

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.03.01 – Информатика и вычислительная техника
Направленность (профиль) образовательной программы: Автоматизированные
системы обработки информации и управления

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов
« _____ » _____ 2017 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка программно-аппаратного комплекса для управления движением модели движущейся платформы посредством сети Wi-Fi

Исполнитель

студент группы 353 об

(подпись, дата)

Ю.А. Сысолятин

Руководитель

доцент, канд. техн. наук

(подпись, дата)

Т.А. Галаган

Консультант

по безопасности и
экологичности

доцент, канд. техн. наук

(подпись, дата)

А.Б. Булгаков

Нормоконтроль

инженер кафедры

(подпись, дата)

В.В. Романико

Благовещенск 2017

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.В. Бушманов

«_____» _____ 2017 г.

З А Д А Н И Е

К бакалаврской работе студента Сысолятина Юрия Альбертовича.

1. Тема бакалаврской работы: Разработка программно-аппаратного комплекса для управления движением модели движущейся платформы посредством сети Wi-Fi

(утверждено приказом от 26.05.2017 № 1189-уч)

2. Срок сдачи студентом законченной работы _____

3. Исходные данные к бакалаврской работе: отчет о прохождении преддипломной практики, техническое задание.

4. Содержание бакалаврской работы: анализ предметной области, проектирование программно-аппаратного комплекса, реализация программно-аппаратного комплекса.

5. Перечень материалов приложения: техническое задание, функциональная схема модулей системы и их взаимодействия, функциональная схема взаимодействия модулей, функциональная схема взаимодействия реализованных классов, диаграммы классов, экранные формы, руководство пользователя.

6. Консультанты по бакалаврской работе:
по безопасности и экологичности – А.Б. Булгаков, доцент, канд. техн. наук.

7. Дата выдачи задания: 20.02.2017.

Руководитель бакалаврской работы: Галаган Татьяна Алексеевна, доцент, канд. техн. наук.

Задание принял к исполнению _____ Ю.А. Сысолятин

РЕФЕРАТ

Бакалаврская работа содержит 97 с., 43 рисунка, 7 приложений, 23 источника.

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС, АППАРАТНАЯ ЧАСТЬ, ARDUINO, ESP8266, WI-FI, ПРОГРАММНАЯ ЧАСТЬ, МОДУЛЬ, ПЛАТФОРМА, АВТОМАТИЗИРОВАННАЯ СИСТЕМА, МЕТОД, КЛАСС, ФУНКЦИИ, АЛГОРИТМЫ, ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА, РЕАЛИЗАЦИЯ, ДВИЖЕНИЕ

Цель работы – разработка универсального комплекса для управления движением модели движущейся платформы, представляющего собой программный продукт и аппаратную часть, реализующие функциональные возможности, представленные в техническом задании. Взаимодействие программной и аппаратной компонент осуществляется посредством сети Wi-Fi.

Объект исследования – робототехнические автоматизированные системы, выполненные на базе аппаратно-вычислительной платформы Arduino.

Результатом работы является разработанный программно-аппаратный комплекс для управления движением модели движущейся платформы.

					<i>ВКР.135149.09.03.01 ПЗ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>				
<i>Разраб.</i>	<i>Сысолятин Ю.А.</i>				РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ УПРАВЛЕНИЯ ДВИЖЕНИЕМ МОДЕЛИ ДВИЖУЩЕЙСЯ ПЛАТФОРМЫ ПОСРЕДСТВОМ СЕТИ WI-FI	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>	<i>Галаган Т.А.</i>					У	3	112
<i>Консульт.</i>	<i>Булгаков А.Б.</i>					АмГУ кафедра ИУС		
<i>Н. контр.</i>	<i>Романико В.В.</i>							
<i>Утверд.</i>	<i>Бушманов А.В.</i>							

СОДЕРЖАНИЕ

Введение	8
1 Описание предметной области	11
1.1 Роботы и робототехнические системы	11
1.2 Мобильные роботизированные системы	14
1.3 Транспортные роботизированные системы	14
1.4 Анализ имеющихся программно-аппаратных комплексов для управления движением платформы-робота	14
2 Требования к проекту	17
2.1 Анализ требований к программно-аппаратному комплексу управления движением платформы	17
2.1.1 Анализ требований технического задания	17
2.1.2 Постановка цели работы и назначение системы	18
2.2 Описание объекта автоматизации	19
2.3 Постановка задач системы	20
2.4 Определение функций системы	20
3.1 Проектирование аппаратной части комплекса	22
3.1.1 Выбор компонентов аппаратной части	23
3.1.2 Блок питания	36
3.1.3 Драйвер двигателя	36
3.1.4 Датчик линии	37
3.1.5 Обвязка Wi-Fi модуля ESP8266-12E	39
3.1.6 Схема подключения компонентов к Arduino Due	40
3.2 Взаимодействие программной части с аппаратной частью комплекса	42
3.3 Проектирование программного обеспечения для компьютера на языке UML	43
3.3.1 Диаграмма вариантов использования системы	43
3.3.2 Диаграмма состояний системы	44
3.3.3 Диаграмма последовательностей системы	44

3.3.4	Проектирование алгоритмов основных функций системы	47
3.3.5	Проектирование функциональных модулей программного приложения	50
3.3.6	Проектирование взаимодействия функциональных модулей программного приложения	53
3.3.7	Проектирование интерфейса программного приложения	54
3.4	Проектирование программного обеспечения для микроконтроллера аппаратной части комплекса	55
3.4.1	Алгоритм выполнения функции движения по нарисованной линии	57
3.4.2	Алгоритм выполнения функции записи траектории движения	58
3.4.3	Алгоритм выполнения функции движения по сохраненной траектории	59
3.4.4	Алгоритм выполнения функции движения в ручном режиме управления	60
3.5	Характеристика обеспечивающих подсистем комплекса	61
3.5.1	Подсистема «Организационное обеспечение»	61
3.5.2	Подсистема «Информационное обеспечение»	61
3.5.3	Подсистема «Программное обеспечение»	62
3.5.4	Подсистема «Техническое обеспечение»	62
4	Разработка программно-аппаратного комплекса	64
4.1	Разработка аппаратной части комплекса	64
4.2	Обоснование выбора средств разработки	64
4.2.1	Выбор языка программирования	64
4.2.2	Выбор сред разработки	65
4.3	Разработка программного приложения	66
4.3.1	Реализация связи программного приложения и платформы через сеть	66
4.3.2	Реализация подмодуля управления платформой	68
4.3.3	Реализация работы с пакетами передаваемых данных	71
4.3.4	Реализация графического интерфейса программного приложения	75

4.4	Разработка программной прошивки для микроконтроллера аппаратной части комплекса	78
4.4.1	Реализация алгоритма функции ручного управления	80
4.4.2	Реализация алгоритмов функций автоматического управления	81
4.5	Разработка руководства пользователя	82
4.5.1	Обоснование необходимости разработки руководства пользователя	82
4.5.2	Определение содержания руководства пользователя	82
4.5.3	Разработка руководства пользователя	83
5	Безопасность и экологичность при эксплуатации программно-аппаратного комплекса для управления движением модели движущейся платформы посредством сети Wi-Fi	84
5.1	Безопасность	84
5.1.1	Общие требования безопасности для операторов	84
5.1.2	Требования к помещениям	85
5.1.3	Требования к микроклимату	86
5.1.5	Требования к освещению	86
5.1.6	Требования к организации и оборудованию рабочих мест	87
5.1.7	Пожарная и электробезопасность на рабочем месте	89
5.1.8	Эргономика программного интерфейса	90
5.1.9	Обеспечение безопасности работы системы	91
5.2	Экологичность	91
5.3	Чрезвычайные ситуации	92
5.3.1	Аварийные ситуации	92
5.3.2	Меры пожарной безопасности на рабочих местах	92
	Заключение	94
	Библиографический список	96
	Приложение А Техническое задание	98
	Приложение Б Функциональная схема модулей системы и их взаимодействия	104
	Приложение В Функциональная схема взаимодействия модулей в программном приложении комплекса	105

Приложение Г Функциональная схема взаимодействия реализованных классов в программном приложении комплекса	106
Приложение Д Экранные формы программного приложения	107
Приложение Е Руководство пользователя	109
Приложение Ж Справка о внедрении комплекса на предприятие	112

						<i>Лист</i>
						7
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>ВКР.135149.09.03.01.ПЗ</i>	

ВВЕДЕНИЕ

Наука не стоит на месте. Еще совсем недавно мысль о том, что машина может заменить человека, была всего лишь мечтой. Но уже сейчас развитие робототехники привело к тому, что роботы и автоматизированные системы заняли свою нишу в жизни нашего общества. Они стали частью современной промышленной революции, характеризующейся широким внедрением адаптивных технологий и роботизацией производства. Каждый год все больше и больше предприятий автоматизируются. Законом, на котором трудятся лишь несколько десятков человек, а всю основную работу выполняют машины, уже никого не удивить.

На сегодняшний день все робототехнические решения моделей передвигающихся роботов разрабатываются узкоспециализировано под конкретные задачи. Такие системы поставляются вместе с шасси и управляющей аппаратной частью уже подобранной под имеющееся на данном шасси силовое оборудование (двигатели, сервоприводы). Следовательно, нет универсальных ПАК для автоматизации любой передвигающейся платформы (робота). А заказ такой автоматизированной системы под поставленные задачи потребует больших капитальных вложений.

Именно поэтому основной целью данной работы является разработка малобюджетного программно-аппаратного комплекса для управления движением некой платформы, который не зависит от конкретной конструкции шасси и имеющихся на нем силовых агрегатов для передвижения. Данная система включает в себя аппаратное обеспечение, устанавливаемое на платформу и управляющая двигательными механизмами, а также программное обеспечение для управления аппаратной частью.

В связи с указанной целью были поставлены следующие задачи выпускной квалификационной работы:

– исследовать и проанализировать предметную область, связанную с роботизированными системами;

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		8

- провести проектирование программно-аппаратного комплекса;
- разработать программно-аппаратный комплекс;
- разработать руководство пользователя;
- проанализировать инструкции по охране труда и техники безопасности.

Актуальность поставленной задачи заключается в расширении и развитии данного направления автоматизации. На данный момент подобных решений, обладающих универсальностью по отношению к установке их на конкретную модель платформы, не существует.

Для разрабатываемого программно-аппаратного комплекса характерна низкая стоимость, а также возможность расширения и совершенствования под определенные задачи. Например, данную систему можно использовать для транспортировки грузов, в охранных целях и для патрулирования территории, если установить на платформу средства сигнализации и видеонаблюдения.

При написании выпускной квалификационной работы были использованы следующие методы исследования:

- анализа нормативно-правовой документации по теме выпускной квалификационной работы;
- сравнение.

Структура работы была сформирована исходя из темы и поставленной цели. Введение содержит в себе раскрытие актуальности темы, краткое определение объекта и предмета исследования, цель и задачи, поставленные в выпускной квалификационной работе, практическую и теоретическую значимость выпускной квалификационной работы.

В первой главе описывается анализ предметной области и имеющихся аналогов разрабатываемого комплекса.

Во второй главе предъявляются необходимые требования к проекту, основанием для которых служит техническое задание к выпускной квалификационной работе.

В третьей главе проводится проектирование функциональных модулей аппаратной и программной частей комплекса, их взаимодействия.

В четвертой главе описано обоснование выбора языка программирования, среды разработки и технологий, применяемых при разработке, также описана разработка функциональных модулей и руководства пользователя программного продукта, программной прошивки для микроконтроллера аппаратной части и самой аппаратной части.

В пятой главе описывается анализ безопасности жизнедеятельности и возможных чрезвычайных ситуаций при эксплуатации программно-аппаратного комплекса, исследование эргономичности разработанного программного продукта.

В заключении подводятся итоги произведенной работы и формируется окончательный вывод по выбранной для рассмотрения теме.

Апробация работы. По теме исследования опубликованы результаты в статье «Проектирование программно-аппаратного комплекса для управления моделью движущейся платформы с помощью сети Wi-Fi» ежемесячного электронного научного журнала «Постулат» (выпуск №12, 2016 год) [1].

<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>

ВКР.135149.09.03.01.ПЗ

Лист

10

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Роботы и робототехнические системы

Робот (чеш. robot, от robota – «подневольный труд») – автоматическое устройство, созданное по принципу живого организма, предназначенное для осуществления производственных и других операций, которое действует по заранее заложенной программе и получает информацию о внешнем мире от датчиков (аналогов органов чувств живых организмов). Робот может как иметь связь с оператором (получать от него команды), так и действовать автономно.

Робототехника, мехатроника и робототехнические системы – область науки и техники, ориентированная на создание роботов, мехатронных и робототехнических систем, предназначенных для автоматизации сложных технологических процессов и операций, в том числе, выполняемых в недетерминированных условиях, для замены человека при выполнении тяжелых, утомительных и опасных работ. «Мехатроника» как отдельная область науки и техники, основана на синергетическом объединении узлов точной механики с электронными, электротехническими и компьютерными компонентами, обеспечивающими проектирование и производство качественно новых модулей, систем и машин с интеллектуальным управлением их функциональными движениями.

Существует большое множество областей применения роботов: промышленность, медицина, бытовая область, военная область, обучение, развлекательная и досуговая области и т.д. В настоящее время основной отраслью применения роботов является промышленность. Их внешний вид по причинам технического и экономического характера далек от внешнего вида живых организмов. Они занимают довольно большие площади, имеют мощные агрегаты в своей конструкции, и их трудно программировать.

Прогресс технологий позволил уменьшить и облегчить компоненты, при этом не утратив их основных свойств и характеристик. Также появились мощные и очень компактные микроконтроллеры, на которых основаны программируе-

мые платы. Использование таких плат упрощает и ускоряет процесс проектирования и быстрого макетирования. К ним можно подключать различные датчики и исполняющие устройства, без особых проблем решая поставленные задачи. Облегчается и процесс программирования подобных устройств: появилась возможность писать программы и прошивки для микроконтроллеров не на специфических языках или машинных кодах, а на языках более высокого уровня, удобных и доступных для понимания даже начинающему.

В основе любого робота или роботизированного устройства лежат механическая, электрическая и программная части. Разработка проекта начинается с проектирования и реализации механической части. Далее для созданной конструкции подбираются необходимые электронные компоненты. Исходя из решаемых задач выбираются нужной мощности и конфигурации исполняющие устройства (моторы, сервомоторы и т.д.) и датчики. Для управления электроникой используется контролирующее устройство (микроконтроллер), которое удовлетворяет необходимой вычислительной мощности. Также немало внимания уделяется энергопотреблению компонентов системы, что в конечном итоге сказывается на автономности и мобильности оной. После окончательной сборки конструкции начинается разработка программной прошивки для выбранного контроллера, ее тестирование и отладка.

Для осуществления движения роботу требуется управление. С этим понятием связано решение следующего комплекса задач: адаптация робота к кругу решаемых им задач, программирование движений, синтез системы управления и её программное обеспечение. По типу управления робототехнические системы подразделяются на:

– биотехнические:

- 1) командные (кнопочное и рычажное управление отдельными звеньями робота);
- 2) копирующие (повтор движения человека, возможна реализация обратной связи, передающей прилагаемое усилие, экзоскелеты);

3) полуавтоматические (управление одним командным органом, например, рукояткой – всей кинематической схемой робота);

– автоматические:

1) программные (функционируют по заранее заданной программе, в основном предназначены для решения однообразных задач в неизменных условиях окружения);

2) адаптивные (решают типовые задачи, но адаптируются под условия функционирования);

3) интеллектуальные (наиболее развитые автоматические системы);

– интерактивные:

1) автоматизированные (возможно чередование автоматических и биотехнических режимов);

2) супервизорные (автоматические системы, в которых человек выполняет только целеуказательные функции);

3) диалоговые (робот участвует в диалоге с человеком по выбору стратегии поведения, при этом, как правило, робот оснащается экспертной системой, способной прогнозировать результаты манипуляций и дающей советы по выбору цели).

Среди основных задач управления роботами выделяют следующие:

– планирование положений;

– планирование движений;

– планирование сил и моментов;

– анализ динамической точности;

– идентификация кинематических и динамических характеристик робота.

Основными классами роботов широкого назначения являются манипуляционные и мобильные роботы. По своему строению роботы делятся на следующие основные типы: промышленный робот, транспортный робот, социальный робот, бытовой робот, зооробот и др.

Разрабатываемую в данной работе платформу можно отнести к мобильным транспортировочным роботам.

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		13

1.2 Мобильные роботизированные системы

Мобильный робот – автоматическая машина, в которой имеется движущееся шасси с автоматически управляемыми приводами, позволяющая ей передвигаться в пространстве. Есть три больших класса мобильных роботов: наземные роботы, воздушные, морские. Разнообразие морских несколько меньше, чем у остальных классов: они бывают подводные (глубоководные погружаемые автоматы. всевозможные военные роботы-саперы и др.) и наводные (катера: радиоуправляемые либо с автономным управлением). К роботам, использующим воздушное перемещение можно отнести беспилотные летательные аппараты (БПЛА, крылатые ракеты), самолеты с автопилотным управлением и другие. Наземные же роботы делятся на колёсных, шагающих, гусеничных, ползающих, а также использующих в качестве опоры шар (шароботы).

Наиболее распространенными типами роботов наземного класса являются четырёхколёсные и гусеничные. Для упрощения конструкции и повышения маневренности робота создаются двух- и одноколёсные шасси. Существует некоторое количество прототипов сферических роботов, которые по маневренности несомненно превосходят все другие типы наземных роботов.

В настоящее время перспективным типом является шагающие роботы. Они имеют свои преимущества перед колесными, потому что для того, чтобы колесо прокручивалось, необходима некоторая дорога, а шагающие роботы могут преодолевать всевозможные препятствия, например, подниматься по лестнице или двигаться по пересеченной местности. Поэтому развитие шагающих роботов представляет также большой интерес.

Но для создания платформы в данной работе было решено выбрать тип колесных наземных роботов, так как они самые простые в проектировании, разработке и эксплуатации, а также требуют меньшей затраты средств.

1.3 Транспортные роботизированные системы

Транспортный робот – определение, идеально подходящее для описания типа разрабатываемой платформы и системы ее управления.

Транспортный – робот автоматическая машина, представляющая собой

совокупность перепрограммируемого устройства управления и ходового устройства. Также она может быть оборудована манипулятором для выполнения различных действий с грузами.

Транспортные роботы являются подклассом промышленных роботов, которые получили широкое распространение в качестве транспортно-загрузочных устройств в автоматизированном производстве. Характерными признаками таких роботов являются: автоматическое управление; способность к быстрому и относительно легкому перепрограммированию (изменению последовательности, системы и содержания команд); способность к выполнению трудовых действий.

Промышленные транспортные роботы зарекомендовали себя как гибкие автоматизированные средства реализации внутрицеховых и межоперационных материальных связей. У них имеется целый ряд преимуществ по сравнению с другими устройствами: малые габаритные размеры подвижного органа; большой диапазон регулирования скорости перемещения; автоматические перемещения; полное высвобождение проездов после прохождения транспортного робота для транспорта других видов; автономность.

Транспортные роботы выступают альтернативой рельсовому транспорту и конвейерным линиям, отличаясь при этом большей маневренностью и большими возможностями по определению маршрута движения.

Можно выделить два класса транспортных роботов: напольные и подвесные. Напольные роботы не требуют никаких крепежных и направляющих конструкций, в отличие от подвесных. Но при этом обязательным условием их установки являются чистые и ровные полы.

1.4 Анализ имеющихся программно-аппаратных комплексов для управления движением платформы-робота

В настоящий момент на рынке готовых программно-аппаратных комплексов для решения задачи управления некой движущейся платформой очень мало. Все подобные системы выпускаются интегрированными в шасси, реализованное под выполнение определенных задач, поэтому об универсальности их применения речи не идет.

Однако, под авторством Кондратьева К.Л. и профессора Харитонов В.И. из МГТУ «МАМИ» имеется проект подобной системы под названием «Аппаратно-программный комплекс для управления платформой-роботом» [2]. Представленная в нем система направлена на универсальное применение к любому виду шасси, имеет режим управления оператором и автономный режим, но при этом обладает избыточным аппаратным наполнением (датчик Холла, применение сразу нескольких видов датчиков-дальномеров: ультразвуковой и инфракрасный, LCD-дисплей и светодиодная индикация), что неизбежно приводит к увеличению ее стоимости. Также в представленном авторами проекте ручное управление оператора происходит с помощью жестов, что является достаточно интересной и принципиально новой идеей, но неудобной для решения задач, поставленных для системы, разрабатываемой в данной работе.

						<i>Лист</i>
						16
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>ВКР.135149.09.03.01.ПЗ</i>	

2 ТРЕБОВАНИЯ К ПРОЕКТУ

2.1 Анализ требований к программно-аппаратному комплексу управления движением платформы

Прежде чем приступить к проектированию комплекса следует проанализировать поставленные в техническом задании требования (приложение А).

2.1.1 Анализ требований технического задания

Основанием для проведения проектирования и разработки программно-аппаратного комплекса управления движением платформы является техническое задание, представленное в приложении Б.

Основные предъявляемые требования к ПАК управления движением платформы:

– требования к структуре и функционированию, определяющие состав модулей программной и аппаратной частей комплекса;

– требования к надежности, определяющие необходимость написания руководства по правильной эксплуатации системы;

– требования к защите от несанкционированного доступа, определяющие необходимость реализации механизма авторизации пользователей;

– требования к эргономике и технической эстетике, определяющие внешний вид интерфейса и рекомендации по оформлению внешнего вида программного приложения для ПЭВМ;

– требования к численности персонала, описывающие состав персонала, необходимого для обеспечения эксплуатации системы.

Результатом анализа требований технического задания является структурная схема программного приложения для ПЭВМ, представленная на рисунке 1, и структурная схема аппаратной части комплекса, представленная на рисунке 2.



Рисунок 1 – Структурная схема программного приложения комплекса

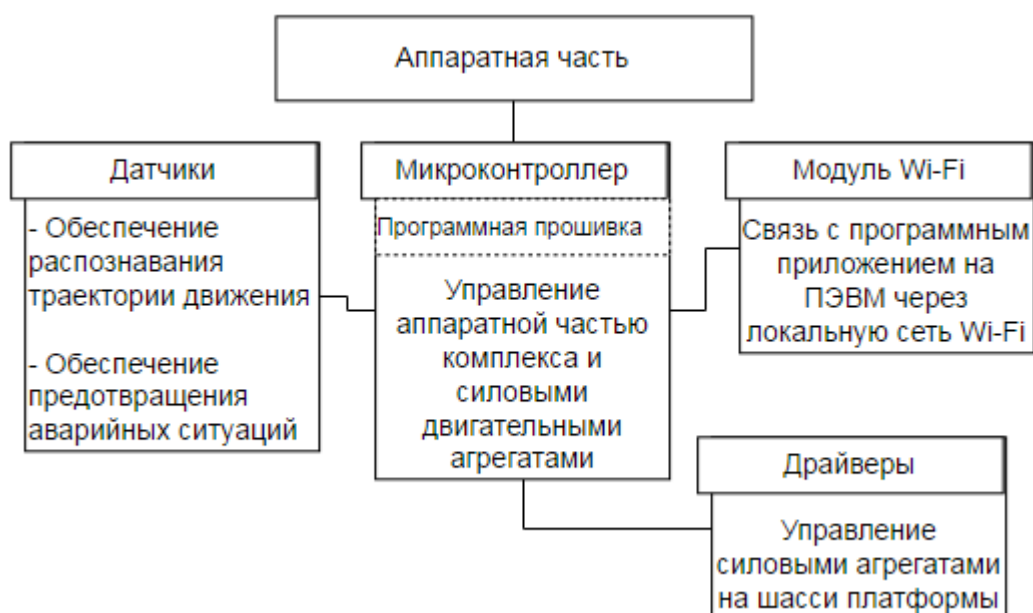


Рисунок 2 – Структурная схема аппаратной части комплекса

2.1.2 Постановка цели работы и назначение системы

Целью данной работы является разработка программно-аппаратного комплекса для автоматизированного управления движением некой движущейся платформы.

Разрабатываемый программно-аппаратный комплекс предназначен для реализации автоматического и ручного управления движением движущейся плат-

формы посредством сети Wi-Fi. В дальнейшем ее можно использовать для автоматизированной транспортировки грузов в пределах складских помещений. Также возможно использование данного комплекса в охранных целях, при установке на платформу специализированных устройств видеонаблюдения, слежения и сигнализации.

2.2 Описание объекта автоматизации

Объектом автоматизации является простейшая платформа, на которую можно установить разрабатываемый управляющий комплекс, оснащенная всего лишь электродвигателем с приводом к ведущему колесу и поворотным механизмом, обеспечивающим достаточную точность управления.

Для демонстрации работы системы было решено разработать демонстрационную модель транспортировочной платформы. В качестве применяемого конструкторского решения для разрабатываемой движущейся платформы было решено выбрать шасси с колесным ходовым устройством. Среди всех имеющихся вариантов (гусеничные, шагающие и т.д.) – этот самый простой в своей реализации. Для того, чтобы платформа могла передвигаться, необходимо оснастить ее шасси двигателем с трансмиссией, связанной с ведущими колесами, и поворотным механизмом, которые должны обеспечить достаточную точность движения [3].

В качестве двигателя подходит любой коллекторный двигатель малой мощности и напряжением питания 5 – 12 вольт, который представлен на рисунке 3.



Рисунок 3 – Двигатель

В качестве поворачивающего устройства удобно использовать сервопривод (рисунок 4), обеспечивающий достаточную точность поворота платформы.

Изм.	Лист	№ докум.	Подпись	Дата



Рисунок 4 – Сервопривод

В дальнейшем конструкцию шасси платформы можно спроектировать для более серьезных нагрузок и оснастить более мощным оборудованием. При этом перенести всю управляющую электронику на модель для опытного применения не оставит труда, так как разработка ведется с уклоном на универсальность применения данного комплекса к различным моделям платформ. Главное, чтобы при установке комплекса на мощные модели было соблюдено правильное питание управляющей части. Также следует заменить управляющие модули для двигателей на более мощные с характеристиками, подходящими для управления силовой частью.

2.3 Постановка задач системы

Исходя из анализа требований технического задания, можно выделить следующую основную задачу разрабатываемой системы: реализация управления движением платформы в ручном и автоматическом режимах.

2.4 Определение функций системы

Анализ поставленной задачи и требований технического задания позволил выделить основные функции разрабатываемого программно-аппаратного комплекса:

- реализация движения платформы по заданной линии;
- запись траектории движения по заданной линии, ее сохранение в локальном хранилище;
- движение по сохраненной траектории;
- управление движением платформой в ручном режиме;
- реализация предотвращения аварийных ситуаций.

Изм.	Лист	№ докум.	Подпись	Дата

ВКР.135149.09.03.01.ПЗ

Лист

20

В результате анализа требований технического задания при общем описании системы была построена функциональная схема модулей системы и их взаимодействия, представленная в приложении Б.

Управление платформой в ручном режиме реализуется достаточно просто: оператор передает управляющие сигналы, а аппаратная часть платформы их выполняет. Совсем по-другому обстоят дела с автоматическим управлением. Платформе необходимо достаточно точно позиционировать себя в пространстве, для того чтобы она могла двигаться по нужному маршруту, избегая при этом аварийных ситуаций. Самым простым способом задания маршрута движения является явное его указание, например, черной линией на полу помещения. А чтобы платформа имела возможность распознавать линию, по которой необходимо двигаться, на днище нужно установить датчик линии.

В целях обеспечения безопасности движения платформы во фронтальной части платформы нужно установить датчик расстояния, который будет сообщать о расстоянии до объектов, находящихся на пути движения платформы. Аппаратная часть будет учитывать это расстояние в процессе движения платформы и останавливать ее в случае опасного приближения к преградам.

Далее для того, чтобы контролировать работу этих устройств, необходим микроконтроллер [3]. Он будет обрабатывать данные, полученные от компьютера, в соответствии с ними руководить поведением платформы, а также отправлять компьютеру данные о состоянии платформы и датчика, координатах ее движения и т.д.

Связь микроконтроллера и компьютера обеспечивается посредством сети Wi-Fi. Это довольно удобный способ беспроводного управления платформой, который не потребует установки дополнительного радиооборудования в помещении, где работает комплекс. К слову, на сегодняшний день большинство предприятий уже имеют локальную Wi-Fi сеть, а если и нет – ее установка не потребует много времени и средств. Для обеспечения доступа микроконтроллера к Wi-

Fi сетям на платформу нужно установить Wi-Fi модуль, через который микроконтроллер и будет отправлять данные в сеть.

Также для компьютера должно быть разработано программное приложение, призванное обеспечить взаимодействие пользователя с платформой, ее настройку и управление. Интерфейс приложения должен быть как можно более простым и интуитивно понятным, при этом он не должен терять своего функционала.

Таким образом, необходимо первым делом приступить к проектированию аппаратной части комплекса, в которую входят следующие модули:

- микроконтроллер;
- Wi-Fi модуль;
- драйвер двигателя;
- датчик линии;
- датчик расстояния.

3.1 Проектирование аппаратной части комплекса

В результате анализа функциональной схемы модулей аппаратной части, получена следующая принципиальная схема их подключения, представленная на рисунке 5.

Так как платформа должна обладать максимально возможной мобильностью, не имея никакого подключения к электросети сети или другому источнику питания, ограничивающего ее передвижение, источником питания для нее был выбран аккумулятор.

Для питания компонентов аппаратной части был добавлен модуль «Блок питания», который призван обеспечить все необходимые линии питания электронных компонентов, потребляя энергию от аккумулятора.

К микроконтроллеру модули подключаются через специализированные интерфейсы, которые будут учитываться при подборе компонентов аппаратной части комплекса.

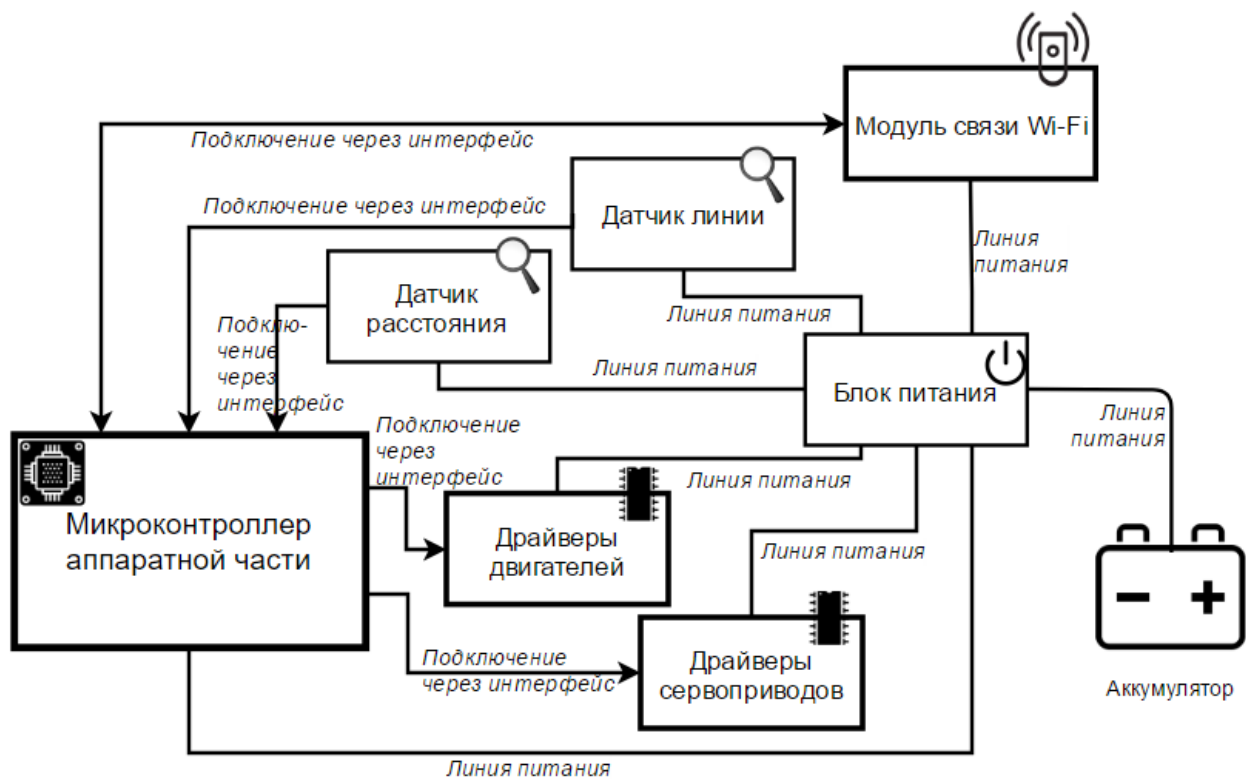


Рисунок 5 – Принципиальная схема подключения модулей аппаратной части

Далее требуется изучить все имеющиеся на сегодняшний день компоненты подобного рода, выбрать из их числа необходимые и спроектировать схему аппаратной части с их использованием.

3.1.1 Выбор компонентов аппаратной части

Главнейшей задачей, лежащей в основе данной работы, было проектирование и реализация как можно более универсальной аппаратной части, которая подошла бы для различных моделей платформ, различных конструкций шасси. Вся электроника управляющей части выбиралась таким образом, чтобы она была более независимой от силовых частей платформы. Такой подход призван обеспечить легкую установку аппаратной части и сопряжение ее с двигательными агрегатами любых мощностей. При этом упор делался и на возможность расширения и совершенствования конструкции платформы под необходимые нужды (например, установка манипулятора для работы с грузами на платформе, установка видеоаппаратуры и т.д.). Помимо этого, уделялось огромное внимание финансовому аспекту: проектирование и разработка велась в условиях достаточно ограниченного бюджета, что в свою очередь позволяет создать не уступающий

всем представленным на рынке системам подобного рода аналог, который будет стоить намного дешевле. При этом он не только не уступает в функциональности рыночным представителям, но во многом даже может превзойти.

На сегодняшний день существует огромное количество всевозможных платформ с микропроцессорами, отдельных микропроцессоров и микроконтроллеров. Среди них есть также и специально предназначенные для решения задач, подобных тем, что предусмотрены проектом, но они либо сложные и дорогостоящие, либо с ограниченным функционалом.

После тщательного изучения предоставленных в продаже микроконтроллеров и микропроцессорных платформ, которые подошли бы для управления платформой, а также всех доступных для них в открытом доступе материалов и документаций, было решено взять в качестве контролирующего всю электронику устройства аппаратную платформу Arduino. Она является довольно удобной платформой быстрой разработки электронных устройств, которая подойдет как новичкам, так и профессионалам. А ввиду своей популярности и принципа открытости (как кода, так и архитектуры) для нее имеется огромное количество комплектующих и плат расширения, предоставляющие возможности работать с любым другим оборудованием.

Далее приступаем к поиску Wi-Fi-адаптера для Arduino. Среди совместимых с Arduino-платформой плат расширения имеются уже готовые адаптеры (Wi-Fi shield), но ввиду большой стоимости, было решено от них отказаться. На сегодняшний день большой популярностью среди разработчиков различных проектов на платформе Arduino пользуется микроконтроллер ESP8266 с интерфейсом Wi-Fi, разработанный компанией Espressif. Он отличается очень малой стоимостью и довольно большим ассортиментом моделей. Такая популярность также привела к появлению большого количества различных материалов в интернете об этом модуле. Поэтому и было решено выбрать именно его.

В качестве распознавателя линии, по которой следует двигаться платформе, было решено не использовать датчики линии заводского производства. Они обладают малой разрешающей способностью, поэтому необходимо было

спроектировать датчик самостоятельно.

Определять наличие объектов на пути следования платформы (для обеспечения безопасности движения) проще всего эхолокационным методом, и для этого идеально подойдет ультразвуковой датчик измерения расстояния. Принцип его работы предполагает излучение короткого ультразвукового импульса, который отражается от объектов на пути его распространения и принимается сенсором. Расстояние рассчитывается исходя из времени от излучения импульса до его приема сенсором и скорости звука в воздухе. Для данного проекта в полной мере хватит возможностей датчика HC-SR04, который представлен на рисунке 6. Напряжение питания у данного модуля равно 5 вольтам, что также создает удобство его использования в данном проекте. А диапазон измерения расстояния составляет 0 – 400 см, чего вполне достаточно [4][12].

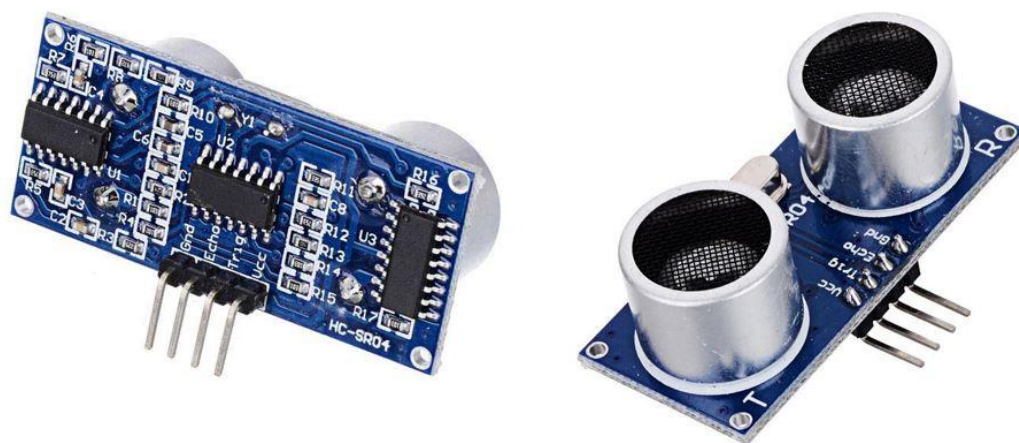


Рисунок 6 – Ультразвуковой датчик измерения расстояния HR-SR04

Далее переходим к рассмотрению имеющейся документации и материалов по платформе Arduino для выбора необходимого микроконтроллера.

3.1.1.1 Аппаратно-вычислительная платформа Arduino

Arduino – это довольно удобный инструмент для проектирования электронных устройств, электронный конструктор для более плотно взаимодействующего с окружающей физической средой оборудования [5]. Данная платформа предназначена для «physical computing» и позволяет компьютеру выходить за рамки виртуального мира в физический и взаимодействовать с ним. Построена

Изм.	Лист	№ докум.	Подпись	Дата

она на простой печатной плате и имеет современную среду разработки программ для микроконтроллера. Принцип разработчиков: открытость архитектуры и программного кода, позволил платформе приобрести огромную популярность и обзавестись широким выбором сопрягаемых комплектующих и плат расширения.

Arduino применяется для создания электронных устройств, которые могут принимать сигналы от различных цифровых и аналоговых датчиков, подключенных к ней, а также для управления различными исполнительными устройствами. Она может взаимодействовать и с различными другими микроконтроллерами и микропроцессорными платформами. Проекты на базе Arduino могут работать самостоятельно или взаимодействовать с программным обеспечением компьютера.

Платы могут быть куплены в сборе или собраны пользователем самостоятельно. Среда разработки программ для Arduino с открытым исходным кодом доступна для бесплатного скачивания с официального сайта.

3.1.1.2 Обоснование выбора микроконтроллера Arduino Due R3

После изучения модельного ряда платформ Arduino самой подходящей для данного проекта была выбрана Arduino Due R3 (представлена на рисунке 7) [6].

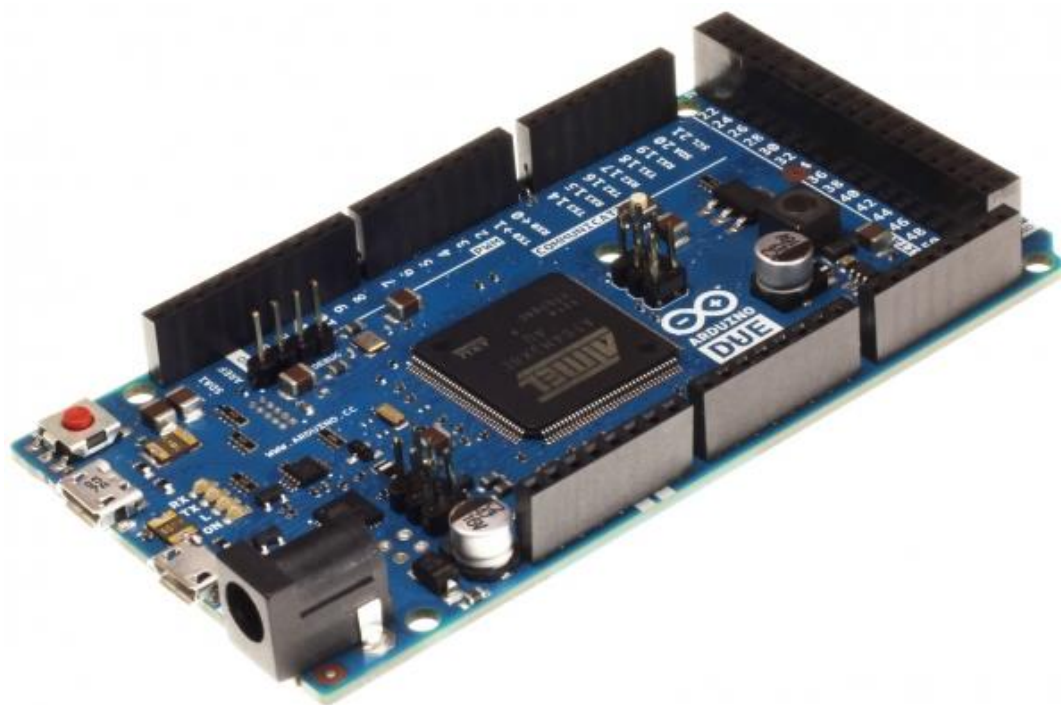


Рисунок 7 – Arduino Due R3

Изм.	Лист	№ докум.	Подпись	Дата

Arduino Due – плата микроконтроллера на базе процессора Atmel SAM3X8E ARM Cortex-M3, обладающего следующими отличительными характеристиками:

- 32-битное ARM ядро с частотой 84 МГц, превосходящее по производительности обычные 8-битные микроконтроллеры и позволяющее выполнять операции с данными шириной 4 байта за 1 такт;
- 96 Кбайт ОЗУ;
- 512 Кбайт флеш-памяти для хранения программ;
- контроллер DMA, который разгружает центральный процессор от выполнения интенсивных операций с памятью.

Это первая плата Arduino на основе 32-битного микроконтроллера с ARM ядром. Она содержит все, что необходимо для поддержки микроконтроллера. Чтобы начать работу с ней, достаточно просто подключить её к компьютеру кабелем микро-USB, либо подать питание с AC/DC преобразователя или батареи.

В отличие от других плат Arduino, Arduino Due работает от 3,3 В. Максимальное напряжение, которое выдерживают вход/выходы составляет 3,3 В. Это очень удобно, так как модуль Wi-Fi, выбранный для данного проекта также работает от 3,3 В, следовательно, нет необходимости организовывать конвертацию сигналов на линии связи между контроллером и ESP8266, как это бы пришлось делать в случае с другими платами Arduino. При этом такое рабочее напряжение не создаст проблем с коммутацией платы с другими контроллерами: имеется вывод IOREF, который позволяет с помощью правильной конфигурации адаптировать присоединенную плату расширения к напряжению, выдаваемому Arduino.

Питается Arduino Due через USB соединитель или с помощью внешнего источника питания (через разъем питания постоянного тока или через вывод Vin на плате). Выбор источника питания выполняется автоматически. Плата может функционировать при внешнем питании от 6 до 20 В. Но, если напряжение питания опускается ниже 7 В, на выводе 5V может оказаться меньше пяти вольт, и

плата будет работать нестабильно. Если же подается напряжение более 12 В, может перегреться стабилизатор напряжения, что приведет к повреждению платы. Рекомендуемый диапазон напряжений – от 7 до 12 В.

У Arduino Due имеются следующие выводы питания:

– Vin – это входное напряжение для платы Arduino, когда она питается от внешнего источника питания (в противоположность 5 вольтам, поступающим через USB соединение или от иного регулируемого источника питания);

– 5V – данный вывод служит выходом регулируемого напряжения 5 В со встроенного стабилизатора на плате. Питающее напряжение через выводы 5V и 3,3V подается в обход стабилизатора, поэтому может повредить плату;

– 3.3V – питание 3,3 В, вырабатываемое встроенным стабилизатором. Максимальный выходной ток 800 мА. Стабилизатор также обеспечивает питание микроконтроллера платы;

– GND – выводы земли;

– IOREF – данный вывод платы Arduino обеспечивает опорное напряжение, при котором работает микроконтроллер.

На Due также имеется самовосстанавливающийся предохранитель, который предназначен для защиты USB портов компьютера от короткого замыкания и перегрузки по току. Несмотря на то, что в большинстве компьютеров есть встроенная защита по току, этот предохранитель дает дополнительную защиту. При токе через USB порт более 500 мА связь автоматически обрывается предохранителем до прекращения перегрузки или короткого замыкания.

Процессор SAM3X, установленный на плате, содержит 512 Кбайт флеш-памяти (2 блока по 256 Кбайт) для хранения программ. Загрузчик (bootloader) записывается Atmel при производстве и хранится в специально отведенном для него ПЗУ. Доступный объем ОЗУ составляет 96 Кбайт в двух смежных банках – 64 Кбайт и 32 Кбайт. Вся доступная память (флеш-память, ОЗУ и ПЗУ) может адресоваться напрямую как плоское адресное пространство.

Существует возможность стереть флеш-память процессора с помощью

встроенной в плату кнопки стирания. При этом из микропроцессора удалится текущая загруженная программа. Для стирания нужно нажать и несколько секунд удерживать кнопку стирания при включенном питании платы.

Due располагает пятьюдесятью четырьмя цифровыми входами-выходами: выводы с 0 по 53. Каждый из них может использоваться в качестве входа или выхода. Выводы работают от 3,3 В. Каждый вывод может выдавать (как источник) ток 3 мА или 15 мА, в зависимости от вывода, или получать (как приемник) ток 6 мА или 9 мА, в зависимости от вывода. На них также имеются внутренние нагрузочные резисторы (по умолчанию они отключены) номиналом 100 кОм.

Кроме того, некоторым выводам на плате назначены специализированные функции:

– Последовательные линии: 0 (RX) и 1 (TX), 19 (RX) и 18 (TX), 17 (RX) и 16 (TX), 15 (RX) и 14 (TX) – эти выводы используются для приема (RX) и передачи (TX) последовательных данных UART интерфейса (с уровнем 3,3 В). Выводы 0 и 1 соединены с соответствующими выводами последовательного контроллера ATmega16U2 USB-to-TTL, что позволяет использовать плату как простой USB-to-TTL конвертер;

– 8-битный ШИМ: выводы с 2 по 13;

– SPI: разъем SPI (разъем ICSP на других платах Arduino) – данные выводы служат для связи по SPI с использованием библиотеки SPI. Сигналы SPI выведены на центральный 6-контактный разъем, который физически совместим с моделями Uno, Leonardo и Mega2560. Этот разъем можно использовать только для связи с другими устройствами SPI, но не для программирования SAM3X по технологии внутрисхемного последовательного программирования (ICSP). SPI на Due также имеет некоторые расширенные функции;

– CAN: CANRX и CANTX – на этих выводах поддерживается протокол связи CAN, но пока его не поддерживают программные интерфейсы (API) Arduino;

- "L" LED: 13 – встроенный светодиод, подключенный к цифровому выводу 13;
- TWI 1: 20 (SDA) и 21 (SCL), TWI 2: SDA1 и SCL1 – эти выводы с использованием библиотеки Wire обеспечивают связь по TWI;
- Аналоговые входы: выводы с A0 по A11 – могут обеспечить разрешение для входящего сигнала в 12 бит (т.е. 4096 различных значений). По умолчанию установлено разрешение 10 бит для совместимости с другими платами Arduino;
- ЦАП: DAC1 и DAC2 – предоставляют достоверные аналоговые выходы с 12-битным разрешением (4096 уровней). Данные выводы можно использовать для создания аудиовыхода;
- AREF – опорное напряжение для аналоговых входов. Он подключен к аналоговому выводу опорного напряжения SAM3X через резисторный мост, поэтому для его активации необходимо отпаять с печатной платы резистор BR1;
- Reset – при низком уровне на этой линии происходит сброс микроконтроллера. Типичное применение вывода Reset – добавление кнопки сброса на плату расширения, которая перекрывает эту кнопку на микроконтроллере.

Arduino Due имеет ряд средств для взаимодействия с компьютером, платами Arduino и другими микроконтроллерами, а также различными устройствами, такими как телефоны, планшеты, фотокамеры и т.п. Установленный на плате процессор SAM3X имеет один аппаратный UART и три аппаратных USARTa для последовательной связи TTL-уровня (3,3 В). Также в отличие от остальных плат Arduino, на Due имеются два разъема USB: Programming port и Native USB port.

Порт программирования (Programming port) соединен с последовательным контроллером ATmega16U2, предоставляющим виртуальный COM порт для программ на подключенном компьютере. Чип 16U2 также соединен с аппаратным UART-ом микропроцессора. Последовательная шина на выводах RX0 и TX0 предоставляет преобразование Serial-to-USB для программирования платы через микроконтроллер ATmega16U2. Светодиоды RX и TX на плате будут мигать, ко-

гда идет передача данных через кристалл ATmega16U2 и через USB подключение к компьютеру (но не во время последовательного обмена по выводам 0 и 1).

Собственный USB порт (Native USB port) подключен напрямую к SAM3X. Через него осуществляется последовательная связь (CDC) посредством USB. Таким образом обеспечивается подключение к монитору последовательной шины, или другим приложениям на вашем компьютере. Также это дает Due возможность эмулировать для присоединенного компьютера USB хост для подключенных периферийных устройств: мыши, клавиатуры и смартфонов.

После изучения доступной технической документации для Arduino Due, было решено, что для данного проекта она подходит лучше всех остальных моделей.

Во-первых, процессор SAM3X на этой плате обладает достаточно большой мощностью и объемом памяти, чтобы выполнять поставленные задачи. При этом есть даже где размахнуться при расширении функционала разрабатываемой аппаратной части.

Во-вторых, сама плата предоставляет широкий выбор интерфейсов связи и контактов ввода-вывода, чего вполне хватает для данного проекта, и с запасом хватает для подключения модулей и исполнительной аппаратуры при расширении функционала системы.

В-третьих, в результате проводившихся испытаний было выявлено, что микроконтроллер ESP8266 (выбранный в качестве модуля Wi-Fi) правильно и без ошибок общается по UART интерфейсу только по установленной производителем по умолчанию скоростью в 115200 бод (baud, бит в секунду). А Due при такой производительности процессора тоже хорошо держит скорость 115200 бод по UART. Также весомым плюсом является рабочее напряжение 3,3 В у Due, т.к. ESP8266 также питается от 3,3 В. Это позволит избежать конвертации уровней сигналов при соединении по UART интерфейсу, а также – питать Wi-Fi модуль прямо от платы Due (выход 3,3V выдерживает ток до 800 мА, такого запаса вполне хватает для ESP8266 даже при пиковой нагрузке).

Возможности конфигурации опорного напряжения для выводов также позволит подключать различное дополнительное оборудование с уровнем логических сигналов и питанием 5 В.

Все эти достоинства Arduino Due в конечном итоге и сказались на его выборе в качестве управляющего устройства для разрабатываемого комплекса.

3.1.1.3 Wi-Fi модуль ESP8266

ESP8266 – микроконтроллер от китайского производителя Espressif с интерфейсом Wi-Fi (представлен на рисунке 8) [7]. Данный модуль был создан для использования в умных розетках, mesh-сетях, IP-камерах, беспроводных сенсорах, носимой электронике и так далее. Одним словом, ESP8266 появился на свет, чтобы стать мозгом грядущего IoT (Internet of Things, «Интернета вещей»).

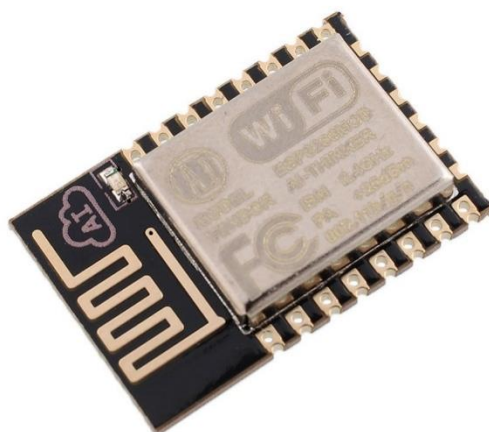


Рисунок 8 – Микроконтроллер ESP8266 ESP-12E

Предполагается два варианта использования этого микроконтроллера [8]:

- в виде моста UART-Wi-Fi, когда модуль на базе ESP8266 подключается к существующему решению на базе любого микроконтроллера и управляется AT-командами, обеспечивая связь решения с инфраструктурой Wi-Fi;
- реализуя новое аппаратное решение, использующее сам чип ESP8266 в качестве управляющего микроконтроллера.

Первый вариант реализуется с помощью любого из недорогих китайских ESP8266-модулей. За простоту использования и отсутствие необходимости программировать модуль он стал так популярен у любителей, разрабатывающих проекты на базе Arduino.

Изм.	Лист	№ докум.	Подпись	Дата

Второй вариант сценария предусматривает написание индивидуальной прошивки для управления чипом «изнутри». На данный момент прошивка должна быть написана для фирменного компилятора, с чем, в основном, и связаны требования к неразглашению информации вокруг этого решения. В обозримом будущем производитель планирует переход на использование GCC, тогда эти ограничения будут сняты. Данный сценарий использования модуля интересен тем, что позволяет создать небольшие устройства, которые реально долго работают от батарей. Это серьезный плюс в сторону автономности и мобильности устройства. Для работы с периферией на борту ESP8266 есть все необходимые возможности.

3.1.1.4 Обоснование выбора микроконтроллера ESP8266-12E

Выбор микроконтроллера ESP8266 обоснован его достаточно малой стоимостью и приличным функционалом. При этом модуль богат документацией и имеющимися в открытом доступе материалами, что очень помогает при его подключении и работе с ним. После изучения предлагаемого списка микроконтроллеров выбор был сделан в пользу версии ESP-12E, так как она является одной из самых свежих на сегодняшний день.

На плате ESP-12E установлен все тот же микроконтроллер, единый для всех остальных версий 32-битный Tensilica's L106 Diamond series с частотой 80 МГц, 80 Кбайт ОЗУ DRAM и высокоскоростной памяти IRAM [9].

Модуль содержит необходимые средства для совместной работы с другими микроконтроллерами. На плате модуля под металлическим кожухом находятся микросхема памяти 4 Мбайт, микросхема ESP8266EX и кварцевый резонатор. Рядом с кожухом находится светодиод показывающий включение питания и миниатюрная антенна из дорожки на верхнем слое печатной платы в виде змейки. Металлический кожух экранирует компоненты модуля и тем самым улучшает электромагнитные свойства. Также в микросхему ESP8266EX интегрирован датчик температуры.

Модуль ESP-12E способен работать совместно с персональным компьюте-

ром или другим аналогичным прибором, имеющим порты USB или COM с интерфейсом RS232. При этом соединение выполняется через преобразователь интерфейсов. Модуль продается с уже установленным программным обеспечением, обеспечивающим работу в режиме моста UART–Wi-Fi при соединении с портом ПК или другим микроконтроллером. Настройка и обмен данными происходят с помощью AT-команд [10]. Прошивку, управляющую работой модуля можно обновить, используя ресурсы интернета.

Ниже приведены характеристики рассматриваемого модуля:

- напряжение питания 3,3 – 3,6 В;
- потребление в пиковом режиме работы 300 мА;
- потребление в режиме ожидания 0,9 мА;
- потребление в режиме глубокого сна 10 мкА;
- потребление в режиме экономии энергии: 1,2 мА (DTIM1) и 0,86 мА (DTIM3);
- предельный ток входов-выходов 12 мА;
- время выхода из режима сна менее 2 мс;
- поддерживаемые типы шифрования: WEP, WPA, WPA2;
- выходная мощность: 802.11b + 16 дБм ± 2 дБм (при скорости 11 МБод), 802.11g + 14 дБм ± 2 дБм (при скорости 54 МБод), 802.11n + 13 дБм ± 2 дБм (HT20, MCS7);
- чувствительность приемника: 802.11b + 93 дБм (при скорости 11 МБод, ССК), 802.11g + 85 дБм (при скорости 54 МБод, OFDM), 802.11n + 82 дБм (HT20, MCS7);

На плате модуля размером всего лишь 16 мм на 24 мм с шагом в 2 мм (что на самом деле не очень удобно при пайке данного модуля на макетные платы со стандартным шагом 2,54 мм) расположены 22 вывода, которые предназначены:

- 1 (RST) – сброс, выполняется при подаче низкого уровня;
- 2 (ADC/TOUT) – вход АЦП разрешающей способностью 10 бит, диапазон измерений 0 – 1 В;

- 3 (EN) – включение модуля, активный уровень высокий;
- 4 (GPIO16) – линия ввода-вывода с открытым коллектором;
- 5 (GPIO14) и 6 (GPIO12) – линии ввода-вывода;
- 7 (GPIO13) – линия ввода-вывода или сигнал RXD интерфейса UART2;
- 8 (VDD) – питание 3,3 В;
- 9 (CS0) – сигнал CS интерфейса SPI;
- 10 (MISO) – сигнал MISO интерфейса SPI;
- 11 (GPIO9) и 12 (GPIO10) – работают только в ESP-12-D;
- 13 (MOSI) – сигнал MOSI интерфейса SPI;
- 14 (SCLK) – сигнал Clock интерфейса SPI;
- 15 (GND) – подключение общего провода (земли);
- 16 (GPIO15) – линия ввода-вывода или сигнал TXD интерфейса UART2;
- 17 (GPIO2) – линия ввода-вывода или подключение светодиода Wi-Fi статуса;
- 18 (GPIO0), 19 (GPIO4) и 20 (GPIO5) – линии ввода-вывода;
- 21 (RXD0) – сигнал RXD интерфейса UART0;
- 22 (TXD0) – сигнал TXD интерфейса UART0.

Распиновка модуля показана на рисунке 9 [9].

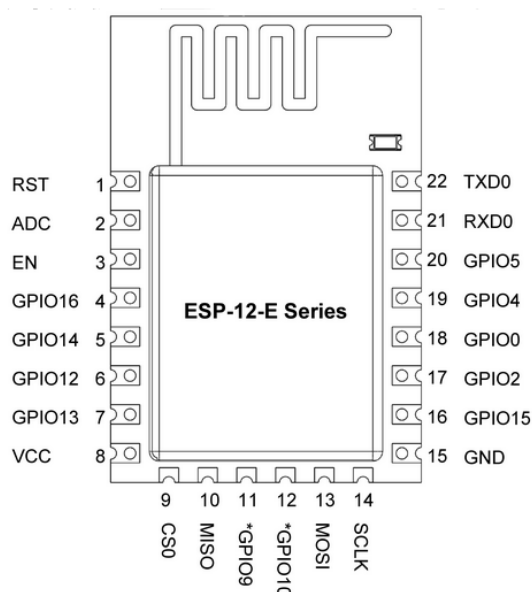


Рисунок 9 – Распиновка модуля ESP8266 – ESP-12E

Изм.	Лист	№ докум.	Подпись	Дата

Ввиду своих размеров и малой стоимости, имея при этом такой широкий функционал и возможности энергосбережения, данный Wi-Fi модуль идеально подходит для разрабатываемого комплекса. Рабочее напряжение, схожее с рабочим напряжением Arduino, позволит при подключении через UART обойтись без конвертации уровней сигналов на линиях связи, а сам модуль сможет питаться напрямую от Arduino Due без подключения стороннего питания.

3.1.2 Блок питания

Для платформы в качестве источника питания было решено выбрать аккумулятор, выдаваемое напряжение которого составляет 12 В. Для питания аппаратной части необходимы две линии питания 5 В и 3,3 В. Чтобы обеспечить такие линии питания можно спроектировать простой блок питания, собранный на стабилизаторах напряжения AMS1117-5.0 и AMS1117-3.3, выдающих на выходе строго 5 В и 3,3 В соответственно, или на аналогичных им. Согласно даташитам данных стабилизаторов на их входы и выходы необходимо установить фильтрующие конденсаторы определенного номинала. Но по результатам тестов было решено установить на вход каждого стабилизатора керамический конденсатор емкостью 10 мкФ, а на выходе – электролитический емкостью 100 мкФ. Переключатель S1 служит для включения и выключения питания платформы. Схема такого блока приведена на рисунке 10.

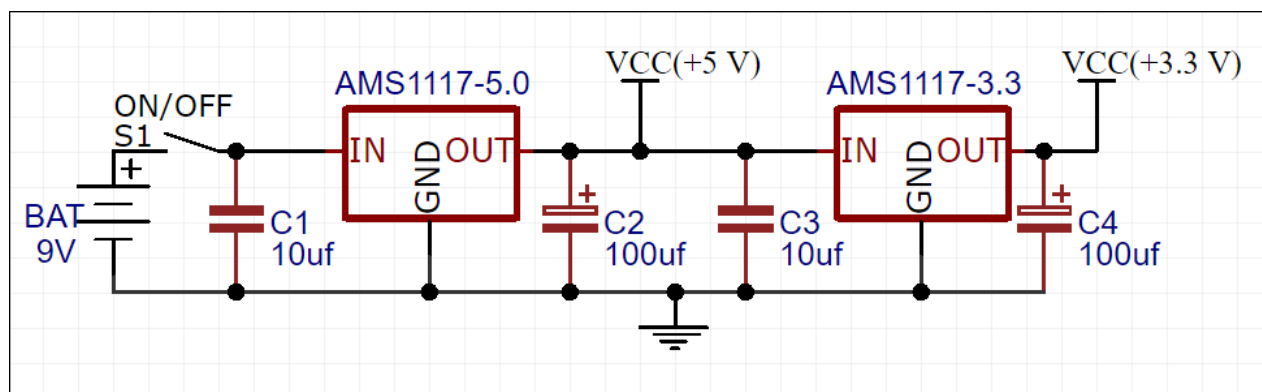


Рисунок 10 – Блок питания для аппаратной части

3.1.3 Драйвер двигателя

Непосредственно от микроконтроллера Arduino управлять коллекторным двигателем не получится, так как она не может выдавать требуемое для выбранного мотора напряжение. Управлять двигателем платформы будет специальный

Изм.	Лист	№ докум.	Подпись	Дата

драйвер, собранный в микросхеме ВА6289F. Данная микросхема не требует особой обвязки, но согласно даташиту, рекомендуется установка фильтрующего электролитического конденсатора емкостью 220 мкФ на вход опорного питания мотора (Vm). Питается схема от линии 5 В, а опорное напряжение мотору будет подаваться прямо от батареи. Через входы Fin и Rin микроконтроллер может управлять направлением движения мотора, подавая на эти входы отличные друг от друга логические уровни (1 и 0, 0 и 1). А подавая на вход Vref напряжение в диапазоне от 0 до 5 вольт (до напряжения питания схемы), Arduino сможет регулировать обороты двигателя. К выходам OUT1 и OUT2 драйвера подключается сам двигатель. Схема подключения драйвера показана на рисунке 11.

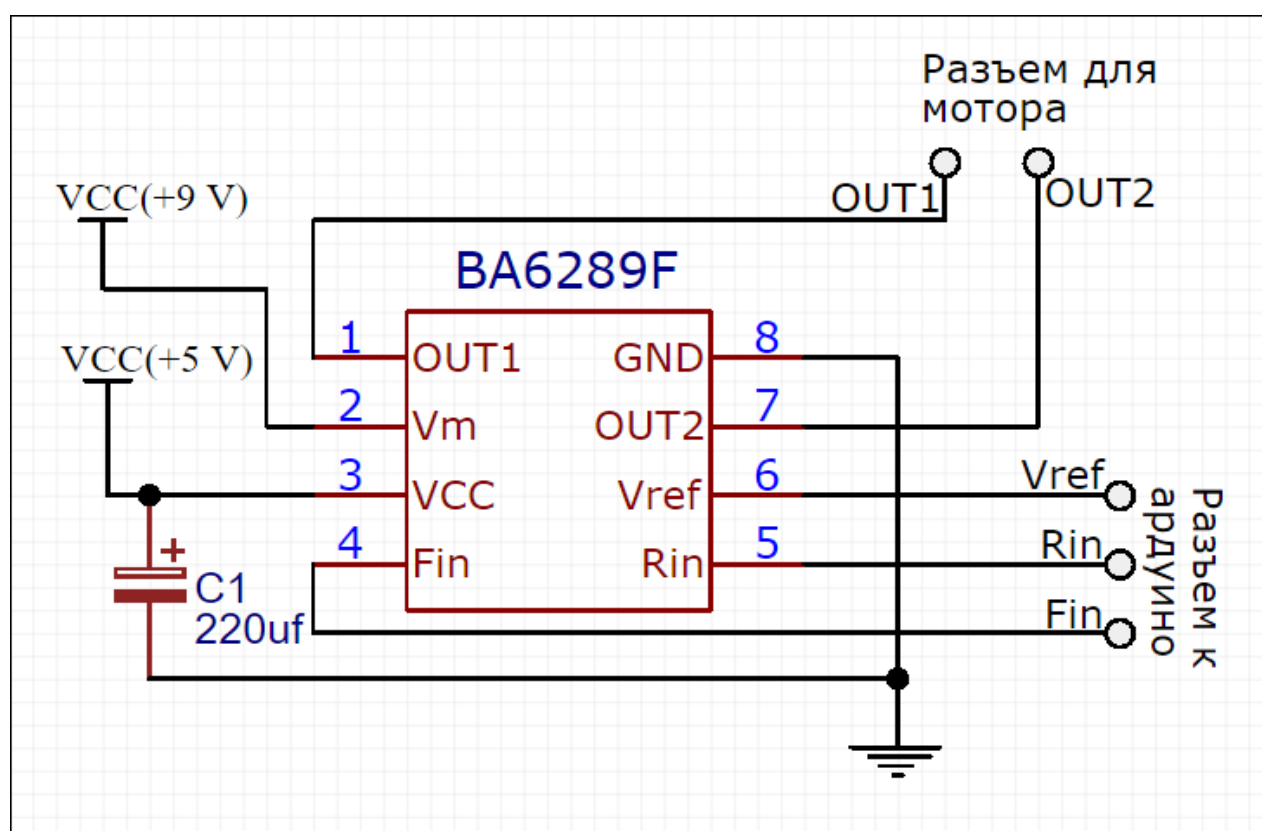


Рисунок 11 – Схема подключения драйвера двигателя ВА6289F

3.1.4 Датчик линии

За основу работы разрабатываемого датчика линии берется поглощение света черным цветом нарисованной линии. Принцип работы заключается в следующем: светодиод излучает свет, который отражается от поверхности и принимается фотодиодом, который при этом открывается. То есть, если свет поглощается черной линией, то фотодиод не откроется и ток через него не потечет, что

позволит явным образом определять наличие линии под датчиком. Целесообразно разделить их перегородкой, которая будет предотвращать паразитное излучение. На рисунке 12 показан принцип работы такого датчика. Таким образом для одиночного датчика достаточно излучающего элемента и приемника излучения. Светодиод, излучающий в инфракрасном диапазоне, и фотодиод (в качестве приемника излучения) вполне для этого подходят.



Рисунок 12 – Принцип работы датчика линии

В проекте также требуется, чтобы датчик мог каким-либо образом показывать величину отклонения нарисованной линии от его центра. Необходимо расширить его разрешающую способность, для чего разместим параллельно несколько датчиков в ряд в одном корпусе (достаточно пяти датчиков: центральный и по два с каждой стороны – для измерения отклонения). Конструкция датчика показана на рисунке 13.

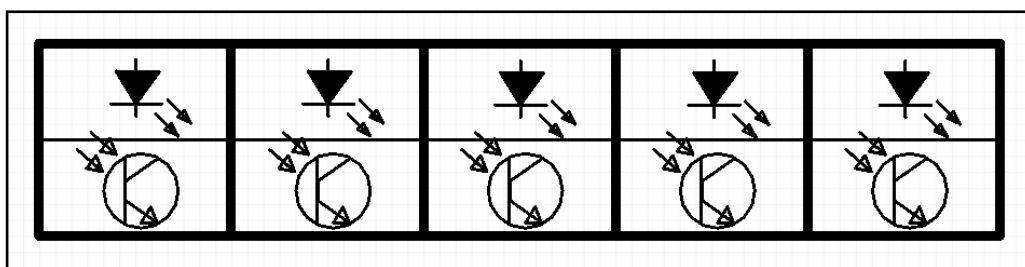


Рисунок 13 – Расширенный датчик линии

Схема датчика линии выглядит следующим образом. От линии 5 В блока питания через ограничивающие ток резисторы 220 Ом питаются инфракрасные светодиоды L-7113SF4C. Длина волны их излучения равна 880 нм. В качестве фототранзисторов были выбраны L-3DP3C с аналогичной длиной принимаемой

Изм.	Лист	№ докум.	Подпись	Дата

волны. Они подключены также к линии 5 В, но уже взятой у платформы Arduino. Сделано это для того, чтобы не использовать для них лишние ограничивающие резисторы во избежание повреждения входов Arduino слишком высоким током, который может потечь от блока питания через эти транзисторы. Их эмиттеры являются выходами датчика и подключаются к Arduino. Схема датчика представлена на рисунке 14.

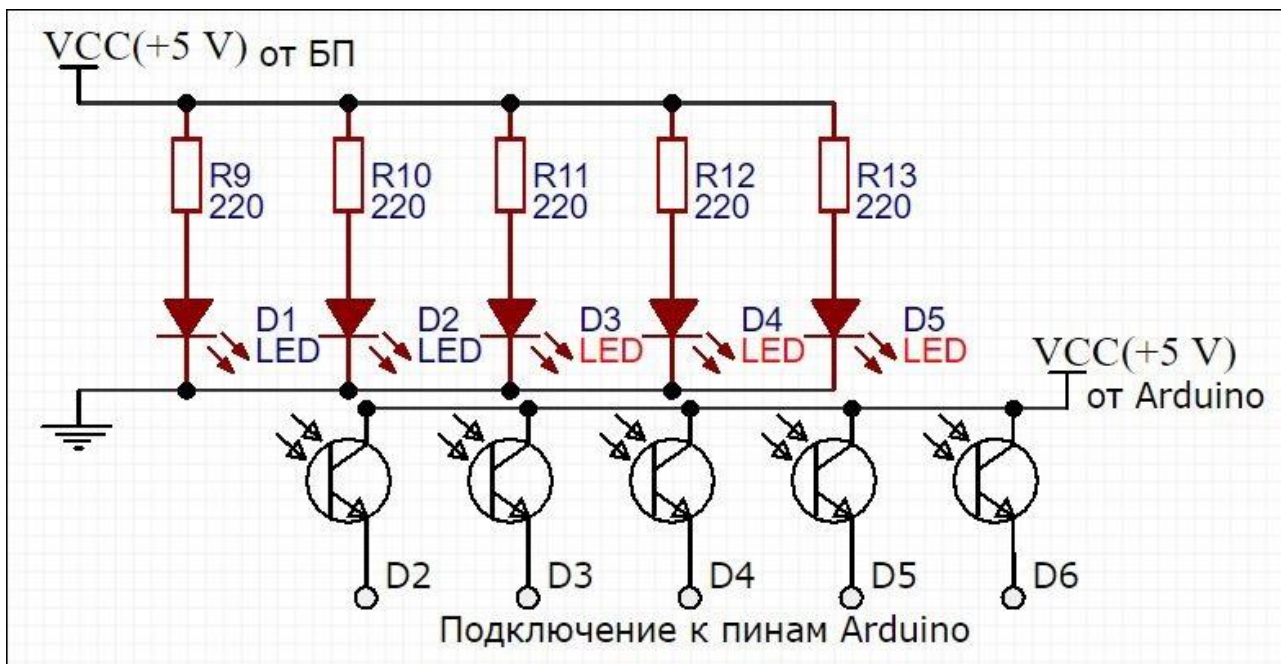


Рисунок 14 – Схема датчика линии

3.1.5 Обвязка Wi-Fi модуля ESP8266-12E

Для правильного запуска Wi-Fi модуля необходимо подать на его вход питания (VCC) напряжение с линии 3,3 В от блока питания, при этом подать это же напряжение на вход GPIO15. Выход RST, предназначенный для перезагрузки модуля подтягивается к напряжению питания через резистор 4 – 10 кОм. Для возможности перезагрузки модуля следует соединить этот выход с землей через кнопку S2. Выход CH_PD, отвечающий за включение схемы, также необходимо подтянуть к напряжению питания через резистор. Выход GPIO0, подключенный к земле, позволяет перевести схему в режим прошивки микроконтроллера. Пере-прошивать его заводскую прошивку нам не потребуется, поэтому можно соединить этот выход с линией питания через резистор, но переключку с землей на всякий случай также организуем. Схема обвязки ESP-12E показана на рисунке 15.

Изм.	Лист	№ докум.	Подпись	Дата

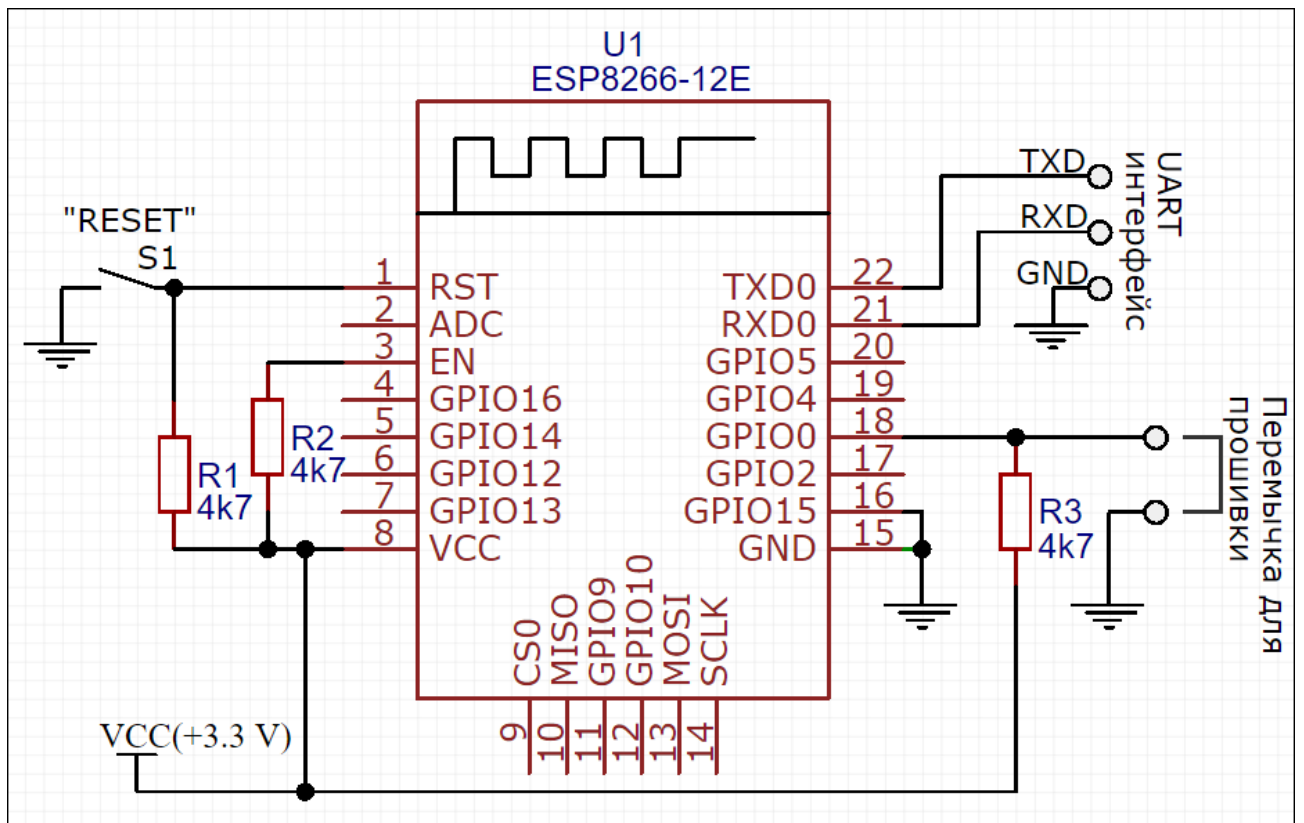


Рисунок 15 – Обвязка модуля ESP8266

3.1.6 Схема подключения компонентов к Arduino Due

Теперь осталось только подключить все эти устройства к Arduino и блоку питания. Выходы RXD и TXD модуля Wi-Fi подключается пинам 18(TX1) и 19(RX1) на плате Arduino, которые отвечают за аппаратный UART интерфейс под номером 1. Подключаются они по принципу: RXD к 18(TX1) а TXD к 19(RX1).

Управлять драйвером двигателя микроконтроллер будет через пины 5, 6 и 7. Пины 7 и 6 управляют логикой вращения двигателя, а пин 5 с помощью широтно-импульсной модуляции будет регулировать напряжение на входе Vref драйвера, и этим самым будет регулировать обороты двигателя.

Датчик измерения расстояния подключается к пинам 4 и 3. А датчик линии подключается к пинам 8 – 12 по порядку следования выходов сенсоров (A1 – A5). Через пин 2 микроконтроллер с помощью ШИМ будет управлять сервоприводом.

Далее все эти устройства подключаются к линиям питания в соответствии с их требованиями. Wi-Fi модуль подключается к линии 3,3 вольта. Питание

Изм.	Лист	№ докум.	Подпись	Дата

драйвера двигателя VCC – к 5 вольтовой линии, а опорное напряжение мотора Vm подключается напрямую к аккумулятору. Сервопривод, датчик измерения расстояния и датчик линии также будут питаться от линии 5 вольт. Сама платформа Arduino также подключается через вывод внешнего питания Vin к линии 5 В. При этом все выходы GND (земля) устройств должны быть подключены к общей земле блока питания.

К пину 13 Arduino будет подключена кнопка, при нажатии на которую платформа будет переходить в режим настройки.

Общая схема аппаратной части комплекса приведена на рисунке 16.

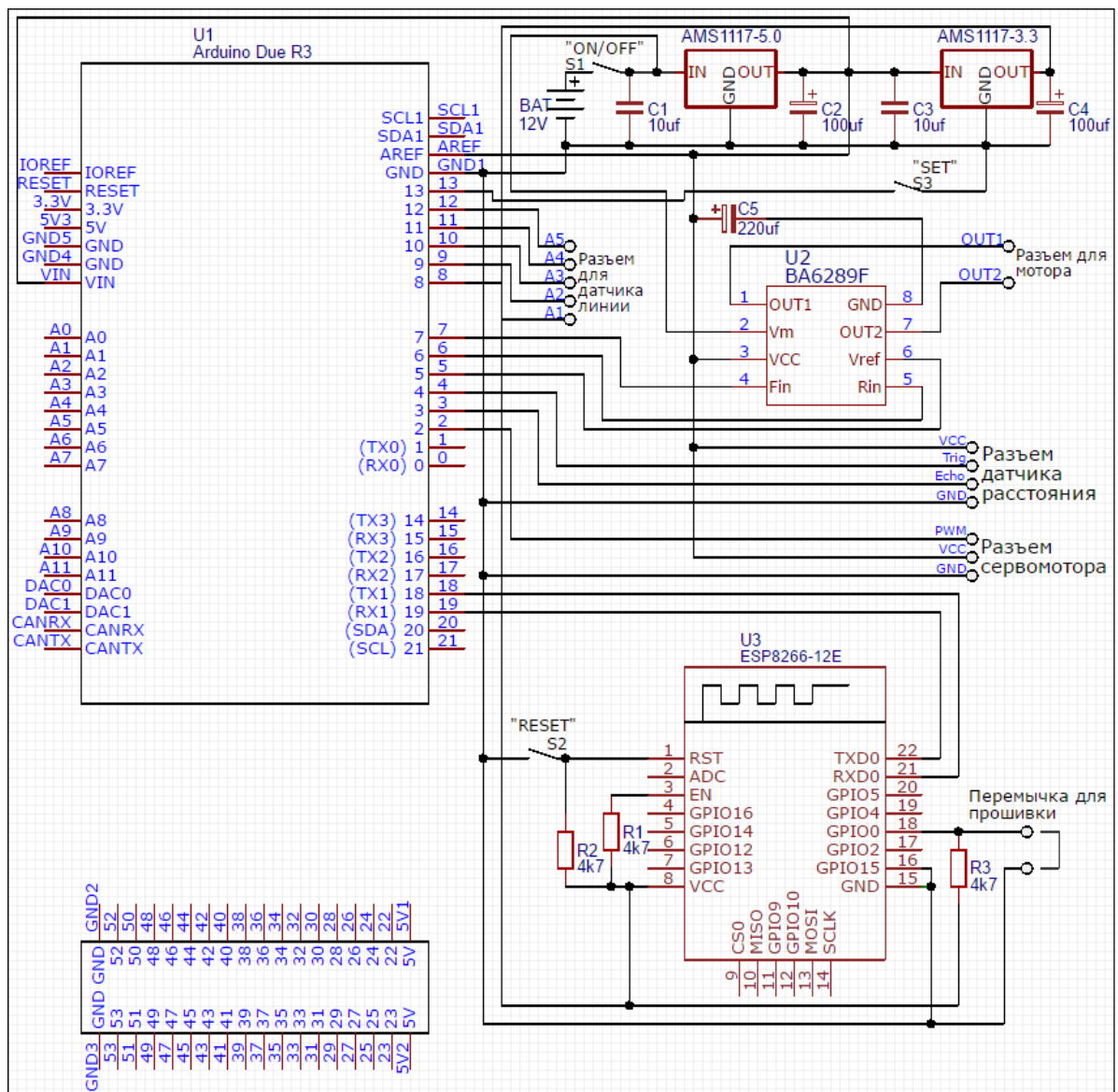


Рисунок 16 – Схема аппаратной части комплекса

Изм.	Лист	№ докум.	Подпись	Дата

3.2 Взаимодействие программной части с аппаратной частью комплекса

Связь микроконтроллера платформы, управляющего ей, и программного обеспечения происходит посредством сети Wi-Fi через роутер или адаптер, установленный в компьютере пользователя. Платформа при включении подключается к сети предприятия и ожидает включения приложения. Приложение при включении производит поиск доступной для работы платформы в данной сети, опознает ее и подключает к себе. Далее под управлением пользователя программа посылает платформе управляющие команды. При потере связи с приложением платформа останавливается и переходит в режим ожидания подключения.

Передача информации по сети происходит пакетами определенного содержания и оформления. Со стороны приложения платформе отправляются управляющие пакеты, сигнализирующие о начале операции, конце операции, промежуточные пакеты с данными для выполняемой операции, а также пакеты запросов о состоянии платформы и запросов передачи данных. Со стороны платформы приложение получает пакеты о состоянии, а также запрошенные данные (координаты траектории движения). Пакеты для одной и другой стороны абсолютно идентичны.

Пакет имеет следующий формат:

#| id платформы | id команды | необходимые данные для команды |#.

Начало и конец пакета обозначены символами решетки. Внутри пакет разделяется вертикальными чертами.

Id платформы – уникальный идентификатор для подключенной платформы позволяет управлять сразу несколькими платформами, а также надежно их опознавать.

Id команды – подразделяет все команды для платформы на классы для ручного управления и для автоматического, а также на команды запросов.

Необходимые данные для команды хранят в себе информацию, которая используется при выполнении той или иной команды, или ответ на запрос. Эта

часть пакета может быть разделена на несколько, например, при передаче информации об угле поворота и скорости платформы.

3.3 Проектирование программного обеспечения для компьютера на языке UML

Программное обеспечение разрабатываемой системы в рамках данной работы спроектировано в специализированном средстве моделирования Rational Rose, использующем язык UML [13].

UML (Unified Modeling Language, унифицированный язык моделирования) – это язык графического описания для объектного моделирования в области разработки программного обеспечения, системного проектирования и отображения организационных структур.

3.3.1 Диаграмма вариантов использования системы

Данный вид диаграмм предназначен для описания функционального назначения системы. Она позволяет определить общие границы моделируемой предметной области, разработать исходную модель системы, а также сформулировать общие требования к функциональному поведению системы.

На рисунке 17 представлена диаграмма вариантов использования для разрабатываемой системы.

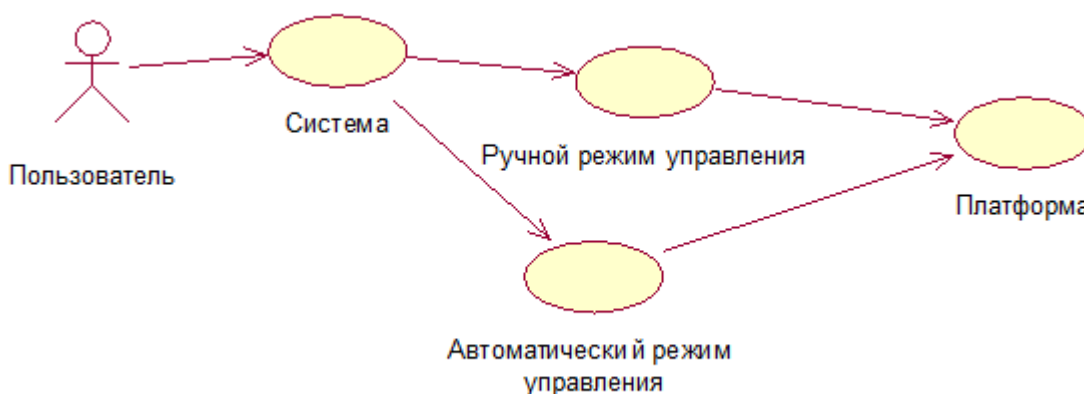


Рисунок 17 – Диаграмма вариантов использования системы

Пользователь может управлять движением платформы в ручном режиме, либо же выбрать автоматический режим.

3.3.2 Диаграмма состояний системы

Диаграмма состояний описывает поведение системы. В ней отображены все возможные состояния системы, а также условия и процессы, при которых происходит смена этих состояний.

На рисунке 18 показана диаграмма состояний UML, отражающая поведение разрабатываемой системы.

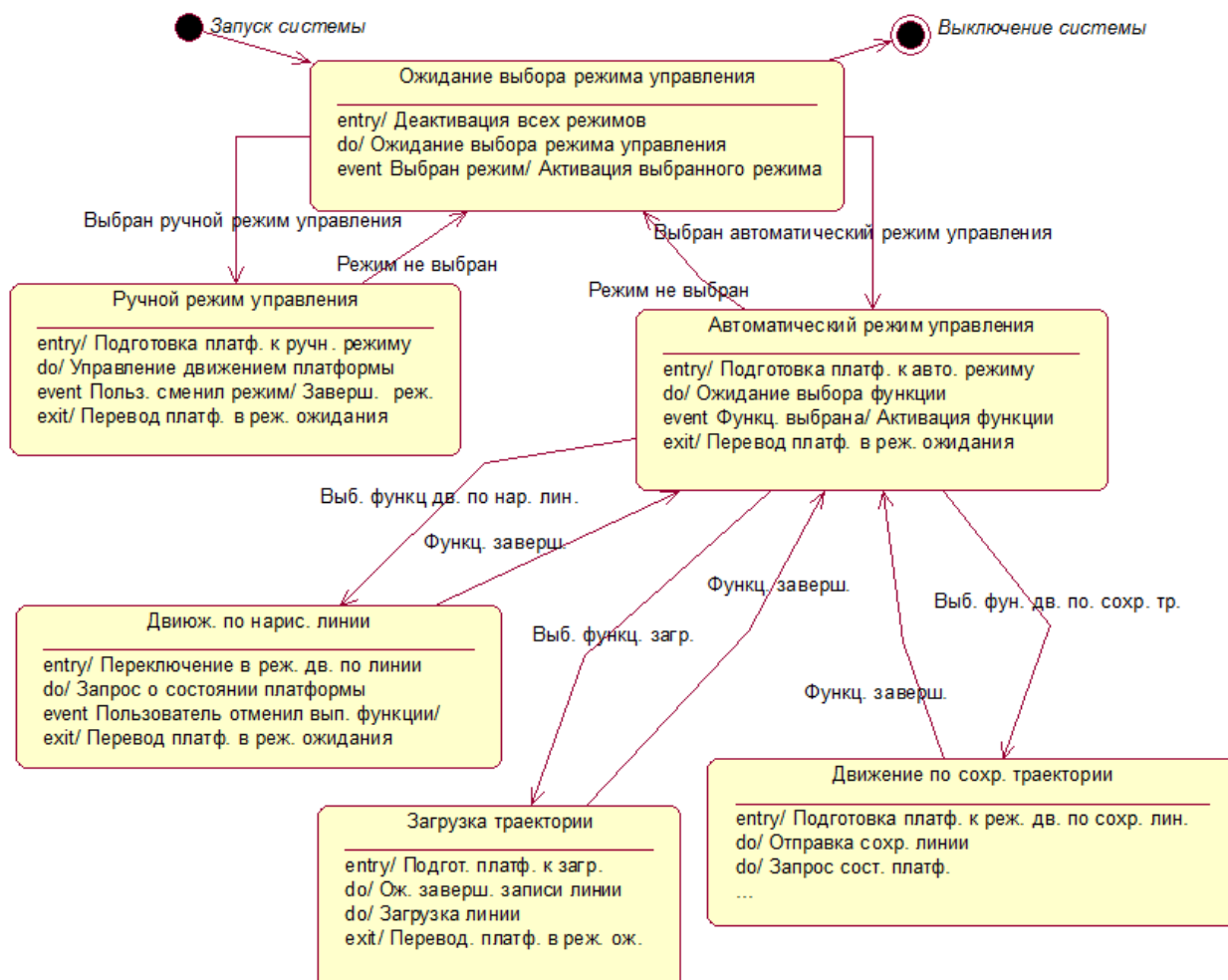


Рисунок 18 – Диаграмма состояний системы

3.3.3 Диаграмма последовательностей системы

Диаграмма последовательностей используется для моделирования процессов взаимодействия объектов системы во времени, т.е. она содержит последовательность выполняемых действий во времени.

На рисунке 19 показана диаграмма последовательностей для разрабатываемой автоматизированной системы в ручном режиме ее работы.

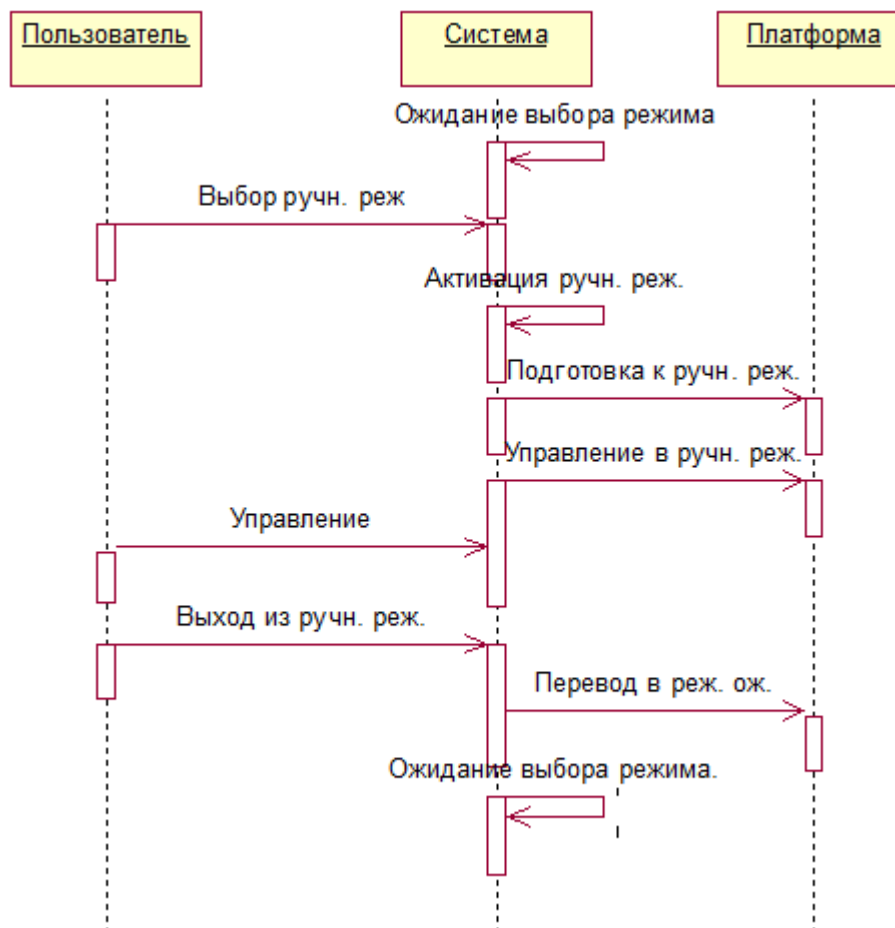


Рисунок 19 – Диаграмма последовательности системы в ручном режиме управления

При включении система ожидает выбора режима. Выбрав ручной режим, пользователь активирует его в системе, далее она подготавливает платформу к ручному режиму управления и начинает цикл управления. В цикле она снимает данные с экранной формы пользовательского интерфейса и клавиатуры, передает их платформе вместе с командой на выполнение. Если пользователь выходит из режима управления, то система переводит платформу в режим ожидания и деактивирует выбранный ранее режим.

На рисунке 20 представлена диаграмма последовательностей системы в автоматическом режиме управления.

В автоматическом режиме управления система ожидает от пользователя выбора функции, которую следует выполнить: движение по нарисованной линии, запись траектории движения, движение по сохраненной траектории. После переводит платформу в соответствующий режим и периодически получает от нее

данные о состоянии выполнения. Если пользователь остановил выполнение функции, то система переводит платформу в режим ожидания. Если пользователь завершил режим, то система переводит платформу в режим ожидания и деактивирует режим.

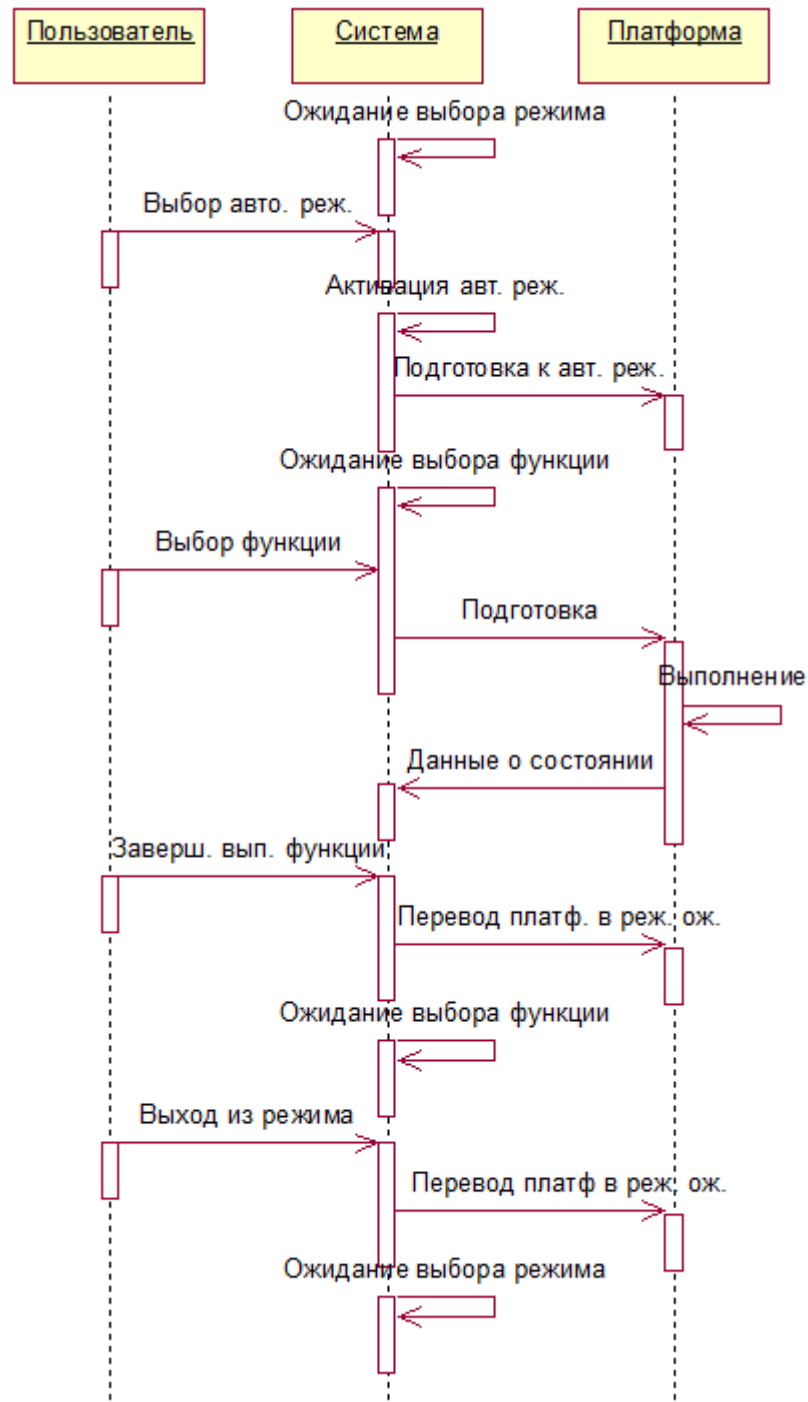


Рисунок 20 – Диаграмма последовательностей системы в автоматическом режиме управления

3.3.4 Проектирование алгоритмов основных функций системы

Функция движения по нарисованной линии реализована следующим образом. При выборе пользователем этой функции платформе отправляется пакет с командой на выполнение соответствующего режима движения. Платформа его активирует и начинает движение. Приложение периодически запрашивает данные о состоянии платформы. Платформа в свою очередь ожидает получения команды завершения процесса. Если пользователь отменил выполнение режима, приложение отправляет платформе пакет с командой завершения. Платформа останавливается, завершает режим и переходит к ожиданию дальнейших указаний от приложения. Алгоритм данной функции представлен на рисунке 21.

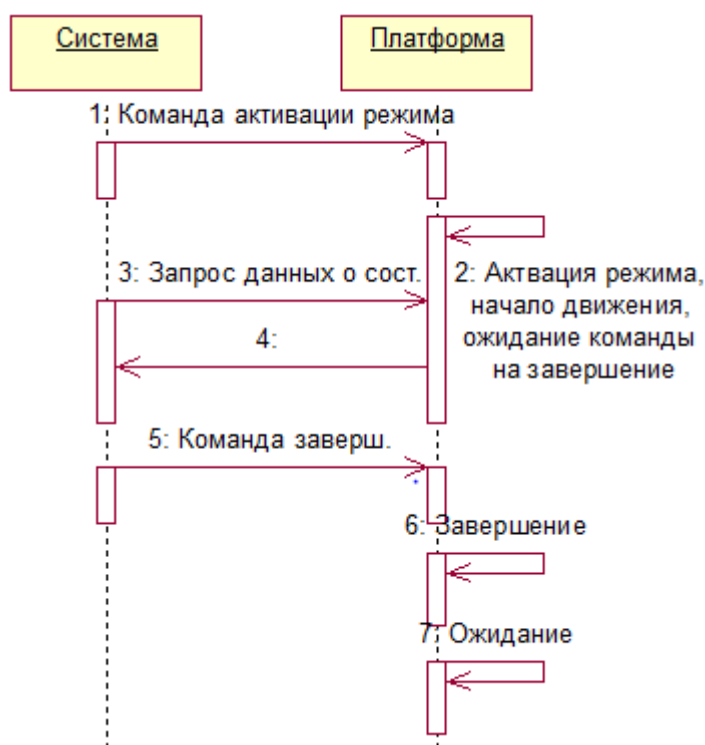


Рисунок 21 – Алгоритм функции движения по нарисованной линии

Если пользователь выбрал функцию записи траектории платформой, и сохранения ее в локальном хранилище, то платформе отправляется команда на активацию соответствующего режима. Платформа его активирует и начинает движение по нарисованной линии, вычисляя координаты и записывая их, а система ожидает ответ о завершении записи. После завершения записи, платформа отправляет ответ о готовности к передаче данных. Начинается загрузка записанной

траектории в приложение и сохранение ее в текстовый файл в локальном хранилище. После завершения операции загрузки платформа переходит в режим ожидания команд от приложения. Алгоритм данной функции представлен на рисунке 22.

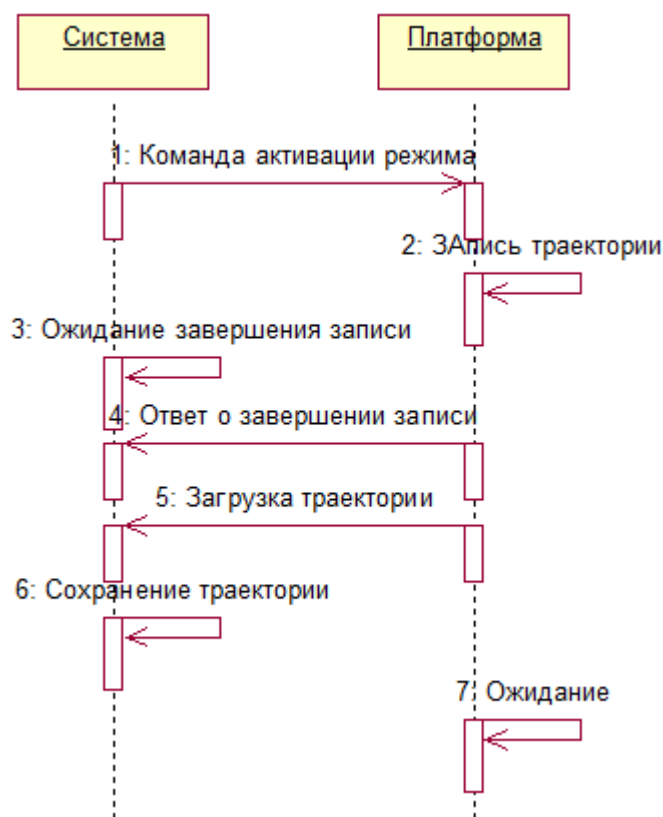


Рисунок 22 – Алгоритм функции записи траектории

Включение функции движения по сохраненной траектории начинается с отправки платформе команды на выполнение соответствующего режима. Платформа, получив ее, отправляет ответ о готовности к загрузке траектории. Далее происходит чтение приложением траектории из выбранного файла и отправка ее платформе. Платформа начинает движение по принятым координатам, а приложение периодически запрашивает данные о ее состоянии. Если пользователь отменил выполнение данного режима, платформе отправляется пакет с командой завершения. Платформа останавливается и завершает режим, переходя к ожиданию дальнейших указаний. Алгоритм данной функции представлен на рисунке 23.

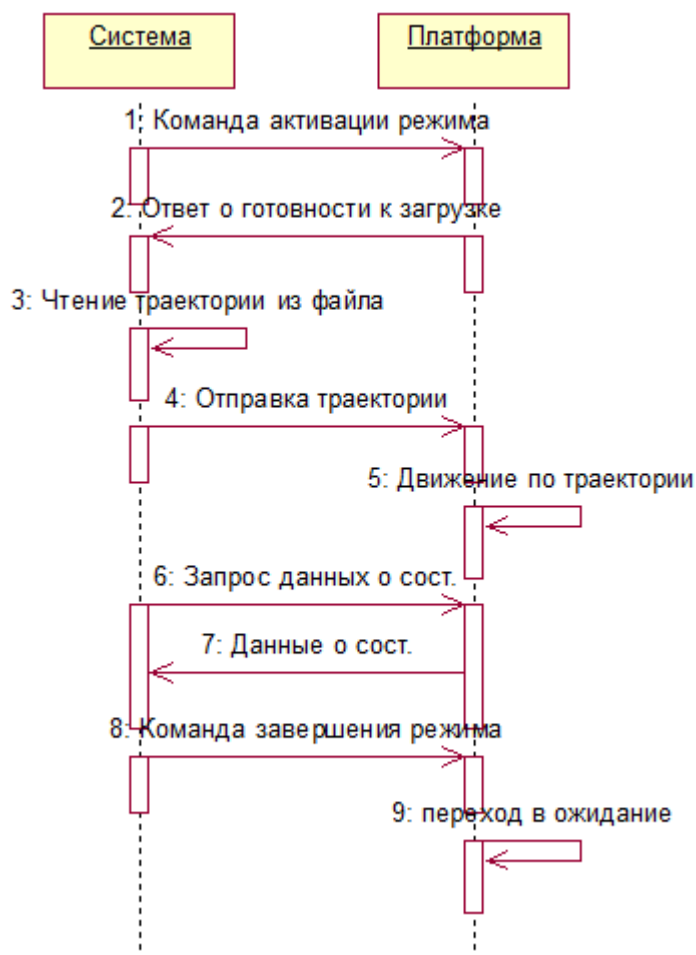


Рисунок 23 – Алгоритм функции движения по сохраненной линии

Режим ручного управления движением платформы реализован следующим образом. При его активации платформе отправляется пакет с командой на выполнение соответствующего режима. Далее приложение в цикле снимает данные о нажатых клавишах, значениях скорости, угла поворота и отправляет их платформе. Платформа, получив пакет с управляющими данными, разбирает его и отдает команды драйверу двигателя и сервомотору на выполнение. В ответ приложению платформа отправляет данные о своем состоянии. Когда пользователь выключает данную функцию, платформе отправляется команда о завершении, и она переходит в режим ожидания последующих указаний. Алгоритм данной функции представлен на рисунке 24.



Рисунок 24 – Алгоритм функции ручного управления

3.3.5 Проектирование функциональных модулей программного приложения

Спроектировав алгоритмы выполнения функций системы, можно приступить к проектированию программного приложения для ПЭВМ. Для того, его было удобнее разрабатывать и тестировать, его необходимо поделить на функционально законченные части, т.е. модули.

Разрабатываемый программный продукт включает в себя три основных модуля, которые были выделены в результате анализа требований технического задания в пункте 2.1. Для каждого модуля был спроектирован свой собственный класс, а также классы состояний, которые позволяют контролировать работу модулей.

Модуль связи представлен на рисунке 25. Он предназначен для осуществления передачи данных между приложением и микроконтроллером аппаратной части комплекса. Это, по своей сути, программный сервер, к которому подключается платформа и через который можно обмениваться с ней сообщениями. Он

хранит в себе текущий IP адрес и порт компьютера, информацию о подключенном клиенте (Wi-Fi модуле платформы), а также свое состояние, которое реализовано объектом сопряженного класса состояния сервера, специально созданного для хранения статуса сервера и состояния подключения клиента. Данный модуль включает в себя следующий набор функций:

- метод запуска сервера;
- метод подключения клиента (Wi-Fi модуля аппаратной части);
- метод приема сообщения;
- метод отправки сообщения;
- метод остановки сервера.

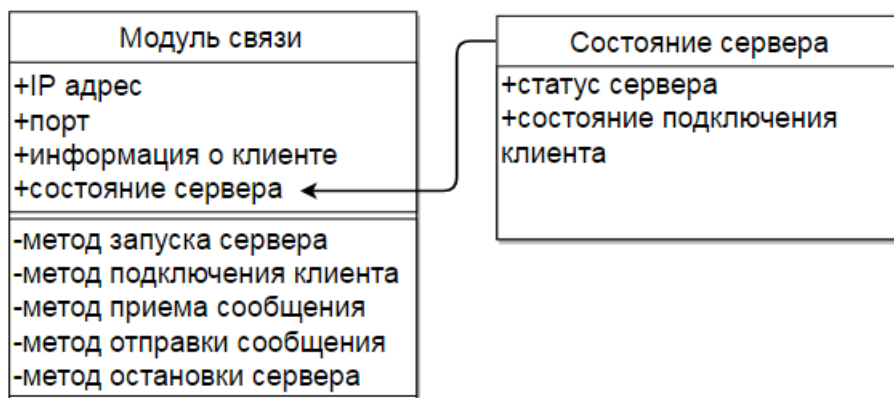


Рисунок 25 – Модель модуля связи

Модуль для управления платформой представлен на рисунке 26. В этом модуле собран основной функционал системы: подключение платформы, ее идентификация, запуск функций на выполнение и их остановка. Он хранит в себе свое состояние, реализованное с помощью объекта сопряженного класса состояния модуля управления, содержащего статус модуля, состояние подключения платформы, состояние цикла управления в ручном режиме и состояния цикла управления в автоматическом режиме. Также модуль содержит пакет передаваемых данных (реализованный объектом класса передаваемых данных) и пакет принимаемых данных (реализованный объектом класса принимаемых данных). Они созданы для облегчения работы с данными, которые передаются или принимаются. К числу рабочих методов данного модуля относятся:

- метод запуска модуля;

- метод подключения (идентификации) платформы;
- метод цикла ручного режима управления;
- метод автоматического управления, который включает в себя циклы выполняемых платформой функций в автоматическом режиме управления (движение по нарисованной линии, запись траектории, движение по сохраненной траектории);
- метод для подготовки пакета к отправке (собирает данные в специально оформленную строку);
- метод парсинга принятого пакета (разбор принятой строки на составляющие);
- метод обработки экранной формы графического интерфейса;
- метод остановки модуля.



Рисунок 26 – Модель модуля управления платформой

Графический интерфейс пользователя (рисунок 27). Призван обеспечить удобную и интуитивно понятную оболочку для приложения, через которую пользователь сможет управлять комплексом. Этот модуль включает в себя:

- экранную форму;
- набор методов-обработчиков нажатия кнопок и методов графических построений;
- методы запуска и остановки автоматического и ручного режима управления в модуле управления платформой;
- методы запуска функций автоматического управления;
- справочную информацию и подсказки по работе с системой.

Графический интерфейс
+экранная форма
+справочная информация
+подсказки
-методы-обработчики нажатия кнопок
-методы графических построений;
-методы запуска и остановки автоматического и ручного режима
-методы запуска функций автоматического управления;

Рисунок 27 – Модель модуля графического интерфейса

3.3.6 Проектирование взаимодействия функциональных модулей программного приложения

Общая схема взаимодействия модулей в программном приложении комплекса представлена в приложении В.

Модуль управления платформой управляется графическим интерфейсом пользователя с помощью вызовов управляющих методов. Интерфейс обрабатывает нажатия кнопок и запускает (останавливает) выполнение функций управления платформой в модуле управления платформой. Перед отправкой сообщений модуль управления подготавливает пакет данных с помощью специального метода и упаковывает их в объект класса «Пакет передаваемых данных» с помощью методов установки данных в нем. При приеме данных модуль управления разбирает принятый от платформы пакет на составляющие с помощью метода парсинга принятого пакета, упаковывает данные в объект класса «Пакет принимаемых данных» с помощью методов установки данных. При обработке данных, они берутся из пакетов с помощью методов получения данных.

Отправка и прием сообщений происходит через специальные функции приема-отправки в модуле связи.

Классы состояний модулей содержат в себе переменные состояний отдельных процессов и всей активности модуля (в каком режиме работы он находится). Установка и получение состояний происходит напрямую через объект этого класса, без использования методов получения и установки.

3.3.7 Проектирование интерфейса программного приложения

С помощью графического интерфейса пользователь может взаимодействовать с системой и управлять ей, поэтому к его разработке необходимо подойти со всей серьезностью. Анализируя требования к эргономике и технической эстетике программного продукта в части внешнего интерфейса пользователя можно выделить следующие основные аспекты: интерфейс должен быть лаконичным и интуитивно понятным пользователю, должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя, должен использоваться удобочитаемый шрифт, его размер должен быть достаточным для зрительного восприятия, цветовая палитра должна быть гармоничной и приятной для глаз, а элементы управления и уведомления должны быть достаточного размера для хорошего восприятия пользователем [14].

Для удобного расположения элементов управления и дополнительной информации в главном окне программы было решено разделить его на две области: для ручного управления платформой и для автоматического. При выборе нужного режима управления, соответствующая ему область должна активироваться, а в смежной области должны выводиться подсказки и справочная информация по работе в выбранном режиме.

В левом верхнем углу располагается выпадающий список, в котором должны быть режимы управления, которые может выбрать пользователь.

Область управления в ручном режиме должна содержать элементы управления в виде кнопок для управления направлением движения платформы, а также переключатель скорости движения. Рядом должна выводиться информация о состоянии платформы (текущая скорость движения и угол отклонения).

В области управления в автоматическом режиме находится поле рисования траектории движения платформы, а также кнопка выбора траекторий из сохраненных. Ниже располагаются кнопки выбора режимов автоматического управления: движение по нарисованной линии, движение по сохраненной траектории и запись траектории движения.

Макет главного окна программы представлен на рисунке 28.

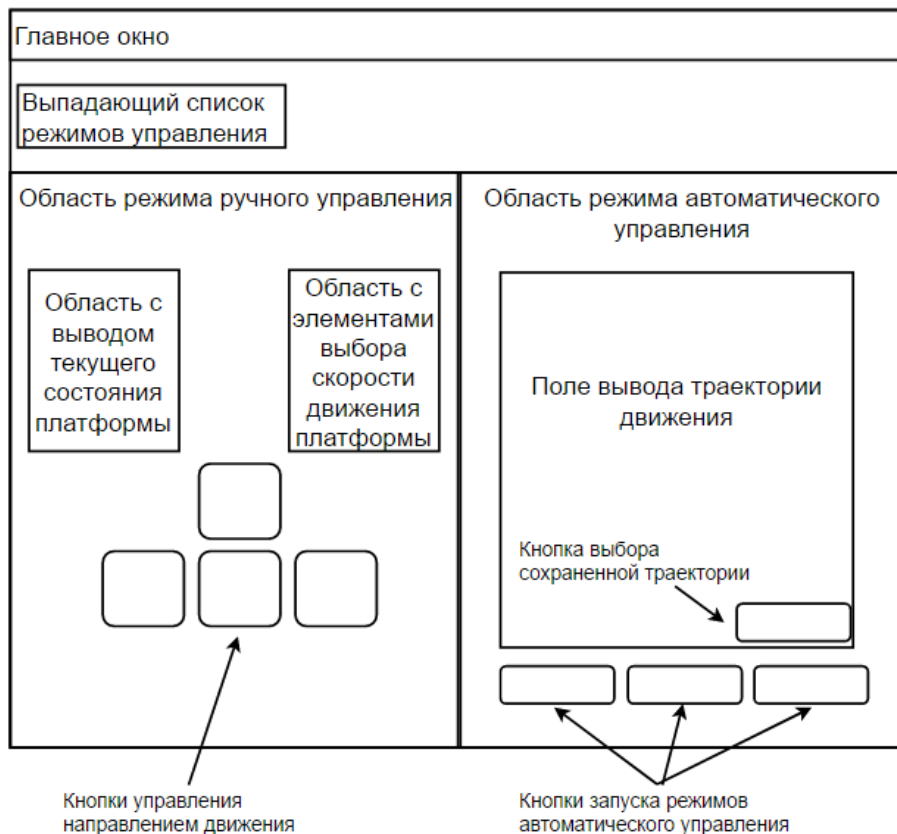


Рисунок 28 – Макет главного окна интерфейса программного приложения

3.4 Проектирование программного обеспечения для микроконтроллера аппаратной части комплекса

Прошивка Arduino по умолчанию включает в себя две основных функции setup и loop. Функция setup выполняется единожды при запуске микроконтроллера. Во время ее выполнения необходимо инициализировать все задействованные пины в соответствии с их назначением, инициализировать последовательное соединение с модулем Wi-Fi. После этого нужно выполнить подключение Wi-Fi модуля к сети, после чего необходимо подключиться к программному приложе-

нию комплекса, а именно к модулю связи. Как только подключение будет установлено, микроконтроллер в цикле начинает посылать приветственные сообщения программному приложению, с помощью которых платформу можно опознать и подключить к модулю управления.

При успешном выполнении этих операций микроконтроллер перейдет к выполнению функции loop, являющейся бесконечным циклом. В ней он принимает входящие сообщения от приложения с запросом на запуск автоматического или ручного режима управления платформой, сформированные модулем управления платформой.

Когда режим выбран, и платформа переведена в него, принимая команды на выполнение определенной функции платформы (движение по записанной траектории, запись траектории и т.д.) микроконтроллер запускает или останавливает циклы, содержащие все необходимые действия.

На рисунке 29 показана общая блок-схема прошивки для микроконтроллера аппаратной части комплекса.

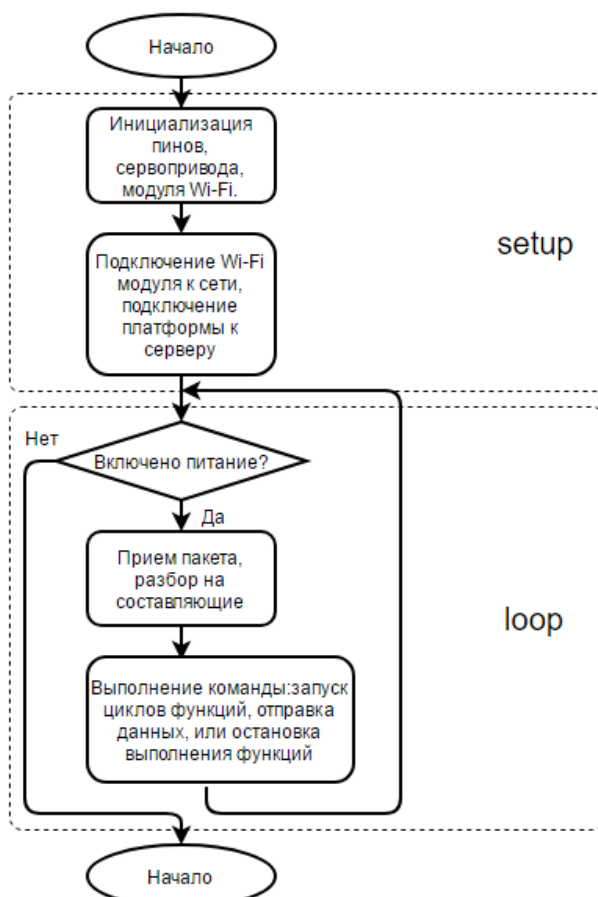


Рисунок 29 – Блок-схема прошивки микроконтроллера

3.4.1 Алгоритм выполнения функции движения по нарисованной линии

При получении платформой команды на выполнение функции движения по нарисованной линии, микроконтроллер переходит к выполнению следующей блок-схемы работы программы (рисунок 30).

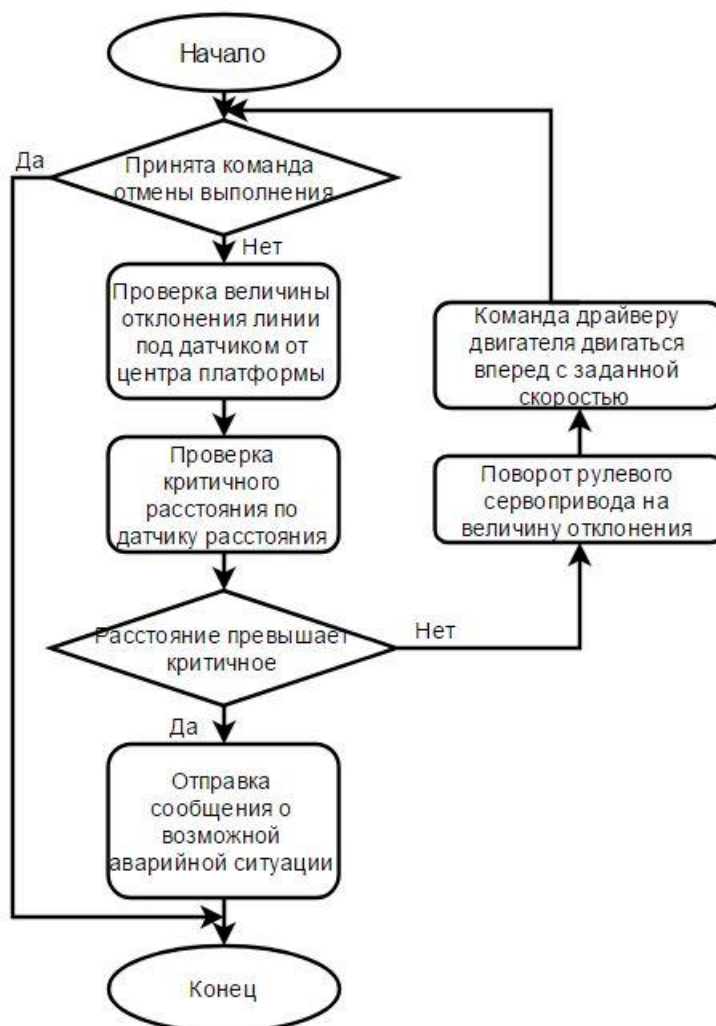


Рисунок 30 – Блок-схема алгоритма выполнения функции движения по нарисованной линии

В первую очередь микроконтроллер проверяет, пришла ли команда на отмену выполнения функции. После этого с датчика расстояния снимается величина отклонения линии движения от центра платформы и сверяется расстояние до ближайших объектов на пути следования платформы. Если это расстояние не превышает критичного (столкновения), то микроконтроллер поворачивает сервопривод на снятую с датчика линии величину отклонения и подает на логиче-

ские входы драйвера двигателя управляющий сигнал на движение вперед с заданной скоростью. Далее цикл вновь повторяется.

3.4.2 Алгоритм выполнения функции записи траектории движения

Если платформа получила команду на выполнение функции записи траектории движения, микроконтроллер переходит к выполнению следующего алгоритма, представленного на блок-схеме ниже (рисунок 31).

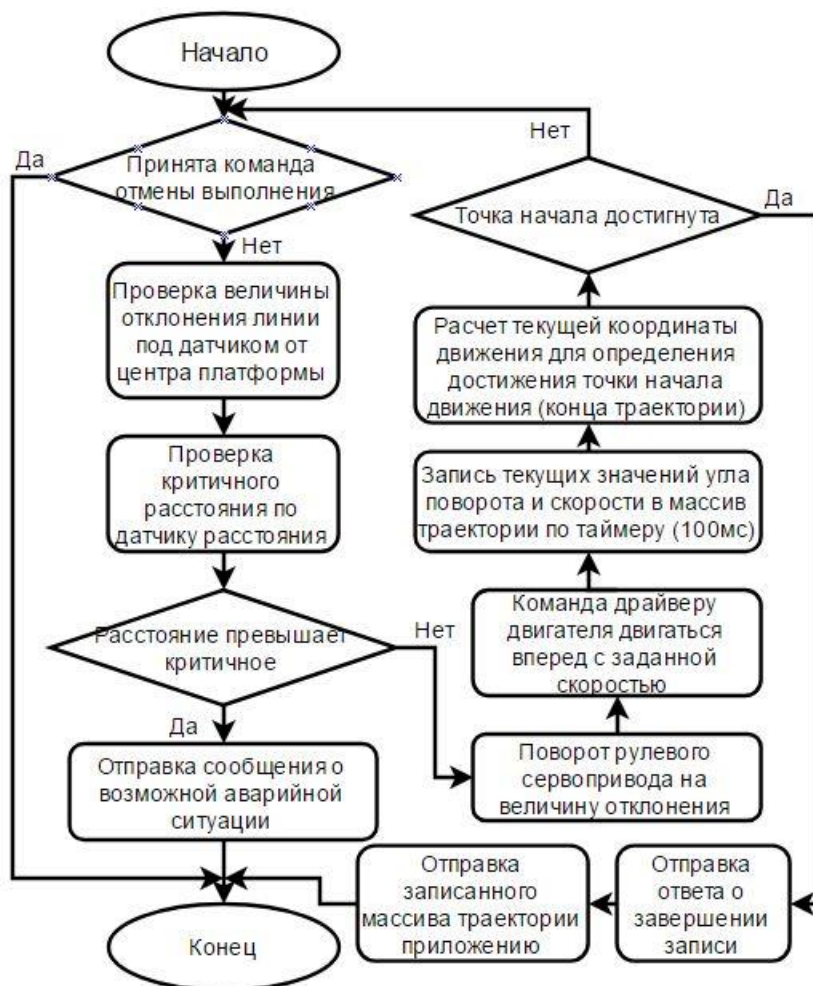


Рисунок 31 – Блок-схема алгоритма выполнения функции записи траектории движения

За основу работы этого алгоритма берется алгоритм движения платформы по нарисованной линии, но расширенный. В этом случае микроконтроллер также сначала проверяет, пришла ли команда на отмену выполнения функции. Если нет, то снимает отклонение с датчика линии, проверяет расстояние до ближайших объектов с помощью датчика расстояния, поворачивает сервопривод в соответствии с отклонением по датчику линии и дает команду драйверу двигателя

для движения платформы. После этого по таймеру в 100 мс значения угла отклонения сервопривода, направление и скорость движения платформы записываются в массив траектории. Вместе с этим выполняется отсчет координаты текущего движения от точки начала движения. Пока точка начала движения не будет достигнута, цикл будет выполняться. Но при ее достижении, микроконтроллер отправляет приложению ответ об окончании записи и пересылает массив траектории. На этом выполнение данной функции завершается.

3.4.3 Алгоритм выполнения функции движения по сохраненной траектории

При получении платформой команды на выполнения движения по сохраненной траектории микроконтроллер приступает к выполнению алгоритма, представленного блок-схемой на рисунке 32.

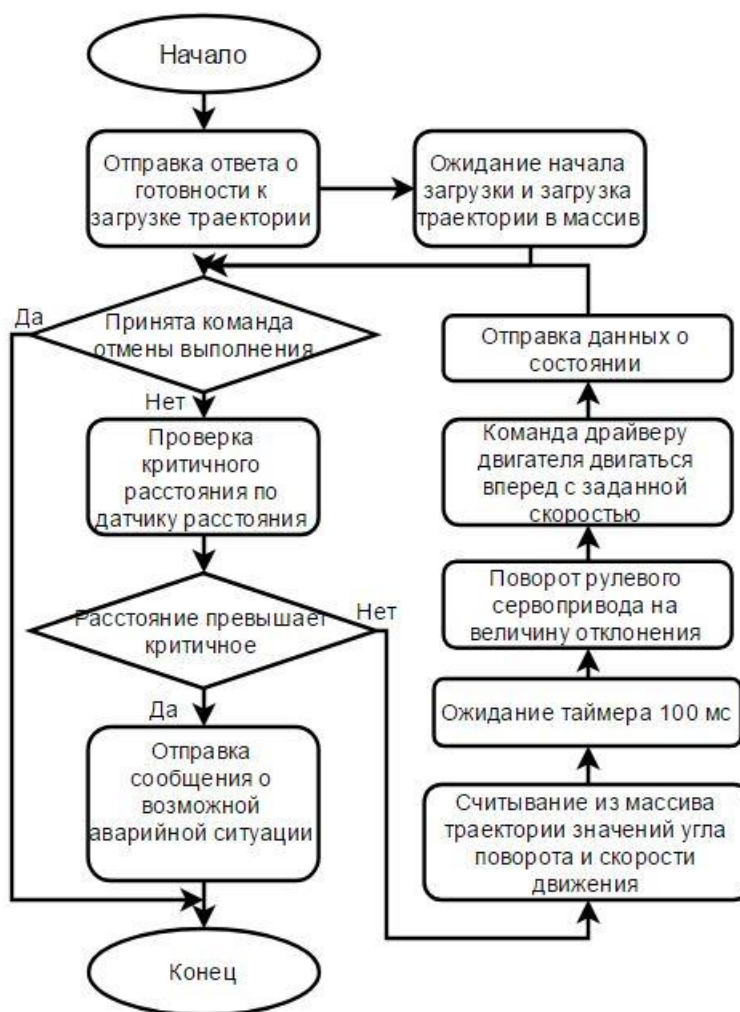


Рисунок 32 – Блок-схема алгоритма выполнения функции движения по сохраненной траектории

В самом начале микроконтроллер отправляет приложению на компьютере ответ о готовности к загрузке траектории движения, после чего происходит сама загрузка траектории в массив, из которого она будет впоследствии читаться. После того, как траектория была загружена, начинается цикл движения. Микроконтроллер проверяет, не пришла ли команда на отмену выполнения функции, проверяет расстояние до ближайших объектов с помощью датчика расстояния. Если расстояние не критично, то из массива траектории считываются значения угла поворота сервопривода и скорости движения. Далее микроконтроллер ждет по таймеру 100 мс и отправляет управляющие сигналы сервоприводу и драйверу двигателя. Ожидание нужно потому, что при записи значения записывались каждые 100 мс. Параллельно приложению отправляются данные о состоянии платформы. Если пришла команда о завершении, либо датчик расстояния зафиксировал опасное приближение к преграде, цикл движения завершается.

3.4.4 Алгоритм выполнения функции движения в ручном режиме управления

Алгоритм данной функции представлен блок-схемой на рисунке 33.

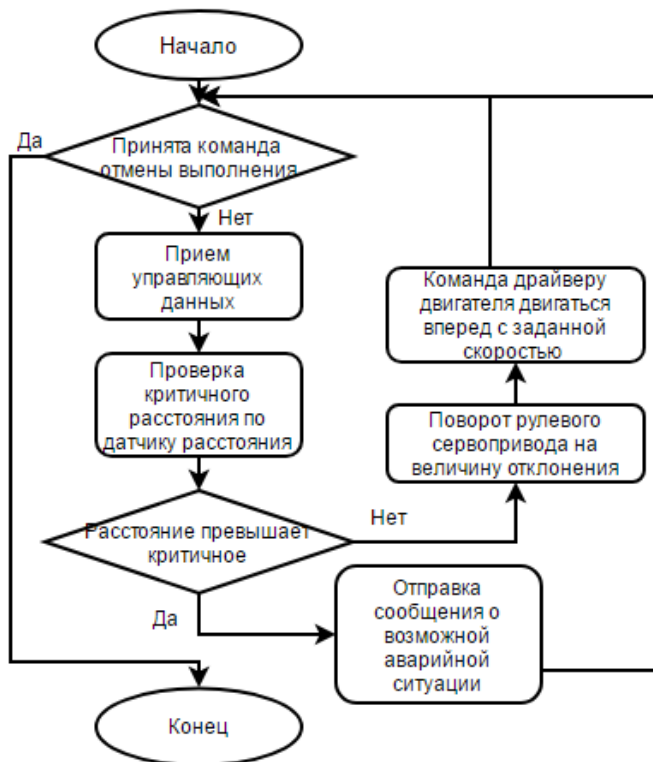


Рисунок 33 – Блок-схема алгоритма выполнения функции ручного управления движением

При активации ручного режима, микроконтроллер переходит к выполнению цикла ручного управления, проверяет, не пришла ли команда на завершение выполнения функции. После этого он принимает управляющие данные (угол поворота сервомотора, скорость и направление движения), проверяет расстояние до объектов по датчику расстояния и отдает управляющие сигналы сервоприводу и драйверу двигателя.

Если микроконтроллер получил команду завершения, то выполнение алгоритма прекращается.

3.5 Характеристика обеспечивающих подсистем комплекса

Выделяемая в соответствии со структурным подходом обеспечивающая часть разрабатываемой автоматической системы управления включает в себя: организационное, информационное, программное, техническое, лингвистическое и правовое [15].

3.5.1 Подсистема «Организационное обеспечение»

Организационным обеспечением называется совокупность средств и методов, регламентирующих взаимодействие работников с программными и техническими средствами, а также между собой в процессе разработки и эксплуатации программно-аппаратного комплекса.

Методы взаимодействия пользователей с соответствующей аппаратурой и программным обеспечением к ней описаны в руководстве пользователя и инструкции по эксплуатации.

3.5.2 Подсистема «Информационное обеспечение»

Информационное обеспечение - это совокупность методов и средств построения информационной базы для системы. Оно определяет способы и формы отображения состояния объекта управления в виде данных внутри информационной системы, документов, графиков и сигналов вне информационной системы. Информационное обеспечение подразделяют на:

– немашинное информационное обеспечение (правила классификации и кодирования, нормативно-справочная информация, оперативная информация, методические и инструктивные материалы);

– внутримашинное информационное обеспечение (входные сигналы и данные, промежуточные информационные массивы, выходные сигналы и документы).

Основными элементами информационного обеспечения разрабатываемого программно-аппаратного комплекса являются данные о траектории движения платформы, а также вводимые пользователем данные в графическом интерфейсе пользователя.

3.5.3 Подсистема «Программное обеспечение»

Программное обеспечение – это совокупность программ, которые предназначены для отладки, функционирования и проверки работоспособности автоматизированной системы. Разрабатываемый программно-аппаратный комплекс включает в себя лишь общее программное обеспечение.

В состав этой подсистемы входит операционная система Windows 7/8/8.1/10 от компании Microsoft.

Windows – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении. Спроектированный комплекс стабильно работает на всех представленных выше версиях этой ОС. Она содержит все необходимые элементы для реализации работы программного приложения комплекса.

Основным элементом реализации работы программного приложения комплекса является включенная в данные ОС платформа .NET Framework, основой которой является общезыковая среда исполнения Common Language Runtime (CLR), поддерживаемая разными языками программирования.

3.5.4 Подсистема «Техническое обеспечение»

Техническое обеспечение состоит из устройств измерения, преобразования, передачи, обработки, хранения, регистрации, отображения, ввода/вывода информации и исполнительных устройств.

К техническому обеспечению разрабатываемого программно-аппаратного комплекса относятся:

- стационарные компьютеры, являющиеся центральным звеном системы обработки данных;
- периферийные технические средства, обеспечивающие ввод и вывод информации (мыши, клавиатуры, мониторы);
- устройства передачи данных и сетевые коммуникации (локальная сеть, Wi-Fi роутер или адаптер).

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		63

4 РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

4.1 Разработка аппаратной части комплекса

В пункте 3.1.6 данной работы была спроектирована общая схема подключения компонентов аппаратной части комплекса. Для ее реализации были созданы печатные платы драйвера двигателя, блока питания и плата с обвязкой для ESP8266. Электронные компоненты запаяны на платы с использованием припоя ПОС-61. Вся аппаратная часть собрана согласно спроектированной схеме.

4.2 Обоснование выбора средств разработки

При переходе от проектирования к этапу разработки программного обеспечения комплекса необходимо выбрать нужные инструменты для реализации, предоставляющие все необходимые возможности.

4.2.1 Выбор языка программирования

Для реализации программного модуля системы был выбран объектно-ориентированный язык программирования C#. Разработан язык в 1988 – 2001 годах компанией Microsoft как язык разработки приложений для платформы Microsoft .NET Framework. C# относится к семье языков с C-подобным синтаксисом, наиболее близким к C++ и Java. У этих двух языков он перенял многие особенности, например, полиморфизм, наследование, перегрузку операторов, статическую типизацию. Его объектно-ориентированный подход позволяет строить крупные, но в то же время гибкие, расширяемые и масштабируемые программные продукты. Его синтаксис прост в изучении и очень выразителен. C# активно развивается, с каждой его новой версией появляется все больше функциональностей, как, например, асинхронные методы, динамическое связывания, лямбда-выражения и так далее [16].

Текущей версией языка является версия C# 7.0, вышедшая 7 марта 2017 года вместе с интегрированной средой разработки Visual Studio 2017.

C# очень тесно связан с платформой .NET. Данный фреймворк представляет собой мощную платформу для разработки приложений, основанной на общезыковой среде исполнения Common Language Runtime (CLR). К основным ее

чертам можно отнести:

– поддержка нескольких языков. Основой .NET является общезыковая среда исполнения Common Language Runtime (CLR), благодаря чему .NET поддерживает несколько языков: C#, VB.NET, C++, F#, Delphi.NET, а также различные диалекты других языков, связанные с .NET;

– кроссплатформенность. .NET является переносимой платформой (с некоторыми ограничениями). Последняя ее версия поддерживается на большинстве современных операционных системах семейства Windows (Windows 10/8.1/8/7/Vista). А благодаря проекту Mono, создаются приложения, работающие и на других ОС семейства Linux, в том числе на мобильных платформах Android и iOS;

– мощная библиотека классов. .NET представляет единую для всех поддерживаемых языков библиотеку классов, которая используется для разработки любых приложений;

– разнообразие технологий. Общезыковая среда исполнения CLR и базовая библиотека классов являются основой для целого стека технологий, используемых разработчиками при построении приложений. Например, для работы с базами данных предназначена технология ADO.NET, а для построения графических приложений с интерфейсом – WPF. Для создания веб-сайтов предназначена ASP.NET и т.д.

4.2.2 Выбор сред разработки

В качестве среды разработки была выбрана интегрированная среда Visual Studio Community 2017. Она предоставляет огромный набор удобных инструментов для разработки как консольных приложений, так и приложений с графическим интерфейсом. Visual Studio включает в себя редактор исходного кода с поддержкой технологии автодополнения IntelliSense, а также возможность простейшего рефакторинга кода. Встроенный отладчик среды работает как с исходным кодом, так и с машинным. В набор инструментов включены редактор форм для упрощения создания графического интерфейса, веб-редактор, дизайнер классов, дизайнер схем баз данных и так далее [18].

Для разработки программной прошивки Arduino использовалась официальная версия Arduino IDE 1.8.2. О данных средствах разработки подробно рассказано в пункте 3.1.2 этой работы.

Все перечисленные качества выбранных средств сыграли решающую роль в их выборе. Данный набор очень удобен для решения поставленных задач при разработке программного продукта.

4.3 Разработка программного приложения

4.3.1 Реализация связи программного приложения и платформы через сеть

В программном приложении для осуществления связи с платформой через локальную сеть [19] используется спроектированный в пункте 3.2.5 модуль связи, для которого реализован класс TCPserver (представлен на рисунке 34). Он предоставляет возможность запуска и остановки сервера для приложения, а также отправку и прием пакета данных. В нем содержатся следующие поля данных:

- `_hostPort` – хранит в себе порт, через который осуществляется подключение;
- `_hostAddr` – содержит IP-адрес компьютера в сети, на котором установлено приложение;
- `_tcpListener` – объект класса TCPListener, который слушает запросы клиентов на подключение к локальной точке доступа приложения;
- `_tcpClient` – объект класса TCPClient, который после подключения клиента принимает от слушателя `_tcpListener` управление доступом;
- `state` – объект пользовательского класса ServerState, специально разработанного для контроля за работой класса TCPServer.

Класс TCPServer содержит следующие методы:

- `StartServer()` – метод, который запускает слушатель входящих подключений для выбранного порта, а также асинхронно выполняет подключение клиентов (платформ);
- `DoAcceptTcpClientCallback()` – callback метод (метод обратного вызова), который запускается после приема слушателем `_tcpListener` нового входящего

подключения. В этом методе управление доступа от слушателя передается `_tcpClient`, через который в последствии и будет вестись передача и прием данных;

- `StopServer()` – выполняет закрытие всех соединений, подготовку к остановке и саму остановку сервера;
- `SendToClient()` – выполняет отправку данных клиенту (платформе);
- `RecieveFromClient()` – выполняет чтение из буфера данных, полученных от клиента (платформы).

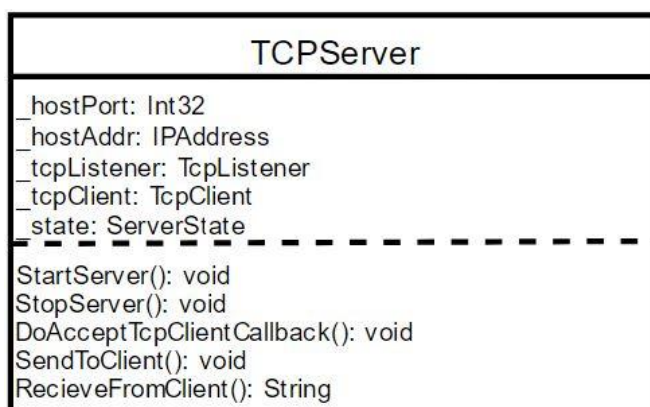


Рисунок 34 – Модель класса TCPServer

Для контроля за работой сервера создан класс состояния `ServerState` (рисунок 35), в котором содержится набор перечислений (тип `enum`) возможных состояний сервера:

- `StatusOfServer` – отображает режим работы сервера: запущен он или остановлен (`_SERVER_STOPED_`, `_SERVER_STARTED_`);
- `StatusOfConnection` – содержит состояние подключения клиента, подключен он или нет (`_NO_CONNECTION_`, `_CLIENT_CONNECTED_`).

В процессе работы устанавливается то или иное состояние, что позволяет отслеживать и контролировать выполняемые сервером операции.

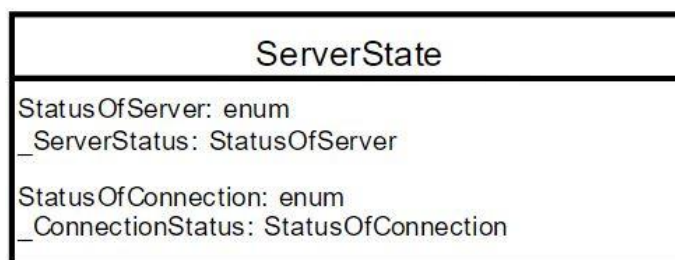


Рисунок 35 – Модель класса состояний ServerState

4.3.2 Реализация подмодуля управления платформой

Для непосредственного управления и работы с платформой служит модуль управления (пункт 3.2.5), на основе которого создан класс PlatformTools, который представлен на рисунке 36. Он предоставляет набор методов для подключения платформ, их опознания, работы системы в автоматическом режиме и в ручном. Для удобства класс разделен несколькими регионами, в которых собраны методы одного функционального назначения.

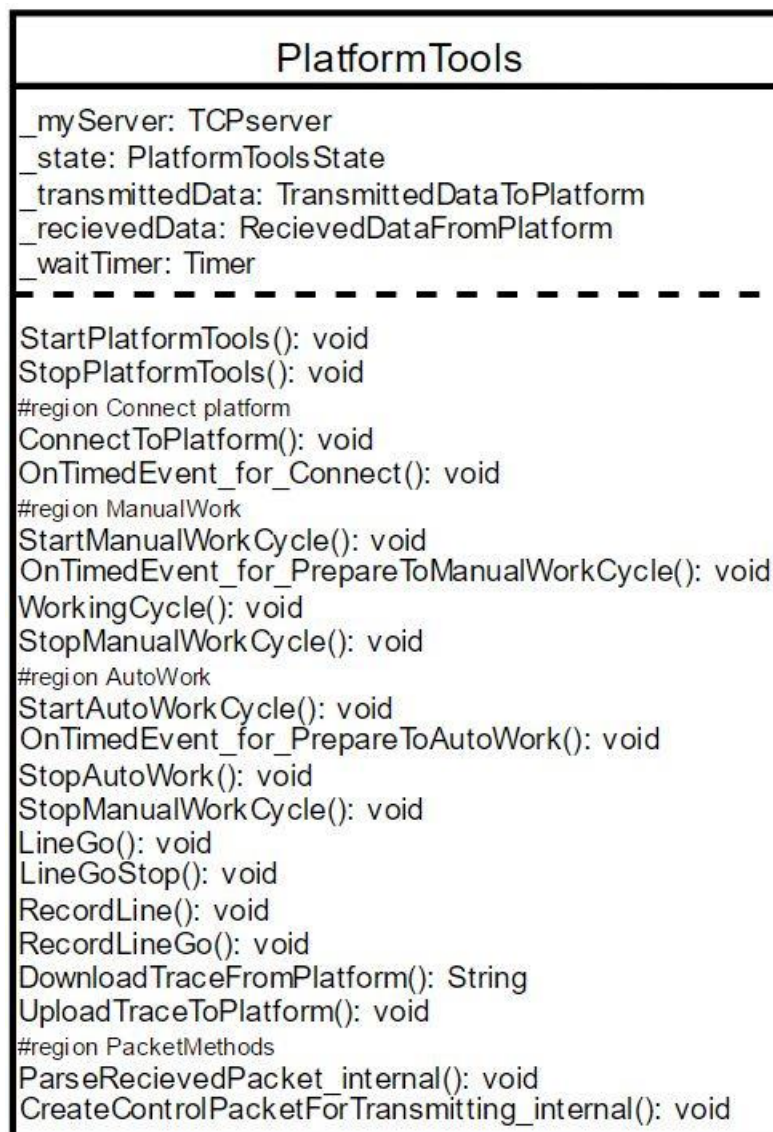


Рисунок 36 – Модель класса управления платформой PlatformTools

Поля данных класса PlatformTools:

- `_myServer` – объект класса TCPserver, который выполняет функции сервера;

Изм.	Лист	№ докум.	Подпись	Дата

- `_state` – объект класса состояний `PlatformToolsState`, который содержит данные о выполняемых операциях;
- `_transmittedData` – объект пользовательского класса `TransmittedDataToPlatform`, соержащий информацию о передаваемом платформе пакете данных;
- `_recievedData` – объект пользовательского класса `RecievedDataFromPlatform`, соержащий информацию о принятом от платформы пакете данных;
- `_waitTimer` – таймер, предназначенный для ожидания завершения операций.

Регион `Connect Platform` – регион, содержащий методы, осуществляющие подключение платформы и ее опознание:

- `ConnectToPlatform()` – метод подключения платформы, в котором запускается асинхронное ожидание (вызов функции `OnTimedEvent_for_Connect()` по таймеру `_waitTimer`) появления в сети доступных платформ (ожидание приветственных сообщений), а при их появлении происходит опознание и подключение;

- `OnTimedEvent_for_Connect()` – метод, который вызывается при каждом проходе таймера `_waitTimer`. Он проверяет, есть ли принятый сервером пакет данных и содержит ли он приветственное сообщение от платформы. Если это так, то таймер останавливается, подключение завершено, в `_state` записывается состояние подключения платформы.

Регион `ManualWork` – содержит методы для запуска и остановки режима ручного управления платформой:

- `StartManualWorkCycle ()` – метод, запускающий режим управления в ручном режиме и осуществляющий настройку платформы в этот режим. Платформе отправляется запрос на включение ручного режима, а после запускается таймер `_waitTimer`, при прохождении которого вызывается метод `OnTimedEvent_for_PrepareToAutoWork()`;

- `OnTimedEvent_for_PrepareToManualWorkCycle()` – метод, проверяющий ответ от платформы с готовностью к ручному режиму управления. Если получен

ответ, то асинхронно запускается метод WorkingCycle;

– WorkingCycle() – метод, содержащий цикл приема-передачи управляющих пакетов данных для платформы, которые управляют ее движением;

– StopManualWorkCycle() – метод, который останавливает цикл приема-передачи для ручного управления и отправляет платформе запрос на прекращение выполнения этого режима.

Регион AutoWork – содержит методы для запуска и остановки функций автоматического управления платформой:

– Метод StartAutoWork() – запускает режим автоматического управления, отправляя платформе запрос на включение данного режима. После по таймеру _waitTimer вызывается метод OnTimedEvent_for_PrepareToAutoWork();

– OnTimedEvent_for_PrepareToAutoWork() – метод, который проверяет ответ от платформы на готовность к управлению в автоматическом режиме;

– LineGo() – метод запуска функции движения платформы по нарисованной линии, путем отправки запроса на выполнение этой функции платформе;

– LineGoStop() – метод остановки режима движения по нарисованной линии, путем отправки запроса на прекращение выполнения этой функции;

– RecordLine() – метод активации режима записи платформой нарисованной линии;

– DownloadTraceFromPlatform() – метод для загрузки траектории движения с платформы, который вызывается после завершения процесса записи траектории платформой;

– UploadTraceToPlatform() – метод отправки платформе траектории, по которой необходимо двигаться;

– RecordLineGo() – метод, отправляющий платформе запрос на выполнение движения по загруженной в нее с помощью метода UploadTraceToPlatform() траектории;

– RecordLineGoStop() – метод остановки движения по загруженной траектории;

– StopAutoWork() – метод, завершающий автоматический режим управления и отправляющий платформе запрос на прекращение работы в данном режиме.

Регион PacketMethods – регион, содержащий методы работы с пакетами данных, описание которых приводится в пункте 3.3.5 данной работы.

Для контроля работы класса PlatformTools был создан класс PlatformToolsState, который представлен на рисунке 37. Он содержит следующие перечисления (тип enum) состояний:

– StatusOfPlatformTools – состояние объекта класса PlatformTools (_STARTED_, _STOPED_);

– StatusOfPlatformConnection – состояние подключения к платформе (_NO_CONNECTION_, _PLATFORM_CONNECTED_);

– StatusOfWorkCicle – указывает на состояние рабочего цикла ручного режима управления платформой (_CICLE_STARTED_, _CIKLE_STOPED_);

– StatusOfAutoWork – состояние циклов управления в автоматическом режиме управления (_AUTO_STARTED_, _AUTO_STOPED_, _LINE_GO_, _RECORD_LINE_, _RECORD_LINE_GO_).

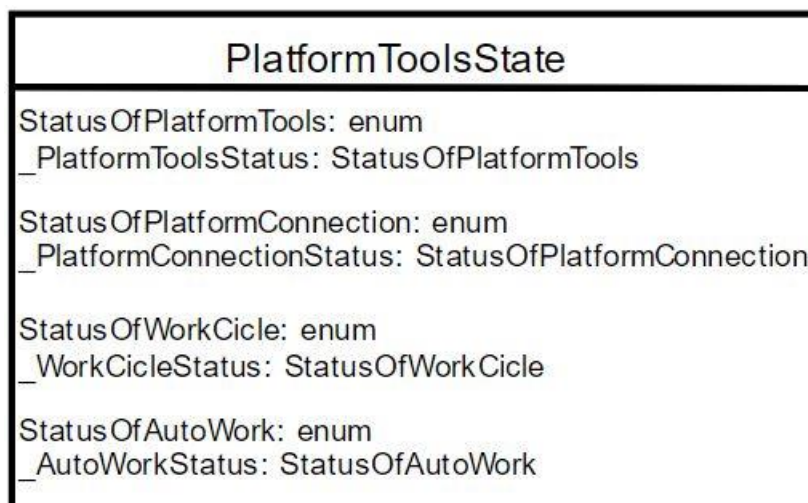


Рисунок 37 – Модель класса состояний PlatformToolsState

4.3.3 Реализация работы с пакетами передаваемых данных

Пакеты данных в системе представлены двумя классами RecievedDataFromPlatform (принятые данные от платформы, которые вносятся

из принятого пакета данных) и TransmittedDataToPlatform (передаваемые платформе данные, которые собираются в пакет из полей этого класса), показанные на рисунках 38 и 39 соответственно.

Объект класса RecievedDataFromPlatform используется для записи в него принятых от платформы данных в следующие имеющиеся в нем поля:

- `_platformID` – идентификационный номер платформы, передавшей пакет;
- `_requestID` – идентификатор запроса, который был отправлен платформе;
- `_answerToRequest` – ответ от платформы на отправленный ей запрос.

Также RecievedDataFromPlatform содержит следующие методы для работы с этими полями данных:

- `GetPlatformID()` – метод получения значения `_platformID` из объекта данного класса;
- `GetRequestID()` – метод получения значения `_requestID` из объекта данного класса;
- `GetAnswerToRequest()` – метод получения ответа `_answerToRequest` из объекта данного класса;
- `SetPlatformID()` – метод установки значения `_platformID` в объекте данного класса;
- `SetRequestID()` – метод установки значения `_requestID` в объекте данного класса;
- `SetAnswerToRequest()` – метод установки ответа `_answerToRequest` в объекте данного класса;
- `Reset()` – метод сброса всех значений полей.

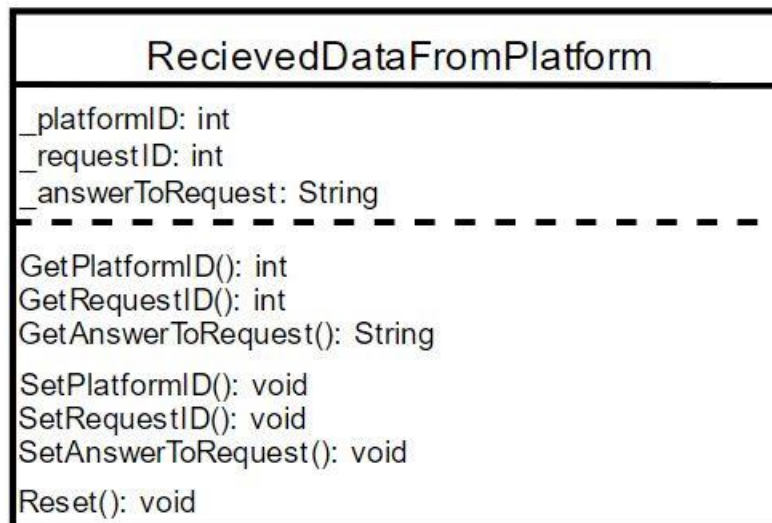


Рисунок 38 – Модель класса для принимаемых данных RecievedDataFromPlatform

Объект класса TransmittedDataToPlatform используется для записи в него передаваемых платформе данных в следующие имеющиеся в нем поля:

- `_platformID` – идентификационный номер платформы, передавшей пакет;
- `_requestID` – идентификатор запроса, который был отправлен платформе;
- `_platformSpeed` – скорость движения платформы;
- `_platformAngle` – угол поворота платформы;

Также TransmittedDataToPlatform содержит следующие методы для работы с этими полями данных:

- `GetPlatformID()` – метод получения значения `_platformID` из объекта данного класса;
- `GetRequestID()` – метод получения значения `_requestID` из объекта данного класса;
- `GetPlatformSpeed()` – метод получения значения `_platformSpeed` из объекта данного класса;
- `GetPlatformAngle()` – метод получения значения `_platformAngle` из объекта данного класса;

- SetPlatformID() – метод установки значения `_platformID` в объекте данного класса;
- SetRequestID() – метод установки значения `_requestID` в объекте данного класса;
- SetPlatformSpeed() – метод установки значения `_platformSpeed` в объекте данного класса;
- SetPlatformAngle – метод установки значения `_platformAngle` в объекте данного класса;
- Reset() – метод сброса всех значений полей.

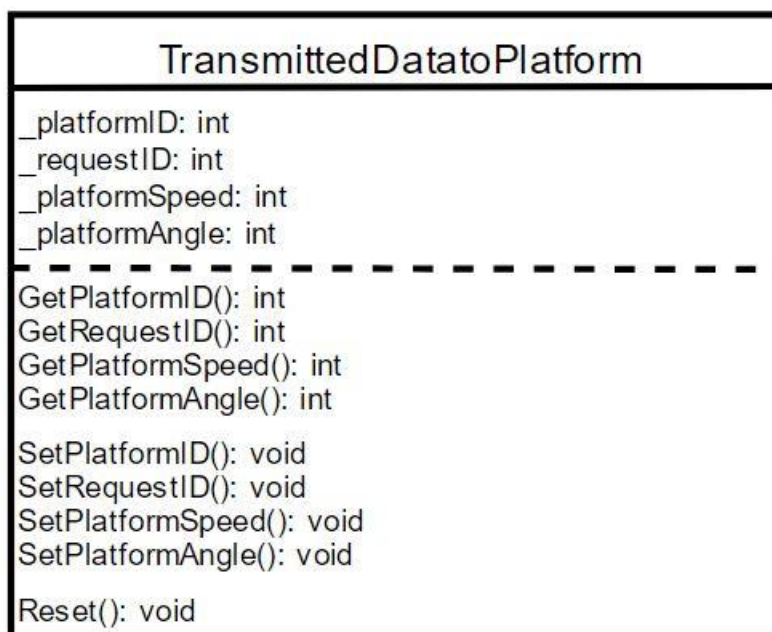


Рисунок 39 – Модель класса для передаваемых данных TransmittedDataToPlatform

Для работы с пакетами данных в классе PlatformTools (регион PacketMethods) реализованы два метода: метод парсинга данных принятого пакета (ParseRecievedPacket_internal()) и метод компоновки пакета, предназначенного к отправке (CreateControlPacketForTransmitting_internal()).

Метод парсинга данных в параметрах принимает строку данных, пришедшую от сервера, и разбирает ее по полям (id платформы, id команды, данные для команды) в специально созданный для принятых данных класс RecievedDataFromPlatform.

Метод компоновки пакета собирает данные для отправки из специально созданного класса TransmittedDataToPlatform. Он формирует строку данных для отправки в нужном формате.

4.3.4 Реализация графического интерфейса программного приложения

Графический интерфейс в программном приложении реализуется в классе GUI, который содержит в себе главное окно программы MainWindow, справочную информацию справка и файл подсказок tips.

К числу его рабочих методов относятся:

- методы обработки нажатий кнопок, собранные в регионе ButtonClickMethods;
- метод отрисовки траектории в поле траекторий DrawCan();
- метод запуска ручного режима управления ActivateManualMode();
- метод запуска автоматического режима управления ActivateAutoMode();
- методы запуска функций автоматического управления, собранные в регионе AutoFunction.

На рисунке 40 представлена модель этого класса.

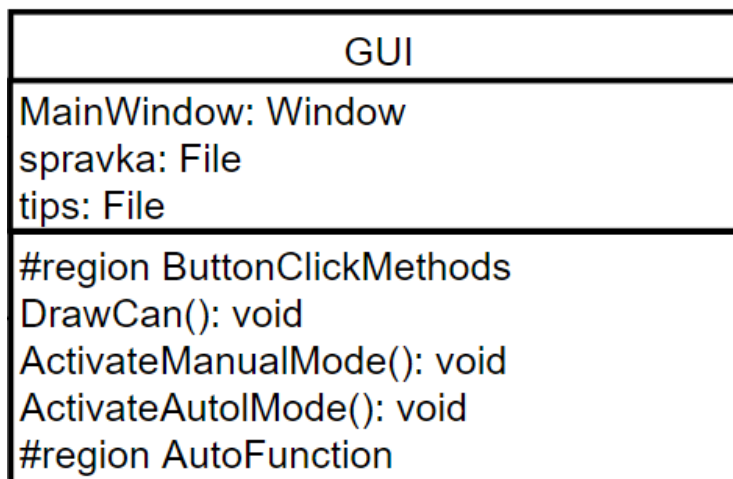


Рисунок 40 – Модель класса графического интерфейса пользователя

В пункте 3.3.7 в результате проектирования графического интерфейса пользователя был получен макет главного окна программы.

Интерфейс программы на основе спроектированного макета реализован с помощью системы для построения клиентских приложений Windows Presentation Foundation (WPF) на языке разметки для приложений XAML [17].

Код главного окна программы на языке XAML написан в файле MainWindow.xaml. В сопряженном управляющем файле MainWindow.xaml.cs находится код на языке C# для реализации взаимодействия интерфейса пользователя системы. В нем реализовано включение определенного режима управления, выбранного пользователем (с помощью выпадающего списка) через методы ActivateManualMode() и ActivateAutoMode(), а также описаны обработчики нажатия всех кнопок и переключателя скорости, методы для вывода траекторий в поле траекторий и отображения данных о состоянии платформы.

Для простоты управления пользовательским интерфейсом он был разделен на две области управления (в ручном и автоматическом режимах) с помощью контейнера Grid. Контейнер Grid является идеальным инструментом для разбиения окна на меньшие области, которыми можно управлять с помощью других панелей.

Далее все пользовательские элементы управления, такие как Button, Slider, CheckBox, Canvas, ComboBox, размещаются в контейнере в соответствии с функциональным назначением. Код разметки окна программного приложения также поделен регионами на два (разметка ручного управления и разметка автоматического управления) для удобства управления их отображением.

В левом верхнем углу программы с помощью выпадающего списка элемента ComboBox пользователь может выбрать режим управления.

В регионе разметки ручного управления находятся кнопки управления движением (стрелки), переключатель скорости и поле отображения угла поворота и скорости движения.

В регионе разметки автоматического управления находятся кнопки включения режимов движения платформы, кнопка выбора траектории движения, а также поле отображения траектории.

На этом построение графического интерфейса окончено, он представлен на рисунке 41.

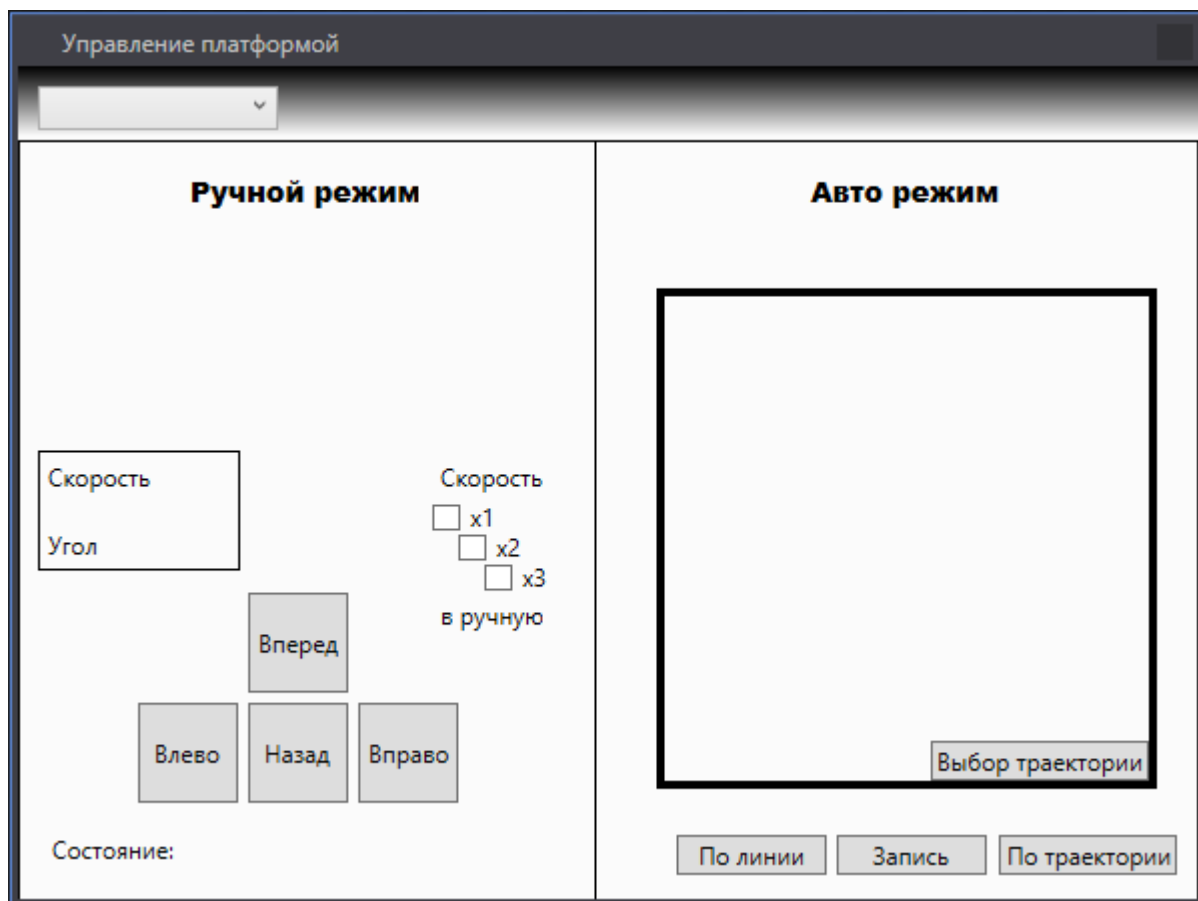


Рисунок 41 – Пользовательский графический интерфейс

На рисунке Д.1 и Д.2 приложения Д показаны экранные формы в автоматическом и ручном режимах управления.

При ручном управлении пользователь может управлять платформой с помощью кнопок со стрелками, меняя при этом скорость движения платформы выбором одного из трех флажков (X1, X2, X3) справа от них, либо ползунком, появляющимся при нажатии на «в ручную». Также в программе реализовано управление с помощью стрелок клавиатуры. Левее от кнопок управления показаны текущие угол поворота рулевого колеса и скорость платформы в процентах.

Автоматический режим позволяет пользователю дать платформе ряд указаний:

- двигаться по нарисованной линии;
- записать эту линию;
- двигаться по линии, выбранной из записанных;

С помощью кнопки «Выбор траектории» можно просматривать и выбирать

сохраненные траектории для загрузки на платформу. Кнопкой «По траектории» – активировать автоматический режим следования по выбранной траектории. В поле траекторий показывается текущая выбранная траектория, либо траектория в процессе записи. Кнопка «По линии» запускает режим следования по нарисованной линии.

В левом нижнем углу располагается поле состояния платформы, куда выводятся все ошибки, возникающие при работе системы.

4.4 Разработка программной прошивки для микроконтроллера аппаратной части комплекса

Язык программирования Arduino (основан на языке Wiring) является стандартным C++ (используется компилятор AVR-GCC) с некоторыми особенностями, облегчающими новичкам написание первой работающей программы. Программирование ведется целиком через собственную программную оболочку Arduino IDE (основана на среде Processing), бесплатно доступную с официального сайта Arduino. В этой оболочке имеется текстовый редактор, менеджер проектов, препроцессор, компилятор и инструменты для загрузки программы в микроконтроллер.

Программы, написанные программистом Arduino, называются наброски (или иногда скетчи – калька от англ. sketch) и сохраняются в файлах с расширением `.ino`. Эти файлы перед компиляцией обрабатываются препроцессором Arduino. Также существует возможность создавать и подключать к проекту стандартные файлы C++.

Обязательную в C++ функцию `main()` препроцессор создает сам, вставляя туда необходимые «черновые» действия. А программист должен написать две обязательные для Arduino функции `setup()` и `loop()`. Первая вызывается однократно при старте, вторая выполняется в бесконечном цикле все время работы контроллера.

В текст своей программы (скетча) программист не обязан вставлять заголовочные файлы используемых стандартных библиотек. Эти заголовочные файлы добавит препроцессор Arduino в соответствии с конфигурацией проекта.

Однако пользовательские библиотеки нужно указывать.

Менеджер проекта Arduino IDE имеет нестандартный механизм добавления библиотек. Библиотеки в виде исходных текстов на стандартном C++ добавляются в специальную папку в рабочем каталоге IDE. При этом название библиотеки добавляется в список библиотек в меню IDE. Программист отмечает нужные библиотеки, и они вносятся в список компиляции.

Arduino IDE не предлагает никаких настроек компилятора и минимизирует другие настройки, что упрощает начало работы для новичков и уменьшает риск возникновения проблем.

Закачка программы в микроконтроллер Arduino происходит через предварительно запрограммированный специальный загрузчик (bootloader). Загрузчик создан на основе Atmel AVR Application Note AN109 и может работать через интерфейсы RS-232, USB или Ethernet в зависимости от состава периферии конкретной процессорной платы.

Так как Wi-Fi модуль подключен к аппаратному UART с номером 1, то в коде программы для удобства заменим стандартное обозначение этого интерфейса (Serial1) на espSerial с помощью директивы #define.

Далее объявляются все необходимые переменные:

- char buffer[256] – буфер данных размером в 256 байт;
- String wifiSSID – строка, хранящая имя точки доступа Wi-Fi;
- String wifiPass – строка, хранящая пароль точки доступа Wi-Fi;
- String id – строка, хранящая идентификатор платформы;
- String ipAddr – строка, хранящая IP адрес компьютера, на котором установлено программное приложение;
- String port – строка, хранящая порт, через который идет подключение к серверу.

Для управления двигателем и сервомотором, а, следовательно, и движением платформы реализовано две функции:

- Motor() – функция передачи управляющих сигналов на пины, к которым подключен драйвер двигателя, для управления двигателем;

– Servo() – функция передачи управляющего сигнала на пин, к которому подключен рулевой сервопривод платформы.

Установочная функция setup() содержит в себе инициализацию необходимых пинов микроконтроллера, функцию подготовки платформы к работе FirstSetup().

Функция FirstSetup() с помощью AT-команд производит настройку Wi-Fi модуля: первоначально перезагружает его, устанавливает режим клиента и подключает к точке доступа. Далее происходит подключение аппаратной части к модулю TCPServer программного приложения. Когда подключение установлено, приложению отправляется приветственное сообщение с идентификатором платформы, а в ответ принимается ответ об успешной идентификации. Как только все эти операции выполнены и функция FirstSetup() отработала, происходит переход к основному циклу приема-передачи сообщений (находится в функции loop()), в котором платформа находится в режиме ожидания выбора режима управления.

Как только пользователь выбрал режим управления в выпадающем списке режимов в главном окне программного приложения, платформа (приняв пакет с соответствующей командой перехода в режим) переходит в цикл выбранного режима управления.

При каждом прохождении любого цикла происходит снятие показаний с датчика расстояния, и сравнение значения расстояния с критическим. Если оно меньше критического, то любой цикл останавливается с помощью оператора break, а приложению отправляется отчет о возникшей ошибке.

4.4.1 Реализация алгоритма функции ручного управления

Данный режим управления реализован циклом приема-передачи управляющих пакетов данный в функции HandleMode() программной прошивки микроконтроллера.

Перейдя к выполнению этой функции микроконтроллер отправляет пакет с ответом об успешном переходе в данный режим программному приложению, а после запускает бесконечный цикл while. При каждом проходе цикла с помощью

специальной AT-команды с буфера приема сообщений Wi-Fi модуля ESP8266 считывается принятый управляющий пакет. Если в нем не содержится команда завершения режима, то в соответствии со значениями скорости и угла поворота, принятых в этом пакете, функцией Motor() передается управляющие сигналы драйверу двигателя, а функцией Servo() – управляющие сигналы сервоприводу рулевого механизма. После этого приложению отправляется ответ об успешной обработке управляющего пакета.

Если же пакет содержал команду завершения, то цикл завершается с помощью оператора break, а платформа переходит в основной цикл ожидания выбора режима loop().

4.4.2 Реализация алгоритмов функций автоматического управления

Данный режим управления реализован циклом приема-передачи управляющих пакетов данный в функции AutoMode() программной прошивки микроконтроллера, который по своей сути является циклом ожидания выбора функции режима автоматического управления.

В этом цикле с помощью AT-команды чтения буфера принятых сообщений модуля ESP8266 считывается пакет с командой на выполнение функции автоматического управления и осуществляется вызов соответствующего цикла управления:

- LineGo() – функция, выполняющая цикл управления движением платформы по нарисованной линии;
- RecordLine() – функция, выполняющая цикл записи траектории движения по нарисованной линии;
- RecordLineGo() – функция, выполняющая цикл движения по сохраненной траектории.

Функция LineGo() в бесконечном цикле while производит считывание показаний с датчика линии, анализ отклонения от центра платформы и подачу управляющего сигнала на двигательные агрегаты с помощью функций Motor() и Servo().

Функция RecordLine() основана на функции LineGo(), но при это по счетчику таймера каждые 100 мс в буфер данных (buffer) производится запись текущих значений скорости движения платформы и угла поворота сервопривода. При этом параллельно идет отсчет от точки начала движения в переменной start-Stop(int). При достижении точки начала движения платформа отправляет записанную траекторию в буфере программному приложению. На этом выполнение цикла окончено и платформа переходит в режим ожидания в цикле функции AutoMode().

Функция RecordLineGo() начинает свое выполнение с приема от программного приложения пакета, содержащего траекторию движения. Она копируется в буфер данных, а после этого, считывая значения угла поворота и скорости движения, каждые 100 мс по счетчику таймера происходит передача управляющих сигналов двигательным агрегатам через функции Motor() и Servo().

4.5 Разработка руководства пользователя

4.5.1 Обоснование необходимости разработки руководства пользователя

При разработке программно-аппаратного комплекса следует помнить о конечных пользователях, которые не имеют представления о работе с данной программой и аппаратной частью, их функциональных возможностях и особенностях. Для ознакомления конечного пользователя с функциями программного продукта, а также основными сведениями о работе программно-аппаратного комплекса необходима разработка и написание руководства пользователя.

4.5.2 Определение содержания руководства пользователя

Наиболее важным аспектом при разработке и написании руководства пользователя является определение содержания. В зависимости от содержания и полноты информации о системе будут складываться знания о системе у конечного пользователя.

При написании руководства пользователя необходимо затронуть следующие темы, интересные для конечного пользователя:

- общие сведения о программном продукте;

- требования к компьютеру для оптимальной работы программного продукта;
- описание возможностей программы;
- правила эксплуатации программного продукта и аппаратной части комплекса.

4.5.3 Разработка руководства пользователя

При разработке руководства пользователя были учтены предлагаемые структуры по следующим стандартам:

- ГОСТ 19.101 – 77 ЕСПД Виды программ и программных документов;
- ГОСТ 19.503 – 79 ЕСПД Руководство системного программиста. Требования к содержанию и оформлению;
- ГОСТ 19.504 – 79 ЕСПД Руководство программиста. Требования к содержанию и оформлению;
- ГОСТ 19.505 – 79 ЕСПД Руководство оператора. Требования к содержанию и оформлению;
- ГОСТ 19.508 – 79 ЕСПД Руководство по техническому обслуживанию. Требования к содержанию и оформлению.

Разработанное руководство пользователя представлено в приложении Е.

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		83

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРИ ЭКСПЛУАТАЦИИ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ УПРАВЛЕНИЯ ДВИЖЕНИЕМ МОДЕЛИ ДВИЖУЩЕЙСЯ ПЛАТФОРМЫ ПОСРЕДСТВОМ СЕТИ WI-FI

5.1 Безопасность

5.1.1 Общие требования безопасности для операторов

Оператор во время работы обязан:

– соблюдать правила охраны труда, техники безопасности и производственной санитарии, промышленной, экологической и противопожарной безопасности, безопасных условий труда;

– немедленно докладывать непосредственному руководителю о всех происшествиях и несчастных случаях на производстве, нарушениях трудовой и технологической дисциплины;

– соблюдать осторожность и быть внимательным вблизи зон повышенной опасности (зон передвижения и маневрирования транспортных средств, погрузочно-разгрузочных работ и др.), а также на проезжей части дорог, обращать внимание на неровности и скользкие места на территории рабочей зоны;

– соблюдать осторожность при перемещении по территории, чтобы не споткнуться и не удариться о камни, строительный мусор и другие предметы, находящиеся в рабочей зоне;

– следует помнить, что в условиях повышенного уличного шума звуковые сигналы, подаваемые транспортными средствами, и шум работающего двигателя приближающегося автомобиля могут быть не слышны;

– нужно быть внимательным при передвижении возле низкорасположенных конструктивных частей здания, дорожно-строительных машин;

– при передвижении следует обращать внимание на неровности на поверхности земли и скользкие места, остерегаться падения из-за спотыкания или подскользывания.

Во время работы с системой на ПЭВМ оператору запрещается:

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		84

- прикасаться к задней панели системного блока при включенном питании;
- переключать разъемы кабелей устройств при включенном питании;
- загромождать верхние панели устройств бумагами и посторонними предметами;
- допускать захламленность рабочего места бумагой во избежание накопления органической пыли;
- допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств;
- включать сильноохлажденное (принесенное с улицы в зимнее время) оборудование;
- производить самостоятельно вскрытие и ремонт оборудования.

По окончании работы, специалист обязан:

- убрать рабочее место;
- доложить непосредственному руководителю о выполнении работ и о возникших в процессе работы неисправностях;

5.1.2 Требования к помещениям

Помещения, в которых находятся рабочее место оператора и аппаратная часть комплекса, должны соответствовать САНПИН 2.2.2/2.4.1340-03 [20].

В процессе эксплуатации разработанного программно-аппаратного комплекса работа с ПЭВМ является основной, так как оператору нужно контролировать работу системы и управлять ей. Поэтому при работе за компьютером предъявляются следующие требования к помещениям:

- не допускается расположение рабочих мест за мониторами для пользователей в подвальных помещениях;
- параметры микроклимата должны обеспечиваться для категории работ 1а и 1б в соответствии с действующими санитарно-эпидемиологическими нормами микроклимата производственных помещений; помещения с компьютерами

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		85

должны оборудоваться системами отопления, кондиционирования и вентиляцией воздуха;

– в помещении с компьютерами поверхность пола должна быть ровной, нескользкой, для удобной очистки и влажной уборки и обладать антистатическими свойствами;

– в помещении обязательно должны находиться средства пожаротушения (углекислотный огнетушитель для тушения пожара) и аптечка первой медицинской помощи [21].

5.1.3 Требования к микроклимату

Требования к микроклимату регулируются СанПиН 2.2.2/2.4.1340-03. В соответствии с этим документом на рабочих местах пользователей должны обеспечиваться оптимальные для работы параметры микроклимата:

– температура воздуха в холодный период года не должна превышать 22-24 °С, а в теплый период года – 20-25 °С;

– относительная влажность воздуха должна составлять 40-60 %;

– скорость движения воздуха – 0,1 м/с;

Для поддержания данных показателей в помещениях должна использоваться система отопления и кондиционирования воздуха.

Уровни шума для производственных помещений регулируются СанПиН 2.2.2/2.4.1340-03 и составляют не более 50 дБА. Посторонние шумы от компьютера и периферийной техники, а также вызванные работой аппаратной части комплекса, не превышают данных показателей. При наличии внешнего шума, помещения должны быть оборудованы шумоизоляцией, достигается уплотнение по периметру притворов окон и дверей.

5.1.5 Требования к освещению

Через оконные проемы должно обеспечиваться естественное освещение, коэффициент КЕО которого должен быть не ниже 1,5%. Световой поток должен падать с левой стороны на рабочее место.

Освещенность на поверхности стола в зоне размещения документа должна быть 300-500 лк. Допускается установка светильников местного освещения для

подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Следует ограничить прямую блескость от источников освещения. Яркость светящихся поверхностей (окна, светильники), находящихся в поле зрения, должна быть не более 200 кд/м².

Для искусственного освещения помещений с персональными компьютерами следует применять светильники типа ЛПОЗ6 с зеркалированными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами.

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

5.1.6 Требования к организации и оборудованию рабочих мест

Для рабочего места с персональным компьютером источник естественного света должен располагаться так, чтобы не мешать работе, желательно слева. Для исключения засветки экранов мониторов прямыми световыми потоками светильники общего освещения располагают сбоку от рабочего места, параллельно линии зрения оператора и стене с окнами. Также размещение светильников позволяет производить их последовательное включение в зависимости от величины естественной освещенности и исключает раздражение глаз чередующимися полосами света и тени, возникающее при поперечном расположении светильников.

Расстояния между персональными компьютерами должны учитывать ГОСТ 21958-76 «Система «Человек-машина». Зал и кабины операторов. Взаимное расположение рабочих мест. Общие эргономические требования». Расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м.

Рабочий стол не имеет конкретных стандартов, он лишь должен отвечать современным требованиям эргономики. На рабочей поверхности должны с лёгкостью разместиться всё необходимое оборудование, с учетом его количества, размеров и характера выполняемой работы. Большим плюсом будет использование на предприятии рабочего стола с регулируемой высотой поверхности. При

отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.

Рабочая поверхность стола не должна иметь острых углов и краев, иметь матовую или полуматовую текстуру. Допускаемая глубина поверхности рабочего места не менее 600 мм, а ширина не менее 1200 мм.

Пространство для ног должно быть длиной не менее 600 мм, шириной не менее 500 мм.

Клавиатуру, манипулятор «мышь» следует располагать в оптимальной зоне – части пространства рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с относительно неподвижным плечом. Эта зона составляет от 300 до 400 мм от точки опоры локтя оператора, не более.

Экран монитора должен размещаться на столе или на подставке так, чтобы расстояние наблюдения информации на его экране не превышало 700 мм, оптимальное расстояние от 450 до 500 мм.

Экран монитора по высоте должен быть расположен на столе или подставке так, чтобы угол между нормалью к центру экрана и горизонтальной линией взора составлял 20 градусов.

Угол наблюдения экрана монитора, а также других средств отображения в горизонтальной плоскости (угол разворота монитора относительно оператора) в общем случае не должен превышать 60 градусов.

Чтобы обеспечить сотруднику физиологически рациональную рабочую зону, создание условий для её изменения в течение дня рекомендуется применять подъемно-поворотные рабочие стулья с возможностью регулировки, высоты, регулировки в области спины, регулировки по наклону и расстоянию спинки от переднего края сиденья. Конструкция стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400-550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов;
- высоту опорной поверхности спинки 300 ± 20 мм, ширину – не менее 380

мм и радиус кривизны горизонтальной плоскости 400 мм;

- угол наклона спинки в вертикальной плоскости в пределах 0 ± 30 градусов;
- регулировку расстояния спинки от переднего края сидения в пределах 260-400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной 50-70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 ± 30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм;
- поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим, не электризующимся, воздухопроницаемым покрытием, легко очищаемым от загрязнения.

Рекомендованная и допустимая реализации рабочего места для эксплуатации разработанного программно-аппаратного комплекса представлена на рисунке 42, где 1 – это дверь, 2 – рабочие кресла, 3 – ПЭВМ, 4 – оконные проемы.

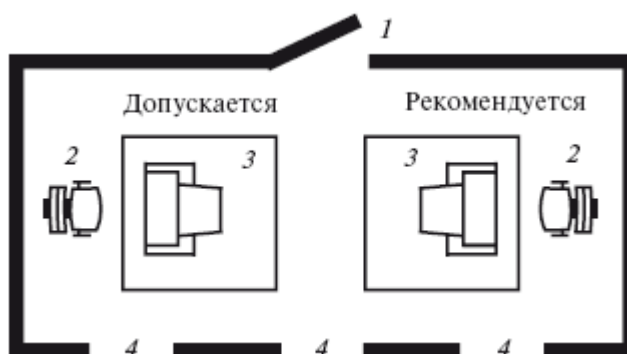


Рисунок 42 – Рекомендуемая и допустимая реализации рабочего места

5.1.7 Пожарная и электробезопасность на рабочем месте

Пожарная безопасность должна обеспечиваться системой предотвращения пожара и системой пожарной защиты. Во всех служебных помещениях обязательно должен быть «План эвакуации людей при пожаре», регламентирующий действия персонала в случае возникновения очага возгорания и указывающий места расположения пожарной техники.

В качестве электробезопасности перед началом работы следует убедиться

Изм.	Лист	№ докум.	Подпись	Дата

ВКР.135149.09.03.01.ПЗ

Лист

89

в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в отсутствии видимых повреждений аппаратуры и рабочей мебели.

Токи статического электричества, наведенные в процессе работы компьютера на корпусах монитора, системного блока и клавиатуры, могут приводить к разрядам при прикосновении к этим элементам. Такие разряды опасности для человека не представляют, но могут привести к выходу из строя компьютера. Для снижения величин токов статического электричества используются нейтрализаторы, местное и общее увлажнение воздуха, использование покрытия полов с антистатической пропиткой.

5.1.8 Эргономика программного интерфейса

Главное окно программы представлено на рисунке 43.

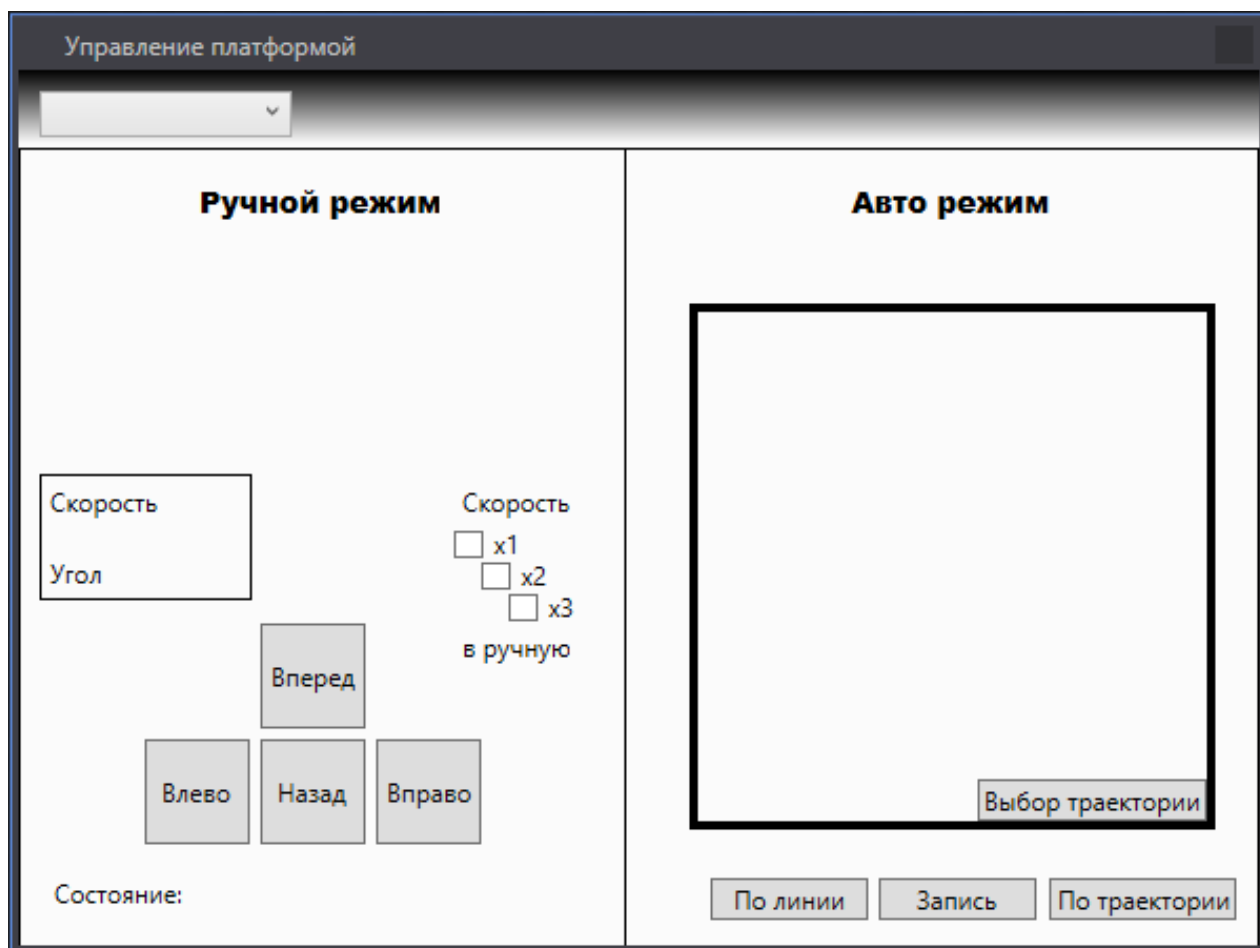


Рисунок 43 – Главное окно программы

Интерфейс разработанного программного приложения для ПЭВМ выполнен в нейтральных тонах, приятных для глаз. Дизайн разрабатывался в формате

Изм.	Лист	№ докум.	Подпись	Дата

близком к стилю стандартных приложений Windows. Было решено не использовать яркие цвета, или сильно контрастирующие, поэтому фон выбран белый, а элементы управления выполнены в серых тонах, что не напрягает зрительный аппарат в течении длительного времени. Размер шрифта подобран оптимальным для чтения и составляет 12 px, а гарнитура – Segoe UI.

5.1.9 Обеспечение безопасности работы системы

Во время движения платформы необходимо обеспечить предотвращение случаев аварийного столкновения с препятствиями на пути движения. Для этого в конструкции аппаратной части предусмотрен эхолокационный датчик расстояния, с помощью которого измеряется расстояние до ближайших объектов на пути следования, и, если оно близко к критическому (т.е. препятствие находится в опасной близости), то платформа прекращает движение.

Конструкция платформы и аппаратной части выполнена из материалов низкой степени пожароопасности, таких как металл и пластмасса. Напряжения питания и токи в цепях аппаратной части не настолько велики, чтобы привести к возгоранию или повреждению человека электрическим током.

Для зарядки аккумулятора используются серийно выпускаемые зарядные устройства, отвечающие всем требованиям безопасности.

5.2 Экологичность

Для обеспечения экологичности на предприятии существует Федеральный закон №89 «Об отходах производства и потребления» от 24.06.1998 (ред. от 28.12.2016). Данным законом регулируются способы утилизации отходов.

Утилизация оргтехники и компьютеров также должна производиться по правилам утилизации, так как в микросхемах техники содержатся детали, с определенной долей драгоценных металлов. Отдел, занимающийся эксплуатацией, обязан вести учет драгоценных металлов, находящихся в технике. Нарушение правил утилизации техники попадает под действие статьи 19.14 КоАП РФ [22].

5.3 Чрезвычайные ситуации

5.3.1 Аварийные ситуации

При работе могут возникнуть следующие аварийные ситуации:

- обрыв проводов питания;
- неисправность заземления;
- повреждение электрооборудования;
- повреждение инженерных коммуникаций;
- повреждение конструктивных элементов здания или помещения.

Во всех случаях обнаружения аварийной ситуации или появления резких ухудшений самочувствия, а также в любых других ситуациях, которые создают непосредственную угрозу жизни или здоровью людей, необходимо:

- остановить производство работ;
- при наличии пострадавших, обеспечить оказание первой помощи;
- при необходимости, обеспечить отключение электроэнергии, а также вывешивание запрещающего плаката «НЕ ВКЛЮЧАТЬ, РАБОТАЮТ ЛЮДИ!»;
- обеспечить открывание аварийных выходов и последующую эвакуацию персонала;
- доложить о принятых мерах руководителю работ и действовать в соответствии с полученными указаниями;
- доложить оперативному дежурному УЭТГСК, начальнику отдела.

Сотрудник, находящийся вблизи места происшествия, несчастного случая, должен оказать доврачебную помощь пострадавшему, доложить об этом оперативному дежурному УЭТГСК, начальнику отдела. При обнаружении человека, попавшего под напряжение, немедленно отключить электропитание и освободить его от действия тока [23].

5.3.2 Меры пожарной безопасности на рабочих местах

При расстановке технологического и другого оборудования должно быть обеспечено наличие свободных проходов к путям эвакуации и эвакуационным выходам.

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		92

Коробки вводов электродвигателей и аппаратов управления должны быть уплотнены и закрыты крышкой. Рубильники должны быть установлены так, чтобы они не смогли замкнуть цепь самопроизвольно под действием силы тяжести.

Электродвигатели должны иметь защиту от токов короткого замыкания и перегрузок. Электропривод должен немедленно отключаться от сети в случаях появления дыма или огня из электродвигателя или его аппарата управления, снижения скорости вращения, перегрева подшипников.

Для дополнительного освещения следует пользоваться переносными светильниками напряжением не более 50 В.

Использованные обтирочные материалы должны собираться в контейнеры из негорючего материала с закрывающейся крышкой и удаляться по окончании рабочей смены.

Специальная одежда лиц, работающих с маслами, лаками, красками и другими легковоспламеняющимися и горючими жидкостями, должна храниться в подвешенном виде в металлических шкафах, установленных в специально отведенных для этой цели местах.

Ответственность за пожарную безопасность рабочего места возлагается на должностных лиц в соответствии с приказом «О закреплении рабочих мест за ответственными должностными лицами и их единой нумерации».

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		93

ЗАКЛЮЧЕНИЕ

На сегодняшний день все робототехнические решения моделей передвигающихся роботов разрабатываются узкоспециализировано под конкретные задачи и поставляются вместе с шасси и управляющей аппаратной частью уже подобранной под имеющееся на данном шасси силовое оборудование (двигатели, сервоприводы). Следовательно, нет универсальных программно-аппаратных комплексов для автоматизации любой передвигающейся платформы (робота).

Именно поэтому в данной работе и была поставлена цель разработки как можно более универсального программно-аппаратного комплекса для управления движением различного рода платформ. Упор делался на применимость аппаратной части для управления любыми силовыми агрегатами, установленными на используемой модели платформы, при условии подбора модулей управления с подходящими характеристиками.

Для разработки системы подобного рода важно разбираться в сфере робототехники. Необходимы знания в микроэлектронике и электротехнике, механике, общее представление о рынке электронных компонентов на сегодняшний день, а также необходимо разбираться в передовых технологиях программирования и управления аппаратными платформами.

В результате анализа предметной области было выявлено, что на сегодняшний день подобного рода систем на рынке очень мало. Разрабатываемые комплексы либо узкоспециализированы под определенную платформу и задачу, либо наоборот обладают излишней аппаратной функциональностью, которая несомненно повышает стоимость их разработки. Разработка же комплекса в данной работе имеет малобюджетную направленность, при этом не теряя необходимого функционала системы и имея возможность расширения.

При проектировании комплекса были учтены все требования технического задания, а также построена основная логика взаимодействия модулей программного обеспечения и аппаратной части, с учетом входящих данных. Для каждого функционального модуля программного приложения были разработаны модели

классов, описывающие: его основной функционал; данные, которые хранит модуль и методы, которыми модуль оперирует.

При реализации проекта были выбраны наиболее развитые и передовые технологии, предоставляемые по открытой лицензии, что в свою очередь позволяет использовать разработанный продукт бесплатно и без установки дополнительного программного обеспечения. Одной из таких технологий для реализации аппаратной части была использована аппаратно-вычислительная платформа Arduino, являющаяся очень удобным конструктором для быстрого макетирования и создания электронных устройств. Спроектированные модели классов программного приложения были реализованы на языке C#. Пользовательский графический интерфейс был разработан с использованием системы построения WPF на языке XAML.

При рассмотрении раздела безопасность и экологичность, были проанализированы требования по безопасности к помещениям, организации рабочих мест и требования к эргономике программного обеспечения, рассмотрены аварийные ситуации и меры пожарной безопасности, знание которых необходимо в случае чрезвычайных ситуаций при эксплуатации разработанного программно-аппаратного комплекса.

Делая вывод, важно заметить, что разработанный комплекс хоть и имеет определенные недостатки, вызванные любительской направленностью использованных компонентов, но его разработка позволила пополнить свои знания и разработать функционально полную автоматизированную систему (в соответствии с поставленными требованиями), которая в будущем может быть доработана и усовершенствована благодаря модульной структуре и возможностям выбранных технологий программирования.

На данный момент разработанный комплекс внедрен в магазине автозапчастей ИП «Волкова Т.Г.» в качестве управления транспортировочной платформой в складском помещении и находится на стадии тестирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Сысолятин Ю.А. Проектирование программно-аппаратного комплекса для управления моделью движущейся платформы с помощью сети Wi-Fi // Электронный научный журнал «Постулат» [Электронный ресурс]. – Режим доступа: <http://e-postulat.ru/index.php/Postulat/article/view/259/274>.

2 Кондратьев К.Л., Харитонов В.И. Аппаратно-программный комплекс для управления платформой-роботом // Научная электронная библиотека КиберЛенинка [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/apparatno-programmnyu-kompleks-dlya-upravleniya-platformoy-robotom>. (Дата обращения 02.05.2017).

3 Майк Предко. Устройства управления роботами: схемотехника и программирование: Учебное пособие/ Майк Предко – М.: БХВ-Петербург, 2004. – 364 с.

4 Том Иго. Arduino, датчики и сети для связи устройств: Пер. с англ. – 2-е изд. – СПб.: БХВ-Петербург, 2015. – 544 с.

5 Arduino.cc Arduino [Электронный ресурс]. Режим доступа: <https://www.arduino.cc>. (Дата обращения 03.05.2017).

6 Arduino.ru Arduino Due (дополнено) [Электронный ресурс]. Режим доступа: <http://arduino.ru/Hardware/ArduinoDue>. (Дата обращения 07.05.2017).

7 Сообщество разработчиков ESP8266 [Электронный ресурс]. Режим доступа: <https://esp8266.ru>. (Дата обращения 07.05.2017).

8 Хабрахабр ESP8266: Что внутри «народного wi-fi»? [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/blog/238>. (Дата обращения 07.05.2017).

9 ESP-12E WiFi Module Version 1.0 [Электронный ресурс]. Режим доступа: <https://mintbox.in/media/esp-12e.pdf>. (Дата обращения 07.05.2017).

10 RAAAR.RU AT-команды [Электронный ресурс]. Режим доступа: http://www.raaar.ru/radio/AT_komand.html. (Дата обращения 13.05.2017).

11 Амосов, В.В. Схемотехника и средства проектирования цифровых устройств: Учебное пособие для студентов высших учебных заведений/ В.В.

ПРИЛОЖЕНИЕ А

Техническое задание

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование системы

1.1.1 Полное наименование системы:

Программно-аппаратный комплекс для управления движением модели движущейся платформы посредством Wi-Fi сети.

1.1.2 Краткое наименование системы:

ПАК для управления движением МДП посредством Wi-Fi сети.

1.2 Основания для проведения работ

– ГОСТ 34.602-89 – техническое задание на проектирование автоматизированной системы;

– требование к системе;

1.3 Наименование организаций – Заказчика и Исполнителя:

1.3.1 Исполнитель: студент Амурского государственного университета, факультета математики и информатики, Сысолятин Юрий Альбертович.

1.4 Плановые сроки начала и окончания работы:

Начало: Январь 2017 г.

Окончание: Июль 2017 г.

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

Разрабатываемый ПАК предназначен для управления движением некой моделью платформы в автоматическом и ручном режимах.

2.2 Цель создания системы:

Целью создания системы является разработка универсального программно-аппаратного комплекса для управления движением модели движущейся платформы.

3 ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ

Программно-аппаратный комплекс для управления движением платформы используется для легкой установки на необходимое шасси с двигательными агрегатами и последующего управления ими для реализации движения платформы в автоматическом и ручном режимах управления. После установки аппаратной части, платформа подключается к программному обеспечению на компьютере через локальную Wi-Fi сеть, опознается, и пользователь может приступать к управлению ей.

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		98

Продолжение ПРИЛОЖЕНИЯ А

4 ТРЕБОВАНИЯ К СИСТЕМЕ

4.1 Требования к системе в целом

4.1.1 Задачи системы:

- управление движением платформы в ручном режиме;
- управление движением платформы в автоматическом режиме;
- построение маршрута движения платформы с помощью датчика линии и его сохранение в локальном хранилище;
- предотвращение аварийных ситуаций при движении платформы с помощью датчика расстояния.

4.1.2 Требования к структуре и функционированию системы

Система должна включать в себя:

- аппаратную часть;
- программную часть

В состав аппаратной части входят:

- модуль Wi-Fi связи;
- модуль управления аппаратной частью (микроконтроллер);
- модуль управления силовыми двигательными агрегатами;
- датчики линии и расстояния до ближайших объектов.

Программное обеспечение системы представляет собой разработанное программное приложение для ПЭВМ и программную прошивку для микроконтроллера управляющего аппаратной частью комплекса.

Программное приложение для ПЭВМ включает в себя следующие модули:

- модуль связи с платформой;
- модуль управления аппаратной частью платформы;
- графический интерфейс пользователя.

Модуль связи с платформой должен реализовывать следующие функции:

- подключение модуля Wi-Fi аппаратной части к программному приложению;
- отправка сообщений модулю Wi-Fi аппаратной части;
- прием сообщений от модуля Wi-Fi аппаратной части.

Модуль управления аппаратной частью платформы должен реализовывать следующие функции:

- опознание платформы;

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		99

Продолжение ПРИЛОЖЕНИЯ А

- управление движением платформы в ручном режиме;
- управление движением платформы по нарисованной линии в автоматическом режиме;
- запись траектории движения платформы и ее сохранение в локальном хранилище;
- управление движением платформы по сохраненной траектории в автоматическом режиме;
- подготовка пакета с данными к отправке;
- разбор принятого пакета данных на составляющие и их обработка.

Модуль графического интерфейса пользователя должен реализовать следующие функции:

- обработка введенных пользователем данных;
- вывод информации о состоянии платформы и выполняемых функциях;
- вывод справочной информации и отчетов об ошибках.

4.1.3 Требования к модулям аппаратной части комплекса

Требования к модулям программной части представлены в таблице А.1.

Таблица А.1 – Требования к модулям аппаратной части комплекса

Модуль	Функции модуля	Задачи модуля
Модуль Wi-Fi связи	Связь с программным приложением через сеть Wi-Fi, передача сообщений.	Реализация клиента, подключаемого к серверу приложения и прием-передача через него сообщений.
Модуль управления аппаратной частью (микроконтроллер)	Управление движением платформы, сбор данных с датчиков.	Реализация управления всей аппаратной частью комплекса.
Модуль управления силовыми двигателями агрегатами	Управление моторами и сервоприводами.	Реализация управления моторами трансмиссии и сервоприводами рулевого механизма, которые установлены на платформе.
Датчик линии	Распознавание линии движения.	Реализация функции распознавания линии, по которой необходимо двигаться платформе.
Датчик расстояния	Определение расстояния до ближайших объектов для обеспечения безопасности движения платформы.	Реализация функции безопасного передвижения за счет вычисления расстояния до ближайших объектов.

4.1.4 Требования к модулям программной части комплекса

Требования к модулям программной части представлены в таблице А.2.

Продолжение ПРИЛОЖЕНИЯ А

Таблица А.2 – Требования к модулям программной части комплекса

Модуль	Функции модуля	Задачи модуля
Модуль связи с платформой	Связь с Wi-Fi модулем аппаратной части комплекса, передача сообщений	Реализация простого сервера, к которому подключается клиент (Wi-Fi модуль платформы), реализация передачи сообщений клиенту (Wi-Fi модулю платформы) и прием сообщений от него.
Модуль управления аппаратной частью платформы	Опознавание подключенной платформы, управление ее движением в ручном режиме, управление ее движением в автоматическом режиме	Реализация опознавания платформы по ID. Реализация возможности управления платформой в ручном режиме. Реализация возможности выполнения функций автоматического управления платформой.
Модуль графического интерфейса пользователя	Обработка введенных пользователем данных, вывод информации о состоянии платформы, ошибках и справочной информации, вывод траектории движения	Реализация ввода данных пользователем для управления платформой. Реализация вывода ошибок в работе системы. Реализация вывода справочной информации о работе с системой.

4.1.5 Требования к численности и квалификации персонала системы и режиму ее работы

Для работы с системой достаточно одного оператора, который должен будет запускать, настраивать и останавливать систему.

Требований к квалификации персонала не предъявляется.

4.1.6 Требования к надежности

Требования к надежности не предъявляются.

4.1.7 Требования к эргономике и технической эстетике

Программное приложение, устанавливаемое на ПК должно обеспечивать удобный для конечного пользователя интерфейс, отвечающий следующим требованиям:

В части внешнего оформления:

- интерфейс должен быть лаконичным и интуитивно понятным пользователю;
- должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя;
- должен использоваться удобочитаемый шрифт, его размер должен быть достаточным для зрительного восприятия.
- цветовая палитра должна быть гармоничной, приятной для глаз, не должна раздражать.

Продолжение ПРИЛОЖЕНИЯ А

– элементы управления и уведомления должны быть достаточного размера для хорошего восприятия пользователем.

В части диалога с пользователем:

– при возникновении ошибок в работе системы на экран монитора должно выводиться сообщение с наименованием ошибки и с рекомендациями по её устранению на русском языке.

– должно быть обеспечено обучение пользователя работе с системой с помощью подсказок.

4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы:

Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы не предъявляются.

4.1.9 Требования к защите информации от несанкционированного доступа

4.1.10.1 Требования к информационной безопасности:

В системе должна быть реализована подсистема ограничения доступа пользователей с помощью формы входа в систему. Она необходима для того, чтобы никто без ведома оператора не смог вмешаться в процесс работы всей системы. Ни один пользователь не может начать работу с системой, не идентифицировав себя и не предоставив системе аутентифицирующую информацию.

4.1.10.2 Требования к антивирусной защите

На ПК, использующем данную систему, должны быть установлены средства антивирусной защиты.

4.1.11 Требования по сохранности информации при авариях:

В системе должно обеспечиваться периодическое резервное копирование данных, чтобы в случае сбоев, она смогла восстановить свое предшествующее состояние и продолжить работу.

4.1.12 Требования по стандартизации и унификации:

Требования по стандартизации и унификации не предъявляются.

4.1.13 Дополнительные требования:

Дополнительные требования не предъявляются.

4.1.14 Требования к безопасности:

Требования к безопасности не предъявляются.

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		102

Продолжение ПРИЛОЖЕНИЯ А

4.2 Состав и содержание работ по созданию системы

Состав и содержание работ по созданию системы представлены в таблице А.3.

Таблица А.3 – состав и содержание работ по созданию системы

№ п/п	Стадии	Этапы работ
1.	Исследование и обоснование создания СПО	- обследование автоматизируемого объекта; - сбор сведений о зарубежных и отечественных аналогах; - разработка требований к системе.
2.	Техническое задание	Разработка технического задания на программно-аппаратный комплекс управления движением модели транспортировочной платформы.
3.	Эскизный проект	Разработка предварительных решений по программно-аппаратному комплексу управления движением модели транспортировочной платформы.
4.	Технический проект	- разработка структуры автоматизированной системы; - разработка окончательных функций автоматизированной системы; - разработка аппаратной части; - разработка программного обеспечения автоматизированной системы.
5.	Рабочая документация	Разработка программной и технической документации.
6.	Производство автономных испытаний ПАК	- выявление ошибок в ходе испытания программно-аппаратного комплекса; - исправление ошибок.
7.	Проведение комплексных испытаний ПАК	- отладка работы системы в складских помещениях «Кладовщик».

4.3 Требования к разрабатываемой документации.

Отчетная документация должна соответствовать требованиям ГОСТ РВ 15.110-2003.

Разработанное в ходе создания комплекса программное обеспечение предоставляется на электронном носителе.

Электронные копии предоставляемых документов исполняются в формате текстового редактора Microsoft Word 2007 или выше.

4.4 Внесение изменений в техническую документацию.

Настоящее ТЗ может при необходимости изменяться и уточняться.

Изм.	Лист	№ докум.	Подпись	Дата

ВКР.135149.09.03.01.ПЗ

Лист

103

ПРИЛОЖЕНИЕ Б

Функциональная схема модулей системы и их взаимодействия

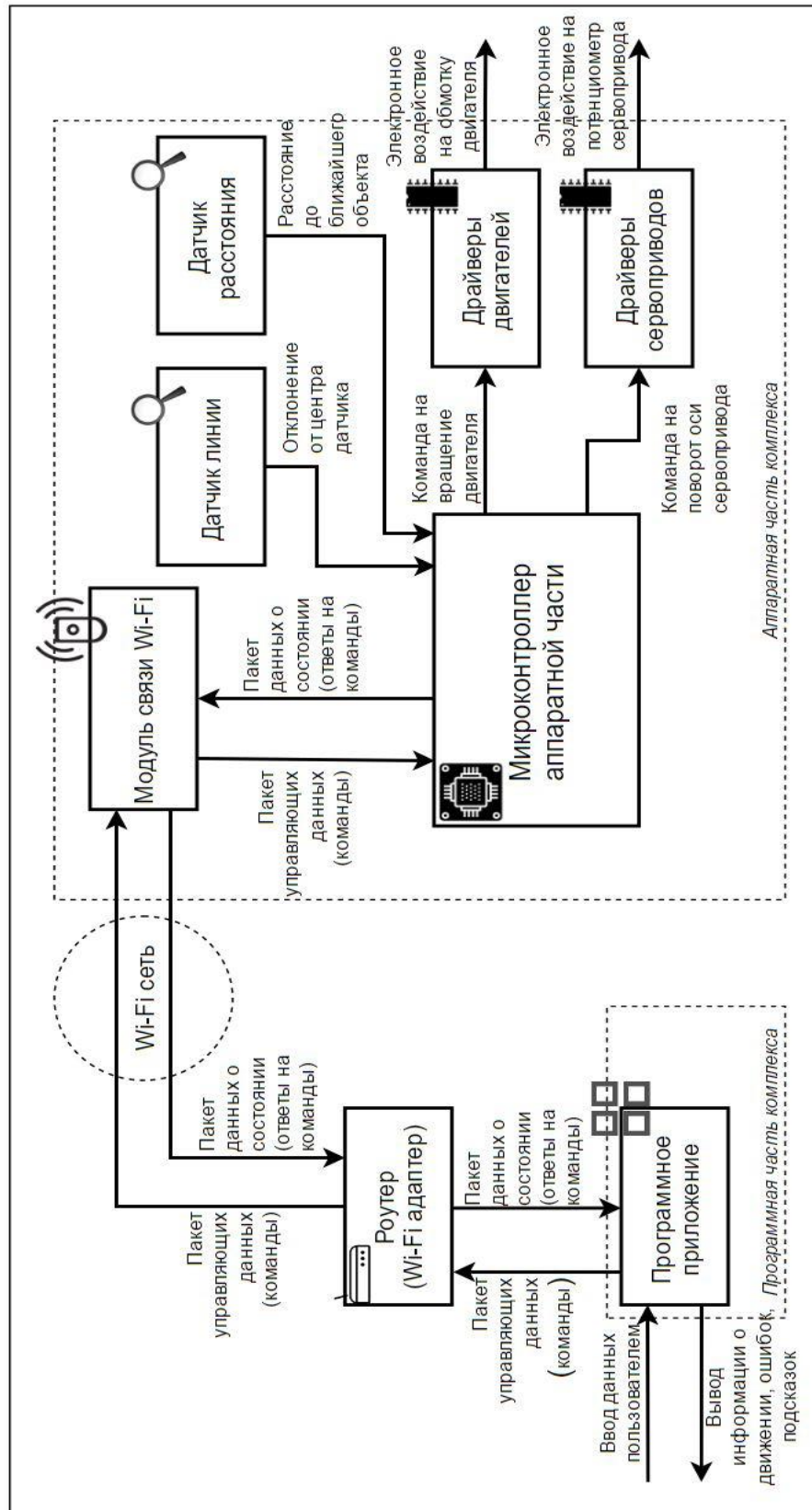


Рисунок Б.1 – Функциональная схема модулей системы и их взаимодействия

Изм.	Лист	№ докум.	Подпись	Дата

ПРИЛОЖЕНИЕ В

Функциональная схема взаимодействия модулей в программном приложении комплекса

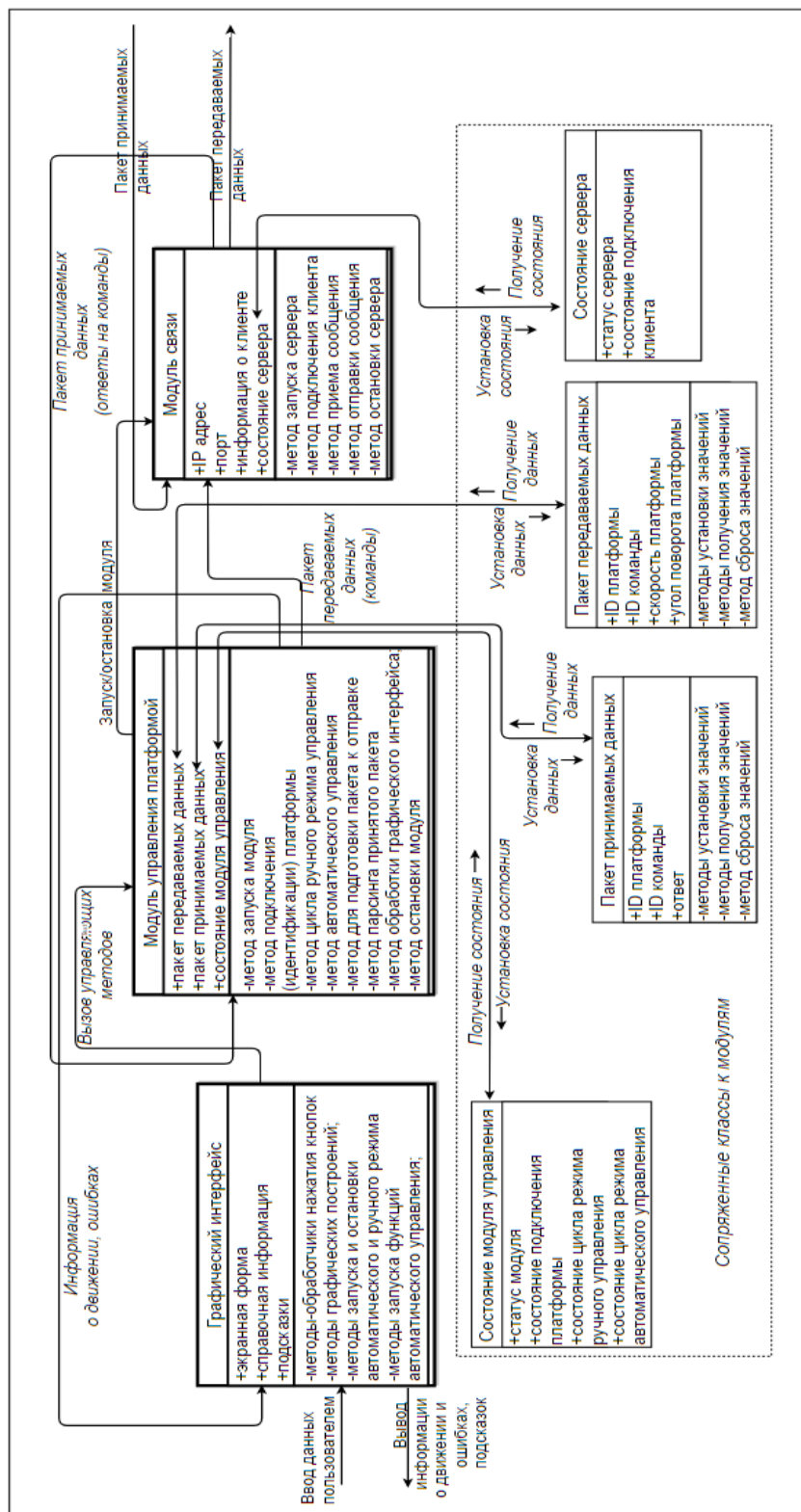


Рисунок В.1 – Функциональная схема взаимодействия модулей в программном приложении комплекса

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ВКР.135149.09.03.01.ПЗ

ПРИЛОЖЕНИЕ Г

Функциональная схема взаимодействия реализованных классов в программном приложении комплекса.

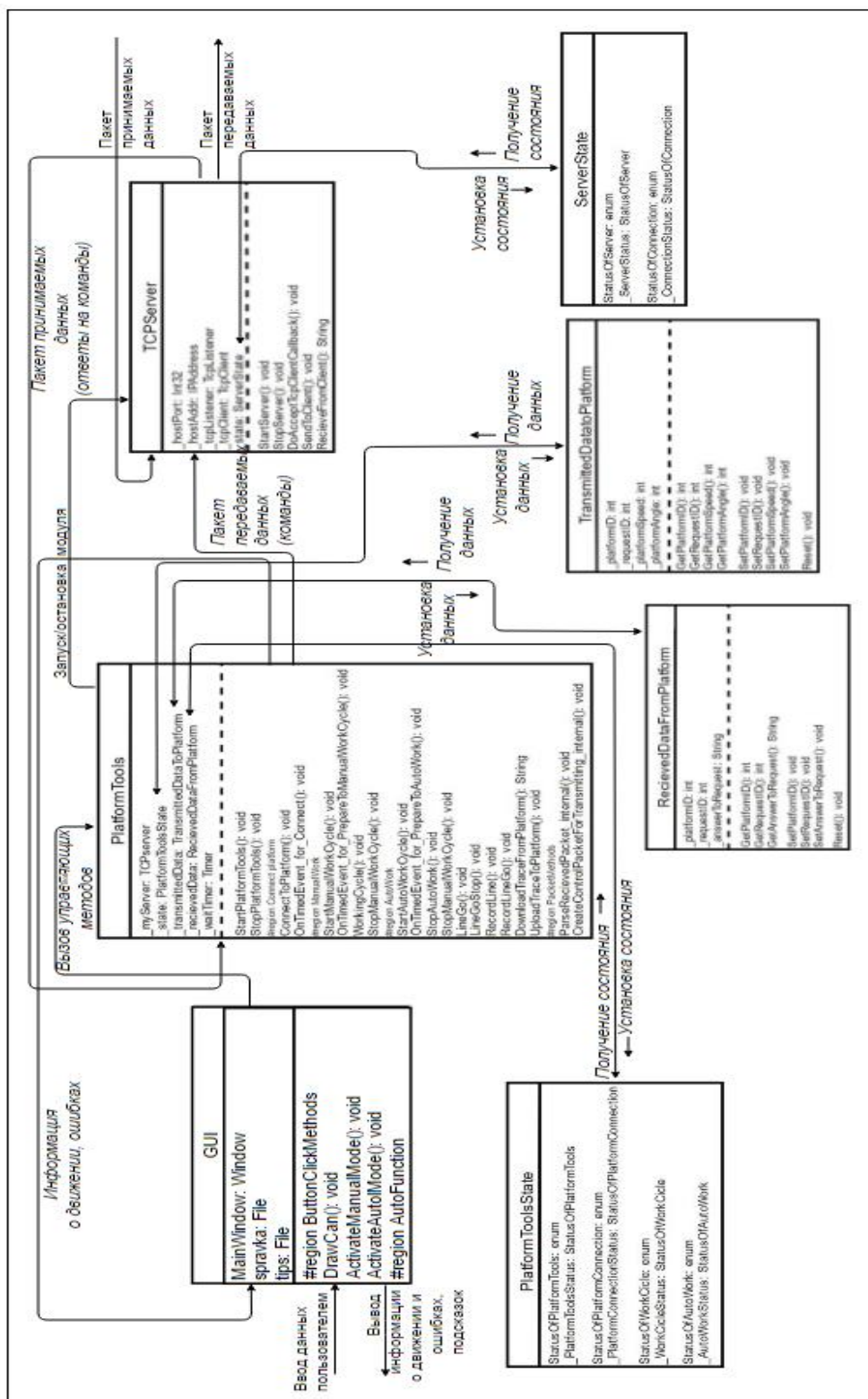


Рисунок Г.1 – Функциональная схема взаимодействия реализованных классов в программном приложении комплекса

Изм.	Лист	№ докум.	Подпись	Дата

ПРИЛОЖЕНИЕ Д

Экранные формы программного приложения

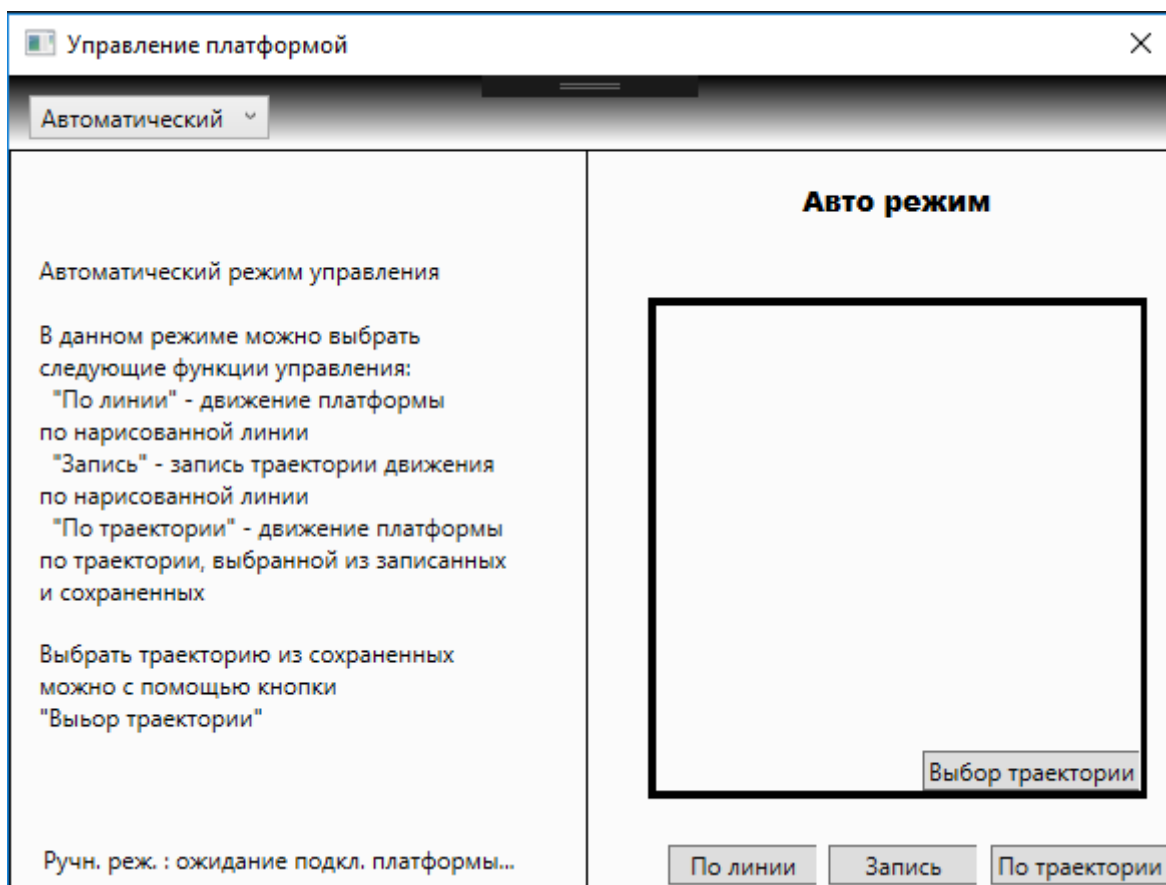


Рисунок Д.1 – Экранная форма приложения в автоматическом режиме управления

Изм.	Лист	№ докум.	Подпись	Дата

ВКР.135149.09.03.01.ПЗ

Лист

107

Продолжение ПРИЛОЖЕНИЯ Д

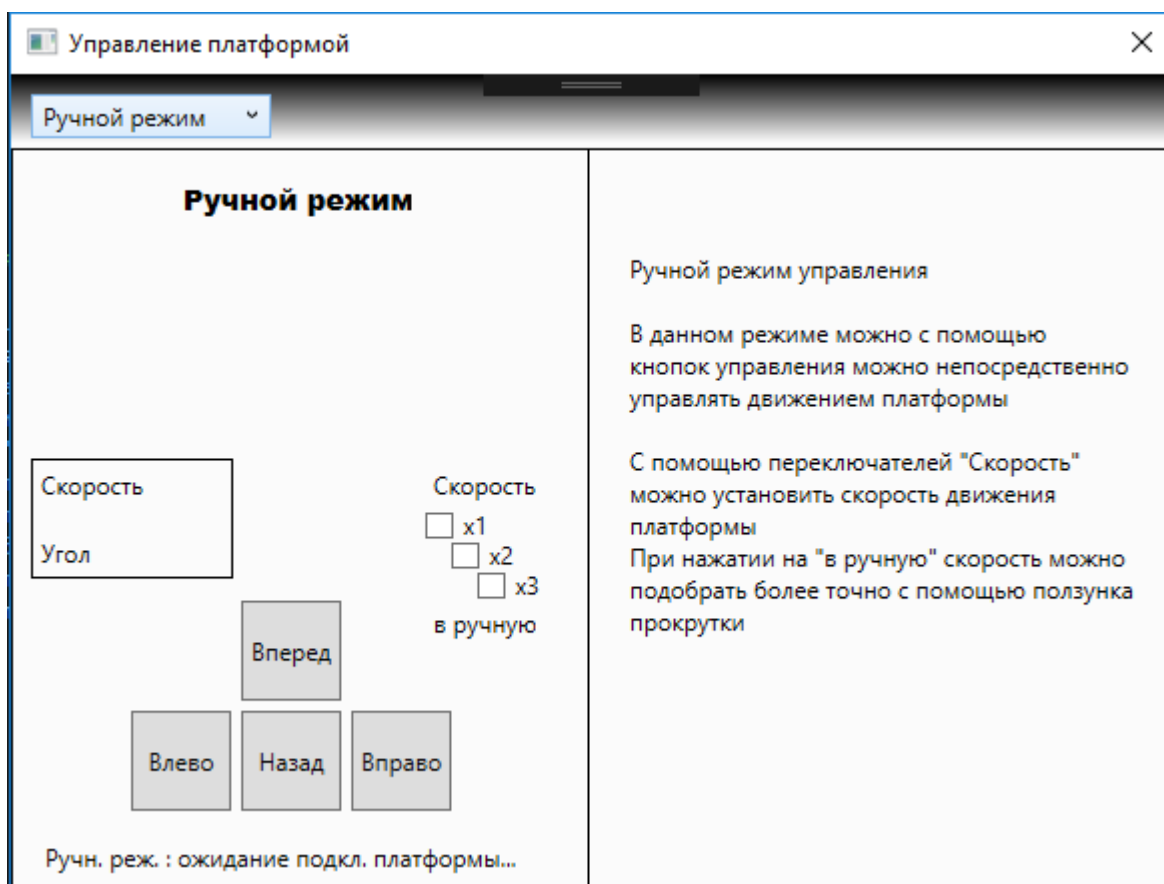


Рисунок Д.2 – Экранная форма приложения в ручном режиме управления

Изм.	Лист	№ докум.	Подпись	Дата

ВКР.135149.09.03.01.ПЗ

Лист

108

ПРИЛОЖЕНИЕ Е

Руководство пользователя

1 АННОТАЦИЯ

1.1 Назначение документа

Данное руководство пользователя предназначено для ознакомления с функциональными возможностями и требованиями к программно-аппаратному комплексу для управления движением модели движущейся платформы посредством сети Wi-Fi.

1.2 Краткое изложение основной части документа

В документе изложены общие сведения о разработанном комплексе. Содержится описание возможностей и функций, выполняемых комплексом, условия его применения и требования к техническому и программному обеспечению.

2 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

2.1 Обозначение и наименование системы

Полное наименование: «Программно-аппаратный комплекс для управления движением модели движущейся платформы посредством сети Wi-Fi».

Краткое наименование: «ПАК для управления движением МДП посредством Wi-Fi сети».

2.2 Языки программирования, на которых написано программное обеспечение сис

Программный код: С# (программное приложение), С++ (прошивка микроконтроллера)

Разметка интерфейса: XAML

2.3 Назначение системы

Программно-аппаратный комплекс предназначен для управления движением некой моделью платформы в автоматическом и ручном режимах.

2.4 Возможности системы

Комплекс разработан для выполнения следующих задач:

- управление движением платформы в ручном режиме;
- управление движением платформы в автоматическом режиме;
- построение маршрута движения платформы с помощью датчика линии и его сохранение в локальном хранилище;
- предотвращение аварийных ситуаций при движении платформы с помощью датчика расстояния.

					<i>ВКР.135149.09.03.01.ПЗ</i>	<i>Лист</i>
						109
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Продолжение ПРИЛОЖЕНИЯ Е

Аппаратная часть комплекса состоит из функциональных модулей, которые в соответствии возложенными на них задачами представлены в таблице Е.1.

Таблица Е.1 – Требования к модулям аппаратной части комплекса

Модуль	Функции модуля	Задачи модуля
Модуль Wi-Fi связи	Связь с программным приложением через сеть Wi-Fi, передача сообщений.	Реализация клиента, подключаемого к серверу приложения и прием-передача через него сообщений.
Модуль управления аппаратной частью (микроконтроллер)	Управление движением платформы, сбор данных с датчиков.	Реализация управления всей аппаратной частью комплекса.
Модуль управления силовыми двигателями агрегатами	Управление моторами и сервоприводами.	Реализация управления моторами трансмиссии и сервоприводами рулевого механизма, которые установлены на платформе.
Датчик линии	Распознавание линии движения.	Реализация функции распознавания линии, по которой необходимо двигаться платформе.
Датчик расстояния	Определение расстояния до ближайших объектов для обеспечения безопасности движения платформы.	Реализация функции безопасного передвижения за счет вычисления расстояния до ближайших объектов.

Программный продукт поделен на функциональные модули в соответствии возложенными на него задачами. Функции и задачи каждого модуля представлены в таблице Е.2.

Таблица Е.2 – Функции и задачи модулей программного продукта

Модуль	Функции модуля	Задачи модуля
Модуль связи с платформой	Связь с Wi-Fi модулем аппаратной части комплекса, передача сообщений	Реализация простого сервера, к которому подключается клиент (Wi-Fi модуль платформы), реализация передачи сообщений клиенту (Wi-Fi модулю платформы) и прием сообщений от него.
Модуль управления аппаратной частью платформы	Опознавание подключенной платформы, управление ее движением в ручном режиме, управление ее движением в автоматическом режиме	Реализация опознавания платформы по ID. Реализация возможности управления платформой в ручном режиме. Реализация возможности выполнения функций автоматического управления платформой.
Модуль графического интерфейса пользователя	Обработка введенных пользователем данных, вывод информации о состоянии платформы, ошибках и справочной информации, вывод траектории движения	Реализация ввода данных пользователем для управления платформой. Реализация вывода ошибок в работе системы. Реализация вывода справочной информации о работе с системой.

2.5 Ограничения области применения комплекса

Данный программно-аппаратный комплекс применяется только для автоматизации движения платформ, имеющих колесное шасси, рулевые механизмы на основе сервоприводов и двигательные агрегаты на основе трансмиссии с коллекторным двигателем.

3 УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

3.1 Сведения о технических и программных средствах, обеспечивающих выполнение программы

К техническим средствам, обеспечивающим правильную работу комплекса, относят:

- ЭВМ;
- сетевое Wi-Fi оборудование.

К программным средствам относят необходимое для правильной работы программы программное обеспечение, такое как:

- операционная система;
- дополнительное программное обеспечение.

3.2 Требования к техническим средствам

3.2.1 Требования к ЭВМ

Требования, предъявляемые к ЭВМ, представлены в таблице Е.3.

Таблица Е.3 – Требования к ЭВМ

Характеристика ЭВМ	Рекомендуемое значение
Процессор	1 ГГц или выше.
ОЗУ	2 ГБ
Видеокарта	AMD Radeon 7680 512 Мб и выше
Место на жестком диске	От 512 МБ

3.3 Программное обеспечение, необходимое для функционирования системы

На ЭВМ должно быть установлено следующее программное обеспечение:

- операционная система Windows 7 и выше.

ПРИЛОЖЕНИЕ Ж

Справка о внедрении комплекса на предприятие


СПРАВКА

о результатах внедрения решений,
разработанных в бакалаврской работе студентом
ФГБОУ ВО «Амурского государственного университета»
Сысолятиным Юрием Альбертовичем

В работе над бакалаврской работой по теме «Разработка программно-аппаратного комплекса для управления движением модели движущейся платформы посредством сети Wi-Fi» студент группы 353об факультета Математики и Информатики, направления 09.03.01 – Информатика и вычислительная техника, Сысолятин Юрий Альбертович принял непосредственное участие в разработке программно-аппаратного комплекса для управления движением модели движущейся платформы посредством сети Wi-Fi.

В настоящее время разработанная система, являющаяся результатом выполнения ВКР, находится на стадии внедрения и тестирования на предприятии ИП «Волкова Т.Г.» в качестве управления транспортной платформой на складе магазина автозапчастей.

Руководитель
индивидуальный предприниматель
Волкова Татьяна Георгиевна



(подпись)



Рисунок Ж.1 – Справка о внедрении

Изм.	Лист	№ докум.	Подпись	Дата

ВКР.135149.09.03.01.ПЗ

Лист

112