнять динамические маневры, что позволяет ему перемещаться по разным ландшафтам и средам. [17]

«Atlas» очень универсален и может быть запрограммирован для выполнения широкого спектра задач и движений. Его человекоподобная форма

и ловкость делают его подходящим для приложений, требующих сложного взаимодействия с окружающей средой, таких как реагирование на стихийные бедствия, поисково-спасательные операции или промышленные задачи. Позволяет выполнять физически сложные задачи, которые были бы сложными или опасными для человека. Его прочная конструкция и приводы позволяют поднимать, переносить и манипулировать объектами со значительной силой и точностью. Также служит платформой для исследований и разработок в области робототехники. Предоставляет из себя исследовательскую и инженерную машину, которую можно настраивать, тестировать и интегрировать с различными алгоритмами и системами управления для развития области робототехники.

Создание и обслуживание гуманоидных роботов, таких как «Atlas», может быть сложным и дорогим из-за сложного оборудования, датчиков и систем управления. Передовые технологии, используемые в их конструкции, могут сделать их недоступными для определенных организаций с ограниченными ресурсами.

Гуманоидные роботы обычно требуют значительного количества энергии для работы. Бортовой источник питания «Atlas» требует тщательного управления, чтобы обеспечить достаточную энергию для продолжительной работы, особенно при выполнении задач, требующих физических усилий.

Баланс и стабильность. Поддержание баланса и стабильности при выполнении динамических движений может быть сложной задачей для роботов-гуманоидов. Хотя у «Atlas» есть передовые датчики и системы управления для решения этой проблемы, он все же может столкнуться с трудностями в определенных ситуациях или на местности.

Стоит отметить, что область робототехники быстро развивается, как следствие это приведет к появлению новых поколений роботов серии Atlas.

1.2.1 Антропоморфный робот Petman

Стоит по больше узнать о антропоморфном роботе «Petman» (Рисунок 1.2.1), разработанный Boston Dynamics. Основная задача данного робота состоит в том, чтобы тестировать специальную одежду и костюмы индивидуальной защиты от различных внешних физических, химических и радиоактивных воздействий. Человекоподобная форма и движения робота позволяют реалистично моделировать взаимодействие этих предметов с человеческим телом в различных сценариях.



Рисунок 1.2.1.1 – Антропоморфный робот Petman

«Petman» служит исследовательской платформой для изучения человеческого передвижения, баланса и динамических движений. Это позволяет исследователям исследовать и улучшать алгоритмы управления, интеграцию датчиков и взаимодействие человека и робота. Робот имитирует размеры, пропорции и движения человека, имеет двуногую структуру с двумя руками и ногами, что позволяет ему выполнять действия, подобные человеческим, и взаимодействовать с окружающей средой. Оснащены датчиками и гидравлическими приводами, что позволяет точно контролировать его движения. Использование гидравлики обеспечивает динамичное и отзывчивое движение, обеспечивая большую силу и маневренность. Данный робот включает в себя ряд датчиков, в том числе датчики силы, камеры захвата движения и датчики окружающей среды. Эти датчики позволяют роботу воспринимать окружающую среду и адаптироваться к ней, сохраняя равновесие и взаимодействуя с объектами. Робот может ходить, бегать, сидеть на корточках и выполнять различные физические движения с естественной плавностью. Его возможности динамического движения делают его подходящим для приложений, требующих человеческого передвижения и физического взаимодействия. [15]

«Petman» обладает реалистичными двигательными функциями. Это позволяет проводить более точные испытания и исследования в областях, связанных с биомеханикой человека и физическим взаимодействием. Технологичное оснащение робота позволяет записывать данные с датчиков в реальном времени и с высокой точностью.

Конструкция робота «Petman» является сложной и весьма специфичной поэтому может не подойти для многих задач. Сложность конструкции делает данного робота очень дорогостоящим устройством. Это может ограничить их доступность и удобство использования для определенных организаций или исследовательских институтов.

Остается надеяться, что увеличится производство расходных материалов и основных узлов подобных роботов для их дальнейшего распространения и большим уровнем доступности.

1.2.3 Антропоморфный робот Pepper

Не мало внимания заслуживает антропоморфный робот «Pepper» (Рисунок 1.2.3.1), разработанный SoftBank Robotics. Он разработан, чтобы быть социальным компаньоном и оснащен различными функциями для взаимодействия с людьми. В основном используется для взаимодействия человека и робота в различных условиях, таких как розничные магазины, гостиничный бизнес, образование и здравоохранение. [16]



Рисунок 1.2.3.1 – Гуманоидный робот Pepper

Данный робот может участвовать в беседах, предоставлять информацию, развлекать и помогать клиентам или пациентам. Развертывается в розничной торговле, чтобы приветствовать клиентов и помогать им, предоставлять информацию о продуктах, направлять их в определенные разделы и обрабатывать простые транзакции. «Pepper» также может собирать отзывы клиентов и выполнять базовый анализ данных. Иногда используется в образовательных учреждениях для облегчения интерактивного обучения. Робот может вовлекать студентов в образовательную деятельность, предоставлять информацию, а также предоставлять услуги репетиторства или помощь в изучении языка. Кроме того, робот «Pepper» способен развлекать людей играми, танцами и рассказыванием историй.

«Pepper» имеет гуманоидную форму ростом примерно 1,2 метра. Корпус робота окрашен преимущественно в белый, имеет округлую голову, выразительные глаза и планшеты с интерфейсом на груди. Оснащен несколькими датчиками, включая камеры, микрофоны, сенсорные датчики и датчики приближения. Эти датчики позволяют ему воспринимать и интерпретировать человеческое выражение лица, жесты и голосовые команды.

«Pepper» обладает возможностями синтеза речи, что позволяет ему общаться с людьми с помощью синтезированного голоса. Робот также может распознавать и понимать человеческую речь с помощью своей системы распознавания голоса. У данного робота есть колесная база, которая позволяет ему автономно перемещаться в окружающей среде

«Pepper» предназначен для создания более увлекательного и интерактивного опыта для людей. Его гуманоидная форма, речевые возможности и выразительные черты способствуют естественному и дружелюбному общению, что делает его подходящим для социальных условий. Помогает при изучении для повышения навыков программирования и разработки программного обеспечения, позволяет настраивать и создавать различные приложения, приспособленные к конкретным потребностям. Эта гибкость позволяет использовать «Pepper» в разных отраслях и адаптироваться к различным вариантам использования.

Мобильность «Pepper» ограничена его колесной базой, что ограничивает его способность перемещаться в определенных средах или выполнять физические задачи, требующие более продвинутых манипулятивных возможностей. Роботу требуется стабильное подключение к Интернету для доступа к облачным службам, обновлениям программного обеспечения и возможностям обработки языка. Потеря подключения может повлиять на его функциональность и производительность.

1.3 Методы управления человекоподобными роботами

Существует несколько способов управления роботами-гуманоидами в зависимости от сложности робота, желаемого уровня автономности и конкретного применения. Самое простое – прямое ручное управление включает в себя человека-оператора, использующего такое устройство, как джойстик, геймпад или экзоскелет, для управления движениями робота в режиме реального времени. Вводы оператора преобразуются в команды, управляющие соединениями и исполнительными механизмами робота.

Телеоперация позволяет человеку-оператору управлять роботом на расстоянии. Действия оператора передаются роботу по линии связи, а робот повторяет движения оператора. Этот метод часто используется в сценариях, когда роботу необходимо выполнять задачи в опасных или недоступных средах. Также существуют системы захвата движения, их можно использовать для отслеживания движений человека-оператора и отображения их на роботе. При

ношении специальных маркеров или использовании камер с определением глубины действия оператора фиксируются и преобразуются в соответствующие движения робота.

Гуманоидных роботов можно запрограммировать на выполнение определенных действий или выполнение predetermined последовательности движений. Языки программирования и программные платформы, специально разработанные для робототехники, такие как ROS (Robot Operating System), предоставляют инструменты и библиотеки для программирования поведения роботов, планирования движения и управления. Гуманоидные роботы оснащены различными датчиками, включая камеры, датчики глубины и датчики силы крутящего момента. Эти датчики позволяют роботу воспринимать окружающую среду и реагировать соответствующим образом. Данные датчиков можно обрабатывать и использовать для управления движениями робота и его взаимодействием с окружающей средой.

Усовершенствованные гуманоидные роботы могут работать автономно, принимая решения и выполняя задачи без прямого вмешательства человека. Автономное управление включает в себя комбинацию алгоритмов восприятия, планирования и управления, что позволяет роботу ориентироваться в окружающей среде, распознавать объекты и адаптировать свои действия для достижения своих целей. Методы машинного обучения и искусственного интеллекта могут использоваться для обучения роботов-гуманоидов выполнению задач и улучшения их контроля. Используя подходы, основанные на данных, роботы могут учиться на демонстрациях, пробах и ошибках или симуляциях, чтобы приобретать новые навыки и адаптировать свое поведение.

Важно отметить, что различные методы управления могут использоваться в комбинации, в зависимости от требований конкретного приложения. Выбор метода управления зависит от таких факторов, как необходимый уровень точности, сложность задачи, взаимодействие с окружающей средой и желаемый уровень автономии гуманоидного робота.

1 Практическая часть

1.1 Разработка гуманоидного робота

Опасная и тяжелая работа на многих участках производства может показаться человеку слишком сложной и опасной, а когда работу человека нужно передать роботам сразу можно представить, что этот робот выглядит и работает также, как человек или даже лучше. Гуманоидные роботы могут быть использованы для исследования и выполнения задач в космосе, подводных областях, местах с повышенным риском или недоступных местах. Также могут быть использованы для проведения экспериментов и исследований в области социального взаимодействия, психологии и когнитивных наук. Они могут помочь в понимании человеческого поведения, эмоций и влияния. Исходя из этого с высокой вероятностью можно сказать, что человек не только создает себе помощника, но и пытается узнать и познать себя.

2.1.1 Разработка технической части гуманоидного робота

Разработка любых устройств и систем всегда берет основу из задокументированного опыта других людей и личного опыта конструктора или изобретателя. Исходя из проведенного исследования и учета возможностей можно приступить к выбору необходимых приводов, контроллеров, комплектующих, модулей и других устройств. [5]

На построение шагающего робота уходит много времени и сил, поэтому в таком случае идеальным вариант – это колесная база, которая будет выступать в роли ходовой части робота. Ходовая часть гуманоидного робота с колесами часто предназначена для обеспечения устойчивости и мобильности робота, позволяя ему двигаться и балансировать на различных поверхностях. [10]

Конструкция ходовой части имеет решающее значение для общей производительности робота, поскольку она влияет на скорость, маневренность и способность робота перемещаться по разным ландшафтам. Так как мы будем иметь небольшой размер прототипа гуманоидного робота, следовательно, мы выберем небольшие модули мотор-редуктора с коллекторным двигателем и колесными парами.

Для осуществления передвижения в пространстве будем использовать мотор-редуктор. В дальнейшем мы сможем менять скорость робота используя программу и ШИМ модуляцию. [11]

Любой робот имеет в своем составе контроллер для обработки поступающей информации и отправки командных сигналов на приводы, модули и устройства. В связи с тем, что имеются много видов различных микроконтроллеры, то стоит акцентировать внимание на более дешевые аналоги. Имея легкий способ программирования и подключения, семейство

микроконтроллеров Arduino (Рисунок 2.1.1.1) является идеальным вариантом для дальнейшего выполнения поставленных задач.



Рисунок 2.1.1.1 – Arduino Mega 2560 R3

Так как прототип состоит из колесной базы поэтому будет считаться мобильным, у мобильных роботов отсутствует питание электрической схемы напрямую от сети, поэтому будем использовать литий-ионные батареи. Четыре литий-ионные батареи мы объединяем в параллельное и последовательное подключение создавая тем самым небольшой аккумулятор с выходным напряжением примерно в 7.5В.

Имея мотор-редуктор нам необходимо управлять скоростью вращения двигателем, то есть управлять передвижением самого робота. Для таких целей используют драйвер двигателя, он поможет не только установить количество оборотов вращения двигателя, но и указать направления вращения. В нашем случае это будет драйвер двигателя L298N (Рисунок 2.1.2.2), так как он наиболее приспособлен для работы с микроконтроллерами семейств Arduino.

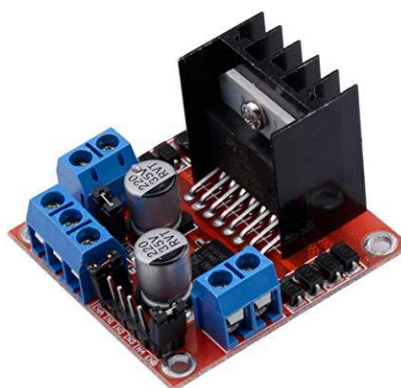


Рисунок 2.1.1.2– Драйвер двигателя L298N

Для обмена информацией между роботом и оператором можно выбрать не мало способов связи, но я решил остановиться именно на радиоканале за счет его большей надежности, экономичности и дальности передачи управляемого сигнала. Для обмена данных между микроконтроллерами семейств Arduino,

можно использовать радио модуль 1NRF24L01 с усилителем и антенной для увеличения дальности связи.

Так как робот у нас считается гуманоидным, значит его исключительной особенностью является наличие манипуляторов, имитирующие руки или хотя бы похожи на них, и выполняют схожую функцию. Для управления такими типами манипуляторов хорошо подойдут серводвигатели или шаговые двигатели, но так как нет дешевых аналогов шаговых двигателей то мы остановимся на простых и компактных серводвигателях таких как SG90 которые отлично работают с микроконтроллерами типа Arduino.

Для создания прототипа разрабатываем основные части корпуса и ходовой части робота. Так как исходя из прочностных и экономических соображений выбор пал на ПВХ пластик то корпус будет изготовлен из него.

Используя штангенциркуль замеряем размеры всех используемых модулей для создания оптимального размера частей корпуса. После того как мерки были сняты расчерчиваем на пластике линии, по которым вырезаем основные детали корпуса.

Используя различные крепежные материалы, сооружаю основную часть корпуса и монтирую колесную базу (Рисунок 2.1.1.3). Основная часть электроснабжения двигателей будет находиться внутри корпуса поэтому вывожу провода питания двигателя во внутрь основной части.

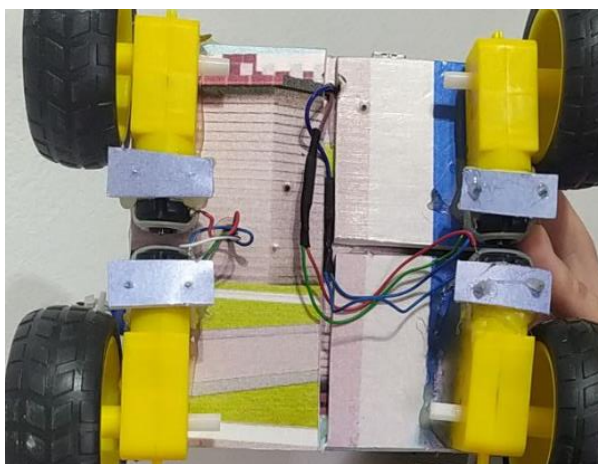


Рисунок 2.1.1.3 – Колесная база робота

После подготовки корпуса можем размещать все электронные и механические компоненты внутри корпуса и фиксировать их положение крепежными материалами (Рисунок 2.1.2.4)

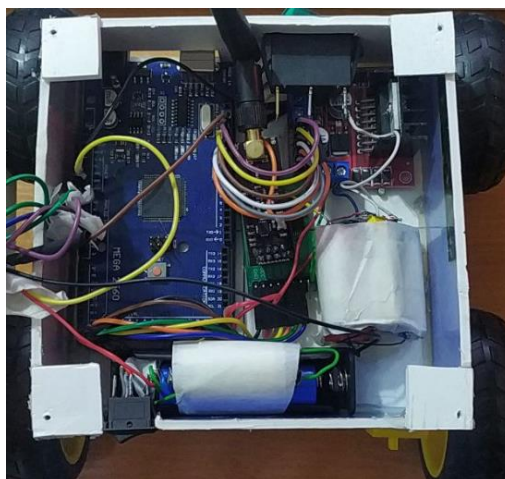


Рисунок 2.1.1.4 – Размещение и подключение электронных компонентов

После установки в корпус всех необходимых комплектующих идет коммутация по следующим схемам (Рисунок 2.1.1.5) (Рисунок 2.1.1.6).[3]

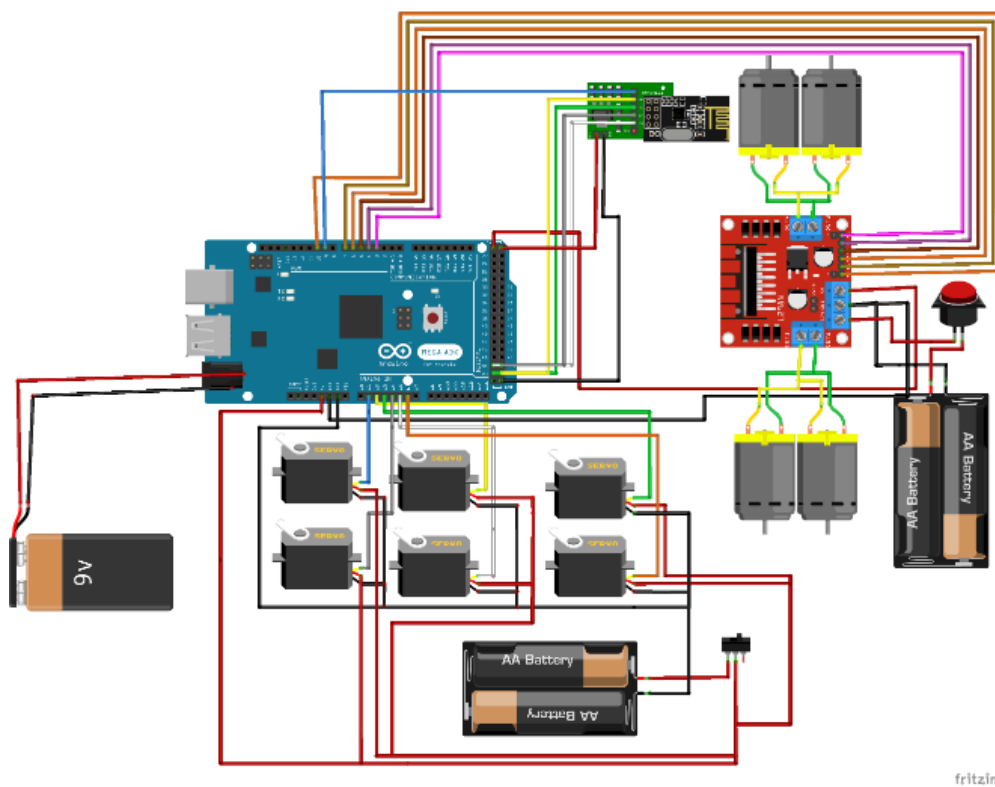


Рисунок 2.1.1.5 – Схема подключения человекоподобного робота

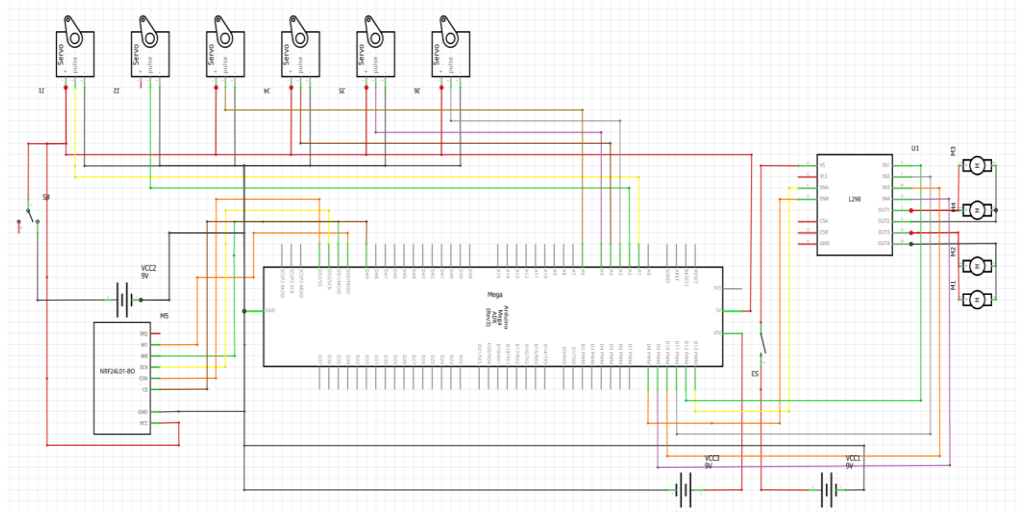


Рисунок 2.1.1.6 – Электрическая схема человекоподобного робота

В итоговом варианте мы получим собранный, но пока не рабочий прототип робота. Для дальнейшей работоспособности готовой модели нам необходим пульт управления. Только после этого можем браться за написание программы. [13]

Пульт управления будем создавать из таких же материалов и по такому же принципу, как самого робота по следующим схемам (Рисунок 2.1.1.7) (Рисунок 2.1.1.8).

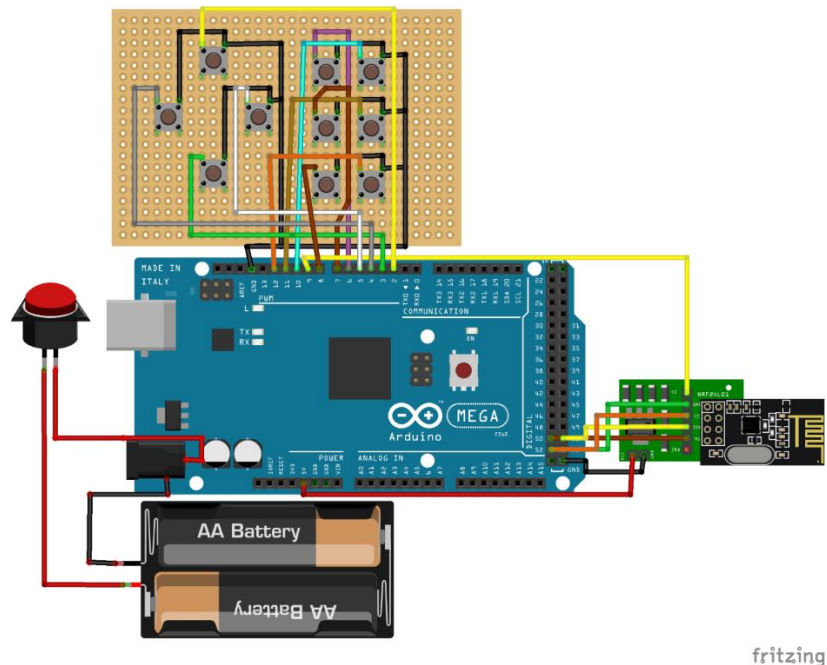


Рисунок 2.1.1.7 – Схема подключения пульта управления

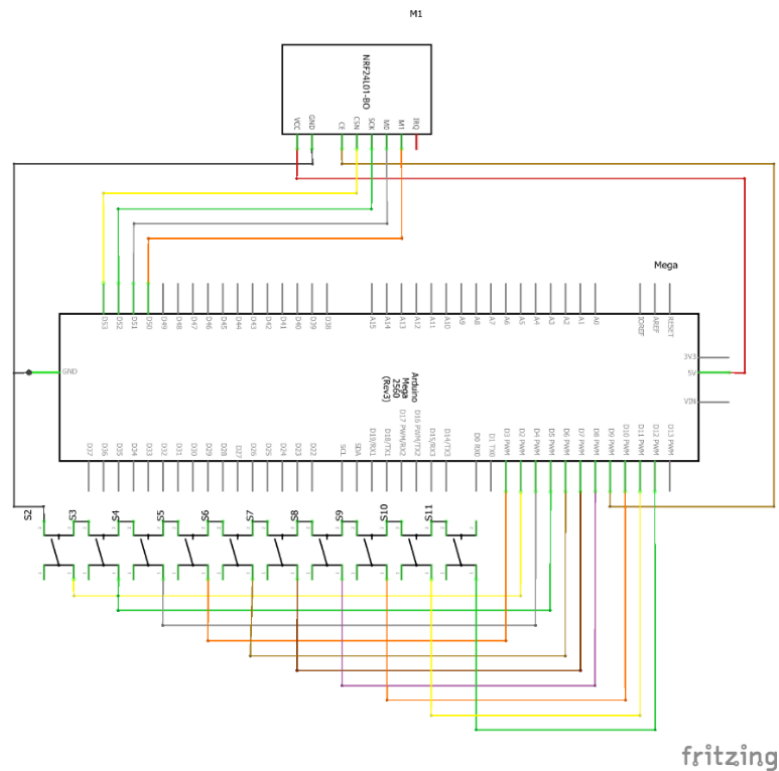


Рисунок 2.1.1.8 – Электрическая схема пульта управления

В итоге мы получаем два устройства, которые необходимо будет лишь настроить и запрограммировать.

2.1.2 Создание программы гуманоидного робота

Для того чтобы начинать управлять передачей команд через радиоканал нам необходимо написать программу, записывающую команду в массив данных которая будет отправлена через радиоканал, предварительно необходимо настроить частоту вещания радиоканала для предотвращения помех и ошибок. Первым делом напишем код для управляющего устройства, а именно для пульта управления. Делать это будем в среде разработки Arduino IDE.[4]

Для работы программы мы инициализируем используемые библиотеки и настраиваем основной и запасной каналы связи (Рисунок 2.1.2.1).

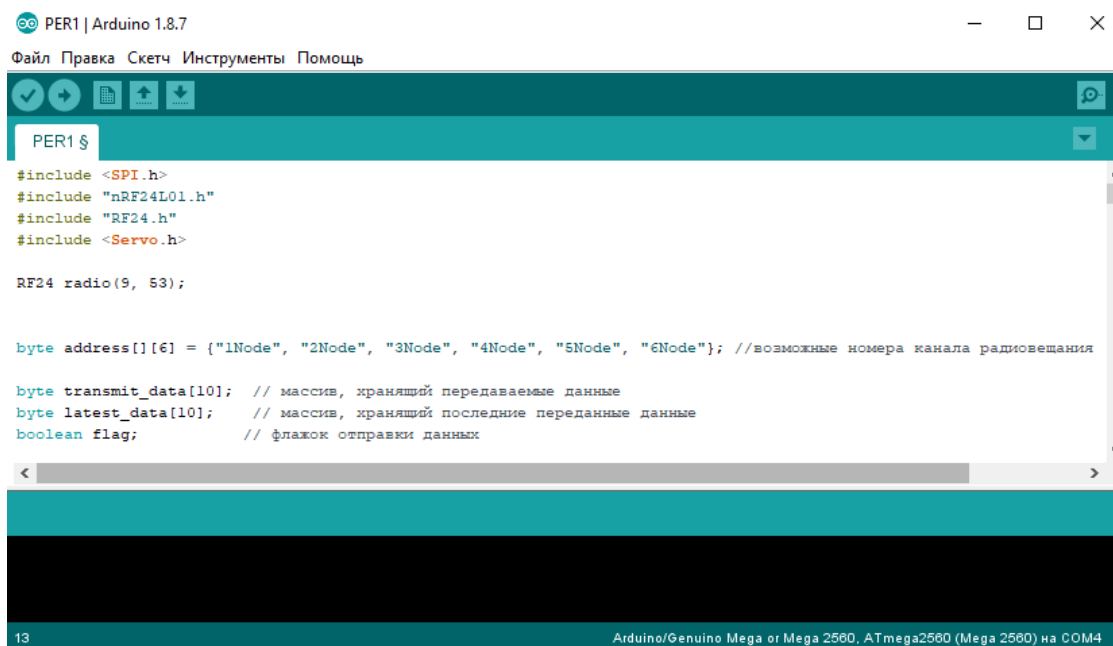


Рисунок 2.1.2.1 – Настройка библиотек и радиоканалов вещания

Следующим шагом нам необходимо указать устройства ввода и указать тип получения или отправки сигнала (Рисунок 2.1.2.2).

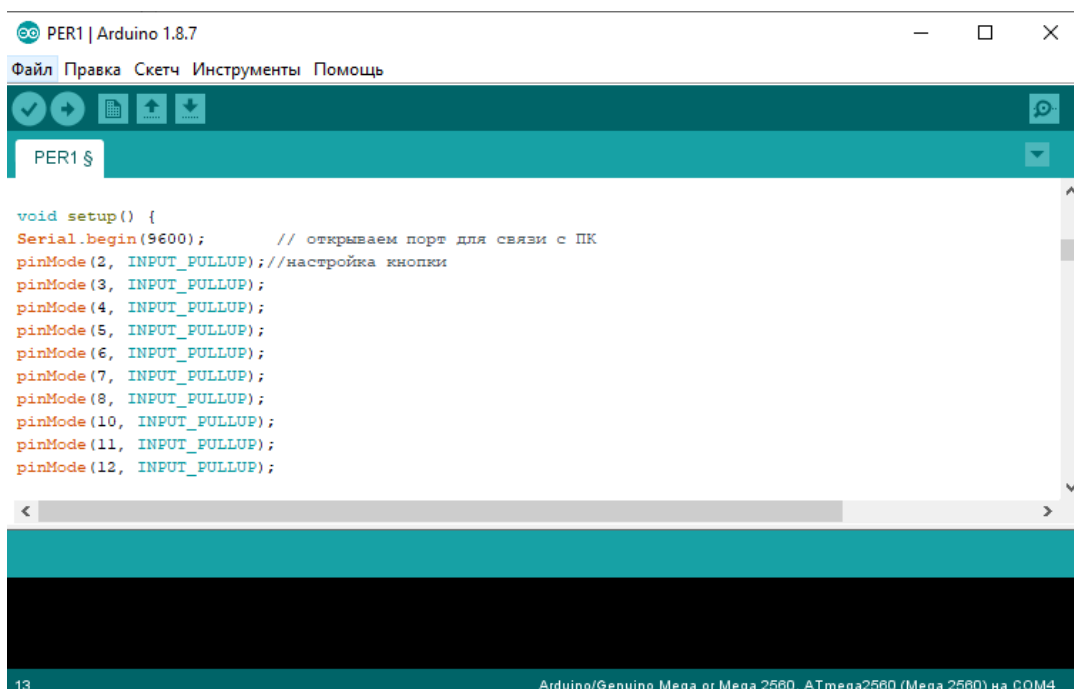
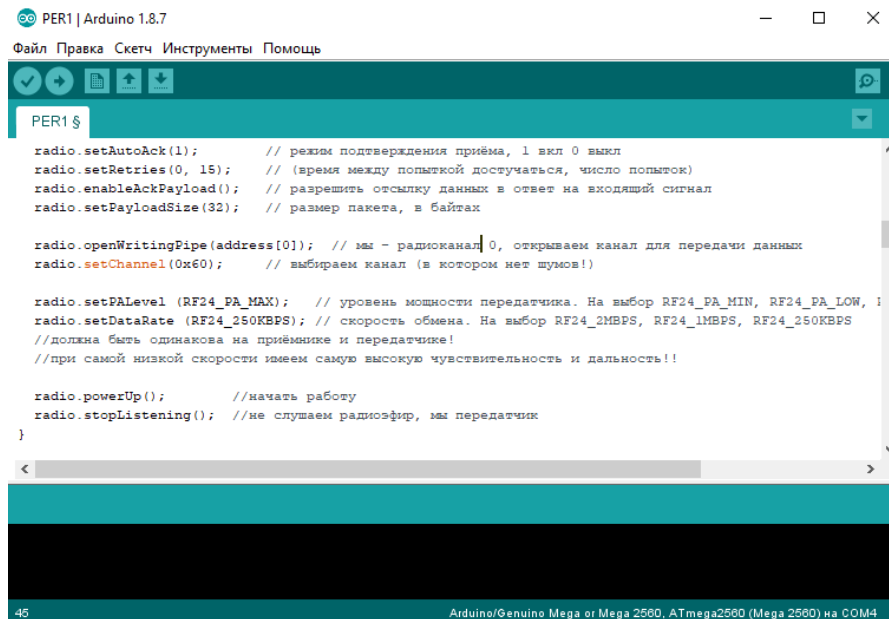


Рисунок 2.1.2.2 – Настройка подключенных устройств ввода

После этого нам необходимо завершить настройку радио-модуля, а именно повысить мощность сигнала, указать время ожидания сигнала, установить канал связи не имеющий шумов при радиовещании и указываем, что мы являемся передатчиком сигнала, а не его приемником (Рисунок 2.1.2.3).



```
PER1 | Arduino 1.8.7
Файл Правка Скetch Инструменты Помощь
PER1 §
radio.setAutoAck(1); // режим подтверждения приёма, 1 вкл 0 выкл
radio.setRetries(0, 15); // (время между попыткой достучаться, число попыток)
radio.enableAckPayload(); // разрешить отсылку данных в ответ на входящий сигнал
radio.setPayloadSize(32); // размер пакета, в байтах

radio.openWritingPipe(address[0]); // мы - радиоканал 0, открываем канал для передачи данных
radio.setChannel(0x60); // выбираем канал (в котором нет шумов!)

radio.setPALevel (RF24_PA_MAX); // уровень мощности передатчика. На выбор RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH, RF24_PA_BOOST
radio.setDataRate (RF24_250KBPS); // скорость обмена. На выбор RF24_2MBPS, RF24_1MBPS, RF24_250KBPS
//должна быть одинакова на приёмнике и передатчике!
//при самой низкой скорости имеем самую высокую чувствительность и дальность!!

radio.powerUp(); //начать работу
radio.stopListening(); //не слушаем радиозфир, мы передатчик
}
45 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) на COM4
```

Рисунок 2.1.2.3 – Настройка передачи данных по радиоканалам

Создаем переменные и присваиваем им сигнал, считывающийся с нажатой кнопки. Записываем условие при которых в переменные будут записываться определенные значения (Рисунок 2.1.3.4).

```
}

void loop() {
  boolean a= !digitalRead(2);
  boolean b= !digitalRead(3);
  boolean c= !digitalRead(4);
  boolean d= !digitalRead(5);
  boolean e= !digitalRead(6);
  boolean f= !digitalRead(7);
  boolean g= !digitalRead(8);
  boolean h= !digitalRead(10);
  boolean i= !digitalRead(11);
  boolean j= !digitalRead(12);

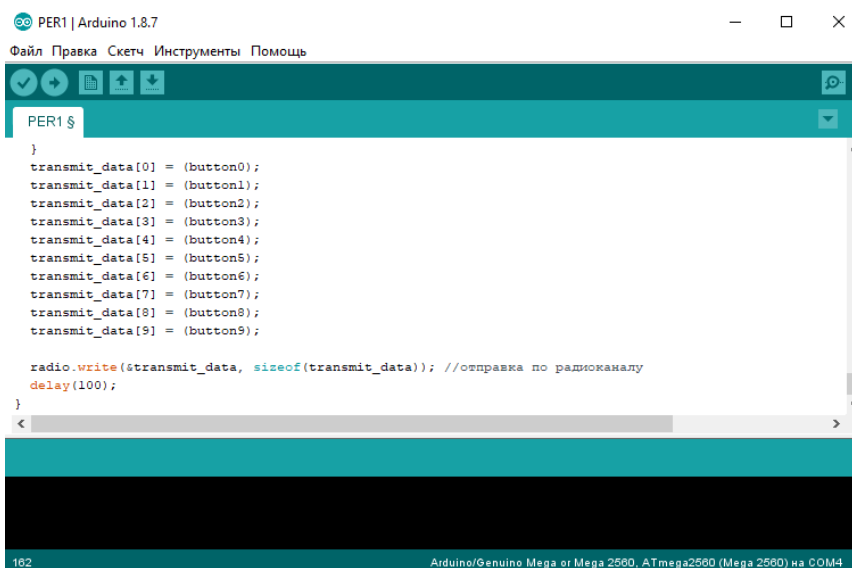
  if (a>0)
  {
    button0 = 1;
  }
  else
  {
    button0 = 0;
  }

  if (b>0)
  {
    button1 = 2;
  }
  else
  {
    button1 = 0;
  }

  if (c>0)
  {
    button2 = 3;
  }
  else
  {
    button2 = 0;
  }
}
```

Рисунок 2.1.3.4 – Запись данных с устройств ввода в переменные

Последним действием мы записываем полученные данные в массив и отправляем их по радиоканалу связанного с управляемым устройством (Рисунок



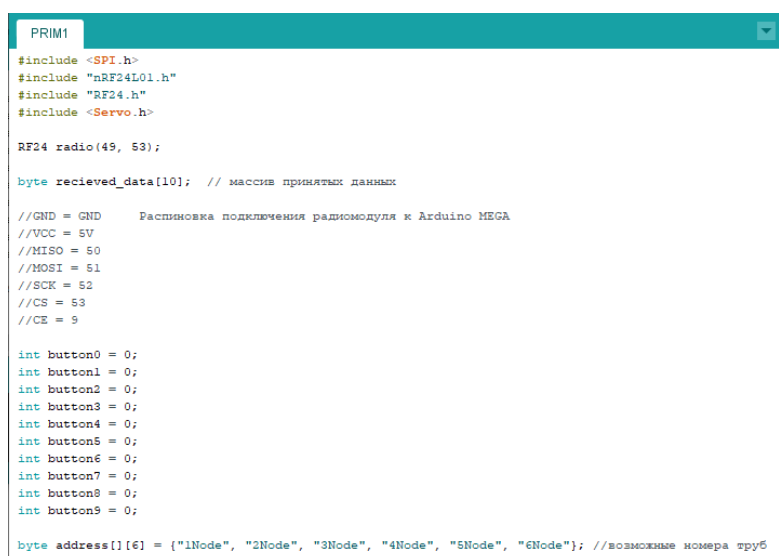
```
PER1 | Arduino 1.8.7
Файл Правка Скетч Инструменты Помощь
PER1 $
}
transmit_data[0] = (button0);
transmit_data[1] = (button1);
transmit_data[2] = (button2);
transmit_data[3] = (button3);
transmit_data[4] = (button4);
transmit_data[5] = (button5);
transmit_data[6] = (button6);
transmit_data[7] = (button7);
transmit_data[8] = (button8);
transmit_data[9] = (button9);

radio.write(&transmit_data, sizeof(transmit_data)); //отправка по радиоканалу
delay(100);
}
162 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) на COM4
```

Рисунок2.1.3.5 – Запись данных в массив и их последующая отправка

Для работы робота гуманоида необходима программа, которая будет обрабатывать данные, приходящие по радиоканалу и отдавать команды за счет управления контроллером, контроллер на основе программы будет исполнять указанный программой алгоритм действий.

Первые шаги программного обеспечения управляемого устройства ничем не отличаются от программы управляющего (Рисунок 2.1.2.6), за исключением использования других типов и названий переменных и настройки контактов подключенного драйвера двигателя (Рисунок 2.1.2.7).



```
PRIM1
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Servo.h>

RF24 radio(49, 53);

byte recieved_data[10]; // массив принятых данных

//GND = GND      Распиновка подключения радиомодуля к Arduino MEGA
//VCC = 5V
//MISO = 50
//MOSI = 51
//SCK = 52
//CS = 53
//CE = 9

int button0 = 0;
int button1 = 0;
int button2 = 0;
int button3 = 0;
int button4 = 0;
int button5 = 0;
int button6 = 0;
int button7 = 0;
int button8 = 0;
int button9 = 0;

byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"}; //возможные номера труб
```

Рисунок 2.1.2.6 – Подключение библиотек и добавление переменных

```

// подключите пины драйвера к цифровым пинам Arduino
// первый двигатель
int enA = 13;
int in1 = 12;
int in2 = 11;
// второй двигатель
int enB = 8;
int in3 = 10;
int in4 = 9;

int ms1 = 0;
int ms2 = 0;
int ms3 = 0;
int ms4 = 0;
int ms5 = 0;
int ms6 = 0;

uint8_t pipe;

Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;

```

Рисунок 2.1.2.7 – Указание контактов подключенных двигателей

Дальнейшая настройка передачи данных через радиоканал отличается лишь тем, что мы являемся приемником, а не передатчиком радиосигнала и устанавливаем контакты драйвера двигателя как выход (Рисунок 2.1.2.8).

```

void setup() {
  Serial.begin(9600);
  // инициализируем все пины для управления двигателями как outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  radio.begin(); //активировать модуль
  radio.setAutoAck(1); // режим подтверждения приёма, 1 вкл 0 выкл
  radio.setRetries(0, 15); // (время между попыткой достучаться, число попыток)
  radio.enableAckPayload(); // разрешить отсылку данных в ответ на входящий сигнал
  radio.setPayloadSize(32); // размер пакета, в байтах

  radio.openReadingPipe(1, address[0]); // хотим слушать трубу 0
  radio.setChannel(0x60); // выбираем канал (в котором нет шумов!)

  radio.setPALevel (RF24_PA_MAX); // уровень мощности передатчика. На выбор RF24_PA_MIN, RF24_PA_LO
  radio.setDataRate (RF24_250KBPS); // скорость обмена. На выбор RF24_2MBPS, RF24_1MBPS, RF24_250KBPS
  //должна быть одинакова на приёмнике и передатчике!
  //при самой низкой скорости имеем самую высокую чувствительность и дальность!!

  radio.powerUp(); // начать работу
  radio.startListening(); // начинаем слушать эфир, мы приёмный модуль

  servo1.attach(A1);
  servo2.attach(A2);
  servo3.attach(A3);
  servo4.attach(A4);
  servo5.attach(A5);
  servo6.attach(A6);
}

```

Рисунок 2.1.2.8 – Настройка контактов подключенных двигателей и серводвигателей

Каждой нажатой кнопке соответствует определенное действие робота. Каждое действие робота записано в виде цикла и будет выполняться при нажатии определенной кнопки на пульте управления (Рисунок 2.1.2.9).


```

void loop() {
  byte pipeNo;
  while ( radio.available(pipeNo)){ // есть входящие данные
    radio.read(&recieved_data, sizeof(recieved_data)); // читаем входящий сигнал

    if (recieved_data[0]>=1)
    {
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);
      analogWrite(enA, 100);
      analogWrite(enB, 100);
      delay(100);
    }
    else
    {
      digitalWrite(in1, LOW);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
      analogWrite(enA, 255);
      analogWrite(enB, 255);
    }
  }

  if (recieved_data[1]>=1)
  {
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(enA, 100);
    analogWrite(enB, 100);
    delay(100);
  }
  else
  {
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    analogWrite(enA, 100);
    analogWrite(enB, 100);
  }
}

```

Рисунок 2.1.2.9 – Циклы управления двигателями

Также, как и для двигателей, записываем программный цикл для серводвигателей (Рисунок 2.1.2.10).

```

PRIM1
{
  if (recieved_data[4]>0)
  {
    ms1=ms1+1;
  }
  if (ms1>1)
  {
    ms1=0;
  }
  if (ms1==1)
  {
    servov1.write(180);
    delay(50);
  }
  else
  {
    servov1.write(50);
  }
}

if (recieved_data[5]>0)
{
  ms2=ms2+1;
}
if (ms2>1)
{
  ms2=0;
}
if (ms2==1)
{
  servov2.write(50);
  delay(50);
}
else
{
  servov2.write(180);
}
}

if (recieved_data[6]>0)
{
  ms3=ms3+1;
}
if (ms3>1)
{
  ms3=0;
}
if (ms3==1)
{

```

Рисунок 2.1.2.10 – Циклы управления серводвигателями

1.2 Кинематика гуманоидного робота

Манипуляторы робота гуманоида имеют две степени свободы и две рабочие точки, точка покоя и точка осуществления манипуляционных действий.

Прямая кинематика манипулятора описывает связь между углами поворота (свободы) звеньев и положением (координатами) рабочего органа. Обратная кинематика, наоборот, позволяет определить углы поворота звеньев на основе заданных координат рабочего органа. [7]

У нас имеется два случая, где необходимо решить задачи на прямую и обратную кинематику, при положении манипулятора в точке покоя (Рисунок

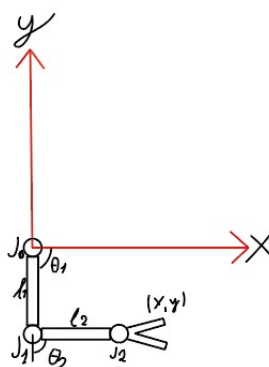


Рисунок 2.2.1 – Манипулятор в точке покоя

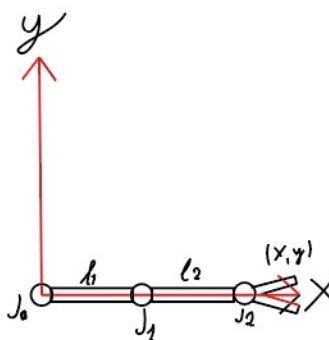


Рисунок 2.2.2 – Манипулятор в рабочей точке

Для манипулятора с 3 степенями свободы и 2 звеньями, находящимися в одной плоскости, можно использовать простую геометрию для расчета прямой и обратной кинематики.

Решим прямую кинематику робота. Допустим, у нас есть углы поворота ($\theta_1, \theta_2, \theta_3$) для каждой степени свободы и длины (L_1, L_2) каждого звена, для решения такой задачи можно использовать метод Денавита-Хартенберга.[8]

Положение (x, y) рабочего органа можно определить следующим образом:

$$x = L1 \times \cos + L2 \times \cos(\theta1 \pm \theta2) \quad (1)$$

$$y = L1 \times \sin(\theta1) + L2 \times \sin(\theta1 + \theta2) \quad (2)$$

После решения прямой кинематике роботу необходимо будет вернуть звенья манипуляторов в обратном порядке. Для определения углов поворота ($\theta1$, $\theta2$, $\theta3$) на основе заданных координат (x , y) рабочего органа, можно использовать обратные тригонометрические функции:

$$\theta2 = \arccos \frac{(x^2+y^2-L1^2-L2^2)}{(2 \times L1 \times L2)} \quad (3)$$

$$\theta2 = \arctan \left(\frac{y}{x} \right) - \arctan \frac{(L2 \times \sin(\theta2))}{(L1 + L2 \times \cos(\theta2))} \quad (4)$$

$$\theta3 = \theta1 + \theta2 - \alpha \quad (5)$$

Здесь α - фиксированный угол между звеньями.

Обратная кинематика может иметь несколько решений, поскольку одному положению рабочего органа могут соответствовать несколько возможных комбинаций углов поворота. В таких случаях для выбора конкретного решения могут использоваться дополнительные критерии или ограничения.

1.3 Расчет надежности гуманоидного робота

Расчет надежности робота-гуманоида может быть сложным процессом, который включает множество факторов, таких как конструкция робота, качество его компонентов, рабочая среда и частота обслуживания.

Один из подходов к расчету надежности робота-гуманоида заключается в использовании показателя, называемого средним временем наработки на отказ. Среднее время безотказной работы – это статистическая мера того, как долго система должна работать, прежде чем произойдет сбой. Он рассчитывается путем деления общего времени работы на количество отказов, произошедших за это время.

Важно отметить, что среднее время безотказной работы — это всего лишь одна мера надежности, не учитывающая другие факторы, которые могут повлиять на производительность и время безотказной работы робота. Другие факторы, такие как график обслуживания робота, условия окружающей среды и поведение пользователя, также могут влиять на его надежность.

Для расчета интенсивности отказа, вероятности безотказной работы и других показателей надежности, нужно знать дополнительные данные, такие как время работы или эксплуатации, чтобы провести более точные расчеты.

При проведении расчета руководствуемся технической документацией используемых модулей. [6]

Возьмем средний показатель безотказной работы из технической документации устройства:

- Arduino Mega 2560. $T_{AM} = 90000$ часов.
- Драйвер двигателя L298N. $T_L = 130000$ часов.
- Модуль NRF24L01. $T_{NRF} = 80000$ часов.
- Сервопривод SG90. $T_{SG} = 60000$ часов.

Каждой из используемых модулей мы присвоим свои обозначения.

Найдем интенсивность отказов каждого элемента по отдельности:

$$\lambda_i = \frac{1}{T_i} \quad (6)$$

По данной формуле нахожу интенсивность отказа для каждого модуля по отдельности.

$$\lambda_{AM} = 1,1 \times 10^{-5} \text{ 1/ч};$$

$$\lambda_L = 7,6 \times 10^{-5} \text{ 1/ч};$$

$$\lambda_{NRF} = 1,25 \times 10^{-5} \text{ 1/ч};$$

$$\lambda_{SG} = 1,6 \times 10^{-5} \text{ 1/ч};$$

Приступаем к следующему расчету, вероятность безотказной работы 1-го элемента при экспоненциальном законе распределения времени до отказа по следующей формуле:

$$P_i(t) = e^{-\lambda_i t} \quad (7)$$

λ_i – интенсивность отказов 1-го элемента, 1/ч.

t – рассматриваемое время, (для расчета принимаем $t=1000$ часов).

Подставляю наши значения и получаем вероятность безотказной работы каждого модуля по отдельности:

$$P_{AM} = 0,999;$$

$$P_L = 0,999;$$

$$P_{NRF} = 0,999;$$

$$P_{SG} = 0,999.$$

После расчета вероятность безотказной работы приступаю к расчёту показателей надёжности системы в целом по следующей формуле:

$$PC = \prod_{i=1}^n P_i(t), \quad (8)$$

$$PC = 0.9994 = 0,996$$

На основе полученных решений найдем интенсивность отказов системы:

$$\lambda_c = \frac{-\ln 0,996}{1000} = 6,9 \times 10^{-3} \text{ 1/ч}$$

Зная интенсивность отказа, мы можем рассчитать, как часто устройство будет выходить из строя, а именно средняя наработка до отказа системы:

$$TC = \frac{1}{\lambda_c} \quad (9)$$

$$TC = \frac{1}{6,9 \times 10^{-3}} = 145 \text{ ч}$$

На основе проведенных расчетов можно сделать вывод, что первые отказы одного из модулей устройства могут происходить каждые 145 часов или не происходить вовсе до полного отказа устройства. Если устройство будет работать 8 часов в сутки, то в таком темпе устройство проработает от 5 до 20 лет. Время работы устройства также напрямую зависит от внешнего воздействия рабочей среды. В целом устройство способно эксплуатироваться на довольно большом промежутке времени без существенных повреждений или отказов.

ЗАКЛЮЧЕНИЕ

Применение роботов-гуманоидов имеет широкий потенциал в различных сферах, начиная от медицины и промышленности, и заканчивая развлекательной индустрией. Они могут быть использованы для выполнения сложных медицинских операций, помощи людям с ограниченными физическими возможностями, проведения исследований и обучения, а также для развлечения и взаимодействия с людьми. [12]

В процессе исследования были изучены существующие достижения в области разработки роботов-гуманоидов, включая механические системы и электронику. Были проанализированы различные методы, применяемые в создании роботов-гуманоидов, и выявлены основные проблемы в разработке проекта.

В практической части нам удалось разработать и собрать прототип робота способный двигаться во всех направлениях, захватывать небольшие объекты и все это осуществляется с помощью пульта управления, который связан с роботом по радиоканалу.

Данная работа может являться отправной точкой в дальнейшем изучении, исследовании и конструировании подобных устройств.

В заключение можно сказать, что построенный прототип гуманоидного робота способен передвигаться в пространстве за счет колесной базы. Манипуляторы робота позволяют ему удерживать, транспортировать и складировать различные объекты, ограничен только в подъемной силе.

Все разработки по созданию человекоподобных роботов направлены на замену человеческого труда и развитие робототехники как передовой науки.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- [1] К. Фу, Р. Гонсалес, К. Ли «Робототехника», 1989г.
- [2] Айлетт, Варгас, Шарки: Жизнь с роботами. Что нужно знать каждому беспокоящемуся человеку, АСТ, 2022г.
- [3] Т. К. Хейс, Искусство схемотехники. Теория и практика, БХВ-Петербург, 2022г.
- [4] Дж. Блум, Изучаем Arduino. Инструменты и методы технического волшебства, ВHV, 2015г.
- [5] А.С. Климчик Р.И. Гомолицкий Ф.В. Фурман К.И. Сёмкин, Разработка управляющих программ промышленных роботов, «Белорусский государственный университет информатики и радиоэлектроники», 2005г.
- [6] Ефремов А.А., Теория надежности, «Национальный Исследовательский Томский Политехнический Университет», 2015г.
- [7] Рыбак Л.А., Ержуков В.В., Чичварин А.В. Эффективные методы решения задач кинематики и динамики робота-станка параллельной структуры, Физматлит, 2011г.
- [8] В. Г. Хомченко, Робототехнические системы, «Омский государственный технический университет», 2016г.
- [9] В. Жандов, Расчет надежности электронных модулей, Солон–Пресс, 2018г.
- [10] Е. И. Юриевич, Основы робототехники, БХВ-Петербург, 2010г.
- [11] Крейг Джон Дж, Введение в робототехнику, «Институт прикладной математики (ИПМ) РАН», 2013г.
- [12] А. Азимов, Я робот, Сборник, 1950г.
- [13] В. А. Глазунова, Новые механизмы робототехники и измерительных систем, Техносфера, 2022г.
- [14] Dragomir N. Nenchev, Atsushi Konno, Humanoid Robots: Modeling and Control, Butterworth-Heinemann, 2019г.
- [15] S. L. Hamilton, Humanoid Robots (Xtreme Robots), A&d Xtreme, 2018г.
- [16] Kathryn Clay, Humanoid Robots: Running into the Future (The World of Robots), Capstone Press, 2014г.
- [17] Lukas Sebastian Kaul, Human-Inspired Balancing and Recovery Stepping for Humanoid Robots, KIT Scientific Publishing, 2019г.

Приложение А

Внешний вид прототипа гуманоидного робота



Внешний вид пульта управления гуманоидным роботом

Приложение Б

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Servo.h>

RF24 radio(49, 53);

byte recieved_data[10]; // массив принятых данных

//GND = GND      Распиновка подключения радиомодуля к Arduino MEGA
//VCC = 5V
//MISO = 50
//MOSI = 51
//SCK = 52
//CS = 53
//CE = 9

int button0 = 0;
int button1 = 0;
int button2 = 0;
int button3 = 0;
int button4 = 0;
int button5 = 0;
int button6 = 0;
int button7 = 0;
int button8 = 0;
int button9 = 0;

byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"}; //возможные номера труб

// подключите пины драйвера к цифровым пинам Arduino
// первый двигатель
int enA = 13;
int in1 = 12;
int in2 = 11;
// второй двигатель
int enB = 8;
int in3 = 10;
int in4 = 9;

int ms1 = 0;
int ms2 = 0;
int ms3 = 0;
int ms4 = 0;
int ms5 = 0;
int ms6 = 0;

uint8_t pipe;
```

Продолжение приложения Б

PRIM1

```
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;

void setup() {
  Serial.begin(9600);
  // инициализируем все пины для управления двигателями как outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  radio.begin(); //активировать модуль
  radio.setAutoAck(1); // режим подтверждения приёма, 1 вкл 0 выкл
  radio.setRetries(0, 15); // (время между попыткой достучаться, число попыток)
  radio.enableAckPayload(); // разрешить отсылку данных в ответ на входящий сигнал
  radio.setPayloadSize(32); // размер пакета, в байтах

  radio.openReadingPipe(1, address[0]); // хотим слушать трубу 0
  radio.setChannel(0x60); // выбираем канал (в котором нет шума!)

  radio.setPALevel(RF24_PA_MAX); // уровень мощности передатчика. На выбор RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH, RF24_PA_BOOST
  radio.setDataRate(RF24_250KBPS); // скорость обмена. На выбор RF24_2MBPS, RF24_1MBPS, RF24_250KBPS
  //должна быть одинакова на приёмнике и передатчике!
  //при самой низкой скорости имеем самую высокую чувствительность и дальность!!

  radio.powerUp(); // начать работу
  radio.startListening(); // начинаем слушать эфир, мы приёмный модуль

  servo1.attach(A1);
  servo2.attach(A2);
  servo3.attach(A3);
  servo4.attach(A4);
  servo5.attach(A5);
  servo6.attach(A6);
}

void loop() {
  byte pipeNo;
  while (radio.available(&pipeNo)){ // есть входящие данные
    radio.read(&recieved_data, sizeof(recieved_data)); // читаем входящий сигнал

    if (recieved_data[0]>=1)
    {
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);
      analogWrite(enA, 100);
      analogWrite(enB, 100);
      delay(100);
    }
    else
    {
      digitalWrite(in1, LOW);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
      analogWrite(enA, 255);
      analogWrite(enB, 255);
    }

    if (recieved_data[1]>=1)
    {
      digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
      digitalWrite(in3, HIGH);
      digitalWrite(in4, LOW);
      analogWrite(enA, 100);
      analogWrite(enB, 100);
      delay(100);
    }
    else
    {
      digitalWrite(in1, LOW);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
      analogWrite(enA, 100);
      analogWrite(enB, 100);
    }

    if (recieved_data[2]>=1)
    {
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, HIGH);
      digitalWrite(in4, LOW);
      analogWrite(enA, 255);
      analogWrite(enB, 255);
      delay(100);
    }
  }
}
```

Продолжение приложения Б

```
    if (recieved_data[4]>0)
    {
        ms1=ms1+1;
    }
    if (ms1>1)
    {
        ms1=0;
    }
    if (ms1==1)
    {
        servol.write(180);
        delay(50);
    }
    else
    {
        servol.write(90);
    }
}
```

```
    if (recieved_data[5]>0)
    {
        ms2=ms2+1;
    }
    if (ms2>1)
    {
        ms2=0;
    }
    if (ms2==1)
    {
        servo2.write(90);
        delay(50);
    }
    else
    {
        servo2.write(180);
    }
}
```

```
    if (recieved_data[6]>0)
    {
        ms3=ms3+1;
    }
    if (ms3>1)
    {
        ms3=0;
    }
    if (ms3==1)
```

Продолжение приложения Б

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Servo.h>

RF24 radio(9, 53);

byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"}; //возможные номера труб

byte transmit_data[10]; // массив, хранящий передаваемые данные
byte latest_data[10]; // массив, хранящий последние переданные данные
boolean flag; // флажок отправки данных

byte button0 = 2;
byte button1 = 3;
byte button2 = 4;
byte button3 = 5;
byte button4 = 6;
byte button5 = 7;
byte button6 = 8;
byte button7 = 10;
byte button8 = 11;
byte button9 = 12;

void setup() {
  Serial.begin(9600); // открываем порт для связи с ПК
  pinMode(2, INPUT_PULLUP); //настройка кнопки
  pinMode(3, INPUT_PULLUP);
  pinMode(4, INPUT_PULLUP);
  pinMode(5, INPUT_PULLUP);
  pinMode(6, INPUT_PULLUP);
  pinMode(7, INPUT_PULLUP);
  pinMode(8, INPUT_PULLUP);
  pinMode(10, INPUT_PULLUP);
  pinMode(11, INPUT_PULLUP);
  pinMode(12, INPUT_PULLUP);

  radio.begin(); // активировать модуль
  radio.setAutoAck(1); // режим подтверждения приёма, 1 вкл 0 выкл
  radio.setRetries(0, 15); // (время между попыткой достучаться, число попыток)
  radio.enableAckPayload(); // разрешить отсылку данных в ответ на входящий сигнал
  radio.setPayloadSize(32); // размер пакета, в байтах

  radio.openWritingPipe(address[0]); // мы - труба 0, открываем канал для передачи данных
  radio.setChannel(0x60); // выбираем канал (в котором нет шумов!)

  radio.setPALevel(RF24_PA_MAX); // уровень мощности передатчика. На выбор RF24_PA_MIN, RF24_PA_
  radio.setDataRate(RF24_250KBPS); // скорость обмена. На выбор RF24_2MBPS, RF24_1MBPS, RF24_250KB
  //должна быть одинакова на приёмнике и передатчике!
  ..
  ..
}
```

Продолжение приложения Б

```
}  
  
void loop() {  
  boolean a= !digitalRead(2);  
  boolean b= !digitalRead(3);  
  boolean c= !digitalRead(4);  
  boolean d= !digitalRead(5);  
  boolean e= !digitalRead(6);  
  boolean f= !digitalRead(7);  
  boolean g= !digitalRead(8);  
  boolean h= !digitalRead(10);  
  boolean i= !digitalRead(11);  
  boolean j= !digitalRead(12);  
  
  if (a>0)  
  {  
    button0 = 1;  
  }  
  else  
  {  
    button0 = 0;  
  }  
  
  if (b>0)  
  {  
    button1 = 2;  
  }  
  else  
  {  
    button1 = 0;  
  }  
  
  button9 = 0;  
}  
transmit_data[0] = (button0);  
transmit_data[1] = (button1);  
transmit_data[2] = (button2);  
transmit_data[3] = (button3);  
transmit_data[4] = (button4);  
transmit_data[5] = (button5);  
transmit_data[6] = (button6);  
transmit_data[7] = (button7);  
transmit_data[8] = (button8);  
transmit_data[9] = (button9);  
//Serial.println(" ");  
  
radio.write(&transmit_data, sizeof(transmit_data));  
delay(100);  
}
```