

титул

задание

задание

план

Реферат

Выпускная квалификационная работа 85 с., 6 глав, 26 рис., 19 табл., 2 приложения, 14 источников.

ПОДСИСТЕМА, МИКРОКОНТРОЛЛЕР, АВТОМАТ, БОРТОВАЯ ЭВМ, ВХОДНОЕ ВОЗДЕЙСТВИЕ, ФУНКЦИЯ, РОБОТ, АНДРОИД, ИСПОЛНИТЕЛЬНЫЕ МЕХАНИЗМЫ

Объектом настоящего исследования является программное и аппаратное обеспечение робота.

Предметом исследования являются методы и алгоритмы, используемые в программном и аппаратном обеспечении робота.

Целью настоящей работы является разработка аппаратного и программного обеспечения системы управления исполнительными механизмами андроидного робота.

Задачи работы:

- исследование предметной области;
- разработка схемотехнических решений;
- разработка ПО микроконтроллера;
- разработка ПО бортовой ЭВМ робота;
- разработка системы управления движением со стабилизацией по курсу, траектории, скорости;
- разработка системы управления «жестикуляцией» робота.

Определения и сокращения

ИМ - исполнительные механизмы

БИНС - бесплатформенная инерциальная навигационная система

ИМС - интегрированная микросхема

МК - микроконтроллер

СТЗ - система технического зрения

АРК - андроидный роботизированный комплекс

АС - автоматизированные системы

НСД - несанкционированный доступ

РД - руководящий документ

СЗИ - система защиты информации

СЗИ НСД - система защиты информации от несанкционированного доступа

Kinect - бесконтактный сенсорный игровой контроллер,

Arduino Mega2560 - отладочная плата микроконтроллера

ArduinoDuemilanove - отладочная плата микроконтроллера

ATMega 328 - микроконтроллер

Ardumoto L298P MotorDriverShield -плата управления двигателями

Содержание

Определения и сокращения.....	6
Введение.....	9
1 Исследовательская часть.....	11
1.1 Исследование существующих подходов к управлению ИМ робота.....	11
1.2 Исследование шасси "Промобота".....	15
1.3 Исследование существующего аппаратного обеспечения.....	17
1.3.1 Характеристики БИНС.....	17
1.3.2 Характеристики сервопривода наклона головы.....	17
1.3.3 Сервоприводы кистей рук.....	18
1.3.4 Характеристики дисплея.....	19
1.3.5 Описание датчиков расстояния.....	19
1.3.6 Характеристики модуля реле.....	20
1.3.7 Используемая светотехника.....	21
1.4 Обзор достоинств и недостатков существующих решений.....	21
1.5 Вывод по разделу.....	23
2 Конструкторская часть.....	25
2.1 Проектирование схмотехнических решений.....	25
2.2 Описание подсистемы управления движением рук.....	26
2.3 Описание подсистемы управления движением ног.....	27
2.4 Описание подсистемы управления движением головы.....	29
2.5 Описание подсистемы управления движением захватов кистей.....	29
2.6 Описание подсистемы управления курсом с помощью гироскопа.....	31
2.7 Описание подсистемы управления питанием бортовых устройств.....	37
2.8 Описание подсистемы управления устройствами ЧМИ.....	38
2.10 Проектирование ПО микроконтроллера.....	38
2.10.1 Автомат работы рук.....	39
2.10.2 Автомат работы головы робота.....	40
2.10.3 Автомат управления питанием компьютера.....	41

2.10.4 Автомат управления светотехникой.....	42
2.10.5 Автомат обхода препятствий.....	44
2.10.6 Обработка датчиков расстояния.....	46
2.10.7 Основной алгоритм программы микроконтроллера.....	50
2.10.8 Инициализация устройств.....	51
2.10.9 Обработчик прерывания по первому таймеру.....	53
2.10.10 Обработчик прерывания по второму таймеру.....	54
2.10.11 Инициализация гироскопа (расчет дрейфа).....	55
2.11 Вывод по разделу.....	57
4 Информационная безопасность.....	58
4.1 Анализ исходных данных.....	58
4.2 Выявление основных признаков АС.....	59
4.3 Сравнение выявленных признаков АС с классифицируемыми.....	60
4.5 Требования к подсистемам класса защищенности 1Д.....	61
5. Экономический раздел.....	64
5.1 Расчет расходов на проект.....	64
5.2 Выводы по экономическому разделу.....	66
6 Технологическая часть.....	67
6.1 Изготовление и сборка схемотехнических решений.....	67
6.2 Разработка ПО микроконтроллера.....	68
6.3 Разработка ПО бортовой ЭВМ робота.....	73
6.4 Разработка системы управления движением.....	80
Список литературы.....	81

Введение

В наши дни тема взаимодействия робота и человека становится все более актуальной. Многие предприятия и заводы едва ли можно представить без использования робототехники. В настоящее время робототехника завоевывает все большие отрасли промышленности и все плотнее внедряется в различные сферы человеческой жизни. И если раньше роботы могли исполнять некоторые функции человека, заменяя его на заводах, где часто требуются однообразные действия при конвейерном производстве, при производстве разнообразных механизмов, то теперь наступили времена, когда роботы способны выполнять всё более сложные задачи. Развивается наука, развивается и техника, следуя за прогрессом.

Также в настоящее время роботы активно внедряются в образовательную и выставочную деятельность. Особое внимание привлекают роботы андроидного типа или антропоморфные роботы, имитирующие поведение человека. В связи с этим ставится вопрос об использовании роботов этого типа для привлечения внимания и выполнения задач их прямого назначения. Данное исследование посвящено разработке робота андроидного типа.

Актуальность темы исследования. В условиях развития информационных ресурсов одним из ключевых направлений современной компьютерной науки является разработка роботов. Процесс решения данных прикладных задач подразумевает, как правило, использование современных методов построения такого рода систем. Управление исполнительными механизмами современного роботизированного комплекса невозможно без применения методов искусственного интеллекта. К таким методам относятся: теория автоматов, нечеткая логика, встраиваемые системы, прикладная лингвистика.

Научная новизна исследования заключается в следующем: создан программный комплекс для управления исполнительными механизмами

андроидного робота. Функционал системы позволяет осуществлять управление и мониторинг состояния подсистем, входящих в аппаратное и программное обеспечение робота. Особенности модульной архитектуры позволяют гибко интегрировать программный комплекс в разрабатываемую систему. В целях осуществления обмена данных в программном комплексе предусмотрены интерфейсы связи между компонентами.

Практическая значимость заключается в создании схемотехнических решений, позволяющих реализовать систему управления автономно передвигающимся андроидным роботом, способным к имитации отдельных человеческих жестов, таких как: движение рук, кистей, головы в двух плоскостях. А также практическая значимость заключается в наделении робота способностью показывать своё "эмоциональное состояние" при помощи дисплея, а намерения передвижения при помощи светотехники.

Объектом настоящего исследования является программное и аппаратное обеспечение робота.

Предметом исследования являются методы и алгоритмы, используемые в программном и аппаратном обеспечении робота.

Целью настоящей работы является разработка аппаратного и программного обеспечения системы управления исполнительными механизмами андроидного робота. Для достижения поставленной цели необходимо сформулировать и решить следующие задачи:

- исследование предметной области;
- разработка схемотехнических решений;
- разработка ПО микроконтроллера;
- разработка ПО бортовой ЭВМ робота;
- разработка системы управления движением со стабилизацией по курсу, траектории, скорости;
- разработка системы управления «жестикulyацией» робота.

1 Исследовательская часть

1.1 Исследование существующих подходов к управлению ИМ робота

Проектирование и разработка андроидных роботов широко практикуется в настоящее время. Обратимся к литературе на данную тематику.

В статье [1] представлена разработка аппаратно-программного комплекса для управления робототехнической платформой и программно алгоритмического обеспечения для автономного движения и бесконтактного управления робототехнической платформой.

В ходе тестирования представленной в этой статье платформы была решена задача обработки данных с камеры Kinect и распознавания жестов оператора, отлажена передача данных на микроконтроллер Arduino Mega2560, разработана программа управления по передвижению робототехнической платформы в необходимых направлениях, апробированы модули ввода данных с датчиков.

Экспериментально подтверждено, что робототехническая платформа выполняет движения в необходимых направлениях с учетом локальных препятствий в режиме реального времени, на основе данных, получаемых с датчиков.

В статье [2] описано создание кисти-манипулятора для андроидного робота. В робототехнике широкое распространение получили клешни для захвата различных предметов. Клешня состоит из двух направляющих, при сближении которых и происходит удержание предмета. Движения направляющих происходит с помощью сервоприводов.

На сегодняшний день большое распространение получили роботы-андроиды. Но практическое применение может иметь и рука (кисть руки) робота отдельно — в качестве протеза для инвалидов. Кисть используют в медицинской технике, в частности области медицинских средств реабилитации

двигательного аппарата человека. Цель - повышение степени приспособляемости при захвате предметов. Пальца кисти андроида могут быть использованы для жестикуляции в развлекательных целях и для рекламы.

В статье [3] рассмотрены перспективы развития мобильных робототехнических комплексов. Автор описывает что в настоящее время в мире наблюдается широкое распространение роботизированных систем и робототехники в целом. Во многих странах, в том числе и в Казахстане, робототехника является уже отдельной дисциплиной изучаемой в учебных заведениях, и не только в профильных вузах, но и в школах, и, соответственно, в педагогических вузах.

Востребованность специалистов, обладающих знаниями в этой области, ежегодно растет. Это обусловлено многими объективными факторами, решающим из которых, конечно же, является бурное развитие робототехники.

Сферы применения роботов и робототехники различны: образование, медицина, строительство, геодезия, метеорология и т.д.

Особое значение так же занимает развитие, так называемой, автономной и мобильной робототехники. В силу того, что робототехника зачастую применяется для выполнения работ в недоступных человеку местах, именно мобильная робототехника часто является приоритетом. Однако, не только в перечисленных выше случаях требуется мобильная робототехника, определенной ветвью ее развития является автономная робототехника, подразумевающая практически полное исключение вмешательства человеческой деятельности для выполнения каких-либо задач. Иными словами, такая робототехника является наиболее перспективной и интересной в плане разработки. Для осуществления автономности, в мобильной робототехнике, все чаще применяется искусственный интеллект, а точнее его элементы.

В статьях [4, 5] рассмотрена система управления движением автономной колесной платформы. В статье описан состав аппаратной и программной частей платформы и даны основные принципы проектирования таких устройств.

Выбранное направление движения преобразуется в команды и передается в систему управления движением. Система включает в себя микроконтроллер ArduinoDuemilanove на основе ATmega 328. Для управления работой двигателей предназначен силовой модуль расширения Arduino L298P MotorDriverShield [6].

Движением аппарата управляет микроконтроллер, микропрограмма которого включает в себя блок инициализации, в котором производится настройка датчиков, а также основной цикл, в котором последовательно производится опрос датчиков аппарата, определение возможности движения в данный момент, остановка движения в случае наличия препятствия в непосредственной близости, передача телеметрической информации в бортовой компьютер, а также прием и анализ команд, поступивших от бортового компьютера.

Для реализации программного обеспечения микроконтроллера в работе [5] была выбрана концепция автоматного программирования. Часть программного кода, ответственная за командно-информационный обмен с бортовым компьютером, представляет собой конечный частично определенный детерминированный автомат-распознаватель (см. рис. 1).

Автомат содержит 13 состояний и 26 переходов между ними. Прием числовых данных реализован отдельной процедурой, причем возникновение ошибки в ходе приема информации немедленно переводит автомат в исходное состояние, что обеспечивает высокую устойчивость системы к ошибкам и сбоям. Программное обеспечение на стороне бортового компьютера, принимающее телеметрические данные от микроконтроллера, реализовано по такому же принципу.

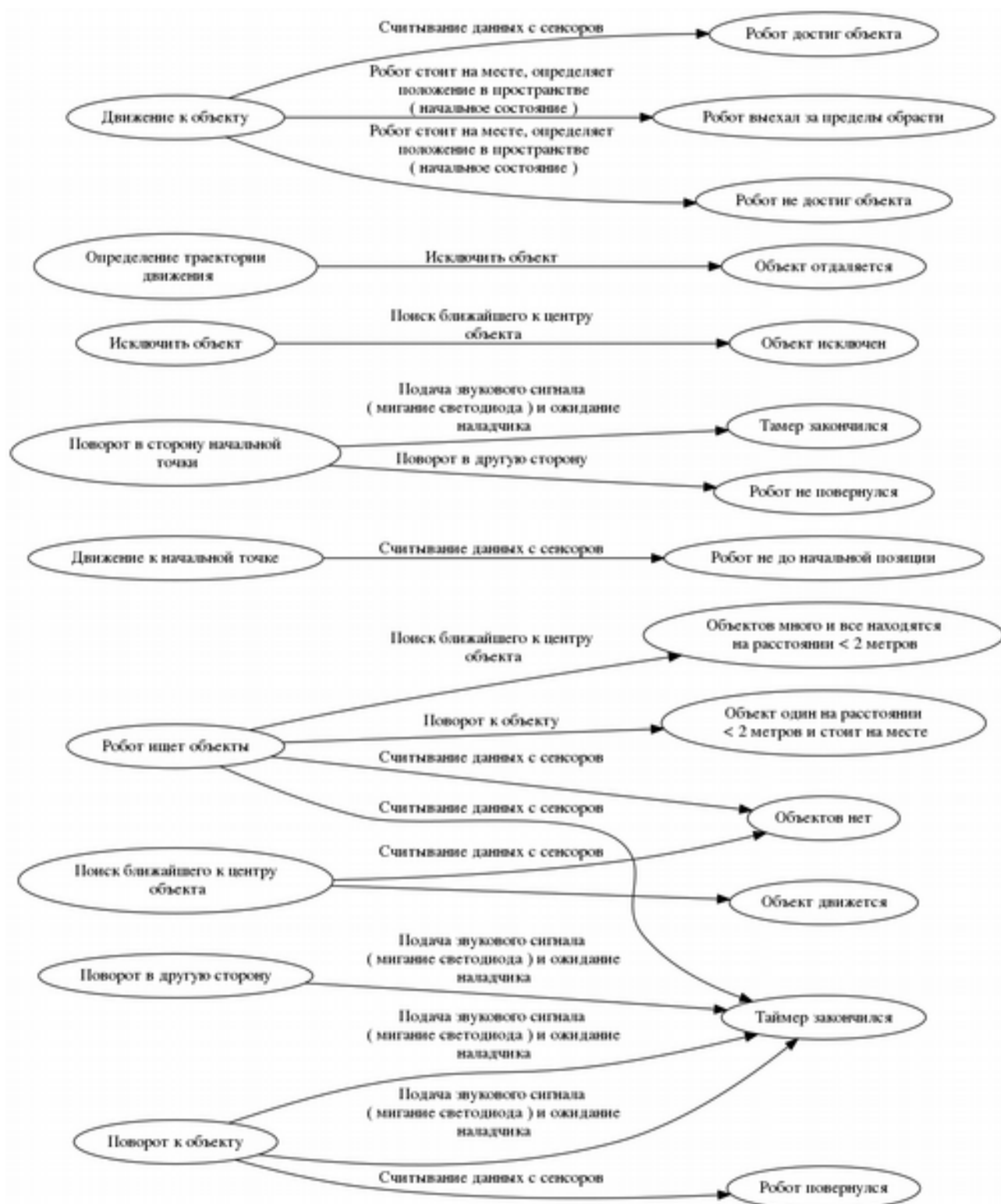


Рисунок 1. Автомат поведения робота [5]

Таким образом, сбой работы микроконтроллера так же не приводит к прекращению обменом информацией между бортовым компьютером и управляющим микроконтроллером или зависанию программы. Для

возобновления нормального функционирования системы достаточно перезагрузить микроконтроллер либо бортовой компьютер, других специальных шагов предпринимать не требуется.

1.2 Исследование шасси "Промобота"

В основе робота было использовано уже имеющееся шасси, представляющее из себя набор исполнительных механизмов и датчиков перечень которых приведен в таблице 1.

Таблица 1. Состав оборудования и ИМ шасси «Промобот»

№ п/п	Наименование прибора	Назначение прибора	Характеристики
1	Привод правого колеса	Перемещение робота	12V 10A DC 1шт.
2	Привод левого колеса	Перемещение робота	12V 10A DC 1шт.
3	Привод правой руки	"Жестикуляция" робота	12V 10A DC 1шт.
4	Привод левой руки	"Жестикуляция" робота	12V 10A DC 1шт.
5	Привод поворота головы	"Жестикуляция" робота	Серво 5V 2A PWM 1шт.
6	Ультразвуковые датчики расстояния	Обнаружение препятствий	10 - 200см 8шт.
7	Инфракрасные датчики расстояния	Обнаружение препятствий	неисправны 6шт.
8	Модуль датчика напряжения питания	Контроль напряжения бортовой сети	Аналоговый
9	Акустический усилитель	Усиление аудио частот	Аналоговый
10	Динамики	Воспроизведение звука	4 Ом

Продолжение таблицы 1.

11	WiFi-адаптер	Связь с интернетом	2.4Ггц
12	Arduino Uno R3	Управление ИМ	Atmega 328 P
13	Драйверы управления двигателями постоянного тока	Управление двигателями	12V, 20А, 2-канальный, цифровое управление, 2шт.
14	Материнская плата	Организация вычислительного процесса	Core i7, SSD
15	LED-матрица	"глаза" робота	RGB, 5V, цифровое управление
16	Блок питания	Питание	12V 20A
17	Камера	Обнаружение человека	Веб, 640x480
18	Концевые выключатели рук	Определение границ перемещения рук	2 контакта

Установлено, что в составе шасси отсутствуют следующие компоненты, необходимые для решения поставленных перед ним задач.

1. БИНС. Необходима для стабилизации курса робота.
2. Серво-привод наклона головы. Нужен для расширения возможностей эмоционального взаимодействия с человеком.
3. Серво-приводы кистей рук. Также необходимы для расширения возможностей эмоционального взаимодействия с человеком.
4. Дисплей. Служит также для расширения возможностей эмоционального взаимодействия с человеком.
5. Датчики расстояния. Должны быть заменены для соответствия заявленным требованиям по ориентированию робота.
6. Реле. Должны быть использованы для корректного управления цепями нагрузки.

Анализ требований показывает, что имеющееся аппаратное обеспечение нуждается в доработке. Для выполнения реинжинирига робота необходимо

произвести анализ доступных на рынке модулей и компонентов, которые могут быть использованы для сборки робота.

1.3 Исследование существующего аппаратного обеспечения

Рассмотрим подробнее список дополнительных модулей и элементов.

1.3.1 Характеристики БИНС

Для того, чтобы робот мог стабилизировать курс, необходимо, чтобы на борту присутствовала бесплатформенная инерциальная навигационная система [7]. Для обеспечения данной функции рационально использовать имеющиеся в продаже модули с совмещенными устройствами, такими, как гироскоп, акселерометр, магнитный компас. Существует несколько разновидностей данных модулей.

В нашем случае удобно использовать модуль GY-85 [8], который имеет малые габариты, интерфейс связи с другими устройствами I2C [9], и имеет гироскоп, акселерометр и магнитный компас на одной плате.

1.3.2 Характеристики сервопривода наклона головы

Для того, чтобы робот мог выражать эмоции с помощью жестов [10], в числе действий по переоборудованию робота присутствует установка сервопривода наклона головы (вперед и назад). Выбранный сервопривод AS3103PG [11] обладает следующими характеристиками (приведены в таблице 3).

Таблица 2. Характеристики БИНС

№ п/п	Характеристика	Значение
1	Напряжение питания	3-5 V
2	Интерфейс	I2C
3	Диапазон измерения акселерометра	+/-2G, +/-4G, +/-8G, +/-16G
4	Диапазон измерения гироскопа	+/-250, +/-500, +/-1000, +/-2000dps
5	Диапазон измерения компаса магнитометра	+/-4800uF
6	Буфер	FIFO 512B
7	Размер	15x25мм

Таблица 3. Характеристики сервоприводов

№ п/п	Характеристика	Значение
1	Угол вращения	180 градусов
2	Вес	38г
3	Длина провода	320мм
4	Скорость	0.22 сек/60 градусов (4.8 В), 0.19 сек/60 градусов (6 В)
5	Крутящий момент	5кг*см (4.8 В), 5.5кг*см (6 В)
6	Напряжение	4.8В - 6В

Они вполне удовлетворяет предъявляемым требованиям, необходимым для наклона и поворота головы робота.

1.3.3 Сервоприводы кистей рук

У Промобота была задумана, но не реализована функциональность захвата предметов с помощью кистей рук. Для этого на руках робота имеются приспособления, которые предназначены для захвата предметов. Для того,

чтобы реализовать эту функциональность, будут установлены серво-приводы, и рычаги, связывающие сервопривод с кистью.

1.3.4 Характеристики дисплея

Для отображения всего спектра эмоций, которые будет способен «ощущать» робот, было принято решение об установке дисплея в голову роботу. При настройке системы дисплей будет выполнять функцию монитора компьютера, а при работе робота будет отображать лицо, и отражать эмоции на нем. Для этой задачи был выбран дисплей Skylarpu [12] (таблица 4) диагональю 9 дюймов, питающийся от 5V и имеющий интерфейс связи с компьютером HDMI.

Таблица 4. Характеристики дисплея Skylarpu

№ п/п	Характеристика	Значение
1	Размер экрана	9 Дюймов
2	Разрешение	800 * 480px
3	Формат	16:9
4	Напряжение питания	3.3/10.4/17.0/-5.0V (DVDD/AVDD/AVDD/VGL)
5	Ток питания	0.3/0.3mA (IGH/IGL)
6	Интерфейс	VGA, HDMI

1.3.5 Описание датчиков расстояния

Для корректного позиционирования робота внутри помещения было принято решение заменить имеющиеся парктроники на ультразвуковые датчики расстояния. Для измерения расстояния до объектов будет установлено 5 датчиков, и еще 1 для измерения расстояния до собеседника робота (будет

установлен в голове). Выбранные датчики имеют больший диапазон измеряемых расстояний, чем уже имеющиеся парктроники, и имеют гораздо более низкую стоимость (см. табл. 5).

Таблица 5. Характеристики датчиков расстояния

№ п/п	Характеристика	Значение
1	Напряжение питания	5 В
2	Потребление в режиме тишины	2 мА
3	Потребление при работе	15 мА
4	Диапазон расстояний	2–400 см
5	Эффективный угол наблюдения	15°
6	Рабочий угол наблюдения	30°

1.3.6 Характеристики модуля реле

Для корректного управления цепями нагрузок было решено использовать модули реле (таблица 6). Они позволяют включать и выключать цепи потребителей, когда это требуется, а также имеют гальваническую развязку между управляющей и силовой частями.

Таблица 6. Характеристики модуля реле

№ п/п	Характеристика	Значение
1	Количество каналов	4
2	Рабочее напряжение	5V
3	Ток силовой части	10A

1.3.7 Используемая светотехника

Модули светотехники представляют собой 8 светодиодных ламп с напряжением питания 12V. Белые устанавливаются на передней части робота, красные - на задней.

1.4 Обзор достоинств и недостатков существующих решений

Имеющееся оборудование обладает многими характеристиками, в том числе различными свойствами, и способами работы. Далее подробнее рассмотрим достоинства и недостатки применяемых компонентов и модулей.

1. БИНС. Необходима для стабилизации курса робота (таблица 7).

Таблица 7. Плюсы и минусы модуля гироскопа

№ п/п	Плюсы	Минусы
1	Низкая цена	Дрейф показаний
2	Малые габариты	Взаимное влияние осей датчика
3	Удобство работы	Напряжение питания влияет на чувствительность
4	-	Геометрические центры различных датчиков не совпадают

2. Сервопривод наклона головы (таблица 8). Нужен для расширения возможностей эмоционального взаимодействия с человеком.

Таблица 8. Плюсы и минусы сервопривода головы

№ п/п	Плюсы	Минусы
1	Низкая цена	Джиттер (вибрация)
2	Малое энергопотребление	Малый крутящий момент
3	Низкий уровень акустического шума	Малая износостойкость шестерней

3. Сервоприводы кистей рук (таблица 9). Также необходимы для расширения возможностей эмоционального взаимодействия с человеком.

Таблица 9. Плюсы/минусы сервоприводов кистей рук/

№ п/п	Плюсы	Минусы
1	Низкая цена	Джиттер (вибрация)
2	Малое энергопотребление	Малый крутящий момент
3	Низкий уровень акустического шума	Малая износостойкость шестерней

4. Дисплей (таблица 10). Служит также для расширения возможностей эмоционального взаимодействия с человеком.

Таблица 10. Плюсы/минусы используемого дисплея

№ п/п	Плюсы	Минусы
1	Низкая цена	Отсутствует тачскрин
2	Диагональ экрана соответствует размерам головы	Неудобное расположение платы управления (приклеена на 2-ух сторонний скотч)

5. Датчики расстояния (таблица 11). Должны быть заменены для соответствия заявленным требованиям по ориентированию робота.

Таблица 11. Плюсы/минусы датчиков расстояния

№ п/п	Плюсы	Минусы
1	Низкая цена	Взаимовлияние датчиков друг на друга
2	Простое подключение	Низкая угловая (15°) разрешающая способность
3	Простое управление	Многократное отражение сигнала

6. Блок реле (таблица 12). Должны быть использованы для корректного управления цепями нагрузки.

Таблица 12. Плюсы/минусы модулей реле/

№ п/п	Плюсы	Минусы
1	Низкая цена	Низкая надежность
2	Гальваническая развязка цепей	Шумность работы
3	Индикация состояния	-

7. Светотехника (таблица 13). Должна быть установлена для предупреждения окружающих о намерении робота совершить движение, и для диагностики неполадок.

Таблица 13. Плюсы/минусы используемой светотехники

№ п/п	Плюсы	Минусы
1	низкое энергопотребление	малая яркость
2	простое крепление и подключение	высокая цена

1.5 Вывод по разделу

В результате анализа вышеприведённых характеристик можно сделать вывод о том, что, несмотря на наличие некоторых недостатков,

рассматриваемое оборудование может быть использовано для изготовления андроидного робота. Таким образом, можно заключить, что выбор компонентов был произведен верно. Описанная выше система является частью мобильной платформы и была использована для обучения студентов по дисциплинам «Основы обработки изображений», «Нейронные сети» и «Нечеткая логика». Использование показало высокую эффективность предложенной системы, отсутствие сбоев во время работы и заинтересованность студентов в процессе обучения.

Данная система может быть использована для проведения исследований и опытно-конструкторских работ в области робототехники, разработки автономных исследовательских систем и систем автоматического управления, а также для проведения практических занятий со студентами, бакалаврами и магистрантами соответствующих направлений.

2 Конструкторская часть

2.1 Проектирование схемотехнических решений

В состав системы управления платформы-робота обычно входят датчики для получения данных об окружающей обстановке и препятствиях, которые представлены ультразвуковыми дальномерами, гироскопом, акселерометром и т.д. Для нормализации сигналов с этих датчиков служат преобразователи, осуществляющие преобразование аналоговых сигналов в унифицированный сигнал.

Преобразование унифицированных сигналов выполняют входные порты МК. К портам МК подключаются аппаратные драйверы [13] управления исполнительными двигателями, обеспечивающие согласование сигналов по уровню токов, и аппаратные и программные интерфейсы, обеспечивающие связь с бортовой ЭВМ.

Также существуют решения, не использующее МК, решающее задачу связи между датчиками и бортовой ЭВМ посредством преобразователей интерфейсов, являющееся альтернативным способом организации внутрисистемных интерфейсов.

В ходе проекта решена инженерная задача создания механических звеньев, обеспечивающих ожидаемые линейные скорости и ускорения, допустимые внешние нагрузки и режимы работы.

Функциональная особенность аппаратно-программного комплекса состоит в интеллектуализации управления движением за счет использования локальных и визуальных датчиков. Система имеет априорную информацию о среде и в процессе функционирования получает текущую информацию как о внешней среде, так и о собственном состоянии. Ее целью является принятие решений, связанных с анализом внешней среды.

На основе проведенного исследования предметной области и анализа наиболее часто используемых компонентов в системах такого рода был сделан вывод о допустимости и достаточности применения микроконтроллера в связке с ЭВМ.

Общая структурная схема андроидного робота представлена в графическом материале (лист «Структурная схема»). Как можно видеть из этой схемы, робот состоит из модулей, связанных между собой различными интерфейсами связи, такими как: I2C (связь гироскопа и МК), RS485, UART, USB, HDMI, Bluetooth, и т.д..

Согласно цели работы и данной структурной схемы необходимо разработать следующие блоки:

- а) подсистема управления движением рук;
- б) подсистема управления движением ног;
- в) подсистема управления движением головы;
- г) подсистема управления движением захватов кистей;
- д) подсистема сбора информации с датчиков расстояния;
- е) подсистема управления курсом с помощью гироскопа;
- ж) подсистема управления питанием бортовой ЭВМ;
- з) подсистема управления устройствами ЧМИ.

2.2 Описание подсистемы управления движением рук

К функциям подсистемы управления движением рук относятся:

- поднять левую руку
- поднять правую руку
- опустить левую руку
- опустить правую руку

Аппаратное обеспечение

- микроконтроллер

- драйвер мотора левой руки
- драйвер мотора правой руки
- концевые выключатели левой руки
- концевые выключатели правой руки

Опишем подробнее вышеперечисленные функции. Чтобы поднять левую руку, нужно выставить высокий уровень на входе драйвера двигателя постоянного тока, подать ШИМ сигнал на вход драйвера при незамкнутых концевых выключателях руки. Необходимо отметить, что для экономии входов микроконтроллера верхний и нижний уровень руки контролируется одним и тем же входом. Это требует выделения дополнительных состояний автомата управления движением рук.

Автомат состояний рук является конечным автоматом. В начале работы рук он находится в состоянии инициализации.

2.3 Описание подсистемы управления движением ног

К функциям подсистемы управления движением ног (см. рис. 2) относятся:

- проехать вперед на X метров
- остановиться
- проехать назад на X метров

Аппаратное обеспечение подсистемы:

- микроконтроллер
- драйвер мотора левой ноги
- драйвер мотора правой ноги

Опишем вышеперечисленные функции подробнее.

Для подсистемы управления движением ног определены следующие команды:

- 20: вернуть состояние подсистемы;
- 21 L C: движение вперед на L метров курсом C;
- 22 L C: движение назад на L метров курсом C;
- 23 : остановка.

Пример обмена данными приведен на листинге 1.

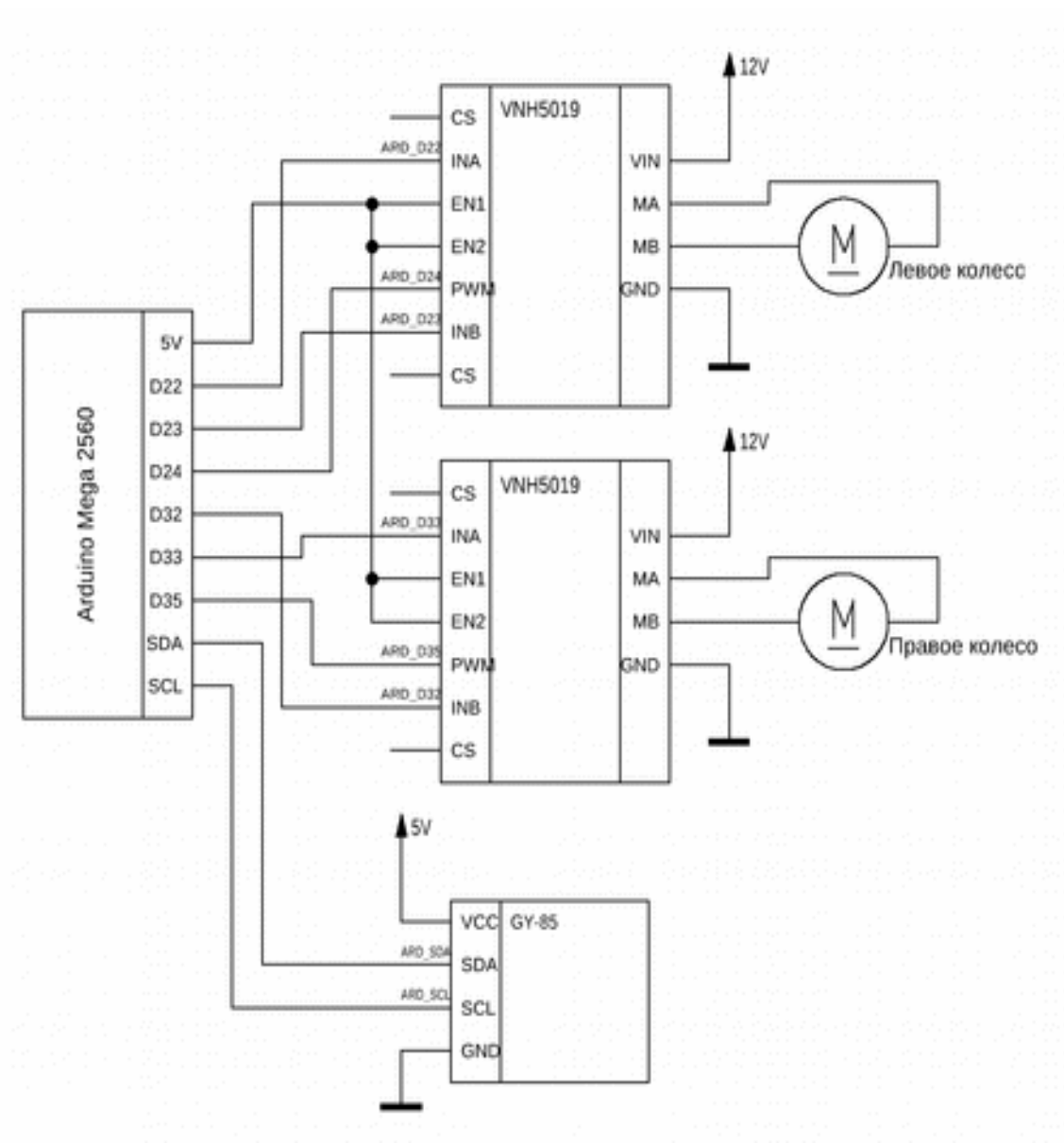


Рисунок 2. Исполнительные механизмы (схема_принципиальная)

Листинг 1. Пример обмена командами:

```
21 320 200 // движение вперед на 320 см курсом 200
// относительно начального положения робота
legs_last_cmd: 21, state: 1 // ответ подсистемы
// с последней командой и состоянием
20 // запрос состояния
legs_last_cmd: 20, avg_pwm: 120, state: 1, direction: 1, trg_course: 200,
course: 202, tgr_distance: 320, distance: 140, lft_pwm: 90, rgt_pwm: 150
```

2.4 Описание подсистемы управления движением головы

Голова робота может двигаться в 2-х плоскостях. Для поворота в горизонтальной плоскости и для наклона головы используются серво-моторы.

К функциям подсистемы управления движением головы относятся:

- повернуть голову на X градусов влево
- повернуть голову на X градусов вправо
- наклонить голову на X градусов вперед
- наклонить голову на X градусов назад

Аппаратное обеспечение:

- микроконтроллер
- сервомоторы
- интерфейсная шина I2C
- драйвер сервомоторов

2.5 Описание подсистемы управления движением захватов кистей

Для управления движением кистей рук (принципиальная схема на рис. 3) используются сервомоторы. К функциям подсистемы управления движением кистей рук относятся:

- сжать кисть левой руки

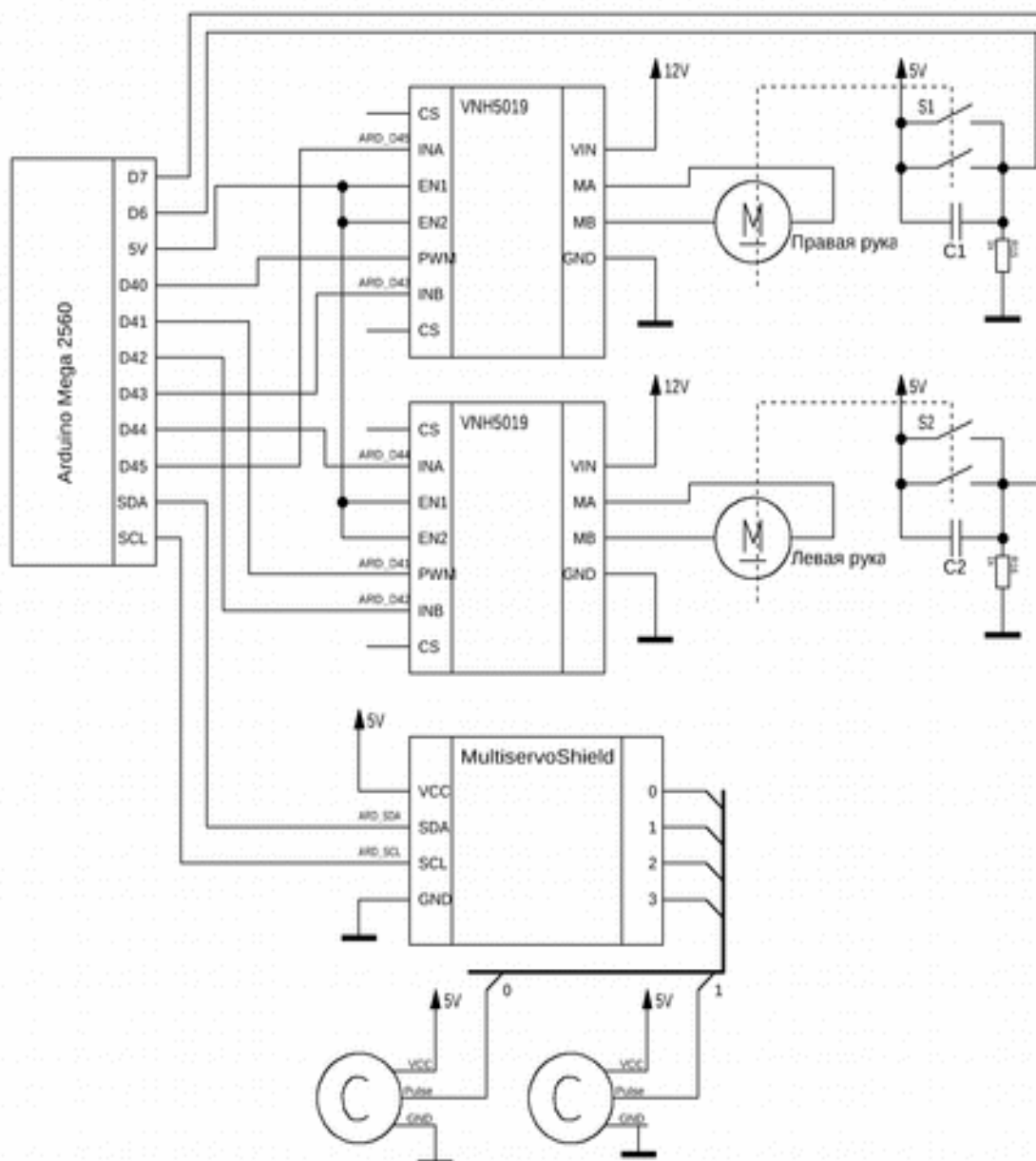


Рисунок 3. Манипуляторные устройства (схема принципиальная)

- сжать кисть правой руки
- отпустить кисть правой руки
- отпустить кисть левой руки

Аппаратное обеспечение

- микроконтроллер
- серво-моторы

- интерфейсная шина I2C
- драйвер серво-моторов

2.6 Описание подсистемы управления курсом с помощью гироскопа

Для сбора информации с гироскопа применяется шина I2C, и библиотека датчика GY_85 [14]. Сначала выполняется калибровка гироскопа, которая заключается в усреднении полученных значений для определения скорости дрейфа гироскопа (алгоритм приведен на рис. 4).

Важнейшим режимом регулирования является «Режим поддержания скорости». При данном режиме мы управляем скоростью перемещения АРК путем регулирования силы тока, подаваемой на ходовой двигатель.

Данный режим активен во время непосредственного движения по ровной местности, в гору и при спуске с наклонной плоскости, при условии, что угол склона не превышает критического значения, когда управление двигателем уже неэффективно. Схема регулятора приведена на рис. 5.

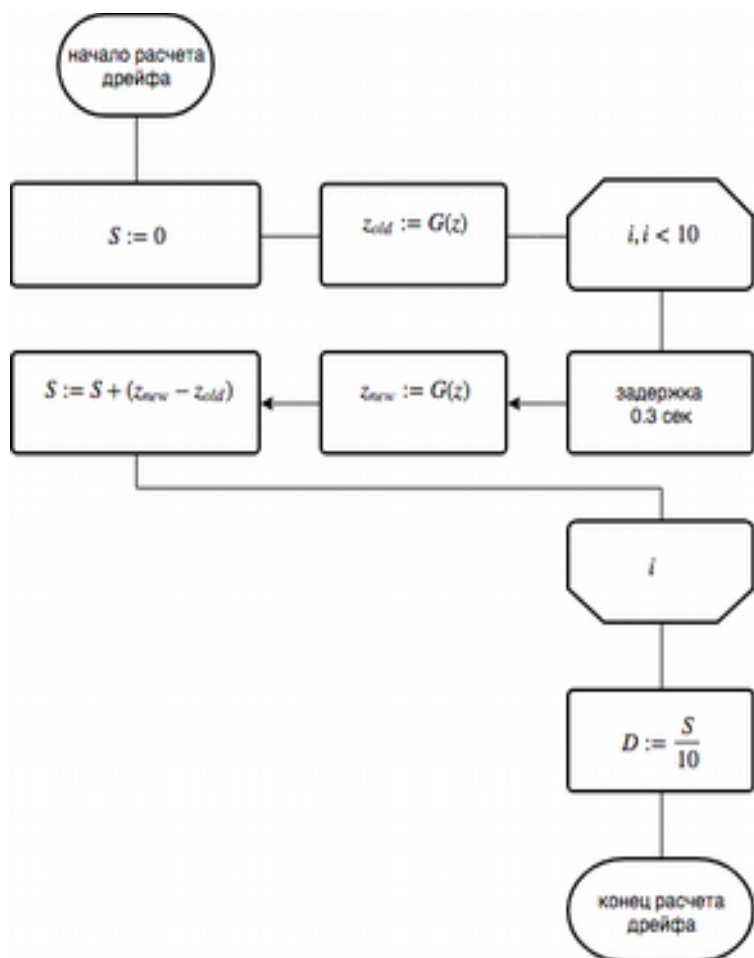


Рисунок 4. Калибровка гироскопа

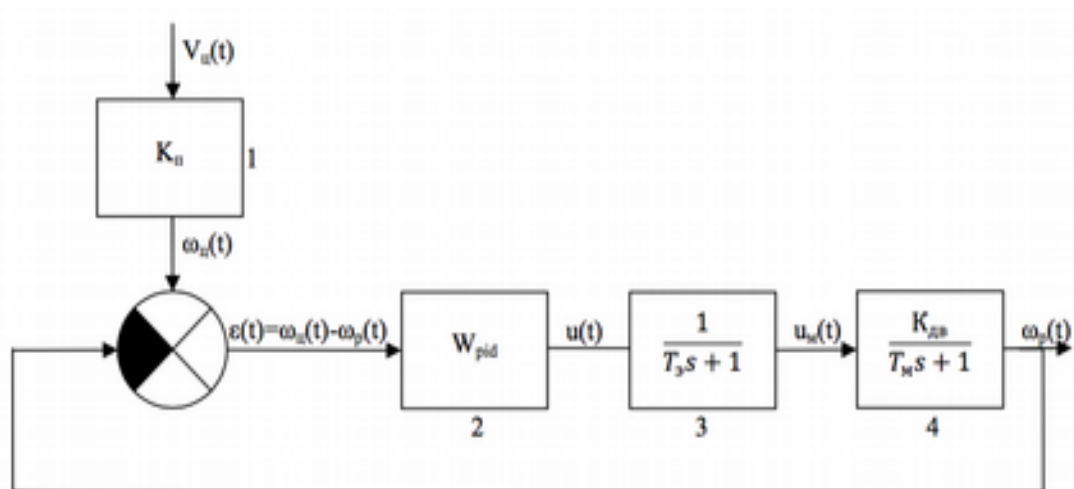


Рисунок 5. Регулятор движения с заданной скоростью

На рисунке 6 отражено функционирование данного режима. В режиме движения робота с заданной скоростью задействованы следующие сигналы и функции:

t – время в с

s – комплексная переменная

$V_{ц}(t)$ – целевая скорость АРК в м/с

$\omega_{ц}(t)$ – целевая скорость вращения колес АРК в оборотах/с

$\omega_{р}(t)$ – реальная скорость вращения колес АРК в оборотах/с

$\varepsilon(t)$ – ошибка управления в оборотах/с

$u(t)$ – сигнал управляющий двигателем

$u_m(t)$ – сигнал, являющийся результатом работы обмотки двигателя

$K_{дв}$ – коэффициент усиления двигателя

T_m – механическая постоянная времени

$T_{э}$ – электрическая постоянная времени

$W_{pid}(s)$ – передаточная функция ПИД-регулятора, отвечающая за управление двигателем

функция 1 – математическое преобразование $V_{ц}(t)$ в $\omega_{к}(t)$

функция 2 – ПИД регулятор, формирующий $u(t)$ исходя из ошибки управления ε

функция 3 – апериодическое звено, отражающее электромагнитные процессы в роторе и статоре двигателя

функция 4 – апериодическое звено, отражающее механические процессы в ходовом двигателе и трансмиссии АРК. Управляющее воздействие $V_{ц}(t)$ поступает в функцию No1, где из м/с преобразуется в $\omega_{к}(t)$ обороты/с, что позволяет использовать естественные для данного процесса сигналы и единицы измерения.

Из результирующего сигнала $\omega_{к}(t)$ вычитается сигнал $\omega_{р}(t)$, являющийся результатом предыдущего цикла регулирования. В результате формируется сигнал $\varepsilon(t)$, представляющий собой ошибку управления.

Функция №2 является ПИД-регулятором. ПИД-регулятор, на базе ошибки управления, формирует управляющий сигнал $u(t)$, который воздействует на двигатель АРК.

Функции №3 и №4 отображают работу электропривода, служащего двигателем АРК. Они представляют собой 2 апериодических звена, которые под действием $u(t)$ вырабатывают сигнал $\omega(t)$ отражающий изменение оборотов двигателя во времени.

Рассмотрим объект управления, в роли которого выступает электропривод, служащий АРК ходовым двигателем.

В упрощенной схеме мы можем приближенно представить его в виде 2х апериодических звеньев, одно из которых отражает работу электромагнитных полей, а второе – механическую составляющую двигателя. При подаче тока на двигатель, сперва происходят процессы, связанные с электромагнитной составляющей привода, и лишь затем вступает в дело механика, что отображено на рисунке 6.

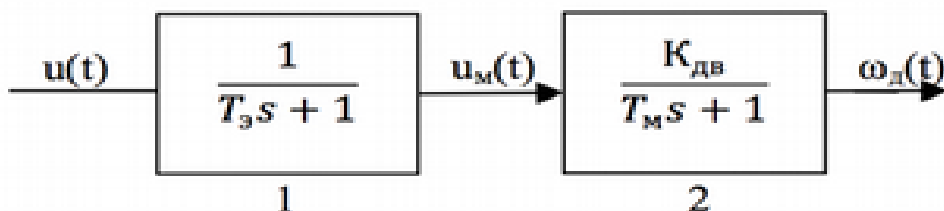


Рисунок 6. Упрощенная схема электропривода

$K_{дв}$ – коэффициент усиления двигателя

T_m – механическая постоянная времени

T_s – электрическая постоянная времени

функция 1 – апериодическое звено, отражающее электромагнитные процессы в роторе и статоре двигателя

функция 2 – апериодическое звено, отражающее механические процессы в двигателе

Рассмотрим представленные величины более подробно.

Коэффициент усиления двигателя $K_{дв}$ определяет пропорциональность между скоростью вращения вала двигателя на холостых оборотах и поданным на вход напряжением. Таким образом он находится по формуле:

$$K_{дв} = \frac{\omega_{хх}}{U_n}$$

Скорость холостого хода $\omega_{хх}$ в оборотах в секунду и U_n задаются в паспорте электродвигателя (номинальное рабочее напряжение). В том случае, если мы не знаем $\omega_{хх}$, она находится экспериментально. Для этого на двигатель подают максимальное напряжение и снимают скорость вращения вала датчиком.

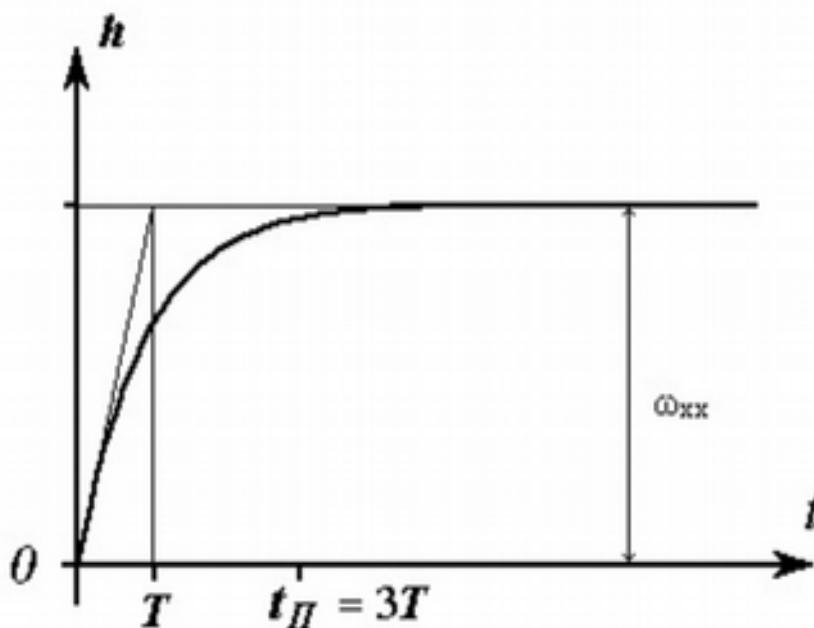


Рисунок 7. График апериодического звена

Механическая постоянная времени T_m зависит, как следует, от инерции всех механических частей входящих в электропривод. Так как в большинстве

случаев это не только детали самого электропривода, но и редуктор, внешняя нагрузка и т.д., то проще всего получит T_m экспериментальным способом.

Для этого на электропривод скачком подают максимальное напряжение, снимая при этом переходной процесс изменения скорости $\omega(t)$.

По данному графику можно определить время регулирования t_P , это время, когда отклонение от целевого значения не превышает 5%, и, как результат, переходной процесс можно считать оконченным.

Если считать влияние первого звена незначительным, в связи с тем, что в электроприводе с механической нагрузкой влияние T_Σ , как правило в 10 раз меньше, то можно принять T_m как T данного переходного процесса.

Учитывая, что T переходного процесса в 3 раза меньше t_P , получим формулу:

$$T_m = \frac{t_P}{3}$$

Следовательно, если мы построим график переходного процесса, то сможем найти T_m . Электрическая постоянная времени T_Σ характеризует инерционность намагничивания якоря двигателя, и ее можно найти по формуле:

$$T_\Sigma = \frac{L_\Sigma}{R_\Sigma}$$

где L_Σ – это индуктивность якоря двигателя, а R_Σ – это его сопротивление, но в практических условиях измерить L_Σ бывает крайне сложно, тогда используется приближение, отраженное в формуле:

$$T_\Sigma \approx \frac{T_m}{10}$$

Передаточной же функцией, записанной, через оператор Лапласа, нашего объекта управления будет формула:

$$W_{\text{дв}}(s) = \frac{K_{\text{дв}}}{(T_\Sigma s + 1)(T_m s + 1)}$$

Следует учесть, что при регулировании нам будут нужны не только текущие показатели АРК, но также и показатели на предыдущем этапе регулирования.

2.7 Описание подсистемы управления питанием бортовых устройств

К функциям подсистемы управления питанием (рис. 8) бортовой ЭВМ относятся:

- включить питание
- замкнуть кнопку включения питания ЭВМ

Аппаратное обеспечение подсистемы:

- микроконтроллер
- модули реле
- бортовая ЭВМ
- блок питания бортовой ЭВМ



Рисунок 8. Система питания бортовой ЭВМ

При включении питания микроконтроллера после инициализации всех подключенных устройств, выставляется низкий уровень на выходе, отвечающем за подачу питания на бортовую ЭВМ (низкий уровень сигнала замыкает реле), и на 100 мс замыкается цепь кнопки включения бортовой ЭВМ.

2.8 Описание подсистемы управления устройствами ЧМИ

К устройствам ЧМИ относятся дисплей, камеры, микрофон, динамики.

К функциям подсистемы управления устройствами ЧМИ относятся:

- включить дисплей
- включить камеры
- включить микрофон
- включить динамики

Аппаратное обеспечение подсистемы:

- микроконтроллер
- модули реле
- камеры
- микрофон
- дисплей
- динамики

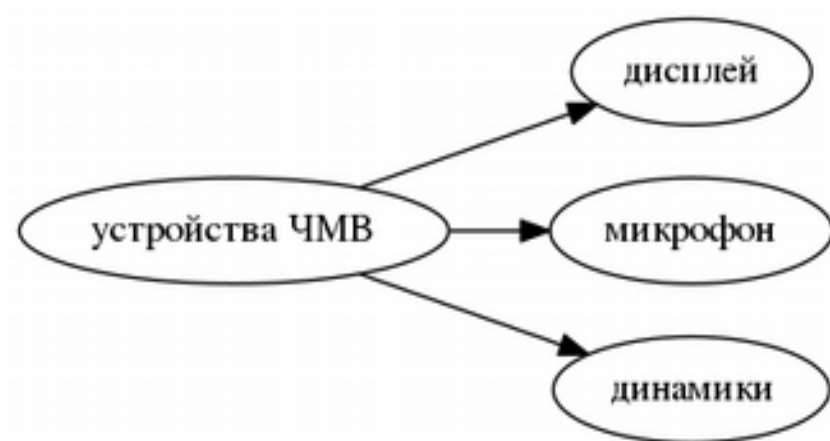


Рисунок 9. Устройства ЧМИ

2.10 Проектирование ПО микроконтроллера

Ниже представлены автоматы и алгоритмы работы подсистем робота.

2.10.1 Автомат работы рук

Граф автомата работы рук приведен в Графическом материале, лист «Автомат рук». Опишем его подробнее. Если автомат в момент 0 находится в состоянии "инициализация", а входное воздействие равно "0", то следующее состояние будет равно "инициально опустить руки". При пребывании автомата в состоянии "инициально опустить руки" в момент времени 1 имея на входе воздействие "нажат к.в.", автомат сменит состояние на "остановка". При пребывании автомата в состоянии "начать движ. вниз" в момент времени 2 имея на входе воздействие "время больше 10 сек", автомат сменит состояние на "ошибка". При пребывании автомата в состоянии "начать движ. вниз" в момент времени 3 имея на входе воздействие "к.в. отпущен", автомат сменит состояние на "продолжить опускать руки". В момент времени 4 автомат перешел в состояние "продолжить опускать руки", и входное воздействие равно "время больше 10 сек", то он перейдет в состояние "ошибка".

Если в момент времени 5 автомат находится в состоянии "продолжить опускать руки" при входном воздействии "к.в. нажат", то он перейдет в состояние "остановка". В момент времени 6 робот находится в состоянии "начать движ. вверх", и при входном воздействии "время больше 10 сек", он переходит в состояние "ошибка". В момент времени 7 робот находится в состоянии "начать движ. вверх", и при входном воздействии "к.в. отпущен", он переходит в состояние "продолжить поднимать руки". Если в момент времени 8 автомат находится в состоянии "продолжить поднимать руки" при входном воздействии "время больше 10 сек", то он перейдет в состояние "ошибка". В момент времени 9 автомат перешел в состояние "продолжить поднимать руки", и входное воздействие равно "к.в. нажат", то он перейдет в состояние "остановка".

В момент 10 автомат пребывает в состоянии "остановка", и, получив воздействие "0", он перейдет в состояние "ожидание". При пребывании автомата

в состоянии "ожидание" в момент времени 11 имея на входе воздействие "принята команда вниз", автомат сменит состояние на "начать движ. вниз". Если автомат в момент 12 находится в состоянии "ожидание", а входное воздействие равно "принята команда вверх", то следующее состояние будет равно "начать движ. вверх".

2.10.2 Автомат работы головы робота

Автомат работы головы робота приведен на рис. 10. Опишем его работу.



Рисунок 10. Автомат движения головы

Если автомат в момент 0 находится в состоянии "инициализация", а входное воздействие равно "0", то следующее состояние будет равно "установка в начальное положение". Если в момент времени 1 автомат находится в состоянии "установка в начальное положение" при входном воздействии "требуемый угол достигнут", то он перейдет в состояние "начальное положение установлено". При пребывании автомата в состоянии "начальное положение установлено" в момент времени 2 имея на входе воздействие "угол меньше требуемый", автомат сменит состояние на "поворот на X градусов в горизонтальной плоскости". Если в момент времени 3 автомат находится в состоянии "начальное положение установлено" при входном воздействии "угол

меньше требуемый", то он перейдет в состояние "поворот на X градусов в вертикальной плоскости".

В момент времени 4 робот находится в состоянии "поворот на X градусов в горизонтальной плоскости", и при входном воздействии "требуемый угол достигнут", он переходит в состояние "поворот завершен". Если в момент времени 5 автомат находится в состоянии "поворот на X градусов в вертикальной плоскости" при входном воздействии "требуемый угол достигнут", то он перейдет в состояние "поворот завершен". При пребывании автомата в состоянии "поворот на X градусов в вертикальной плоскости" в момент времени 6 имея на входе воздействие "требуемый угол достигнут", автомат сменит состояние на "поворот завершен". Если в момент времени 7 автомат находится в состоянии "поворот на X градусов в вертикальной плоскости" при входном воздействии "требуемый угол достигнут", то он перейдет в состояние "поворот завершен".

2.10.3 Автомат управления питанием компьютера



Рисунок 11. Автомат управления компьютером

Если автомат (рис 11) в момент 0 находится в состоянии "инициализация", а входное воздействие равно "0", то следующее состояние будет равно "компьютер выключен". В момент времени 1 автомат перешел в состояние "компьютер выключен", и входное воздействие равно "0", то он

перейдет в состояние "компьютер включается". В момент 2 автомат пребывает в состоянии "компьютер включается", и, получив воздействие "0", он перейдет в состояние "компьютер загружается". В момент времени 3 робот находится в состоянии "компьютер загружается", и при входном воздействии "получено подтверждающая последовательность", он переходит в состояние "компьютер загрузился". Если автомат в момент 4 находится в состоянии "компьютер загружается", а входное воздействие равно "превышено время ожидания", то следующее состояние будет равно "компьютер не загрузился".

2.10.4 Автомат управления светотехникой

Принципиальная схема модуля светотехники приведена на рис. 12, автомат, описывающий ее работу — на рис. 13. Рассмотрим работу автомата.

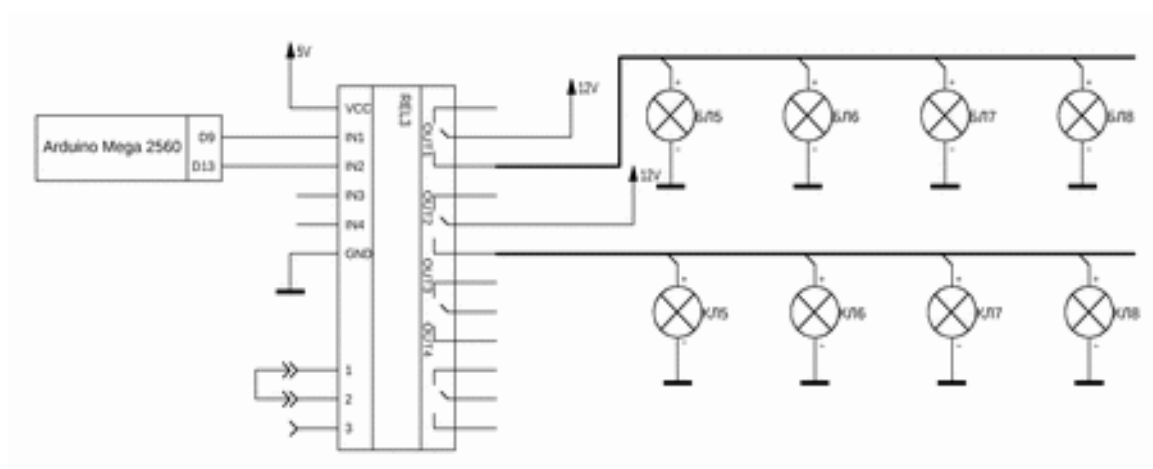


Рисунок 12. Светотехника (схема принципиальная)

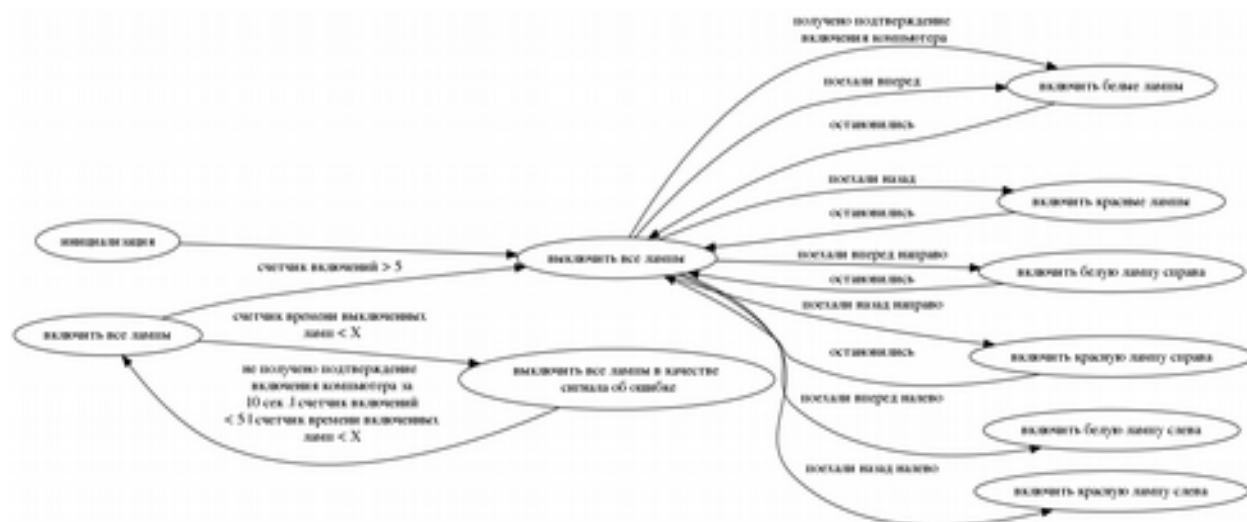


Рисунок 13. Автомат управления светотехникой

Если автомат в момент 0 находится в состоянии "инициализация", а входное воздействие равно "0", то следующее состояние будет равно "выключить все лампы". В момент 1 автомат пребывает в состоянии "выключить все лампы", и, получив воздействие "получено подтверждение включения компьютера", он перейдет в состояние "включить белые лампы". В момент времени 2 робот находится в состоянии "выключить все лампы в качестве сигнала об ошибке", и при входном воздействии "не получено подтверждение включения компьютера за 10 сек. | счетчик включений меньше 5 | счетчик времени включенных ламп меньше X", он переходит в состояние "включить все лампы". В момент времени 3 автомат перешел в состояние "включить все лампы", и входное воздействие равно "счетчик времени выключенных ламп меньше X", то он перейдет в состояние "выключить все лампы в качестве сигнала об ошибке".

При пребывании автомата в состоянии "включить все лампы" в момент времени 4 имея на входе воздействие "счетчик включений больше 5", автомат сменит состояние на "выключить все лампы". При пребывании автомата в состоянии "выключить все лампы" в момент времени 5 имея на входе воздействие "поехали вперед", автомат сменит состояние на "включить белые лампы". Если автомат в момент 6 находится в состоянии "выключить все

лампы", а входное воздействие равно "поехали назад", то следующее состояние будет равно "включить красные лампы". При пребывании автомата в состоянии "включить белые лампы" в момент времени 7 имея на входе воздействие "остановились", автомат сменит состояние на "выключить все лампы".

При пребывании автомата в состоянии "включить красные лампы" в момент времени 8 имея на входе воздействие "остановились", автомат сменит состояние на "выключить все лампы". При пребывании автомата в состоянии "выключить все лампы" в момент времени 9 имея на входе воздействие "поехали вперед направо", автомат сменит состояние на "включить белую лампу справа". При пребывании автомата в состоянии "выключить все лампы" в момент времени 10 имея на входе воздействие "поехали назад направо", автомат сменит состояние на "включить красную лампу справа". В момент 11 автомат пребывает в состоянии "выключить все лампы", и, получив воздействие "поехали вперед налево", он перейдет в состояние "включить белую лампу слева".

Если автомат в момент 12 находится в состоянии "выключить все лампы", а входное воздействие равно "поехали назад налево", то следующее состояние будет равно "включить красную лампу слева". При пребывании автомата в состоянии "включить белую лампу справа" в момент времени 13 имея на входе воздействие "остановились", автомат сменит состояние на "выключить все лампы". В момент 14 автомат пребывает в состоянии "включить красную лампу справа", и, получив воздействие "остановились", он перейдет в состояние "выключить все лампы".

2.10.5 Автомат обхода препятствий

Обход препятствий также реализован при помощи автомата (см. рис. 14). Его описание приведено ниже.



Рисунок 14. Автомат обхода препятствий

Если автомат в момент 0 находится в состоянии "стоим", а входное воздействие равно "потребность", то следующее состояние будет равно "захотели ехать". Если автомат в момент 1 находится в состоянии "захотели ехать", а входное воздействие равно "нет_препятствий(прямо)", то следующее состояние будет равно "ехать прямо". В момент времени 2 робот находится в состоянии "захотели ехать", и при входном воздействии "нет препятствий справа", он переходит в состояние "ехать направо". Если автомат в момент 3 находится в состоянии "захотели ехать", а входное воздействие равно "нет препятствий слева", то следующее состояние будет равно "ехать налево". В момент 4 автомат пребывает в состоянии "ехать прямо", и, получив воздействие "нет препятствий справа", он перейдет в состояние "ехать направо". В момент 5 автомат пребывает в состоянии "ехать прямо", и, получив воздействие "нет препятствий слева", он перейдет в состояние "ехать налево". При пребывании автомата в состоянии "ехать направо" в момент времени 6 имея на входе воздействие "нет препятствий прямо", автомат сменит состояние на "ехать прямо".

Если в момент времени 7 автомат находится в состоянии "ехать налево" при входном воздействии "нет препятствий прямо", то он перейдет в состояние "ехать прямо".

2.10.6 Обработка датчиков расстояния

Работу датчиков расстояния, которая также реализована автоматом удобнее, тем не менее рассмотреть в алгоритмической форме в виду того, что число состояний этой подсистемы очень велико. Алгоритм приведен на рис. 15.

Ветка ожидание:

Блок 1: проверка состояния датчика расстояния.

Входы: состояние датчика

Выходы: направление дальнейшей работы (ожидание, триггер, эхо).

Блок 2: проверка времени ожидания.

Входы: время ожидания t (мс).

Выходы: если t больше 1000мс, то истина, иначе ложь

Блок 3: состояние датчика := триггер

Входы: переменная состояния

Выходы: состояние датчика

Ветка триггер:

Блок 4: проверка состояния триггера

Входы: состояние датчика

Выходы: фронт вниз; фронт вверх; ожидание;

Блок 5: проверка времени импульса

Входы: фронт вниз

Выходы: если время импульса больше 10 мс, то истина, иначе - ложь

Блок 6: выставить низкий уровень на выходе триггера

Входы: -

Выходы: состояние выхода триггера

Блок 7: состояние триггера := ожидание

Входы: -

Выходы: состояние триггера

Блок 8: состояние := эхо

Входы: -

Выходы: состояние датчика

Блок 9: выставить высокий уровень на выходе триггера

Входы: фронт вверх

Выходы: состояние выхода триггера

Блок 10: состояние триггера := фронт вниз

Входы: -

Выходы: состояние триггера

Блок 11: состояние триггера := фронт вверх

Входы: -

Выходы: состояние триггера

Ветка 3:

Блок 12: проверка состояния эхо

Входы: состояние датчика

Выходы: фронт вверх; фронт вниз; ожидание;

Блок 13: проверить уровень на входе эхо по фронту вверх

Входы: уровень

Выходы: если низкий, то истина, иначе - ложь

Блок 14: проверить уровень на входе эхо по фронту вниз

Входы: уровень

Выходы: если низкий, то истина, иначе - ложь

Блок 15: вычислить расстояние

Входы: длительность импульса

Выходы: расстояние

Блок 16: состояние эхо := фронт вниз

Входы: состояние эхо

Выходы: состояние эхо

Блок 17: запомнить момент прихода импульса

Входы: импульс

Выходы: время прихода импульса

Блок 18: состояние эхо := фронт вверх

Входы: состояние эхо

Выходы: состояние эхо

2.10.7 Основной алгоритм программы микроконтроллера

Основной алгоритм микроконтроллера, как следует из названия, служит для реализации основного набора функций. В нем происходит вызов всех обработчиков всех подсистем и их инициализация (см. рис. 16).

Блок 1: инициализация устройств и подсистем

Входы: -

Выходы: состояние портов ввода/вывода, вектора прерываний

Блок 2: основной управляющий алгоритм

Входы: состояние датчиков, команды с бортовой эвм

Выходы: управляющее воздействие, ответы о выполнении команд



Рисунок 16. Основной алгоритм

2.10.8 Инициализация устройств

Инициализация устройств служит для настройки портов ввода-вывода и перемещения исполнительных механизмов в начальное состояние.

Блок 1: Инициализация последовательного порта

Входы: -

Выходы: состояние последовательного порта (RX/TX), прерывания

Блок 2: Инициализация реле компьютера

Входы: -

Выходы: состояния пинов реле

Блок 3: Инициализация светотехники

Входы: -

Выходы: состояния пинов реле светотехники

Блок 4: Инициализация шины I2C

Входы: -

Выходы: состояние шины I2C

Блок 5: Инициализация сети 5V

Входы:

Выходы: состояние пинов реле цепи 5V

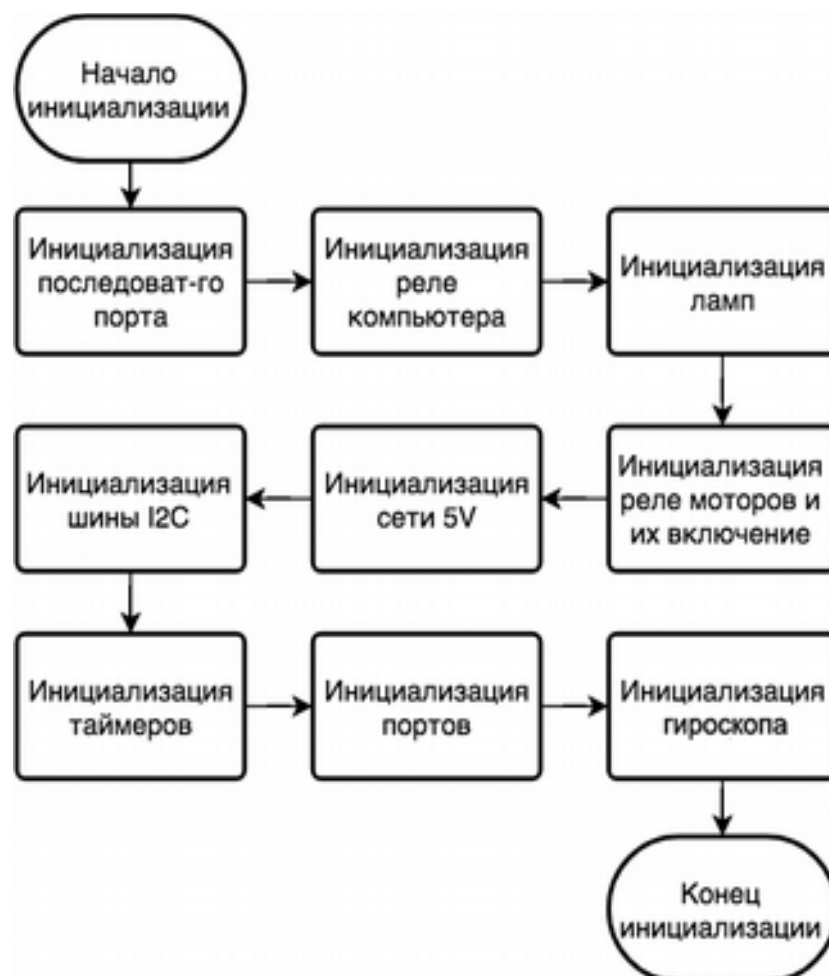


Рисунок 17. Инициализация устройств

Блок 6: Инициализация реле моторов и их включение

Входы:

Выходы: состояние пинов драйверов моторов

Блок 7: Инициализация таймеров

Входы:

Выходы: состояние регистров таймеров

Блок 8: Инициализация портов

Входы:

Выходы: состояние портов

Блок 9: Инициализация АЦП

Входы:

Выходы: состояние регистров АЦП

2.10.9 Обработчик прерывания по первому таймеру

Блок 1: Сброс счетчика регистра прерываний

Входы: Счетчик

Выходы: Счетчик = 0

Блок 2: Обработка команд управления компьютером

Входы: Команда

Выходы: Результат выполнения

Блок 3: Обработка команд управления светотехникой

Входы: Команда

Выходы: Результат выполнения

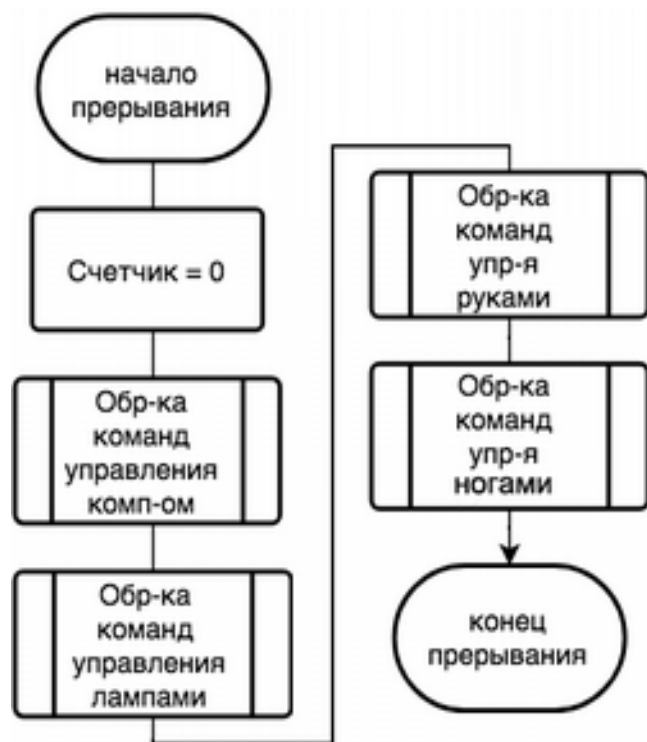


Рисунок 18. Обработчик прерывания по первому таймеру

Блок 4: Обработка команд управления руками

Входы: Команда

Выходы: Результат выполнения

Блок 5: Обработка команд управления ногами

Входы: Команда

Выходы: Результат выполнения

2.10.10 Обработчик прерывания по второму таймеру

Блок 1: Сброс счетчика регистра прерываний

Входы: Счетчик

Выходы: Счетчик = 0



Рисунок 19. Обработчик прерывания по второму таймеру

Блок 2: Счетчик импульсов датчиков расстояния += 1

Входы: Счетчик

Выходы: Значение счетчика, увеличенное на единицу

Блок 3: Обработка команд управления датчиками расстояния

Входы: Команда

Выходы: Результат выполнения

2.10.11 Инициализация гироскопа (расчет дрейфа)

Блок 1: Инициализация

Входы: Переменная суммы значений

Выходы: Переменная суммы значений

Блок 2: первичное измерение вращательного ускорения по оси z

Входы: переменная для хранения

Выходы: измеренное значение

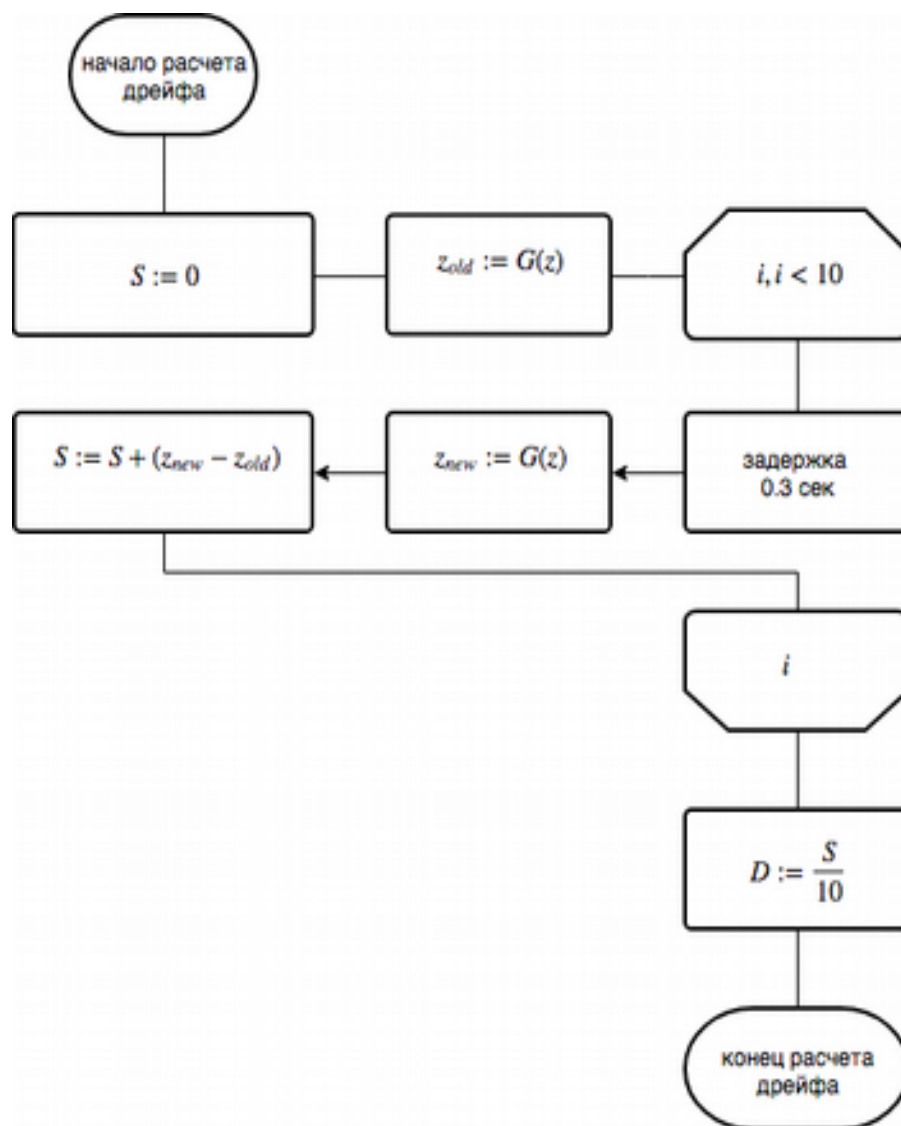


Рисунок 20. Калибровка гироскопа

Блок 3: инициализация цикла

Входы: переменная счетчик цикла

Выходы: -

Блок 4: задержка 0.3 сек

Входы: счетчик задержки

Выходы: -

Блок 5: новое измерение ускорения по z

Входы: переменная для хранения значения

Выходы: значение

Блок 6: подсчет суммы измерений

Входы: переменная суммы

Выходы: значение суммы

Блок 7: конец цикла

Входы: -

Выходы: переменная счетчик

Блок 8: подсчет дрейфа

Входы: переменная для хранения значения дрейфа

Выходы: значение дрейфа

2.11 Вывод по разделу.

Таким образом, нами получены описания работы всех основших подсистем робота в виде конечных автоматов и/или алгоритмов. Реализация предполагается в Технологическом разделе. Также разработаны принципиальная схема всего робота и отдельных его подсистем.

4 Информационная безопасность

1. Определение класса АС

Определим класс нашей системы, основываясь на руководящем документе "Классификация АС и требования по защите от несанкционированного доступа к информации".

Предлагаемая система предназначена для общения с людьми и привлечения внимания к Университету.

Система имеет следующие возможности:

передвижение в автоматическом и/или дистанционно-управляемом режимах с требуемой скоростью и точностью по заданному маршруту,

имитирование человеческих эмоций

реагирование на раздражители

поддержание разговора - распознавать речь и отвечать на вопросы.

4.1 Анализ исходных данных

Таблица 1 перечень защищаемых информационных ресурсов АС и их уровень конфиденциальности

1. модель распознавания речи; секретная информация
2. модель синтеза речи; секретная информация
3. прошивка контроллера; внутренняя информация
4. настройки операционной системы; внутренняя информация
5. файлы исходного кода приложения для управления исполнительными механизмами; внутренняя информация
6. файлы исходного кода приложения для управления эмоциями и рефлексам; конфиденциальная информация
7. база знаний; секретная информация
8. база данных эмоций и рефлексов; секретная информация

9. база данных голосвых записей для воспроизведения речи;
конфиденциальная информация

10. интерфейс пользователя; открытая информация

11. речевое API; открытая информация

Таблица 2 перечень лиц, имеющих доступ к штатным средствам АС, с указанием их уровня полномочий

1. Администратор; доступ к секретной информации;

2. Разработчик; доступ к секретной информации;

3. Тестировщик; доступ к внутренней информации;

4. Пользователь-собеседник; доступ к открытой информации.

Таблица 13. Матрица доступа или полномочий субъектов доступа по отношению к защищаемым информационным ресурсам АС

	1	2	3	4	5	6	7	8	9	10	11
1. Администратор	+	+	+	+	+	+	+	+	+	+	+
2. Разработчик	+	+	+	+	+	+	+	+	+	+	+
3. Тестировщик	-	-	+	+	+	-	-	-	-	+	+
4. Собеседник	-	-	-	-	-	-	-	-	-	+	+

Режим обработки данных в АС

Работы с системой могут производиться одновременно только одним человеком, следовательно режим обработки данных является индивидуальным.

4.2 Выявление основных признаков АС

Таблица 14. Основные признаки АС.

Наличие в ас информации различного уровня конфиденциальности	+
Уровень полномочий субъектов доступа ас на доступ к конфиденциальной информации	+

Наличие в ас информации различного уровня конфиденциальности	+
Режим обработки данных в ас	Индивидуальный

4.3 Сравнение выявленных признаков АС с классифицируемыми

Устанавливается девять классов защищенности АС от НСД к информации.

Каждый класс характеризуется определенной минимальной совокупностью требований по защите. Классы подразделяются на три группы, отличающиеся особенностями обработки информации в АС. В пределах каждой группы соблюдается иерархия требований по защите в зависимости от ценности (конфиденциальности) информации и, следовательно, иерархия классов защищенности АС. Третья группа включает АС, в которых работает один пользователь, допущенный ко всей информации АС, размещенной на носителях одного уровня конфиденциальности. Группа содержит два класса - 3Б и 3А.

Вторая группа включает АС, в которых пользователи имеют одинаковые права доступа (полномочия) ко всей информации АС, обрабатываемой и (или) хранимой на носителях различного уровня конфиденциальности. Группа содержит два класса - 2Б и 2А. Первая группа включает многопользовательские АС, в которых одновременно обрабатывается и (или) хранится информация разных уровней конфиденциальности. Не все пользователи имеют право доступа ко всей информации АС. Группа содержит пять классов - 1Д, 1Г, 1В, 1Б и 1А. 1.4 присвоение АС соответствующего класса защиты информации от НСД

На основе анализа исходных данных, можно сделать вывод, что разрабатываемая система относится к первой группе АС, и классу защищенности 1Д.

4.5 Требования к подсистемам класса защищенности 1Д

В общем случае, комплекс программно-технических средств и организационных (процедурных) решений по защите информации от НСД реализуется в рамках системы защиты информации от НСД (СЗИ НСД), условно состоящей из следующих четырех подсистем:

- управления доступом;
- регистрации и учета;
- криптографической;
- обеспечения целостности.

Технические характеристики робота:

- Микроконтроллер (Arduino 2560)
- Платы расширения (модули реле, H-моста, БИНС, серво-приводы, электродвигатели, провода, ультразвуковые датчики расстояния, камеры, и т.д.)
- Бортовая ЭВМ, ОС - Debian 8

Требования к классу защищенности 1Д:

Подсистема управления доступом:

Идентификация и проверка подлинности таких субъектов как администратор осуществляется средствами операционной системы. Доступ возможен через ssh по сети или локально, через gdm. Gdm - графический дисплейный менеджер для X Window System. Эта программа регистрирует пользователей в операционной системе, предлагая им ввести их логин и пароль. GDM также позволяет выбирать графическую среду (сеанс) и локаль (набор языковых и региональных настроек).

Идентификация и проверка подлинности пользователей-собеседников осуществляется визуально при помощи системы распознавания лиц и голоса.

Подсистема регистрации и учета:

Регистрация факта входа всех пользователей, кроме собеседников осуществляется средствами ОС в файле /var/log/auth.log, а остальных - в

журналах программы. Регистрация выхода из системы или останова не проводится в моменты аппаратурного отключения АС. В параметрах регистрации указываются:

- дата и время входа (выхода) субъекта доступа в систему (из системы) или загрузки (останова) системы;
- результат попытки входа: успешная или неуспешная - несанкционированная;
- идентификатор (код или фамилия) субъекта, предъявленный при попытке доступа;

Подсистема обеспечения целостности:

Целостность программных средств СЗИ НСД достигается за счет подключения файловой системы /usr в режиме "только для чтения" командой `mount /dev/{device} /home/{directory}`

Целостность обрабатываемой информации обеспечивается с помощью исключения взаимодействия пользователя-собеседника непосредственно с операционной системой в обход программного обеспечения посредством визуального интерфейса.

Неизменность программной среды при работе пользователя-собеседника обеспечивается настройкой прав доступа на каталоги программы и ее модулей средствами Linux командой `chmod -X {directory}/`.

При этом:

- целостность СЗИ НСД проверяется при загрузке системы по контрольным хэшам компонент СЗИ запуском скрипта `tc.local`. Используются хэши sha256. Хэши компонент хранятся в файловой системе `/var/ashes`, которая отключается из системы после проверки. Т.о. злоумышленник не может подменить или рассчитать коллизии хэшей.
- целостность программной среды обеспечивается использованием трансляторов с языков высокого уровня и отсутствием средств модификации

объектного кода программ в процессе обработки и (или) хранения защищаемой информации;

- физическая охрана СВТ работа, предусматривающая контроль доступа к ним посторонних лиц, осуществляется самим роботом, так как он оборудован руками и клешнями, и может произвести самозащиту в случае опасности;

- периодическое тестирование функций СЗИ НСД при изменении программной среды и персонала АС с помощью тест - программ, имитирующих попытки НСД производится;

- имеются средства восстановления СЗИ НСД, предусматривающие ведение двух копий программных средств СЗИ НСД и их периодическое обновление и контроль работоспособности. Резервное копирование выполняется с помощью планировщика задач cron и скриптов на языке shell.

5. Экономический раздел

5.1 Расчет расходов на проект

Ориентировочная экономическая эффективность на данный момент не рассчитывается, т.к. робот является прототипом и не планируется поступать в продажу.

Расходы на создание прототипа андроидного робота приведены в таблице 3.

Так же рассчитаем сумму амортизационных отчислений, за 5 лет, на следующее оборудование: микроконтроллер Arduino, экран, камера, аккумулятор, драйверы моторов. Расчет произведем по формуле: $A = \frac{K \cdot C}{n}$, где: K – норма амортизации в процентах к первоначальной (восстановительной) стоимости объекта; C – стоимость объекта; n - срок полезного использования данного объекта.

Для Arduino на срок службы 5 лет, норма амортизации равна или 112 руб.

Для экрана на срок службы 5 лет, норма амортизации равна или 420 руб.

Для камер на срок службы 5 лет, норма амортизации равна или 70 руб.

Для аккумулятора на срок службы 5 лет, норма амортизации равна или 500 руб.

Для драйверов мотора на срок службы 5 лет, норма амортизации равна или 240 руб.

Данные расчеты, для наглядности, сформируем в таблицу 15 и произведем подсчет сумм на количество оборудования.

Таблица 15. Расходы на используемые технические средства

Наименование	Параметры	Количество (шт.)	Цена (за шт.)
Компьютер	На базе Core i7, 4 Гб ОЗУ, SSD 64 Гб	1	5800 руб.
Контроллер	Arduino Atmega 2560	1	560 руб.
Экран	SkyLarpu 9 дюймов	1	2100 руб.
Камера	Logitech CX900	2	700 руб.
Датчик движения	HC-SR501	1	70 руб.
Аккумулятор	12V, 60Ah	1	2500 руб.
Динамик	Dialog HR-1	2	400 руб.
Микрофон	Dialog 300-M	1	300 руб.
Приводы рук	6K4.959.802.F	2	1270 руб.
Концевые выключатели	ZCMD	2	300 руб.
Приводы колес	6K4.959.802.F	2	1200 руб.
Сервоприводы кистей	FS5519M	2	540 руб.
Сервоприводы головы	FS5600M	2	495 руб.
Плата расширения сервоприводов	Multiservo Shield	1	990 руб.
Драйверы моторов	Pololu VNH5019	2	2400 руб.
Модули реле	4 Relay Module	2	900 руб.
Гироскоп	GY-85	1	150 руб.
Преобразователь	AC/DC	1	130 руб.
Итого			30090 руб.

Таблица 16. Сумма амортизационных отчислений

Наименование	Амортизационные отчисления на 1 единицу (руб.)	Сумма амортизационных отчислений (руб.)
Контроллер Arduino	112/5 лет	112/5 лет
Экран	420/5 лет	420/5 лет
Камера	70/5 лет	140/5 лет
Аккумулятор	500/5 лет	500/5 лет
Драйвер мотора	240/5 лет	480/5 лет

Также необходимо рассчитать стоимость доработки оборудования (таблица 17).

Таблица 17. Расходы на дополнительное оборудование.

Наименование	Параметры	Количество (шт.)	Цена (за шт.)
Датчик расстояния	HC-SR04	6	180 руб.
Детектор звука	LM400	4	300 руб.
Датчик освещенности	LM393	2	170 руб.
Датчик температуры и влажности	DHT11	2	100 руб.
Итого			2820 рублей.

5.2 Выводы по экономическому разделу

Для обоснования экономической целесообразности проекта были произведены расчеты необходимых материальных затрат для проекта, а так же затраты на доработку существующей системы.

В данной главе были рассчитаны затраты на амортизацию основного оборудования.

Как видно из таблицы 4, проект является экономически выгодным.

6 Технологическая часть

6.1 Изготовление и сборка схемотехнических решений

Исходя из предложенного в разделе 2.1 деления блоков по выполняемым действиям и выдвинутых в разделе 2 требований была разработана принципиальная электрическая схема андроидного робота. Она показана в Графическом материале, лист «Принципиальная схема». Для удобства дальнейшего описания разберём отдельные подсистемы робота.

В их число входят:

- а) подсистема управления движением рук;
- б) подсистема управления движением ног;
- в) подсистема управления движением головы;
- г) подсистема управления движением захватов кистей;
- д) подсистема сбора информации с датчиков расстояния;
- е) подсистема управления курсом с помощью гироскопа;
- ж) подсистема управления питанием бортовой ЭВМ;
- з) подсистема управления устройствами ЧМИ.

Для разработки аппаратной части использовались плата arduino с микроконтроллером Atmega 2560, гироскоп, акселерометр, плата драйверов двигателя, блок реле, ультразвуковые датчики измерители расстояния, преобразователь интерфейсов USART → rs485 → USB, серво-моторы, электродвигатели постоянного тока и другие компоненты. Также в системе присутствует бортовая ЭВМ для выполнения части (какой?) функций робота.

Для разработки программной системы использовался язык программирования C и библиотеки Arduino.

6.2 Разработка ПО микроконтроллера

В качестве приводов рук и ног используются электродвигатели постоянного тока от стеклоподъемников ВАЗ 2110. Эти двигатели (см. рис. 23) обладают большим крутящим моментом, достаточным для поворота рук и перемещения робота.

Скорость вращения двигателей может изменяться под действием ШИМ-сигнала. Направление вращения задается полярностью подключения. Таким образом, для управления этим двигателем идеально подходит плата VNH5019A.

Это сдвоенный мостовой драйвер, предназначенный для управления электродвигателями постоянного тока. ИМС имеет возможность управления выходными цепями с помощью ШИМ-сигнала. Схема включения — типовая, взята из официальной документации.

Одна микросхема VNH5019A предоставляет два независимых «мостовых» выхода или четыре обычных силовых выхода нагрузочной способностью до 30 ампер при напряжении до 50 вольт. Это больше, чем требуется, и опыту практического применения этой микросхемы сказать, что она выдерживает токи до 30 ампер на канал без каких-либо последствий при наличии качественного охлаждения. В схеме блока применены две такие микросхемы; таким образом, число силовых выходов может варьироваться от четырех до восьми.

Каждый из мостовых выходов VNH5019A (рис. 22) имеет возможность измерения величины проходящего через него постоянного тока за счет наличия вывода — Current sensing A/B \parallel .

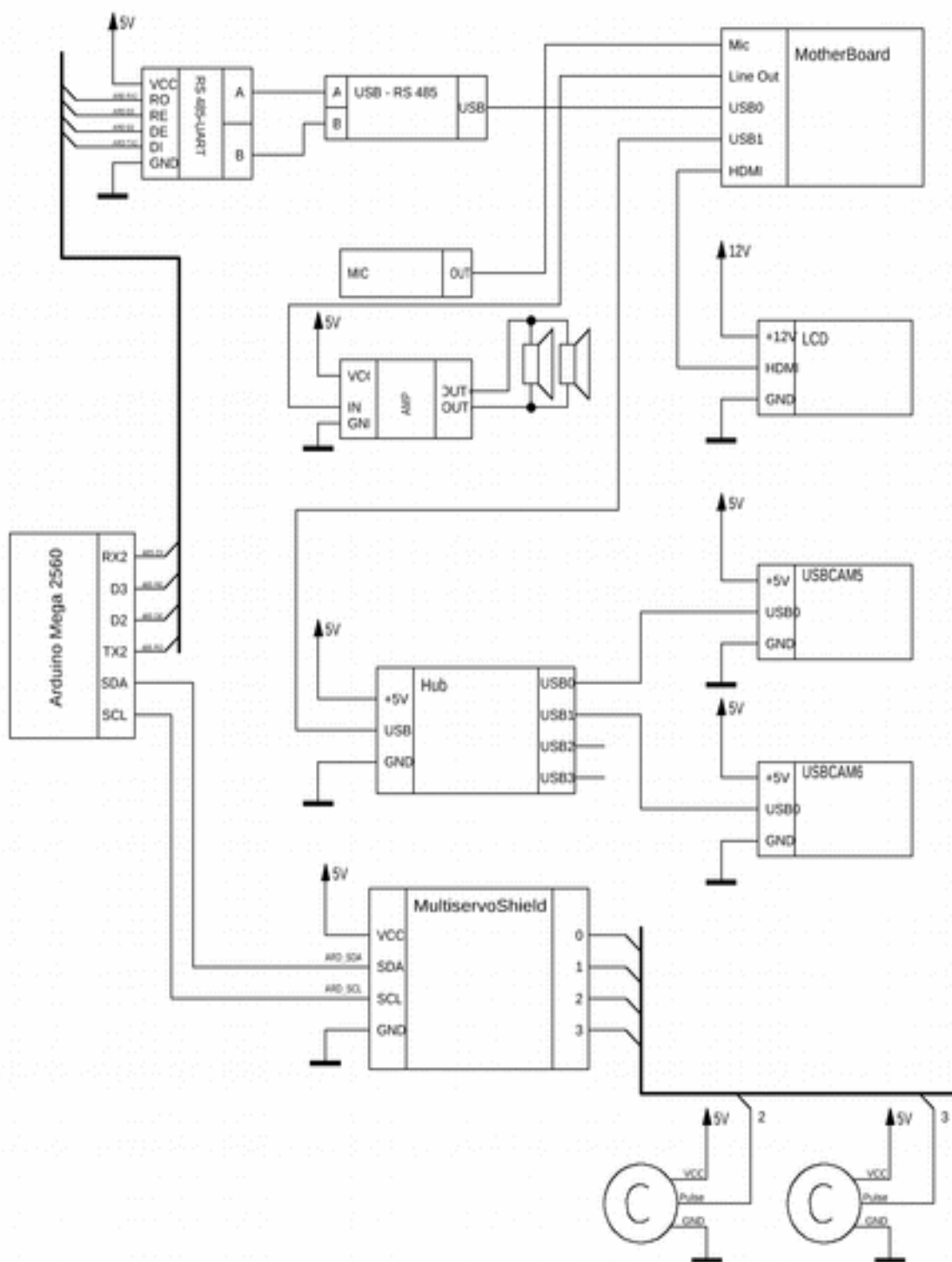


Рисунок 21. Устройства ЧМВ (схема принципиальная)

Технически это общая точка драйверов обоих полумостов канала, которая должна подключаться к общему проводу силового питания через токовый шунт или напрямую, если измерение тока не требуется.

Для защиты силовых выходов от всплесков ЭДС самоиндукции при коммутации индуктивной нагрузки каждый из восьми выходов подключен к средней точке пары диодов, включенных последовательно между цепями силового питания ИМС; таким образом, выброс напряжения любой полярности извне блока будет выровнен по величине с соответствующим уровнем питающих цепей и не будет представлять угрозы для ИМС.

К каждому из восьми управляющих входов ИМС подключен индикаторный светодиод для отображения текущего состояния выхода.

Плата arduino с микроконтроллером Atmega 2560. В качестве управляющего микроконтроллера был использован МК Atmega2560 (рисунок 24). Это высокопроизводительный (до 16 MIPS) малопотребляющий микроконтроллер с разрядностью 8 бит и RISC - архитектурой. Он имеет два 8-битных и четыре 16-битных таймера-счетчика, которые могут генерировать ШИМ-сигнал на соответствующих выводах микроконтроллера. Также имеются 16 каналов АЦП разрядностью 10 бит, модули TWI и UART, 16 входов внешних прерываний.

Выбор именно Atmega2560 обусловлен тем, что для работы всех устройств необходимо большое количество входов и выходов, что как раз и отличает этот контроллер от, например, Atmega328 и др. подобных. Общее количество GPIO-выводов МК равняется 64, что за вычетом выводов, занятых различными интерфейсами, вполне достаточно для обеспечения управляющими сигналами внешних устройств. Внешний вид гироскопа представлен на рисунке 25.

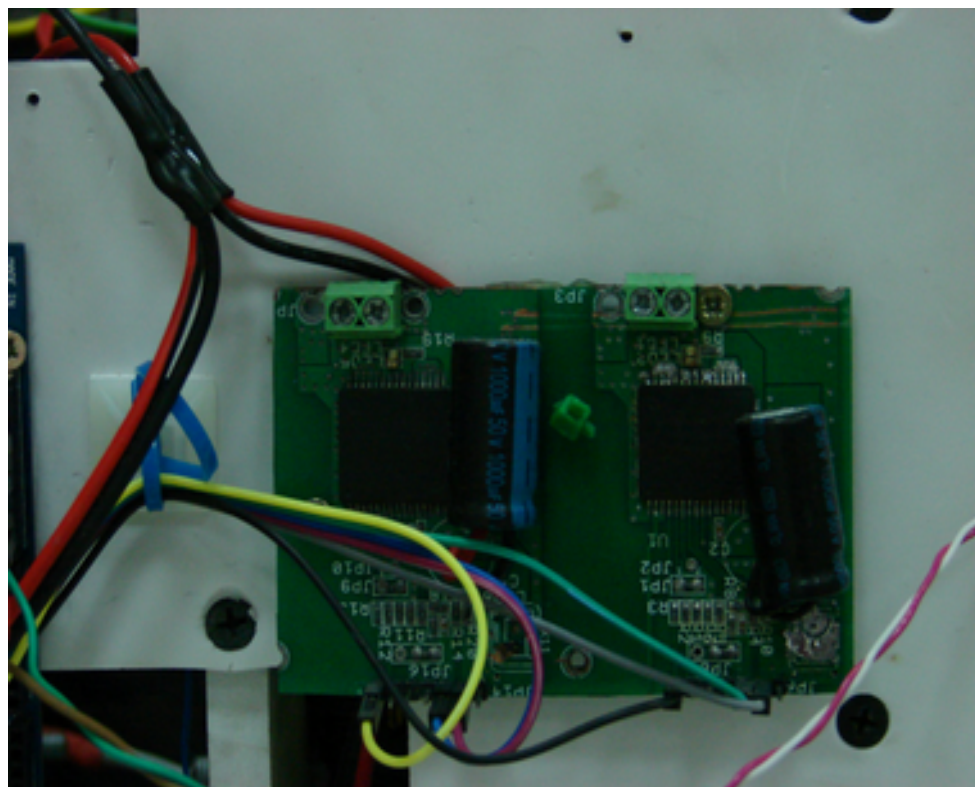


Рисунок 22. Внешний вид VNH5019A

Плата гироскопа с микросхемой MPU5060 удовлетворяет требованиям, предъявляемым к модулю, а именно наличие гироскопа, акселерометра и магнитного компаса в составе единого устройства с интерфейсом I2C.

Блок реле

Блок реле присутствует и представляет из себя плату, на которой размещены 4 реле марки Китай, 4 блока оптической развязки (гальванической), управляющие входы, управляющиеся инверсным сигналом и индикаторные светодиоды, которые показывают состояние реле.

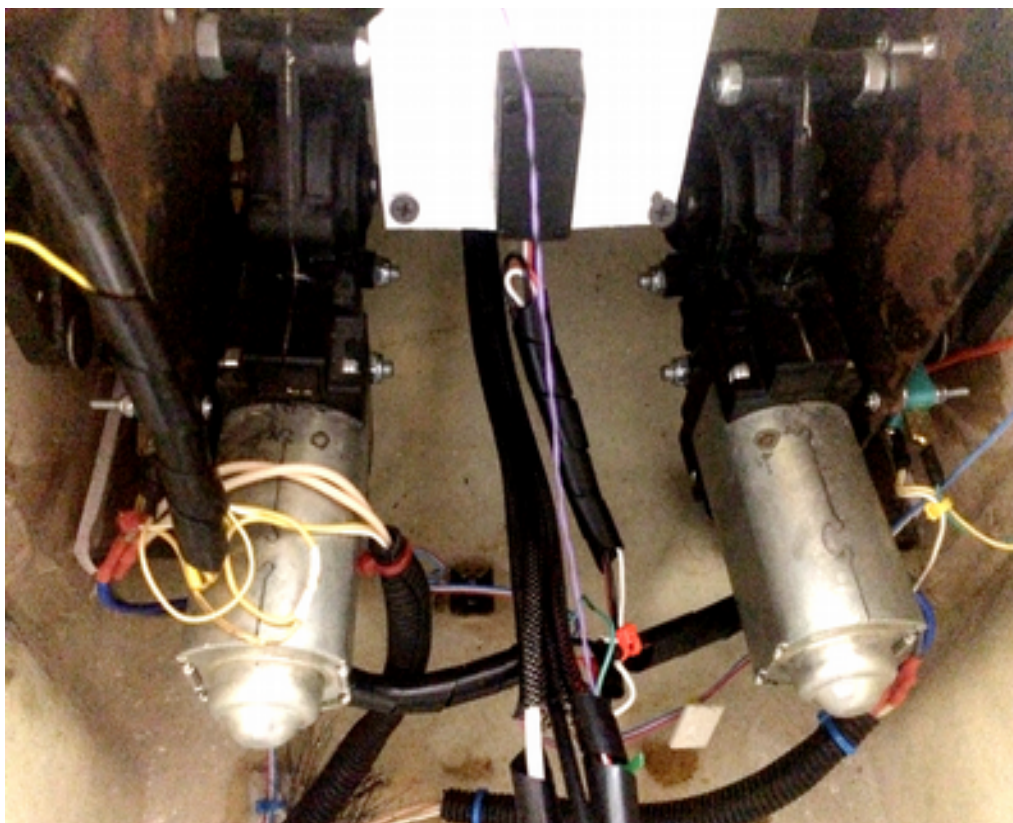


Рисунок 23. Двигатели установленные в робота

Ультразвуковые датчики измерители расстояния

В системе используются ультразвуковые измерители расстояния HC-SR04. Их выбор обусловлен простотой работы с протоколом, который они используют. Преобразователь интерфейсов USART → rs485 → USB

Для обеспечения связи между бортовой ЭВМ и контроллером используется шина rs485, и, соответственно, преобразователь интерфейсов.

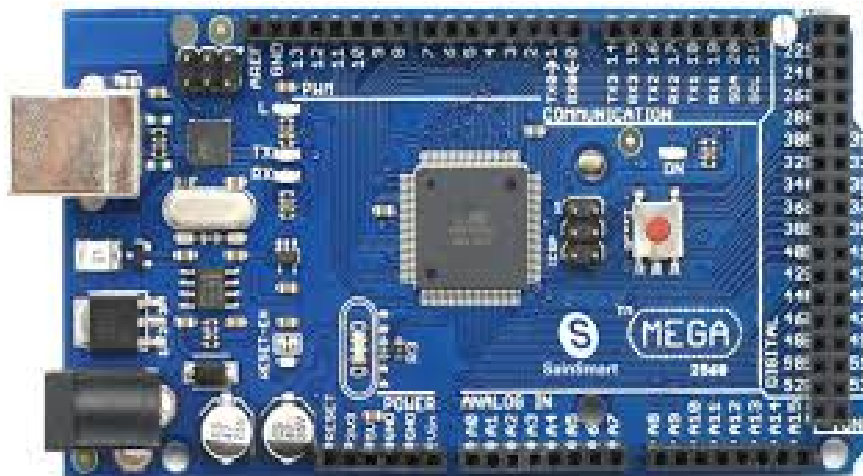


Рисунок 24. Внешний вид МК Atmega2560



Рисунок 25. Внешний вид гироскопа

Он содержит в себе контроллер USB, микросхему MAX485, и микросхему для работы с USART. Его применение обусловлено необходимостью создания единой ЛВС робота, в которую входят все бортовые устройства.

6.3 Разработка ПО бортовой ЭВМ робота

В основе бортового ПО робота, разрабатываемого в рамках данной работы, лежит генерация текстов для произношения. Существуют несколько способов решения этой задачи, в числе которых шаблонные генераторы, и генератор на основе цепей Маркова. В данном случае рассмотрена теория и

создание генератора на базе цепей Маркова, поскольку этот способ является наиболее результативным.

Цепью Маркова называют такую последовательность случайных событий, в которой вероятность каждого события зависит только от состояния, в котором процесс находится в текущий момент и не зависит от более ранних состояний. Конечная дискретная цепь определяется:

Генерацию текста можно провести следующим образом. Мы будем формировать отдельно каждое предложение, выполняя следующие шаги:

- выбираем слово, наиболее вероятное для начала предложения;
- подбираем наиболее вероятное слово-продолжение в зависимости от двух предыдущих слов;
- повторяем предыдущий шаг до тех пор, пока не встретим символ конца предложения.

Реализация программы

Сначала необходимо подготовить текст, с помощью которого будет обучаться модель. Из данного текста выделим необходимую последовательность слов (лист.

Листинг 2. Программа подготовки текста

```
def gen_lines(corpus):  
    data = open(corpus)  
    for line in data:  
        yield line  
  
def gen_tokens(lines):  
    for line in lines:  
        for token in r_alphabet.findall(line):  
            yield token
```

Получившийся генератор tokens будет выдавать последовательность слов и знаков препинания. Однако, простая последовательность не подходит в рамках рассматриваемой задачи. Необходимо найти тройки токенов (под

токеном здесь понимается слово или знак препинания, т.е. некоторые атомарные элементы текста). Для этого существует функция `get_trigrams`, которая возвращает три токена по порядку.

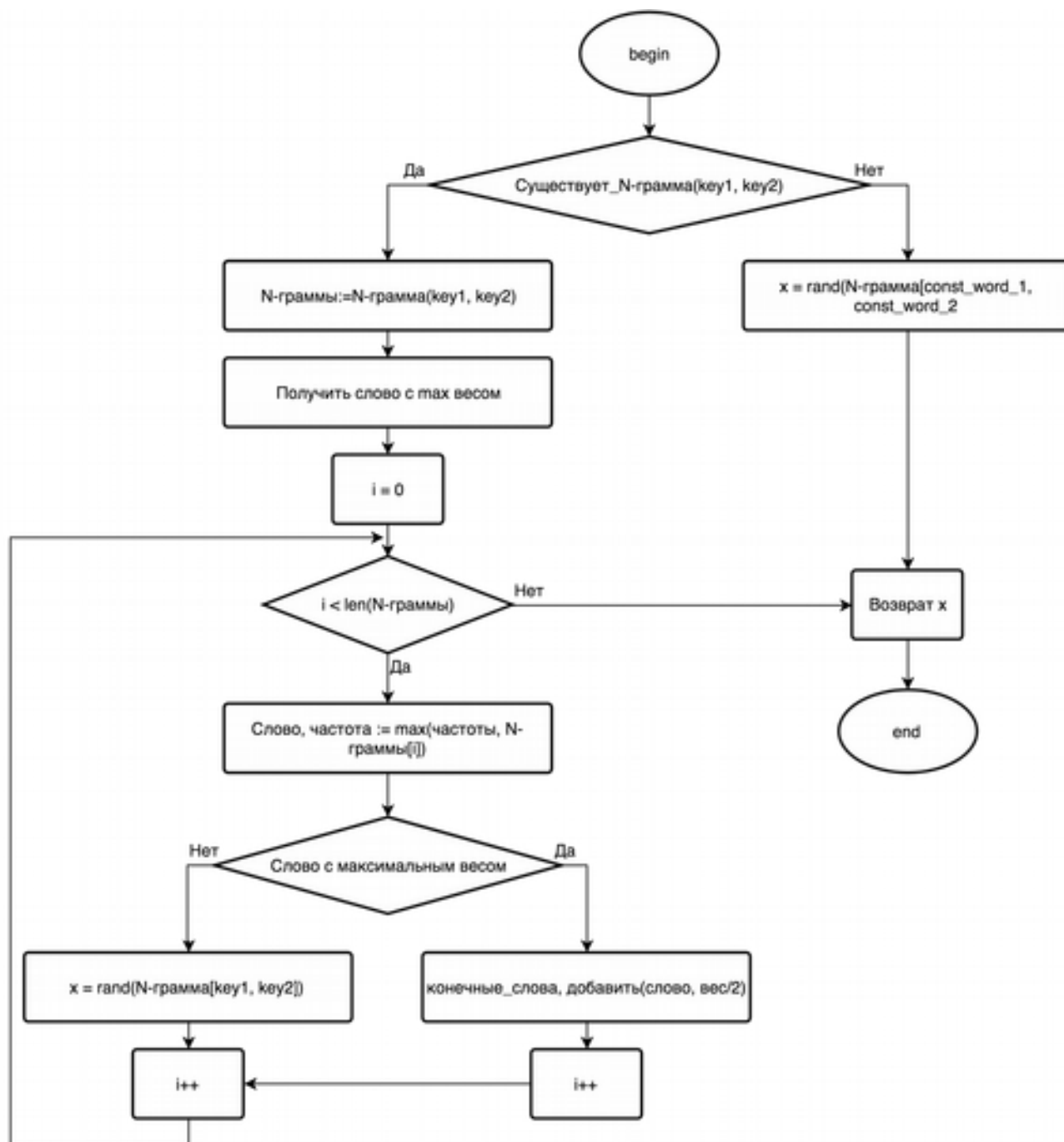


Рис. 26. Блок-схема генератор текста

Листинг 3. Генерация N-грамм

```
def gen_trigrams(tokens):
    t0, t1 = '$', '$'
    for t2 in tokens:
        yield t0, t1, t2
        if t2 in '!.?':
            yield t1, t2, '$'
            yield t2, '$', '$'
            t0, t1 = '$', '$'
        else:
            t0, t1 = t1, t2

lines = gen_lines('corpus.txt')
tokens = gen_tokens(lines)
trigrams = gen_trigrams(tokens)
```

Метод `gen_trigrams` требует пояснения. Символы '\$' используются для маркировки начала предложения. В дальнейшем, это делает проще выбор первого слова генерируемой фразы. В целом, метод действует следующим образом: он возвращает три подряд идущих токена, на каждой итерации сдвигаясь на один токен:

Листинг 4. Входы и выходы программы

На входе:
'Счастье есть удовольствие без раскаяния'

На выходе:
итерация токены
0: \$ \$ счастье
1: \$ счастье есть
2: счастье есть удовольствие
3: есть удовольствие без

Далее, рассчитаем триграммную модель:

В первой части этого метода мы задаем генераторы. Далее, рассчитываем биграммы и триграммы (фактически, мы считаем количество одинаковых пар и троек слов в тексте). Далее, вычисляем вероятность слова в зависимости от двух предыдущих, помещая данное слово и его вероятность в словарь.

Листинг 5. Обучение модели

```
def train(corpus):
    lines = gen_lines(corpus)
    tokens = gen_tokens(lines)
    trigrams = gen_trigrams(tokens)

    bi, tri = defaultdict(lambda: 0.0), defaultdict(lambda: 0.0)

    for t0, t1, t2 in trigrams:
        bi[t0, t1] += 1
        tri[t0, t1, t2] += 1

    model = {}
    for (t0, t1, t2), freq in tri.iteritems():
        if (t0, t1) in model:
            model[t0, t1].append((t2, freq/bi[t0, t1]))
        else:
            model[t0, t1] = [(t2, freq/bi[t0, t1])]
    return model

model = train('model.txt')
```

Необходимо учесть, что текст должен быть сгенерирован на заданную тему. Для этого функция `uni_not_rand` добавляет из модели слова по следующему правилу:

- 1) Попытаться получить из модели слова, которые могут следовать за словами, которые идут за токенами `key1` и `key2`.

- 2) Если это не получится, выполнить функцию `unirand`, описанную выше. В этом случае в предложение добавится слово, наиболее вероятное в данном контексте.

- 3) Если на шаге 1 удастся получить необходимые слова, то выбираем слово с максимальным весом из тех, которые надо вывести.

- 4) Для каждого `w` и `freq` (слово и его частота в корпусе) проверяем, не совпадает ли оно с требуемым.

- 5) Если совпадение найдено, то уменьшаем вес выведенного слова в 2 раза.

- 6) После этого возвращаем слово, которое было получено на предыдущем шаге.

Листинг 6. Генерация «неслучайного» слова

```
def uni_not_rand(words, model, key1, key2):
    try:
        seq = model[key1, key2]
        max_fig = max(words.items(), key=itemgetter(1))[0]
        for w, freq in seq:
            if w == max_fig:
                words.update({max_fig: words[max_fig]/2})
                return w
        return unirand(model[key1, key2])
    except KeyError:
        return unirand(model["для", "этого"])
```

Функция, которая возвращает предложение описана ниже.

Листинг 7. Генерация предложения

```
def generate_sentence(words, model):
    phrase = []
    t0, t1 = '$', '$'
    while 1:
        t0, t1 = t1, uni_not_rand(words=words, model=model, key1=t0, key2=t1)
        if t1 == '$': break
        phrase += [t1]
    return phrase
```

Суть метода в том, что последовательно выбираем наиболее вероятные слова или знаки препинания до тех пор, пока не встречаем признак начала следующей фразы (символа \$). Первое слово выбирается как наиболее вероятное для начала предложения из набора model['\$', '\$'].

Здесь необходимо отметить важный момент. Словарь model для каждой пары слов содержит список пар (слово, вероятность). Нам же необходимо выбрать из этого набора только одно слово. Вариант «в лоб» — выбрать слово с максимальной вероятностью. Но тогда все сгенерированные фразы были бы похожи друг на друга. Более подходящий способ — выбирать слова с некой хаотичностью, которая бы зависела от вероятности слова (мы же не хотим чтобы наши фразы состояли из редких сочетаний). Это и делает метод unirand, который возвращает случайное слово с вероятностью, равной вероятности данного слова в зависимости от двух предыдущих.

После генерации каждого нового предложения выполняется сравнение слов, использованных в нем со словарем слов с вычисленными весами, для того, чтобы убедиться в соответствии темы предложения заданной.

Листинг 8. Код генерации текста

```
words_of_words_initial = [ x for x in open('hands_desc.txt').readlines() ]
words_of_words = []
for w in words_of_words_initial:
    words_with_weight = {}
    wSplitted = [ x for x in re.split('[:, ]', w) if x ]
    print(wSplitted)
    for w1 in wSplitted:
        words_with_weight.update({w1.strip(): 1})
    words_of_words.append(words_with_weight)

for words in words_of_words:
    sent = generate_sentence(words, model)
    cnt = 0
    while compare_sentence(sent, list(words.keys())) < 0.5 and cnt < 10:
        sent = generate_sentence(words, model)
        cnt += 1

    print(sent, words)
```

Пример текста, порождаемого программой приведен ниже.

Если автомат в момент 0 находится в состоянии "инициализация", а входное воздействие равно "0", то следующее состояние будет равно "установка в начальное положение". При пребывании автомата в состоянии "установка в начальное положение" в момент времени 1 имея на входе воздействие "требуемый угол достигнут", автомат сменит состояние на "начальное положение установлено". Если автомат в момент 2 находится в состоянии "начальное положение установлено", а входное воздействие равно "угол меньше требуемый", то следующее состояние будет равно "поворот на X градусов в горизонтальной плоскости". При пребывании автомата в состоянии "начальное положение установлено" в момент времени 3 имея на входе воздействие "угол меньше требуемый", автомат сменит состояние на "поворот на X градусов в вертикальной плоскости". Если автомат в момент 4 находится в состоянии "поворот на X градусов в горизонтальной плоскости", а

входное воздействие равно "требуемый угол достигнут", то следующее состояние будет равно "поворот завершен".

При пребывании автомата в состоянии "поворот на X градусов в вертикальной плоскости" в момент времени 5 имея на входе воздействие "требуемый угол достигнут", автомат сменит состояние на "поворот завершен". В момент 6 автомат пребывает в состоянии "поворот на X градусов в вертикальной плоскости", и, получив воздействие "требуемый угол достигнут", он перейдет в состояние "поворот завершен". Если автомат в момент 7 находится в состоянии "поворот на X градусов в вертикальной плоскости", а входное воздействие равно "требуемый угол достигнут", то следующее состояние будет равно "поворот завершен".

Внимательный читатель заметит, что часть текста пояснительной записки сгенерирована при помощи именно этой модели.

6.4 Разработка системы управления движением

Для получения более точных данных о параметрах движения, при включении робота выполняется калибровка гироскопа (листинг 2).

Листинг 9. Пример кода калибровки гироскопа

```
S = 0;
old_z = гироскоп(z);
for (i=0; i 10; i++)
{
    задержка 0.3 сек;
    new_z = гироскоп(z);
    S = S + (new_z - old_z);
}
дрейф = S / 10;
```

Согласно описанному в конструкторской части ПИД регулятору, был написан код регулирования курса, траектории и скорости.

Список литературы

1. Кондратьев К.Л., проф. Харитонов В.И. Аппаратно-программный комплекс для управления платформой-роботом // Известия МГТУ «МАМИ». - 2014. - №2(20).
2. Михед А.Д., Смирнова Е.Е., Липатова Ю.В. Моделирование манипулятора кисти андроида // Известия ТулГУ. Технические науки.. - 2016. - №11. Ч. 2.
3. Колыванов К.Ю. Перспективы мобильной робототехники // Северо-Казахстанский государственный университет им. М. Козыбаева. - 2016.
4. Шилов М.М., Воробьева О.С., Кракау Т.К. Применение систем технического зрения и средств нечётких вычислений в интеллектуальных устройствах автоматики и робототехники // Научно-технические ведомости СПбГТУ. - 2010. - №2.
5. Саяпин А.В., Зотин А.Г. Система управления движением автономной колесной платформы // Решетневские чтения. - 2015.
6. Motor Shield 2A L298P 2-motors for Arduino // URL: <http://arduino-kit.ru/catalog/id/motor-shield-2a-l298p-2-motors-for-arduino> (дата обращения: 21.04.2018).
7. Инерциальная навигационная система // URL: www.nngasu.ru/geodesy/classification/chastnye-klassifikatsii/11_Inercialnie_sistemi.php (дата обращения: 10.07.2017).
8. Описание библиотеки и примеры работы с платой GY-85 // URL: <https://github.com/madc/GY-85> (дата обращения: 17.09.2017).
9. Описание интерфейсной шины I2C протокола передачи данных по ней // URL: <http://easyelectronics.ru/interface-bus-iic-i2c.html> (дата обращения: 28.10.2017).
10. Анишин Н.С., Булатникова И.Н., Гершунина Н.Н. Алгоритмические проблемы применения микропроцессорных средств в робототехнике // Известия

ТРТУ Тематический выпуск Компьютерные технологии в инженерной и управленческой деятельности. - 2008. - №3.

11. AS3103PG, аналоговый сервопривод [360 градусов] // URL: <http://roboarts.ru/products/as310pg> (дата обращения: 18.04.2018).

12. Дисплеи планшетные skylarpu из Китая // URL: <https://sellercheck.ru/c/202005759/b/4051> (дата обращения: 11.05,2018).

13. Евстифеев А.В. Микроконтроллеры AVR семейств Tiny и Mega фирмы Atmel. - М.: Додэка, 2008.

14. Разработка программы управляющей платы для работы с магнитными энкодерами колес роботизированного комплекса / Е. В. Долгова, С. А. Злобин, Е. В. Головкова, Д. В. Кондаков, Д. С. Курушин // В мире научных открытий. - 2015. - № 10.2(70) : Естественные и технические науки.