

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименование института полностью)

Кафедра Промышленная электроника
(наименование)

ЗАДАНИЕ
на выполнение бакалаврской работы

Студент Пименов Евгений Евгеньевич, Элб-1601а

1. Тема Автоматизированная мобильная платформа

2. Срок сдачи студентом законченной бакалаврской работы _____

3. Исходные данные к бакалаврской работе Робототехнический набор TETRIX MAX с робототехническим контроллером TETRIX PRIZM. Движение по черной линии по маршруту; проезд плавных перекрестков; остановка перед препятствием.

4. Содержание бакалаврской работы (перечень подлежащих разработке вопросов, разделов)

Аннотация

Введение

1. Состояние вопроса

1.1 Формулирование актуальности, цели и задач проекта

1.2. Анализ исходных данных и известных решений

2. Аппаратная часть

2.1. Разработка структурной схемы

2.2. Выбор необходимых компонентов

2.3. Разработка электрической принципиальной схемы

3. Программная часть

3.1 Разработка алгоритма работы

3.2 Разработка программной части устройства

4. Конструкторско-экспериментальный раздел

4.1 Сборка устройства

4.2 Отладка устройства и экспериментальные исследования

Заключение

Список используемой литературы

5. Ориентировочный перечень графического и иллюстративного материала

1. Обзорный лист

2. Структурная схема устройства

3. Схема электрическая принципиальная

4. Блок-схема алгоритма программы

5. Плакат, иллюстрирующий работу платформы

6. Вид готовой платформы

6. Консультант по разделам _____

7. Дата выдачи задания «28» февраля 2020 г.

Руководитель бакалаврской работы

(подпись)

М.В. Позднов

(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименование института полностью)

Кафедра Промышленная электроника
(наименование)

КАЛЕНДАРНЫЙ ПЛАН
выполнения бакалаврской работы

Студент Е.Е. Пименов
по теме Автоматизированная мобильная платформа

Наименование работ	Плановый срок выполнения	Фактический срок выполнения	Отметка о выполнении	Подпись руководителя / консультанта
Анализ актуальности проекта, написание введения и обзорной главы. Разработка структурной схемы разрабатываемого устройства	20.02.2020	20.02.2020		
Разработка электрической принципиальной схемы . Написание остальных разделов работы.	30.04.2020	30.04.2020		
Оформление пояснительной записки и плакатов	10.06.2020	10.06.2020		

Руководитель бакалаврской работы

М.В.Позднов
(подпись) (И.О. Фамилия)

Аннотация

Объем бакалаврской работы 72 стр., 40 рисунков, 16 таблиц, 20 источников, 2 приложений.

В данной бакалаврской работе была создана «автоматизированная мобильная платформа», которая передвигается по трассе из черной линии по определенному маршруту.

Цель работы: создание автоматизированной мобильной платформы с микроконтроллерным управлением, в которую входят следующие функции: езда по черной линии, проезд плавных перекрестков согласно маршруту, остановка на метках, остановка при обнаружении препятствий.

Задачами работы являются:

1. Анализ исходных данных и известных решений;
2. Выбор необходимых компонентов;
3. Разработка структурной и принципиальной схем;
4. Разработка алгоритма передвижения и программы;
4. Сборка и отладка платформы и экспериментальные исследования.

Работа состоит из четырех глав, в которых решены упомянутые задачи.

Для оформления всех диаграмм и чертежей использовался пакет КОМПАС-3D V16.

Для разработки программы, управляющей логикой работы, использовалась среда разработки Arduino IDE.

Областью применения «автоматизированной мобильной платформы» могут быть различные складские помещения, а так же цеха различных производств.

Abstract

The title of the graduation work is: «Automated mobile platform».

The graduation work consists of an explanatory note on 72 pages, including 40 figures, 16 tables, the list of 20 reference including 5 foreign source, and two appendices.

The key issue of this work is the creation of an automated mobile platform that can ride on the highway from the black line. We touch upon the problem of automation of large-area warehouses, as well as various large-scale industries.

The aim of the work is to assemble the platform and create an algorithm for its movement.

This work is divided into the following chapters: studying the status of the issue; hardware development of the device; software development of the device; device assembly and experimental studies.

In the end, we present the data of experimental studies.

In conclusion, we'd like to emphasize that this theme of the graduation work is one of the relevant areas of development of robotics in the world. These technical developments can be applied in various warehouses and industries, which will greatly facilitate the work of staff.

Содержание

Введение.....	3
1. Состояние вопроса	4
1.1 Формулирование актуальности, цели и задач проекта	4
1.2 Анализ исходных данных и известных решений	5
2. Аппаратная часть	24
2.1 Разработка структурной схемы.....	24
2.2 Выбор необходимых компонентов.....	25
2.3 Разработка электрической принципиальной схемы.....	39
3. Программная часть.....	42
3.1 Разработка алгоритма работы.....	42
3.2 Разработка программной части устройства.....	47
4. Конструкторско-экспериментальный раздел.....	55
4.1 Сборка устройства.....	55
4.2 Отладка устройства и экспериментальные исследования.....	58
Заключение.....	63
Список используемой литературы.....	64
Приложение А Перечень элементов к схеме принципиальной.....	66
Приложение Б Программа автоматизированной мобильной платформы.....	67

Введение

Наука не стоит на месте. На данный момент уровень развития робототехники достиг немалых высот. Ситуация с развитием робототехники в настоящее время складывается таким образом, что остановить это развитие в данном направлении уже невозможно. Объясняется это тем, что роботы уже заняли свою нишу в жизни общества. Они стали частью современной промышленной революции, характеризующейся широким внедрением адаптивных технологий и роботизацией производства. Рост количества предприятий, которые автоматизируют свои процессы, с каждым годом только увеличивается, поэтому завод, на котором работает всего пара десятков человек, а всю основную работу выполняют роботы, на данный момент уже никого не удивляет. Промышленные роботы выпускаются тысячами. Несмотря на то, что этот рынок достаточно давно сформировался, ситуация здесь только обостряется. Стимул для активной роботизации складских помещений - оптимизация пространства, ускорение операций, исключение человека из операций на складе.

В России полная автоматизация промышленности и производства – вопрос далекого будущего. Пока что есть тренд на частичную автоматизацию процессов, без долгосрочных затрат. Внедряются погрузчики и роботизированные тележки, которые заменяют человека в одной цепочке процесса логистики.

Сложным вопросом является обеспечение возможности навигации внутри складских помещений. Для этого используются различные способы. Некоторые требуют подготовки склада - разметки на полу или установки специальных меток (беспроводных или отражательных) на стенах и полках.

В данной работе будет создана роботизированная платформа, способная перемещаться по разметке согласно определенному маршруту.

1 Состояние вопроса

1.1 Формулирование актуальности, цели и задач проекта

Автоматизация складских процессов и роботизация складов является одним из актуальных направлений развития робототехники. Системы, разрабатываемые в данном направлении, позволяют ускорить проведение различных операций. Благодаря их внедрению увеличивается производительность, так как машина может обрабатывать количество продукции на порядок выше, чем специально обученный человек за один и тот же промежуток времени. Увеличивается безопасность персонала. Это особенно актуально при работе с вредными и опасными материалами. Уменьшаются затраты на оплату труда.

Несмотря на то, что данное направление актуально, в мире существует очень ограниченное количество готовых решений, позволяющих создавать мобильные платформы, роботов и систем, способных на выполнение складских задач.

Целью работы является разработка автоматизированной мобильной платформы, перемещающейся путем отслеживания черной линии по определенному алгоритму с заданным маршрутом.

Задачами работы являются:

- Разработка структурной схемы;
- Выбор необходимых компонентов;
- Разработка электрической принципиальной схемы;
- Разработка алгоритма работы;
- Разработка программной части устройства;
- Сборка устройства;
- Отладка устройства и экспериментальные исследования.

1.2 Анализ исходных данных и известных решений

Рассмотрим пару готовых решений. Компания Amazon Robotics является одной из первых, кто начал внедрять робототехнические комплексы на складах. В компании решили, что люди должны оставаться в одном определенном месте, а не бегать по складу в поисках товара. Эту работу за них должны делать роботы. Это существенно облегчит труд человека. Вскоре они представили систему умного склада, на котором роботизированные платформы перевозят небольшие стеллажи с определенным товаром от места хранения до работников склада. Тем, в свою очередь, остается лишь взять товар со стеллажа, отсканировать его и положить на конвейер. При этом доступ сотрудникам на территорию склада, где передвигаются платформы и хранятся стеллажи, запрещен. Роботы представляют из себя небольшие тележки на колесах с механизмом, который может поднимать стеллажи на небольшое расстояние от пола для того чтобы перевозить их. Позиционирование платформы происходит с помощью специальных меток нанесенных на пол. Так же у них присутствуют датчики препятствий, чтобы случайно не врезаться в другие платформы. Работают платформы от аккумуляторной батареи и заряжают ее в течение пяти минут в конце каждого часа. Система умного склада отслеживает часто запрашиваемые товары и располагает их ближе к сотрудникам. По подсчетам компании, данная система увеличила производительность персонала примерно в два раза. На рисунке 1 представлен робот от компании Amazon Robotics.



Рисунок 1 – Робот от компании Amazon Robotics

В России тема разработки роботизированных систем управления пока только набирает обороты. Но уже сейчас у нас имеется одна интересная разработка. Робот Ronavi от компании Ronavi Robotics. Это прямоугольный по своей форме робот, который выполнен практически так же как и робот от компании Amazon. Функционал его заключается в том, чтобы перемещать грузы массой до полутора тонн. Это могут быть такие же стеллажи, как у Amazon, либо любой другой складской груз. Перемещается робот с помощью разметки, нанесенной на пол. Но компания на заказ сможет сделать робота, который будет ориентироваться на территории по безразметочной навигации с помощью машинного зрения. Одной из особенностей робота являются специальные колеса, благодаря которым он может разворачиваться на одном месте или ехать боком. Робот работает на аккумуляторной батарее, которой хватает на шесть часов. Разработанное программное обеспечение для

системы управления позволяет работать сразу с несколькими роботами одновременно. На рисунке 2 представлен робот от компании Ronavi Robotics.



Рисунок 2 – Российский робот Ronavi

В исходных данных мы будем использовать робототехнический набор для создания дистанционно управляемых моделей серии TETRIX MAX, который представлен на рисунке 3, и входящий в его комплектацию робототехнический контроллер TETRIX PRIZM.



Рисунок 3 – Робототехнический набор TETRIX MAX

Набор состоит из таких компонентов как:

- алюминиевые и металлические детали и крепежи для сборки каркаса платформы;
- инструменты для сборки;
- колеса для передвижения;
- 2 электродвигателя для движения платформы;
- 2 сервопривода;
- цифровой инфракрасный датчик линии;
- ультразвуковой датчик расстояния;
- перезаряжаемая Ni-MH (никельметаллогидридная) аккумуляторная батарея на 12 В;
- зарядное устройство для аккумуляторной батареи;
- соединительные провода;

- робототехнический контроллер TETRIX PRIZM;
- документация.

Рассмотрим каждый интересующий нас компонент по отдельности, а так же его технические характеристики.

Электродвигатель TorqueNADO.

На рисунке 4 представлен Электродвигатель TorqueNADO.



Рисунок 4 – Электродвигатель TorqueNADO

Это электродвигатель постоянного тока с редуктором и со встроенным высокочувствительным энкодером на эффекте Холла, который точно регулирует положение вала электродвигателя.

Технические характеристики:

- вес: 324 г;
- рабочее напряжение: 12 В;

- длина: 134 мм;
- передаточные числа: 60:1;
- диаметр: 37 мм;
- диаметр вала: 6 мм;
- крутящий момент: 4,9 Нм;
- скорость вращения: 100 об/мин.

Аккумуляторная батарея.

На рисунке 5 представлена аккумуляторная батарея TETRIX.



Рисунок 5 – Аккумуляторная батарея TETRIX

Перезаряжаемая Ni-MH (никельметаллогидридная) аккумуляторная батарея на 12 В, 3 А·ч. Состоит из десяти соединенных аккумуляторных элементов. Для безопасности встроен сменный плавкий предохранитель на 20 А.

Технические характеристики:

- вес: 580 г;
- размеры: 115×45×50 мм;

- выдаваемое напряжение: 12 В;
- емкость: 3 А×ч;
- номинал предохранителя: 20 А.

Ультразвуковой датчик расстояния.

На рисунке 6 представлен ультразвуковой датчик расстояния TETRIX.

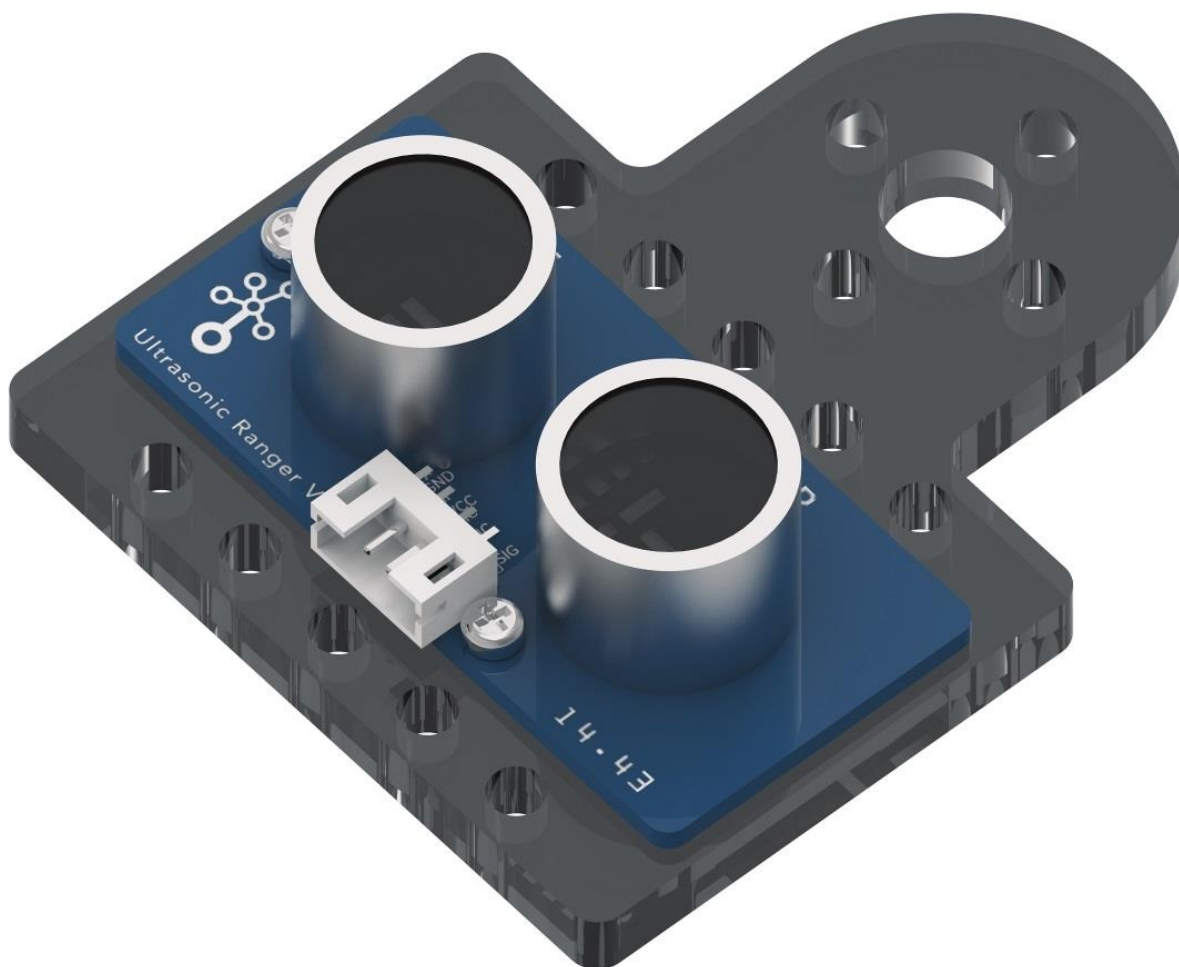


Рисунок 6 – Ультразвуковой датчик

Технические характеристики:

- частота: 42 кГц;
- минимальное расстояние: 3 см;

- максимальное расстояние: 400 см;
- напряжение питания: 5 В.

Зарядное устройство для аккумуляторной батареи.

На рисунке 7 представлено зарядное устройство для аккумуляторов TETRIX.



Рисунок 7 – Зарядное устройство

Зарядное устройство используется для перезарядки 12-вольтового аккумулятора TETRIX. Данное устройство автоматически определяет напряжение аккумуляторной батареи, настраивает правильное зарядное напряжение и автоматически отключает питание, когда аккумулятор полностью заряжен. А так же защищает аккумулятор от перегрузки по току, короткого замыкания и обратной полярности.

Технические характеристики:

- входное напряжение: переменное 100-240 В, 50/60 Гц;
- входной ток: 0.6 А;
- выходное напряжение: постоянное 16 В;
- выходной ток: 0.9/1.8 А;

Робототехнический контроллер TETRIX PRIZM.

На рисунке 8 представлен контроллер TETRIX PRIZM.

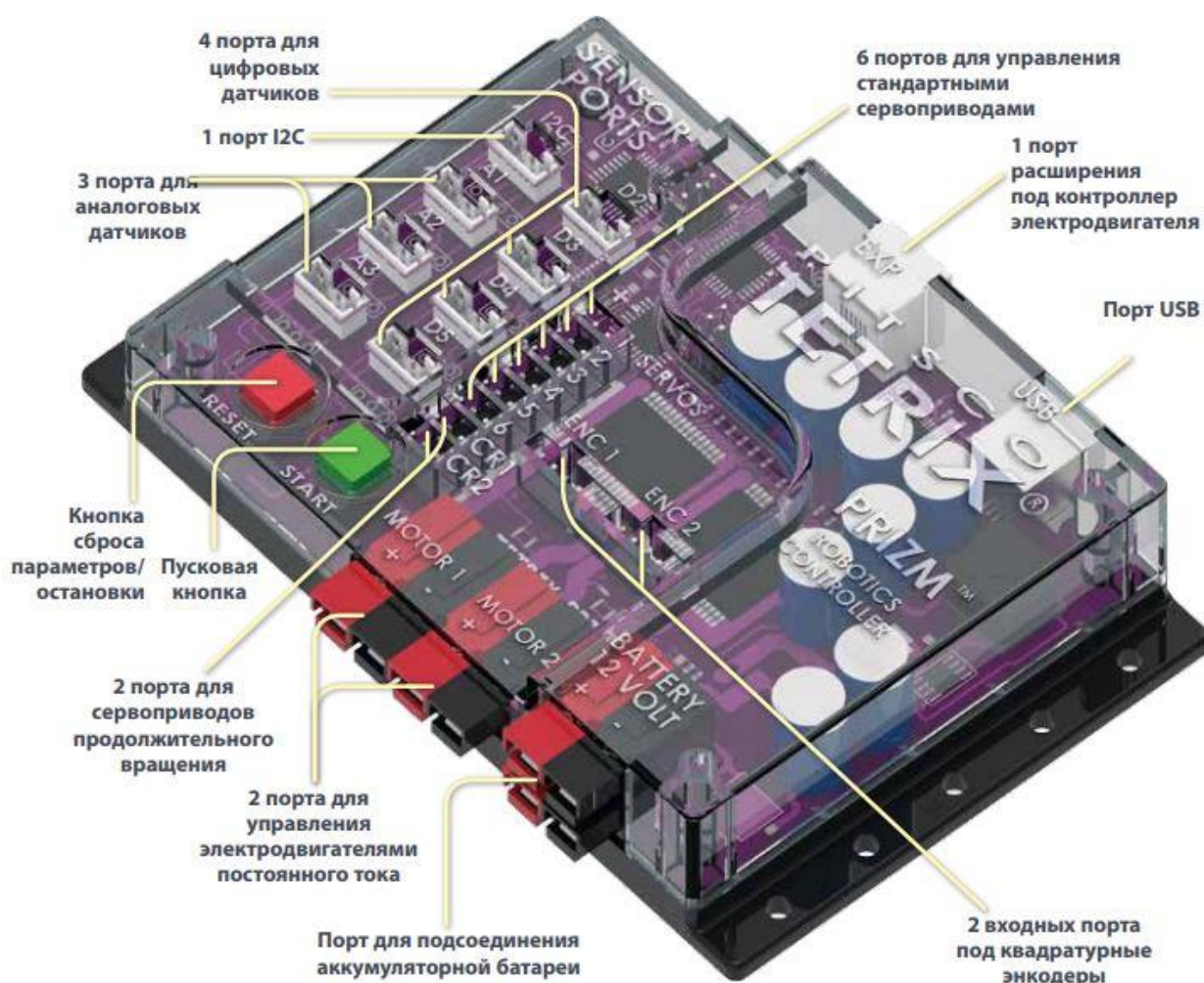


Рисунок 8 – Внешний вид контроллера и назначение его портов

Порты для датчиков на контроллере PRIZM.

В контроллере PRIZM используются схемы назначения контактов, совместимые с Arduino UNO. Датчики, к которым в библиотеке на языке Arduino для контроллера PRIZM есть программные коды, настраиваются автоматически благодаря библиотечным функциям. Однако функции составления программных кодов на языке Arduino позволяют напрямую пользоваться всеми портами, если нужно с ними поработать. За исключением порта I2C все остальные можно настроить на ввод или вывод сигналов при помощи функции Arduino pinMode. На рисунке 9 представлен один из портов для подключения датчиков к контроллеру. В таблицах 1-8 расписаны функции каждого контакта всех портов для подключения датчиков контроллера TETRIX PRIZM.

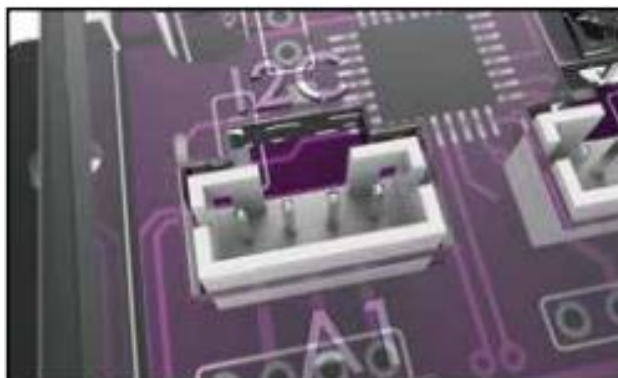


Рисунок 9 – Порт для датчиков на контроллере PRIZM
(Контакты слева направо: 1, 2, 3, 4.)

Таблица 1: Назначение контактов порта I2C

Номер контакта	Функция	Назначение контакта в ПО Arduino (IDE)
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	SDA (последовательная передача данных через шину связи I2C)	Канал ввода ADC4 (A4)
Контакт 4	SCL (тактовый сигнал шины связи I2C)	Канал ввода ADC5 (A5)

Таблица 2: Порт для аналоговых датчиков (A1)

Номер контакта	Функция	Назначение контакта в ПО Arduino (IDE)
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	Неиспользуемый	Отсутствует
Контакт 4	Ввод аналоговых или ввод-вывод цифровых сигналов	Ввод аналоговых сигналов (A1), ввод-вывод цифровых сигналов (15)

Таблица 3: Порт для аналоговых датчиков (A2)

Номер контакта	Функция	Назначение контактов в ПО Arduino (IDE) ()
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	Неиспользуемый	Отсутствует
Контакт 4	Ввод аналоговых или ввод-вывод цифровых сигналов	Ввод аналоговых сигналов (A2), ввод-вывод цифровых сигналов (16)

Таблица 4 Порт для аналоговых датчиков (A3)

Номер контакта	Функция	Назначение контактов в ПО Arduino (IDE) ()
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	Неиспользуемый	Отсутствует
Контакт 4	Ввод аналоговых или ввод-вывод цифровых сигналов	Ввод аналоговых сигналов (A3), ввод-вывод цифровых сигналов (17)

Таблица 5: Порт для цифровых датчиков (D2)

Контакт	Функция	Назначение контактов в ПО Arduino (IDE) ()
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	Ввод-вывод цифровых сигналов	Ввод-вывод цифровых сигналов (9)
Контакт 4	Ввод-вывод цифровых сигналов	Ввод-вывод цифровых сигналов (2)

Таблица 6: Порт для цифровых датчиков (D3)

Контакт	Функция	Назначение контактов в ПО Arduino (IDE) ()
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	Неиспользуемый	Отсутствует
Контакт 4	Ввод-вывод цифровых сигналов	Ввод-вывод цифровых сигналов (3)

Таблица 7: Порт для цифровых датчиков (D4)

Контакт	Функция	Назначение контактов в ПО Arduino (IDE) ()
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	Неиспользуемый	Отсутствует
Контакт 4	Ввод-вывод цифровых сигналов	Ввод-вывод цифровых сигналов (4)

Таблица 8: Порт для цифровых датчиков (D5)

Контакт	Функция	Назначение контактов в ПО Arduino (IDE) ()
Контакт 1	Заземление	Отсутствует
Контакт 2	+5 В, 100 мА	Отсутствует
Контакт 3	Неиспользуемый	Отсутствует
Контакт 4	Ввод-вывод цифровых сигналов	Ввод-вывод цифровых сигналов (5)

Порты для энкодеров на контроллере PRIZM.

У контроллера PRIZM есть два входа под квадратурные энкодеры, которые показаны на рисунке 10, позволяющие точно регулировать параметры в каналах управления электродвигателями постоянного тока № 1 и № 2. Энкодеры на электродвигателях постоянного тока TETRIX отслеживают угловое положение их валов; на основе этих данных программный код обеспечивает быстрое и точное регулирование угловой скорости и положения валов электродвигателей постоянного тока. Через порт ENC1 поступают сигналы энкодера для электродвигателя постоянного тока,

подключенному к каналу № 1, а через порт ENC2 — сигналы энкодера в канал управления электродвигателем постоянного тока № 2.

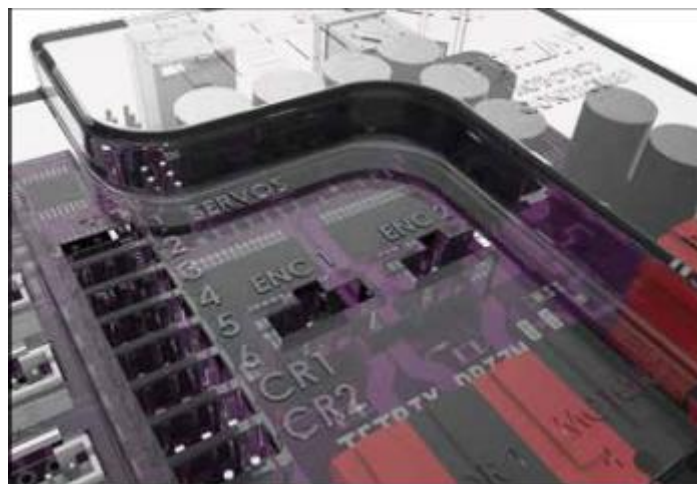


Рисунок 10: Порты под квадратурные энкодеры на контроллере PRIZM (Контакты слева направо: 1, 2, 3, 4.) (Обозначение: ENC1 и ENC2)

Контакт 1: источник постоянного напряжения +5 В.

Контакт 2: сигнал с показаниями счётчика энкодера (А).

Контакт 3: заземление энкодера.

Контакт 4: сигнал с показаниями счётчика энкодера (В).

Порт USB на контроллере PRIZM.

Порт USB на контроллере PRIZM, который показан на рисунке 11, служит для связи между самим контроллером и компьютером под управлением ОС Windows, Mac или Linux. Его главное назначение — загрузка программного кода в память микроконтроллера. Может также обеспечивать обмен данными между контроллером PRIZM и компьютером с использованием протокола последовательной связи. Например, в ПО Arduino (IDE) есть прибор контроля последовательной передачи данных, в окне которого отображаются значения данных, полученных от контроллера PRIZM. Здесь можно отображать показания обычных датчиков или

энкодеров, а при необходимости и другие виды данных о программах. Возможность видеть значения данных зачастую очень помогает при отладке программного кода. Прибор контроля последовательной передачи данных можно использовать и для отправки информации в контроллер PRIZM. Примером этого может служить приведение контроллером PRIZM в движение электродвигателей постоянного тока или сервоприводов в ответ на нажатие кнопок на клавиатуре. Для задействования функций отслеживания данных или управления продвинутое в программировании пользователи имеют возможность создать графические интерфейсы на платформах Java, Python или других платформах программирования.



Рисунок 11 – Порт USB на контроллере PRIZM, предназначенный для программирования и связи

Кнопка сброса параметров на контроллере PRIZM.

Красная кнопка сброса параметров, которая показана на рисунке 12, служит двум целям. Нажатие на неё прекращает выполнение любого программного кода и возвращает показания всех энкодеров и других датчиков в исходное состояние. Нажатие на неё точно так же перезапускает всю систему, как и выключение-включение питания.



Рисунок 12 – Кнопка сброса параметров на контроллере PRIZM

Пусковая кнопка на контроллере PRIZM.

Зелёная пусковая кнопка (Start), которая показана на рисунке 13, служит для запуска программного кода, загруженного в контроллер PRIZM. Когда вы создаёте программный код посредством ПО Arduino (IDE), функция PrizmBegin выполняет все команды, необходимые для запуска программы. Вызов этой функции приводит необходимые параметры в надлежащее состояние и задерживает выполнение программного кода до тех пор, пока не будет нажата зелёная пусковая кнопка.



Рисунок 13 – Пусковая кнопка на контроллере PRIZM

Порты для электродвигателей постоянного тока на контроллере PRIZM.

На контроллере PRIZM два порта для подсоединения электродвигателей постоянного тока. Они обозначены надписями Motor 1 и Motor 2. Каждый канал используется для управления угловой скоростью и

направлением вращения электродвигателей постоянного тока TETRIX при помощи команд, определение которых есть в ПО Arduino (IDE) контроллера PRIZM. На каждый канал управления электродвигателями подаётся постоянный ток силой 10 А и напряжением 12 В. Каждый канал управления электродвигателем снабжён красно-чёрным разъёмом, в котором красная часть имеет положительный потенциал, а чёрная — отрицательный. Силовые кабели к электродвигателям постоянного тока TETRIX снабжены парными цветными разъёмами, которые следует вставлять в часть разъёма того же цвета, чтобы всё работало. На рисунке 14 можно увидеть эти порты для подсоединения электродвигателей постоянного тока.

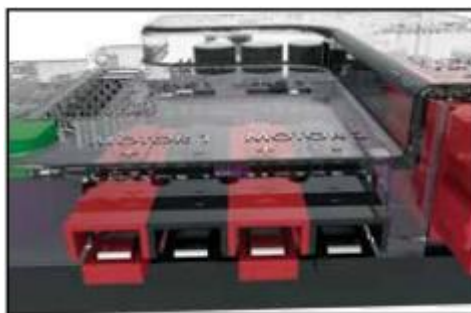


Рисунок 14 – Порты для электродвигателей постоянного тока на контроллере PRIZM

Порт для подсоединения аккумуляторной батареи на контроллере PRIZM.

Для питания контроллера PRIZM служит перезаряжаемая никельметаллогидридная аккумуляторная батарея на 12 В серии TETRIX. К контроллеру PRIZM прилагается комплект выключателя, предназначенный для подсоединения аккумуляторной батареи к соответствующему порту. Аккумуляторную батарею можно подсоединить либо к верхнему либо к нижнему ряду разъёмов, которые показаны на рисунке 15. Порт расширения предназначен для подсоединения дополнительных контроллеров

электродвигателей к контроллеру PRIZM и раздачи им напряжения по гирляндной цепи.

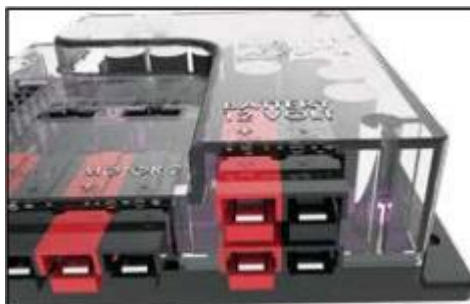


Рисунок 15 – Порт ввода/вывода напряжения аккумуляторной батареи контроллера PRIZM

В таблице 9 представлены технические характеристики контроллера TETRIX PRIZM.

Таблица 9 – Технические характеристики контроллера TETRIX PRIZM

Микроконтроллер:	Atmega328P с установленным загрузчиком операционной системы Arduino Optiboot
Память:	программируемое ЭПЗУ объемом 32 кб (Atmega328P)
Питающее напряжение:	9...18 В пост. тока
Порты для управления электродвигателями постоянного тока:	2 соединителя с разъёмами Powerpole; мостовая схема управления при помощи широтно-импульсной модуляции; каждый канал рассчитан на постоянный ток силой 10 А и максимальную нагрузку по току до 20 А
Рекомендованный электродвигатель:	Электродвигатель постоянного тока на 12 В TETRIX (39530)
Режимы управления электродвигателями постоянного тока:	Постоянная мощность (от -100 % до 100 %) Постоянная скорость при помощи ПИД-регулирования (от -720 до 720 градусов в секунду) ПИД-регулирование постоянной скорости для достижения и удержания заданного конечного положения с учётом показаний энкодера ПИД-регулирование постоянной скорости для достижения и удержания заданного конечного положения с учётом показаний энкодера в градусах Остановка в режиме торможения или выбега Отслеживание тока электродвигателя постоянного тока (все режимы)

Продолжение таблицы 9

Рекомендованный энкодер:	Комплект энкодера для электродвигателя серии TETRIX MAX (38000)
Порты под энкодеры для электродвигателей:	2 квадратурных, 5 В пост. тока, не более 50 мА; Технические характеристики: 360 отсчётов на оборот, 1440 импульсов на оборот; ENC 1 и ENC 2
Разъём USB:	USB типа B
Накопитель USB:	FTDI
Порты для стандартных сервоприводов:	Всего 6: каналы 1-6 для сервоприводов
Порты для сервоприводов продолжительного вращения (CR):	Всего 2: Каналы CR1 и CR2
Совокупное ограничение по току сервопривода:	постоянный ток не более 6 В, 6 А
Режимы управления сервоприводами:	Задание скорости сервопривода (0...100 %) Задание положение вала сервопривода (0...180 градусов) Задание состояния сервопривода продолжительного вращения (вращение по ЧС/против ЧС)
Отслеживание напряжения аккумуляторной батареи:	в пределах 0...18 В
Порты для цифровых датчиков (D2-D5):	Каждый можно настроить на ввод или вывод цифровых сигналов. D2 можно превратить в порт последовательной связи.
3 порта для аналоговых датчиков (A1-A3):	Каждый порт можно настроить на ввод аналоговых сигналов или вводвывод цифровых сигналов.
1 порт I2C (I2C):	скорость 100 кГц. Используется шина связи I2C, которая используется также для внутренних микроконтроллеров, управляющих электродвигателями постоянного тока и сервоприводами. Адреса I2C с 0x01 по 0x06 оставлены контроллером PRIZM в запасе.
Порт расширения под дополнительный контроллер электродвигателя (EXP):	К этому порту при помощи гирляндной схемы можно подключить дополнительные модули управления электродвигателями постоянного тока и сервоприводами TETRIX.
Порт для подсоединения аккумуляторной батареи:	подходит к разъёмам Powerpole; дополнительный порт для раздачи напряжения аккумуляторной батареи по гирляндной цепи контроллерам электродвигателей, подсоединяемых к порту расширения.
1 зелёная пусковая кнопка (START)	Программируемая кнопка
1 красная кнопка сброса параметров/остановки (RESET)	Непрограммируемая кнопка
1 красный светодиод:	Программируемый светодиод, служащий индикатором
1 зелёный светодиод:	Программируемый светодиод, служащий индикатором
1 синий светодиод:	Включается на время подачи в цепь электропитания

Выводы по разделу

Подведем следующие выводы:

- автоматизация складских процессов и роботизация складов является одним из актуальных направлений развития робототехники;
- несмотря на актуальность данного направления, в мире существует пока очень мало готовых рабочих решений;
- робототехнический набор для создания дистанционно управляемых моделей серии TETRIX MAX и входящий в его комплектацию робототехнический контроллер TETRIX PRIZM являются очень перспективным средством для выполнения задач выпускной квалификационной работы;
- технические характеристики набора удовлетворяют всем требованиям для создания на его основе автоматизированной мобильной платформы.

2. Аппаратная часть

2.1 Разработка структурной схемы

Структурная схема была разработана на основе технического задания на бакалаврскую работу.

Для выполнения задания потребуется:

- Микроконтроллер, который будет управлять платформой;
- Инфракрасные датчики для отслеживания черной линии;
- Ультразвуковой датчик для обнаружения препятствий;
- Электродвигатели для передвижения платформы;
- Дисплей для отладки и вывода информации;
- Аккумуляторная батарея для питания платформы.

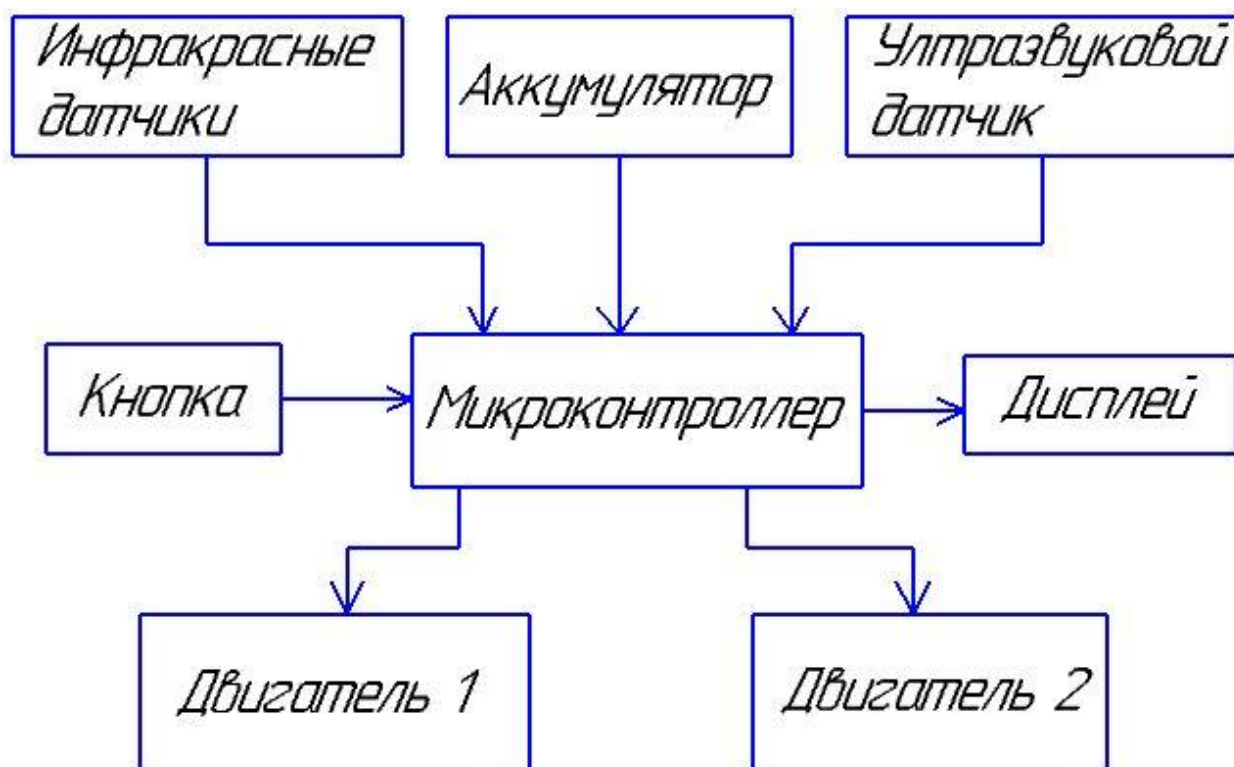


Рисунок 16 – Структурная схема платформы

2.2 Выбор необходимых компонентов

По техническому заданию мы используем робототехнический набор для создания дистанционно управляемых моделей серии TETRIX MAX. При анализе исходных данных мы выяснили, что в составе набора есть: корпус для платформы, микроконтроллер, электродвигатели, аккумуляторная батарея с зарядным устройством и ультразвуковой датчик.

Исходя из разработанной структурной схемы, для корректной работы платформы нам не хватает таких элементов как: инфракрасные датчики линии, кнопка и дисплей.

Обзор, анализ и выбор инфракрасных датчиков линии.

По заданию платформа должна передвигаться по территории по черной линии, поэтому было принято решение использовать инфракрасный датчик линии. На рисунке 17 можно увидеть один из таких датчиков.

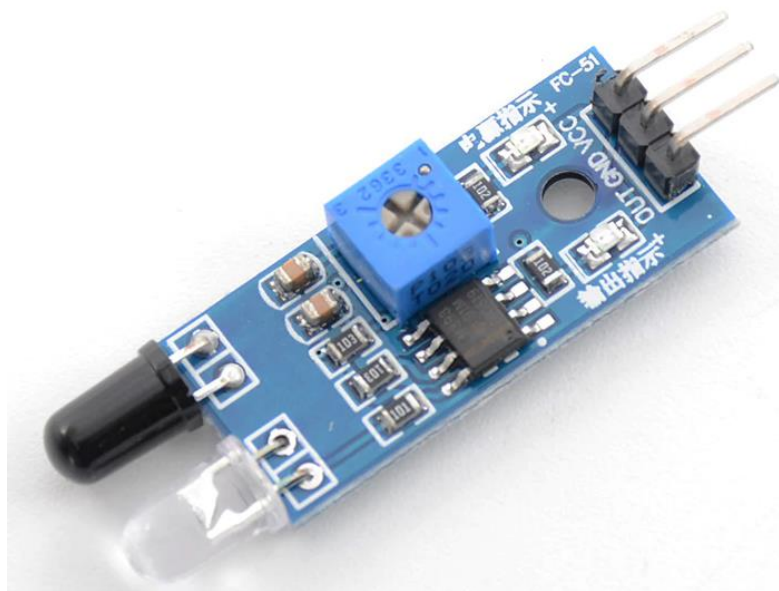


Рисунок 17 – Пример инфракрасного датчика линии

Инфракрасный датчик линии – это оптический модуль, предназначенный для обнаружения препятствий в виде белых или чёрных линий. Основным его элементом является оптопара, состоящая из инфракрасного светодиода и элемента, улавливающего инфракрасный свет. Это может быть фотодиод, фототранзистор или фоторезистор. Инфракрасный диод излучает свет с длиной волны 940-950 нм, что позволяет достоверно определять препятствия на различном расстоянии, в зависимости от типа датчика и используемой оптопары.

Принцип работы датчика, который графически изображен на рисунке 18, основан на детектировании инфракрасного света, отраженного от поверхности.



Рисунок 18 – Принцип работы оптопары

Инфракрасные датчики линии можно подразделить на две группы:

а) по типу выходного сигнала:

- 1) аналоговые;
 - 2) цифровые;
- б) по типу фотоэлемента, используемого в качестве детектора инфракрасного света:
- 1) на фотодиоде;
 - 2) на фототранзисторе.

Аналоговый датчик линии состоит из оптопары и резисторов, ограничивающих ток светодиода и детектирующего фотоэлемента. Пример схемы аналогового датчика представлен на рисунке 19.

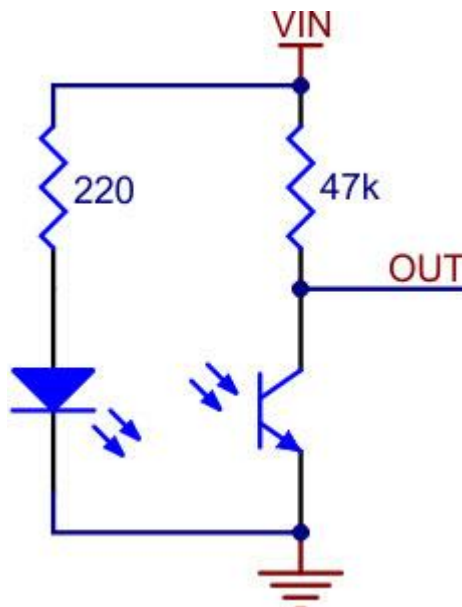


Рисунок 19 – Одна из возможных схем аналогового датчика

Аналоговый датчик способен не только отличать чёрную поверхность от белой, но и также различные оттенки серого. Это даёт возможность точно контролировать процесс перехода границы от чёрного к белому и наоборот.

Когда модуль датчика получает питание, на рабочую поверхность направляется луч инфракрасного светодиода, излучающего свет с длинной

волны 940-950 нм. Отражаясь от поверхности напротив, луч возвращается назад, и попадает на расположенный рядом с инфракрасным светодиодом фотоэлемент. На выходе мы получаем аналоговый сигнал. Чем поверхность светлее, тем сигнал больше. Чем поверхность темнее, тем сигнал меньше. Но может быть и обратная ситуация. Все зависит от схемотехнического решения, а именно от выбранного фотоэлемента, и от того, откуда снимается сигнал. Если взять схему на рисунке 19, то в качестве фотоэлемента используется NPN-фототранзистор. Аналоговый сигнал снимается с его коллектора. Соответственно, чем темнее поверхность, тем отраженный свет меньше, и тем меньше его попадает на затвор фототранзистора. Из-за этого фототранзистор закрывается, и сигнал становится больше. И наоборот, чем светлее поверхность, тем больше света отражается и попадает на затвор. Фототранзистор открывается, и сигнал становится меньше. Пример аналогового датчика представлен на рисунке 20.

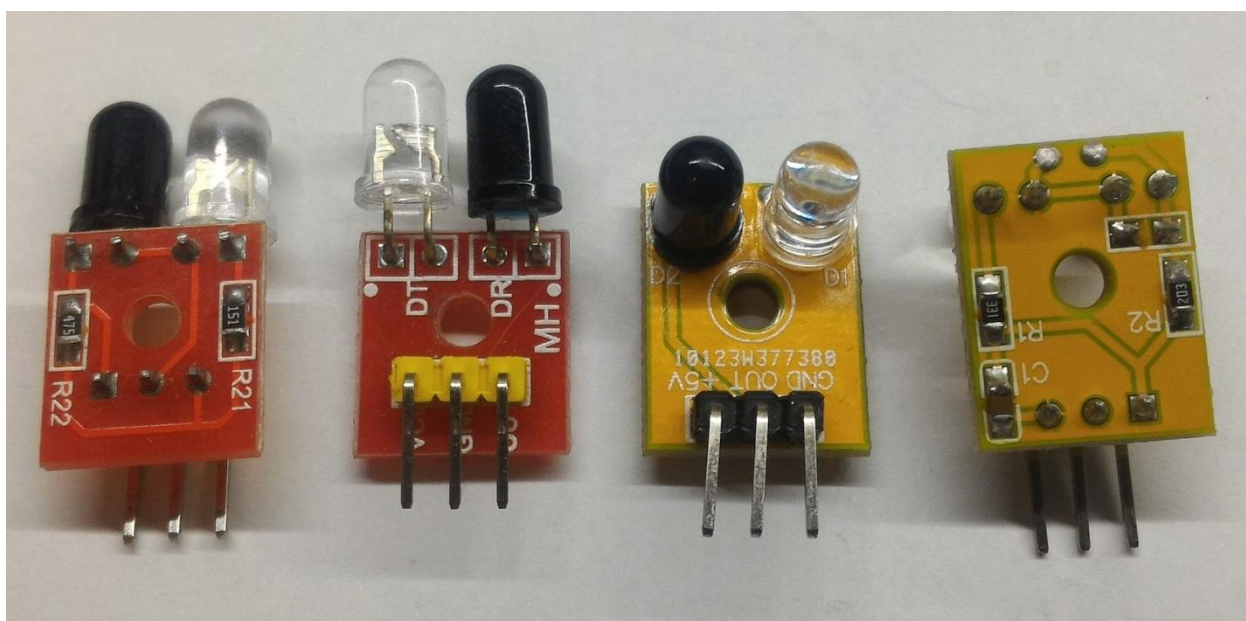


Рисунок 20 – Аналоговые инфракрасные датчики линии

В цифровом датчике, также как и в аналоговом есть оптопара и ее токоограничивающие резисторы. Помимо этого в цифровом датчике обычно устанавливается подстроечный резистор, компаратор и индикаторный светодиод. Пример цифрового датчика представлен на рисунке 21.



Рисунок 21 – Цифровой инфракрасный датчик линии

В нем, как и в аналоговом датчике, инфракрасный светодиод излучает свет с длиной волны 940-950 нм (в инфракрасном диапазоне). Инфракрасный луч отражается от поверхности напротив и попадает на фотоэлемент. Далее сигнал идет на компаратор. В нем сравниваются два аналоговых сигнала: один сигнал с фотоэлемента, другой, его называют опорным, с подстроенного резистора.

Когда сигнал с фотоэлемента будет больше опорного, тогда на выходе появится логическая единица (напряжение высокого уровня), и наоборот, когда сигнал с фотоэлемента будет меньше опорного сигнала, тогда на выходе будет логический ноль (напряжение низкого уровня). Принцип работы цифрового датчика линии визуально показан на рисунке 22.

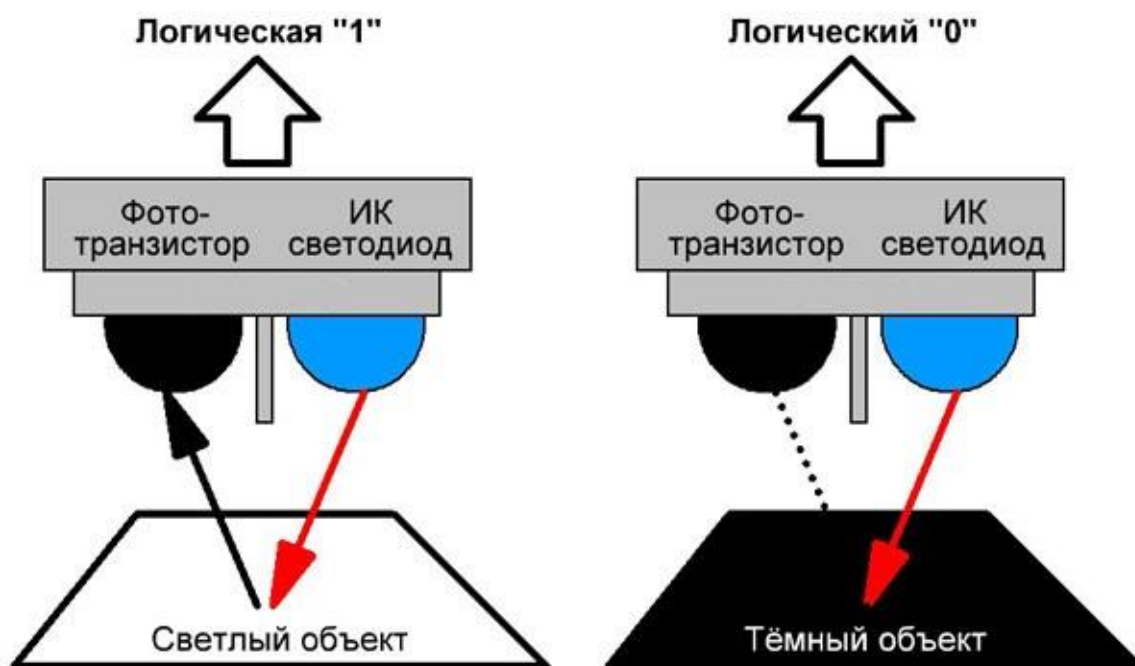


Рисунок 22 – Принцип работы цифрового датчика линии

С помощью подстроечного резистора датчик можно откалибровать на определенный оттенок серого или любой другой цвет, а также на максимальное расстояние. Для калибровки (регулировки чувствительности) ручку подстроечного резистора поворачивают в ту или иную сторону. Так можно добиться срабатывания только на самый темный оттенок или на самый светлый, либо если цвет преграды напротив датчика неизменен – только на расстояние не дальше установленного.

Во время настройки датчика можно ориентироваться на индикаторный светодиод, который загорится тогда, когда луч обратно принят и его интенсивность соответствует калибровке. Если датчик расположить слишком низко, то прямой ИК-луч отразится рано и попадет сразу обратно или на перегородку между светодиодом и фототранзистором, поэтому есть некое минимальное расстояние. Если же датчик установлен слишком далеко – луч

преждевременно рассеется, не дойдя обратно. Поэтому есть максимальное расстояние.

По техническому заданию платформа должна плавно следовать по черной линии, определять и проезжать различные повороты и перекрестки.

Для плавного передвижения будет использоваться ПИД регулирование. Для этой задачи нам нужны аналоговые датчики линии. Для обнаружения поворотов и перекрестков будет достаточно цифровых датчиков линии.

Ниже представлены различные инфракрасные датчики и модули, которые подходят под решение вышеупомянутых задач, и их характеристики, указанные производителем.

Инфракрасный датчик линии KY-032, представленный на рисунке 23. Технические характеристики представлены в таблице 10.

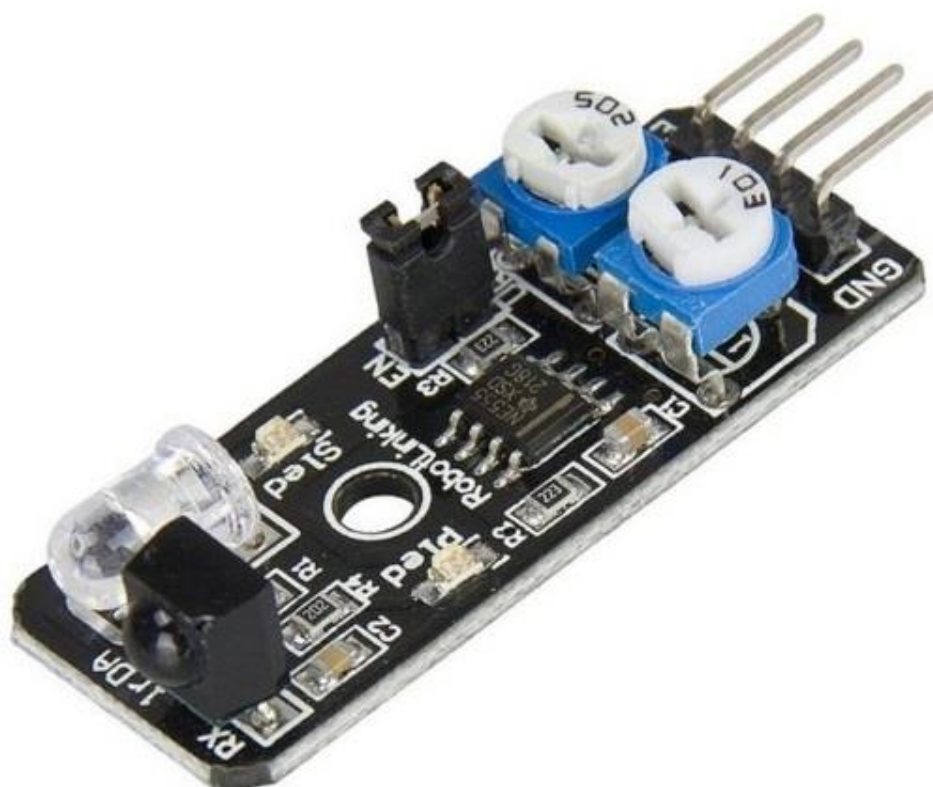


Рисунок 23 – Инфракрасный датчик линии KY-032

Таблица 10 – Технические характеристики датчика линии KY-032

Режим работы	цифровой
Напряжение питания, В	3.3 – 5
Ток потребления, мА	20
Дальность срабатывания, см	2 – 40
Частота излучения, кГц	38

Инфракрасный датчик линии FC-51, представленный на рисунке 24. Технические характеристики представлены в таблице 11.

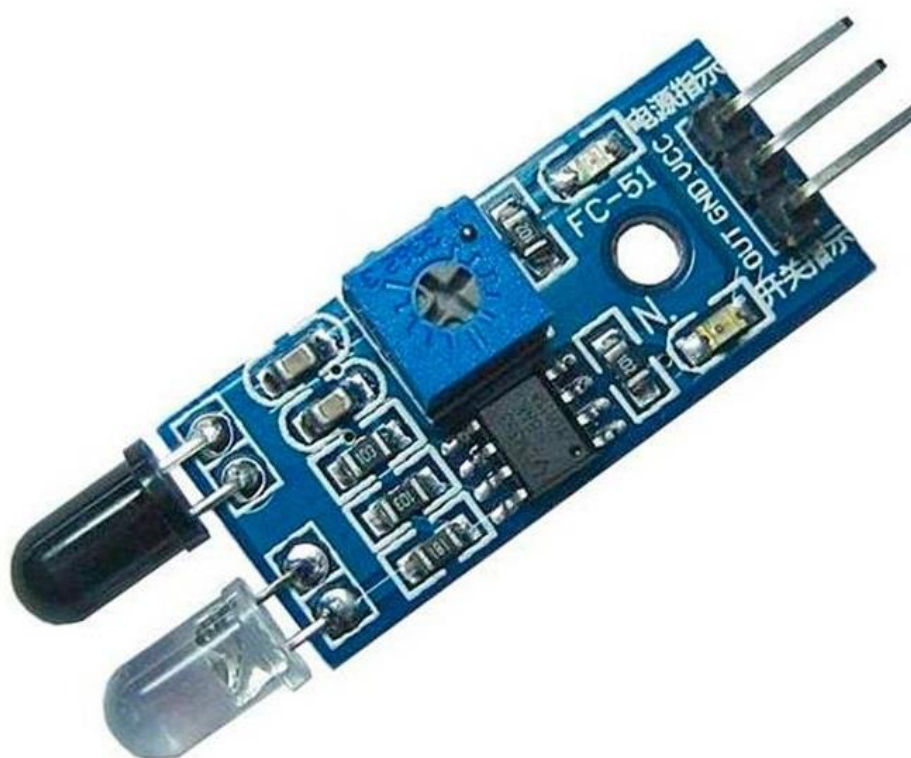


Рисунок 24 – Инфракрасный датчик линии FC-51

Таблица 11 – Технические характеристики датчика линии FC-51

Режим работы	цифровой
Напряжение питания, В	3.3 – 5
Дальность срабатывания, см	2 – 30

Инфракрасный датчик линии TRT5000, представленный на рисунке 25. Технические характеристики представлены в таблице 12.

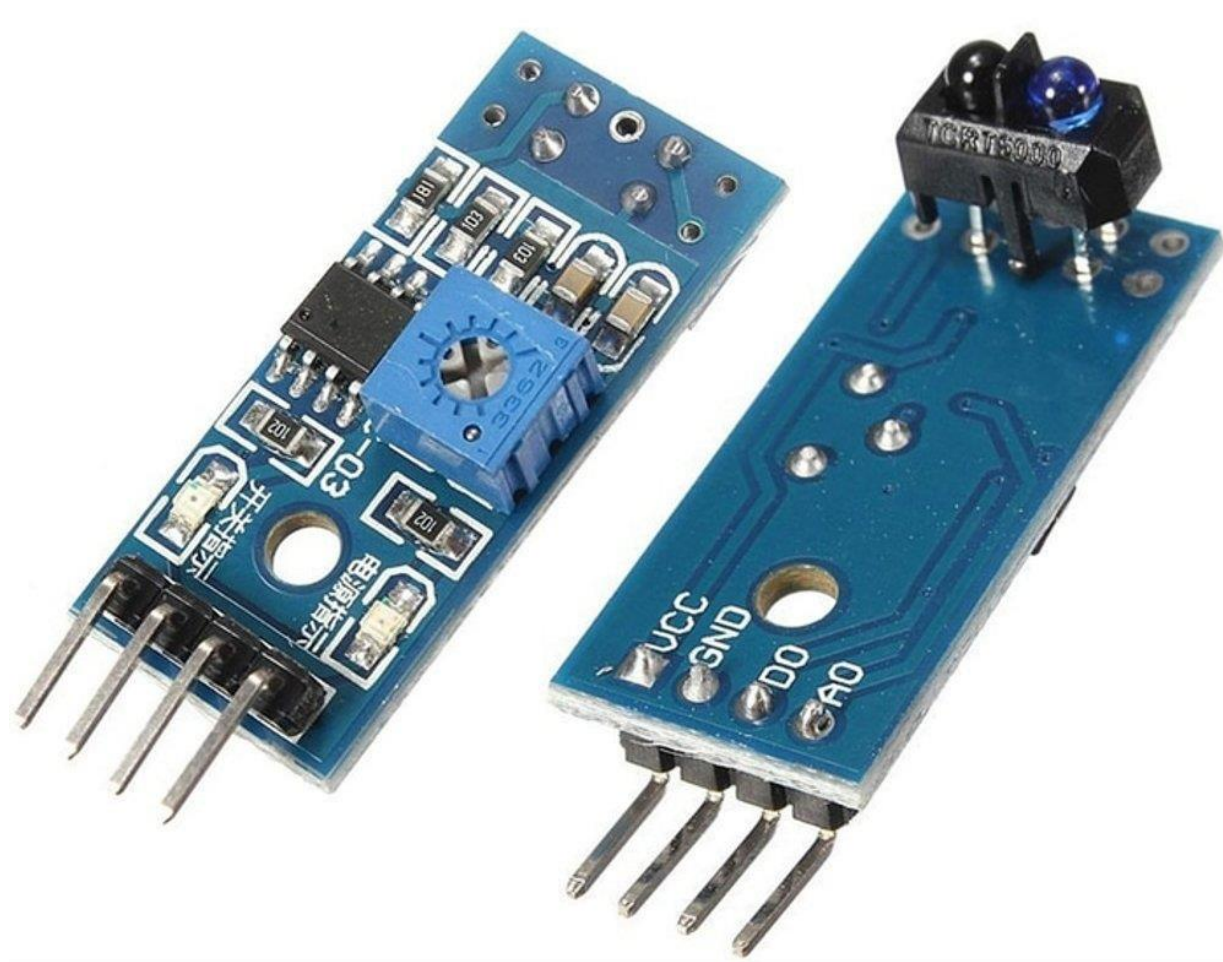


Рисунок 25 – Инфракрасный датчик линии TRT5000

Таблица 12 – Технические характеристики датчика линии TRT5000

Режим работы	аналоговый / цифровой
Напряжение питания, В	3.3 – 5
Ток потребления, мА	15
Дальность срабатывания, см	0.1 – 2.5

Пятиканальный модуль TCRT5000, представленный на рисунке 26. Технические характеристики представлены в таблице 13.

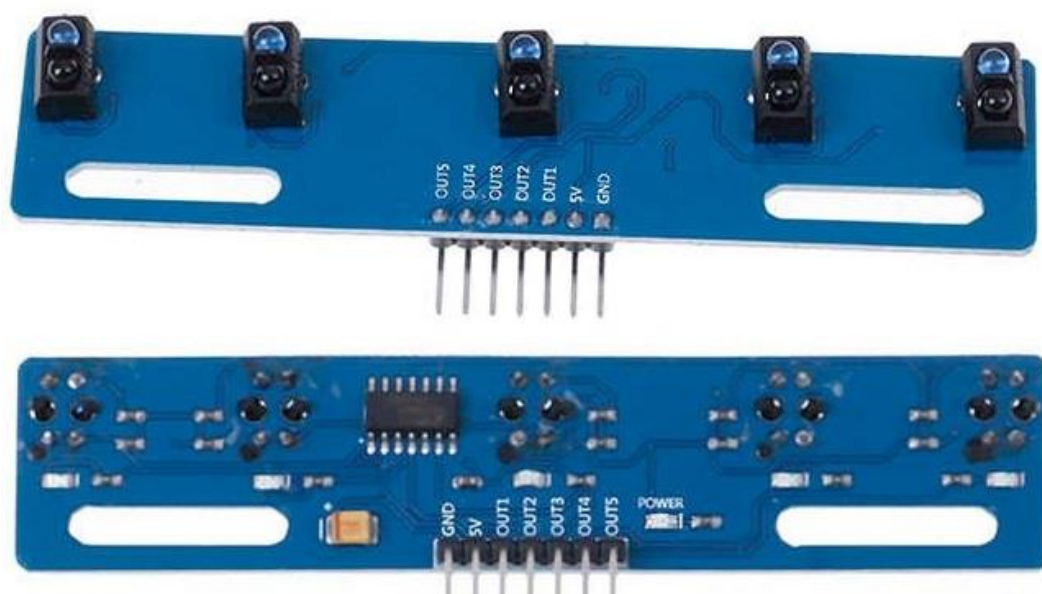


Рисунок 26 – Пятиканальный модуль TCRT5000

Таблица 13 – Технические характеристики пятиканального модуля TCRT5000

Режим работы	цифровой
Напряжение питания, В	3.3 – 5
Ток потребления, мА	15
Дальность срабатывания, см	0.1 – 2.5

Четырехканальный модуль YL-70 с датчиками YL-73, представленный на рисунке 27. Технические характеристики представлены в таблице 14.

Таблица 14 – Технические характеристики четырехканального модуля YL-70

Режим работы	аналоговый / цифровой
Напряжение питания, В	3.3 – 5
Дальность срабатывания, см	2 – 7

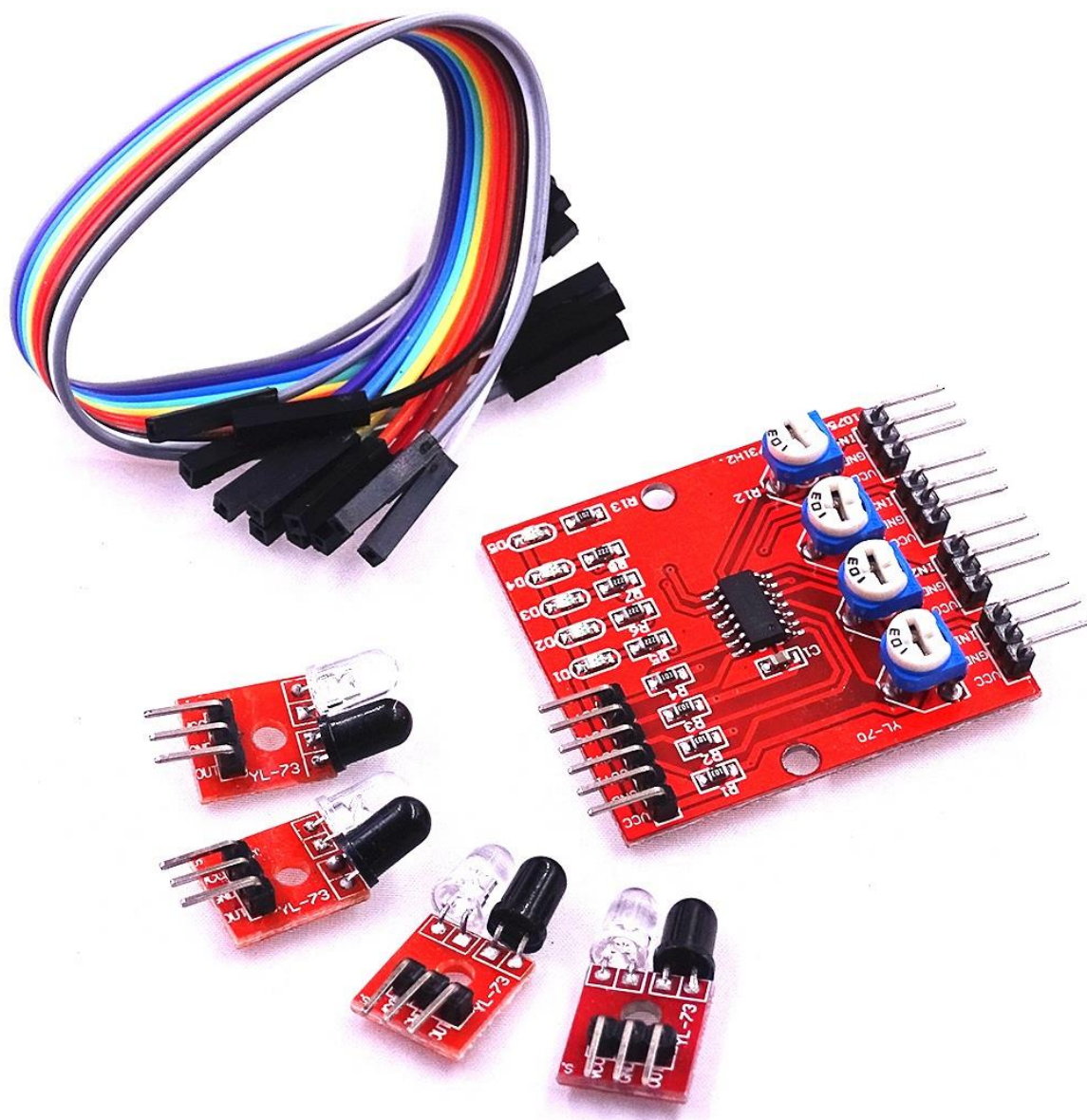


Рисунок 27 – Четырехканальный модуль YL-70 с датчиками YL-73

Восьмиканальный модуль МН, представленный на рисунке 28. Технические характеристики представлены в таблице 15.

Таблица 15 – Технические характеристики восьмиканального модуля МН

Режим работы	аналоговый / цифровой
Напряжение питания, В	3.3 – 5
Дальность срабатывания, см	2 – 7

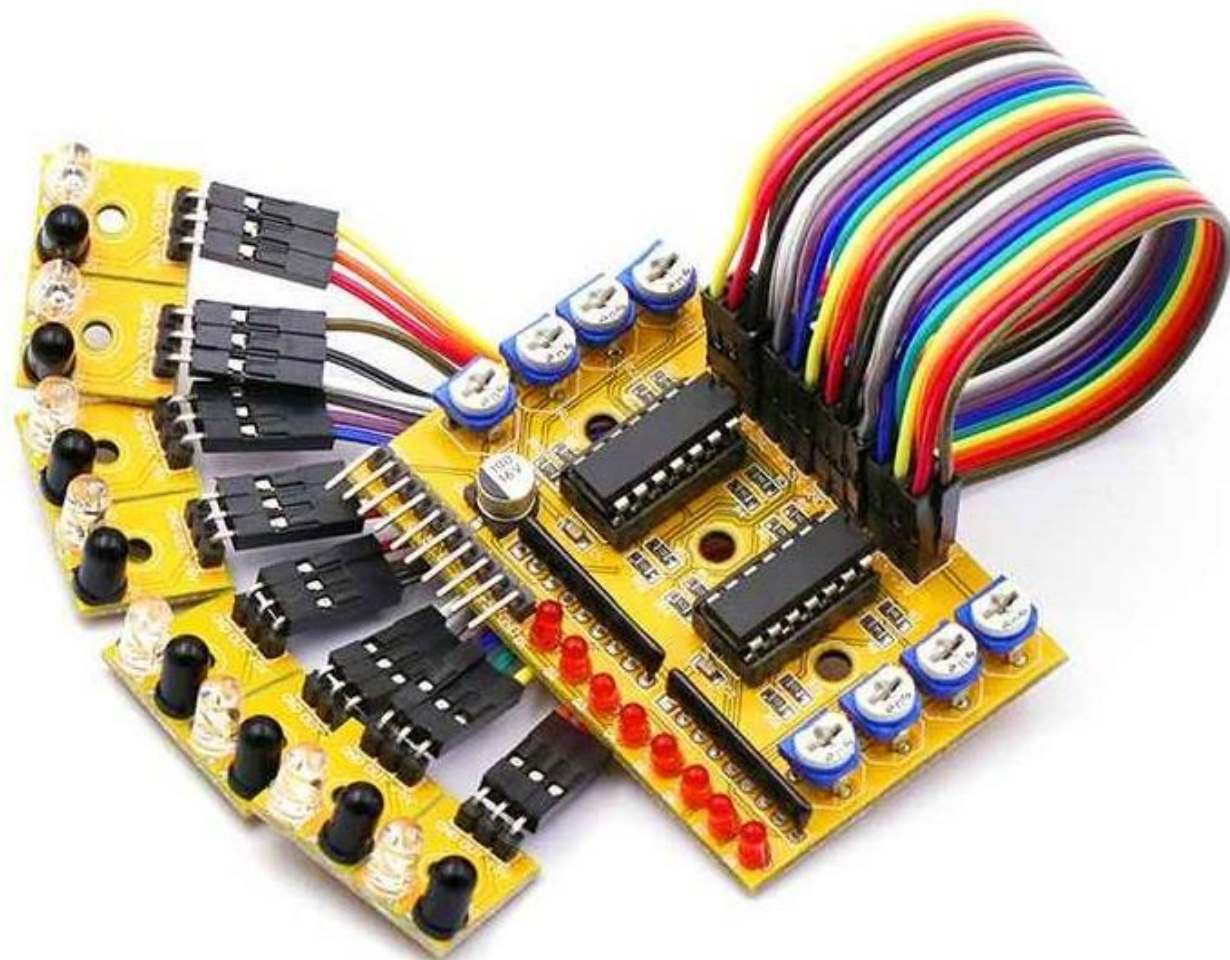


Рисунок 28 – Восьмиканальный модуль МН

Из всех перечисленных вариантов наибольший интерес вызвали четырехканальный модуль YL-70 и восьмиканальный модуль МН.

Модуль YL-70 состоит из четырех аналоговых датчиков линии YL-73 и платы с электроникой. На данной плате установлена микросхема LM339, которая является четырехканальным компаратором. С ее помощью можно использовать аналоговые датчики в качестве цифровых. Так же на плате есть четыре подстроечных резистора и четыре индикаторных светодиода, которые позволяют настраивать и калибровать каждый отдельный датчик на определенный уровень срабатывания. Еще одним плюсом этого модуля является то, что плата в нем разводит питание сразу на все датчики.

Следовательно, если сравнивать этот модуль со всеми остальными датчиками, то для его подключения понадобится меньше проводов и используемых выводов на плате управления платформы. При сравнении YL-70 с пятиканальным модулем TCRT5000, последний будет хуже по характеристикам ввиду его малой дальности срабатывания. Также у TCRT5000 все оптопары расположены на одной плате электроники, и их нельзя расположить в удобном для себя порядке, в отличие от YL-70.

Восьмиканальный модуль МН отличается от четырехканального модуля YL-70 только тем, что он может использовать сразу восемь датчиков. Но при более детальном исследовании были выявлены и отличия. Касаются они только аналоговых датчиков, входящих в модули. У них разный номинал токоограничивающих резисторов оптопары. В документации не указаны их типы, поэтому можно предположить что и оптопары у них различаются.

При дальнейшем сравнении восьмиканального модуля МН и четырехканального модуля YL-70 было выявлено, что последний очень сильно подвержен «засветке» со стороны. Иными словами, показания с датчиков сильно меняются, в зависимости от освещенности. У восьмиканального модуля МН показания датчиков из-за освещенности практически неизменны.

Рассмотрев и сравнив выше представленные модули и датчики и их характеристики, было принято решение использовать в дальнейшей разработке платформы именно восьмиканальный модуль МН.

Для вывода информации было принято решение использовать жидкокристаллический дисплей LCD 1602 с модулем I2C, который представлен на рисунке 29, так как он является самым популярным дисплеем, и вследствие чего стоит дешевле всех известных аналогов. Он имеет две строки по шестнадцать символов в каждой. Плюсом этого дисплея является модуль I2C, благодаря которому подключение дисплея осуществляется всего по четырем выводам.

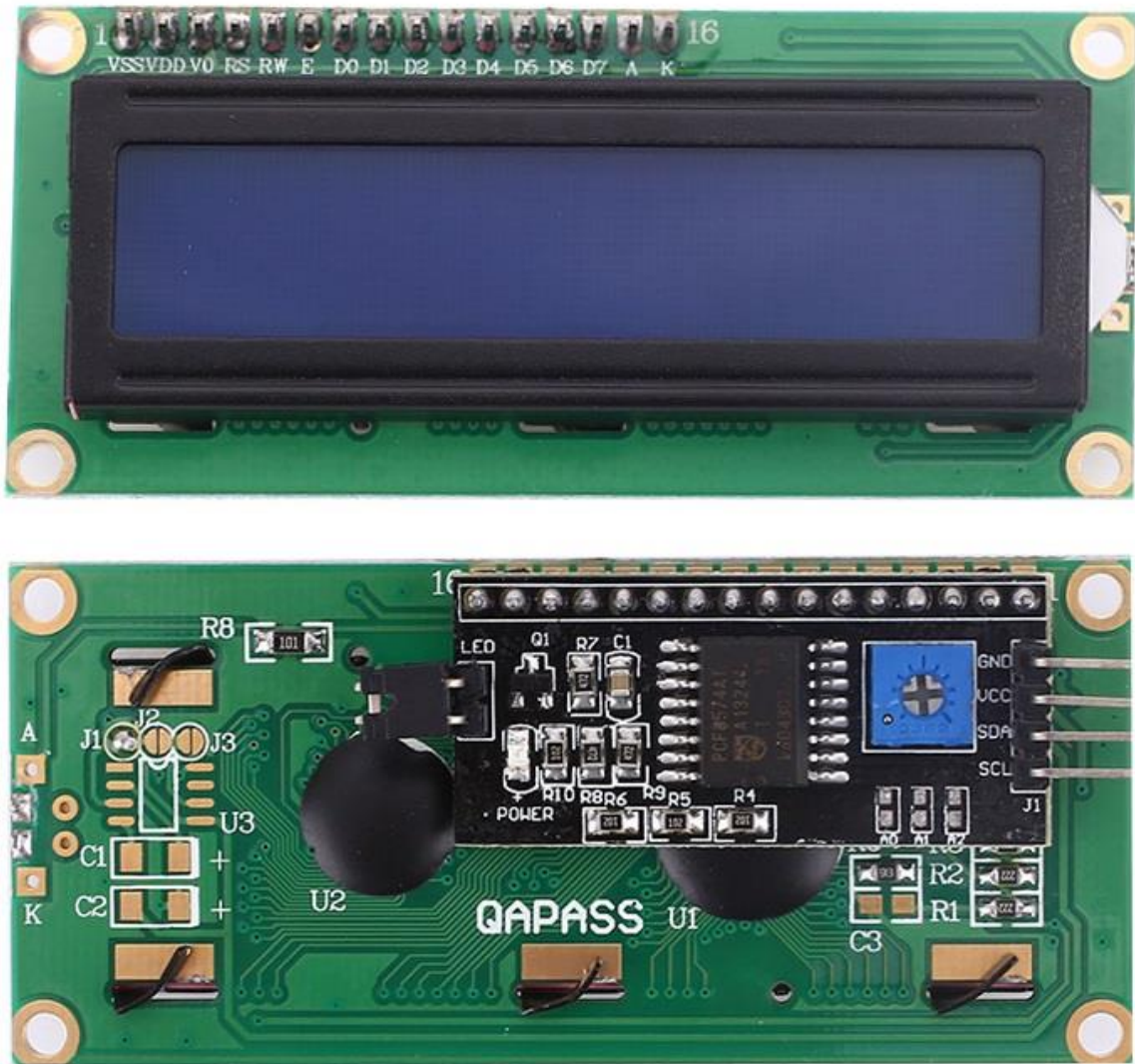


Рисунок 29 – Дисплей LCD 1602 с модулем I2C

Технические характеристики дисплея:

- тип дисплея: символьный с возможностью загрузки символов;
- контраст: настраивается потенциометром;
- напряжение питания: 5 В;
- ПС/I2C преобразователь: есть;
- интерфейс: I2C;
- размер: 80×36×12 мм;
- вес: 35 г.

Для управления платформой и корректировки ее действий во время движения было решено использовать обычную тактовую кнопку без фиксации, которая представлена на рисунке 30.



Рисунок 30 – Тактовая кнопка

2.3 Разработка электрической принципиальной схемы

В соответствии с исходными данными, структурной схемой и выбранными компонентами была разработана схема электрическая принципиальная. Схема электрическая соединений представлена на рисунке 31. Оформление схемы производилось в системе автоматизированного проектирования КОМПАС 3D V16.

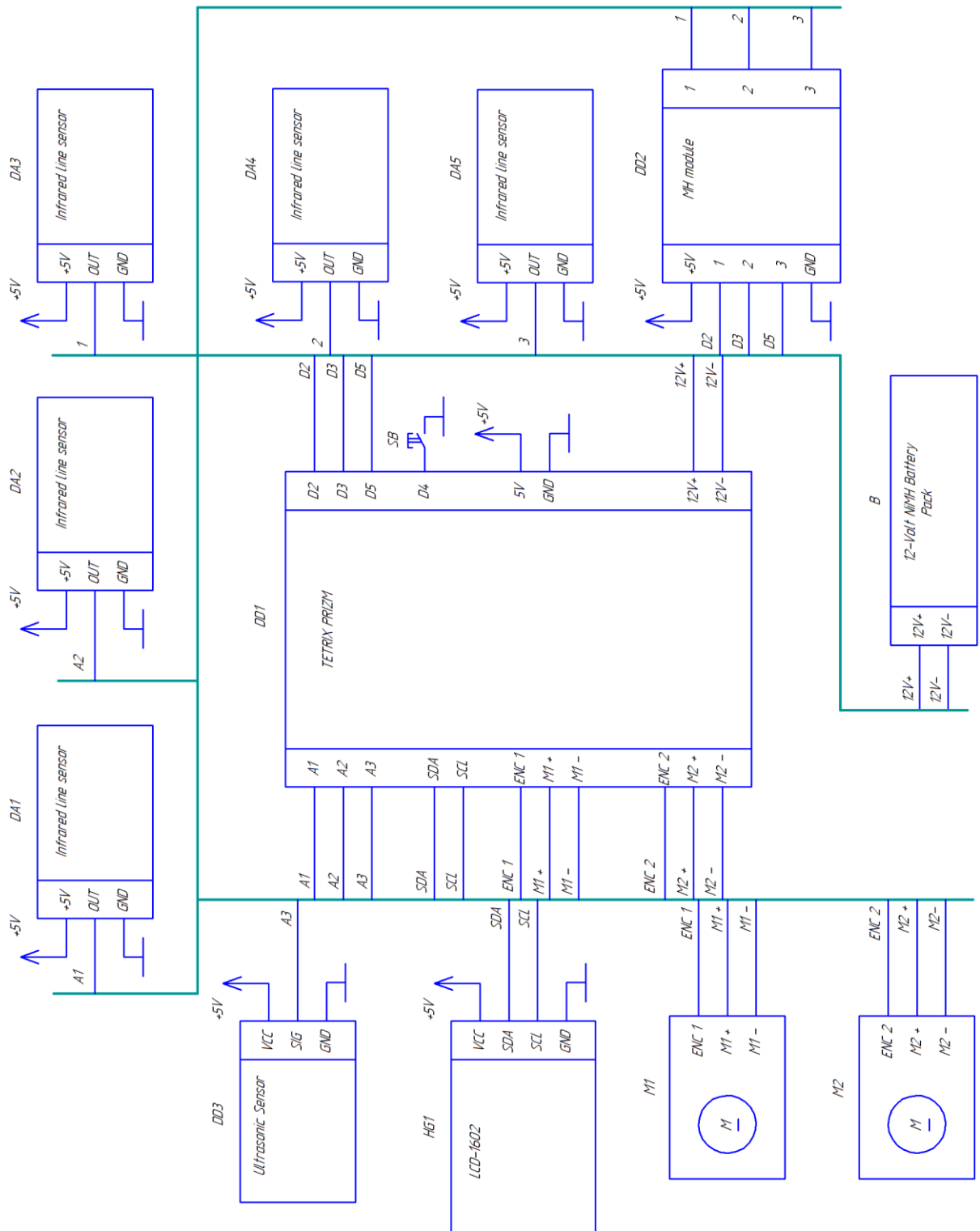


Рисунок 31 – Принципиальная схема автоматизированной мобильной платформы

Перечень элементов к схеме представлен в приложении А.

Выводы по разделу

Таким образом, в данной главе была создана структурная схема и схема принципиальная электрическая. Так же был изучен принцип работы инфракрасных датчиков линии. На основе изученных данных был произведен подбор, сравнение и выбор датчиков для дальнейшего использования в платформе. Выбор был сделан в пользу восьмиканальный модуля МН. Достоинства данного модуля:

- восемь аналоговых датчиков, которые могут работать в цифровом режиме;
- датчики не связаны между собой, и их можно располагать в различном порядке;
- требуется меньше проводов для подключения;
- датчики не подвержены засветке.

3. Программная часть

3.1 Разработка алгоритма работы

По техническому заданию платформа должна плавно передвигаться по трассе из черной линии, распознавать перекрестки с плавными поворотами и проезжать их в соответствии с маршрутом, а так же останавливаться на определенных метках и выводить на дисплей определенную надпись.

Сами метки будут выполнены из черной линии, расположенной перпендикулярно основной трассе. Движение после остановки на метках платформа должна начинать только после нажатия на кнопку.

При всем при этом платформа должна следить за пространством перед собой, чтобы в случае возникновения неожиданного препятствия остановиться и предотвратить столкновение.

На рисунке 32 изображена блок-схема разработанного алгоритма.

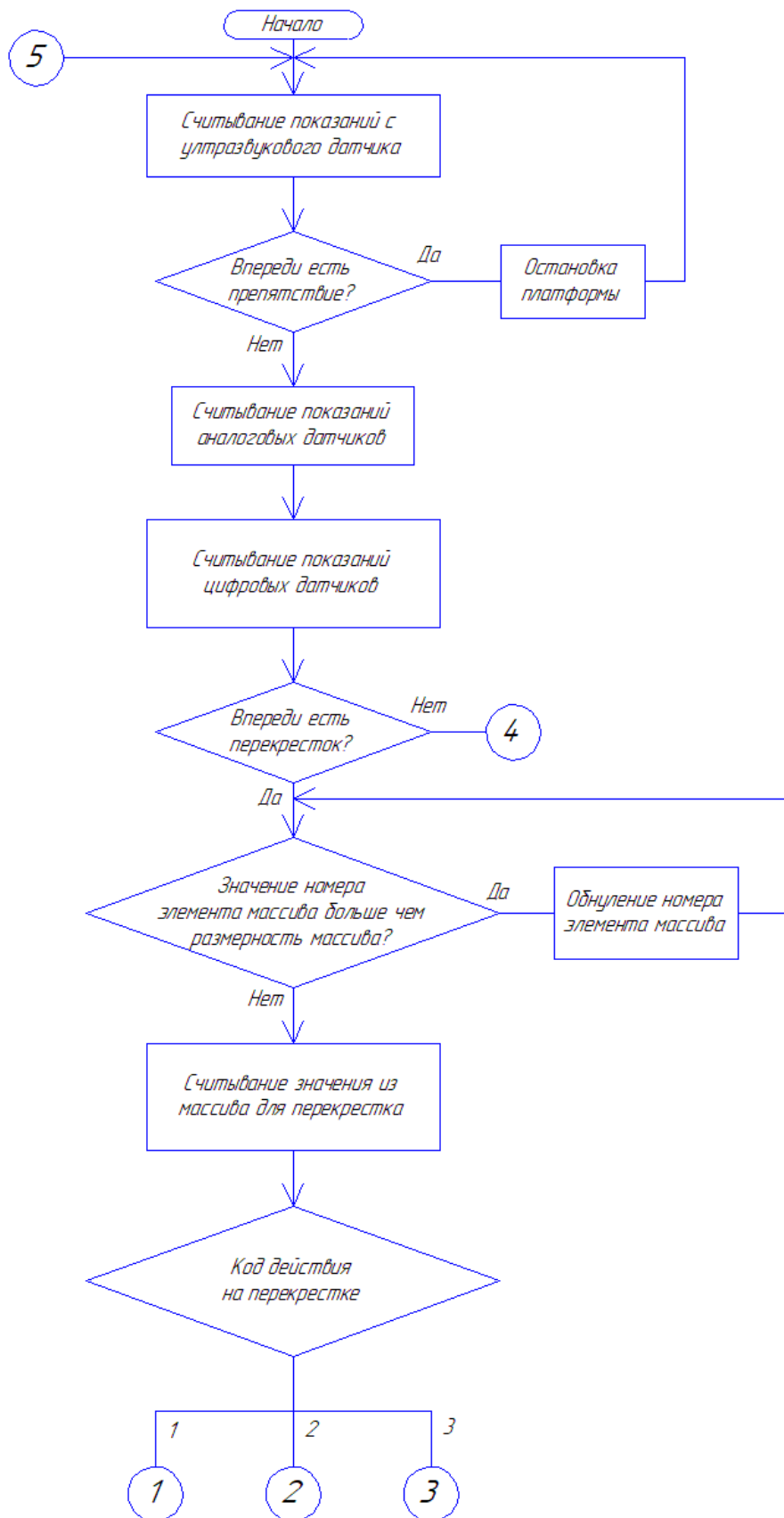


Рисунок 32а – Блок-схема алгоритма работы платформы

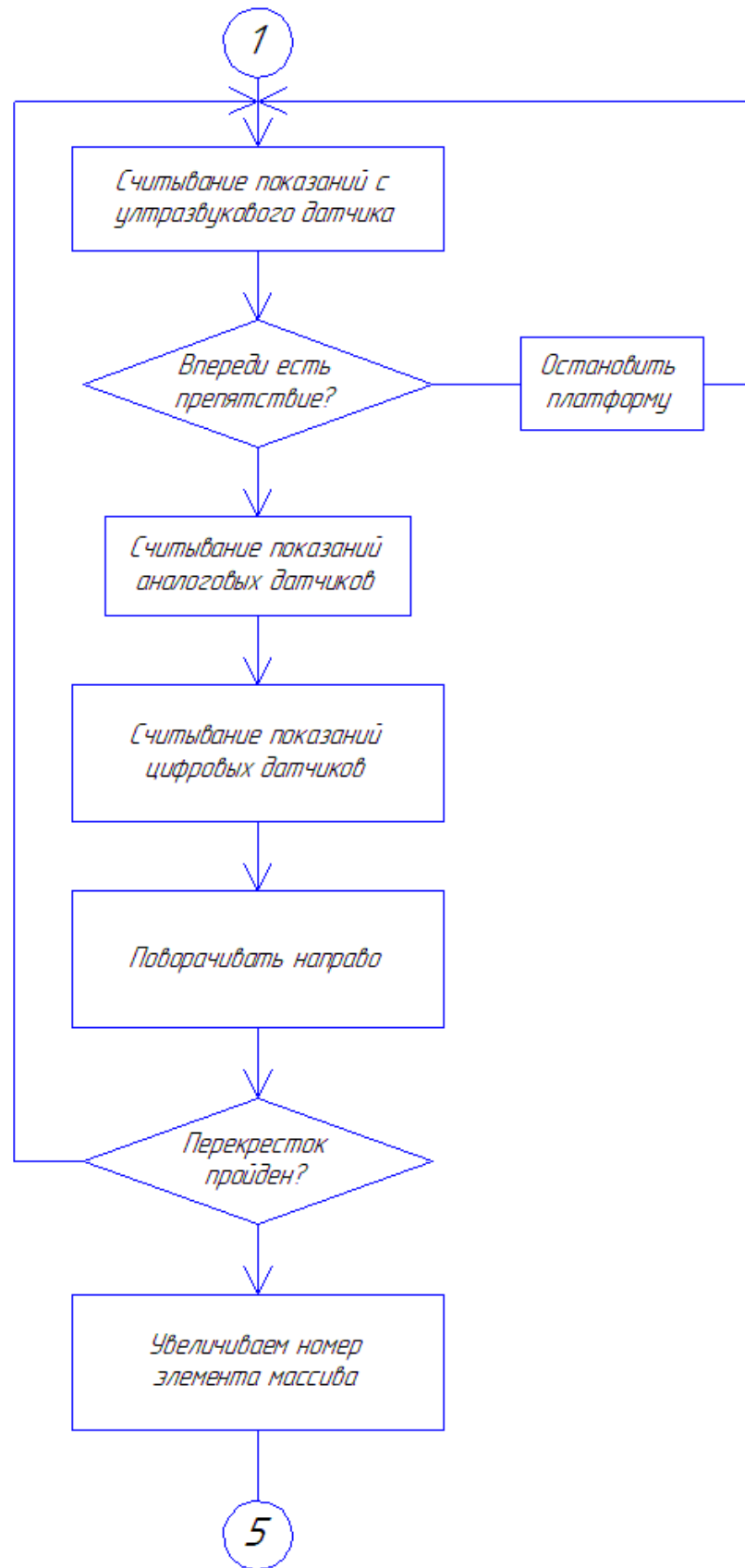


Рисунок 32б – Блок-схема алгоритма работы платформы

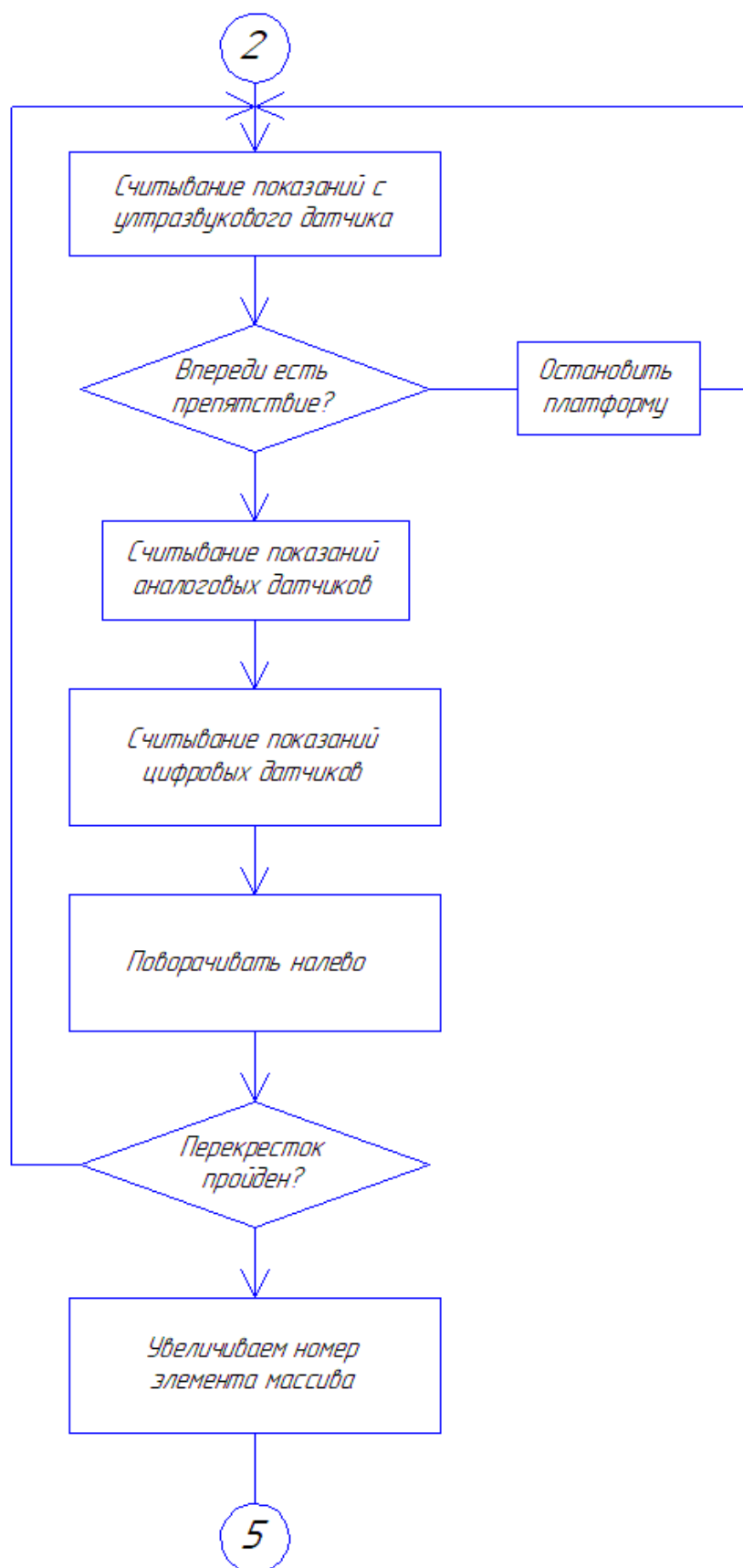


Рисунок 32в – Блок-схема алгоритма работы платформы

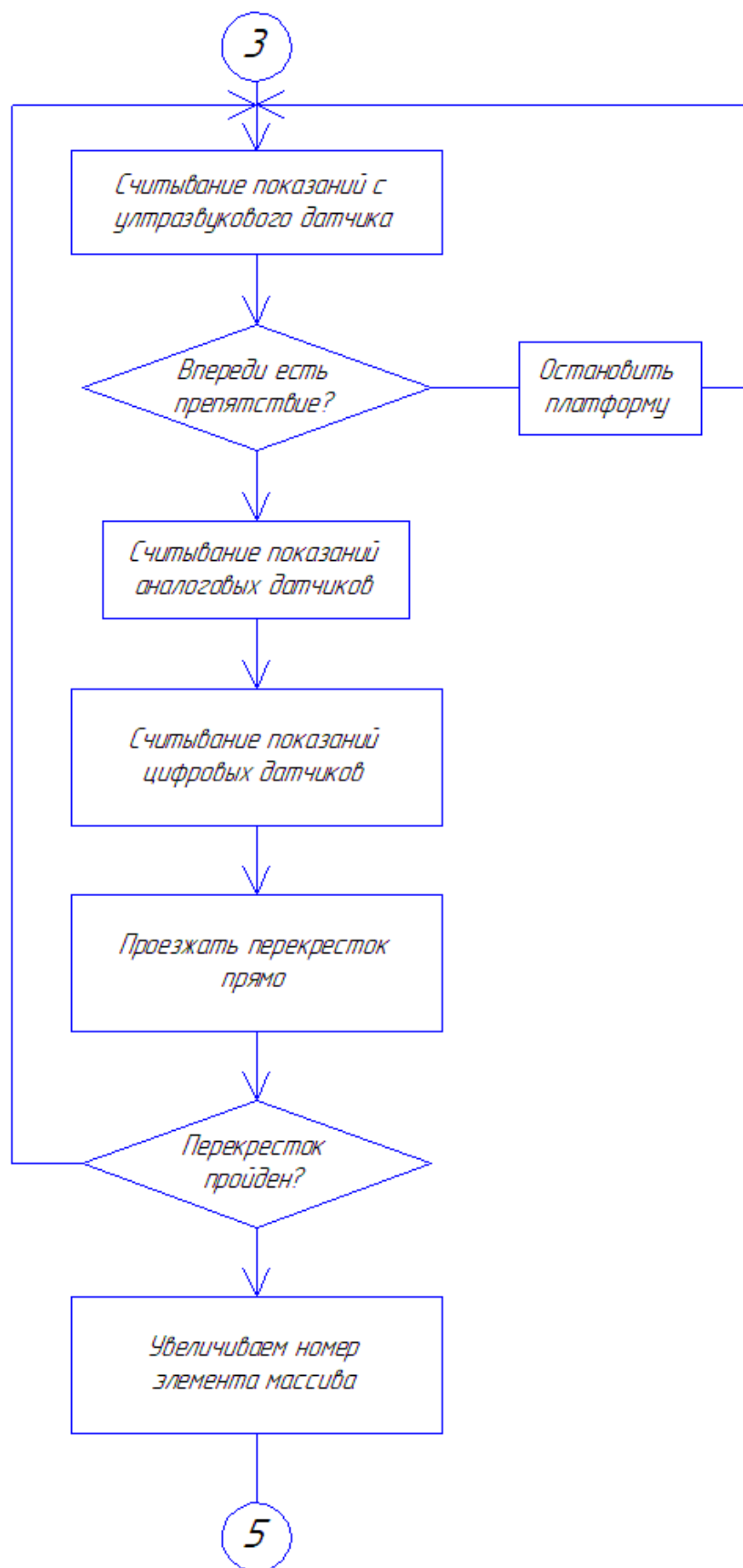


Рисунок 32г – Блок-схема алгоритма работы платформы

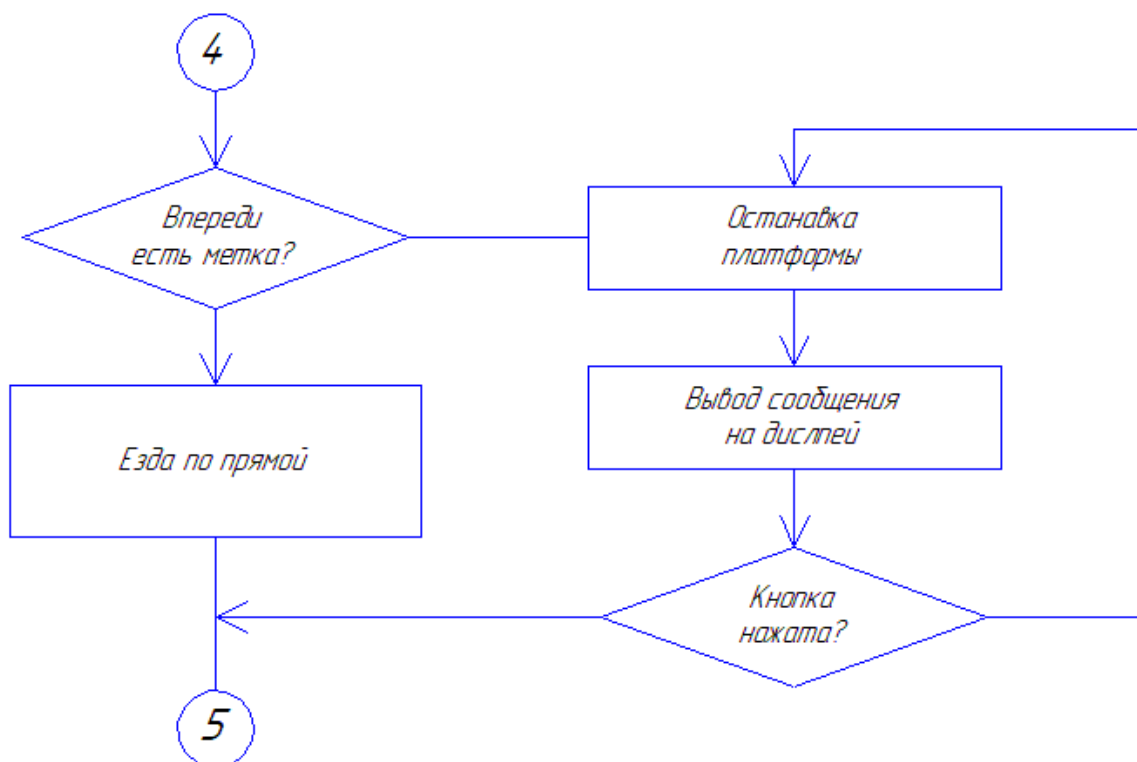


Рисунок 32д – Блок-схема алгоритма работы платформы

3.2 Разработка программной части устройства

Контроллер PRIZM подсоединяют к компьютеру при помощи типового кабеля с разъёмом USB. Напряжение подается из внешней перезаряжаемой, никельметаллогидридной аккумуляторной батареи TETRIX, рассчитанной на 12 В. В контроллере есть система привода сдвоенных высокоточных электродвигателей постоянного тока, каждый электродвигатель снабжён средствами поддержки квадратурного энкодера, благодаря чему осуществляется точное ПИД регулирование скорости и положения валов электродвигателей постоянного тока. Кроме того, есть шесть портов для стандартных сервоприводов и два порта для сервоприводов продолжительного вращения. Для внешних датчиков есть четыре цифровых порта, два аналоговых и один порт I2C. Цифровой порт D2 тоже можно

настроить на последовательную передачу данных при помощи соответствующей библиотеки в составе ПО Arduino. Все порты для цифровых датчиков можно настроить на ввод или вывод сигналов. Порты для аналоговых датчиков можно перевести в режим ввода аналоговых и вывода цифровых сигналов. Есть также два встроенных светодиода — по одному красного и зелёного цвета, — которые служат для подачи зрительных сигналов. Функциями контроллера PRIZM управляет микроконтроллер Atmega328P. Для загрузки в него программного кода через разъём USB используется начальный загрузчик Optiboot. Есть ещё два процессорных микроконтроллера, которые управляют функциями электродвигателей: микроконтроллер, управляющий электродвигателями постоянного тока и обеспечивающий сопряжение с энкодерами, и микроконтроллер, управляющий сервоприводами. Эти микроконтроллеры обмениваются сигналами с главным процессорным микроконтроллером через шину связи I2C. В каждый микроконтроллер, управляющий электродвигателями, встроено ПО, которому подчиняются все сложные функции управления электродвигателями постоянного тока, энкодерами и сервоприводами, так что главный процессор этим не обременён и может свободно управлять выполнением программного кода. Зелёная пусковая кнопка (Start) служит для запуска программы, подгруженной в контроллер. Красная кнопка сброса параметров (Reset) служит для остановки программы. В результате нажатия на красную кнопку сброса параметров главный процессорный микроконтроллер, микроконтроллеры управления электродвигателями постоянного тока и сервоприводами возвращаются в исходное состояние, а в памяти обнуляются все сохранённые значения.

Контроллер PRIZM построен на схеме Arduino, поэтому может работать с языками программирования, совместимыми с языком Arduino.

Единая среда разработки (IDE) Arduino представляет собой программное обеспечение с открытыми исходными кодами, позволяющее

составлять управляющие программы и отправлять их в исполнительные устройства. Удобный для разработчиков инструмент создания программ в виде текста, ПО Arduino (IDE) можно использовать на различных устройствах под управлением операционных систем Windows, Macintosh и Linux. Благодаря этому ПО Arduino (IDE) универсально в применении. Для диалога с PRIZM в ПО Arduino (IDE) используется язык программирования на основе языка Си. В ПО Arduino (IDE) отдельная программа называется скетчем.

Для программирования контроллера платформы и работы самой платформы мы будем пользоваться библиотеками. Библиотека является файлом (набором файлов), содержащим точно такой же код, на котором мы пишем скетч. Мы можем подключить библиотеку в свой код и использовать возможности, которые она даёт, а вариантов там весьма много: готовые “инструменты” для работы с внешними датчиками и модулями, для работы с внутренней периферией микроконтроллера (таймеры, АЦП, память), библиотеки различных математических инструментов и многое другое.

Плюс работы с библиотекой заключается в том, что нам необязательно знать, каким образом работает код внутри неё, мы пользуемся готовыми инструментами, которые предоставил разработчик библиотеки. Это экономит время на разработку определенных частей кода, а так же разгружает память микроконтроллера. Очень часто к библиотекам есть описание или документация и примеры использования.

Производитель робототехнического набора для создания дистанционно управляемых моделей серии TETRIX MAX так же разработал свою собственную библиотеку PRIZM для работы с контроллером, датчиками и двигателями.

Для работы с дисплеем мы тоже будем использовать библиотеку LiquidCrystal_I2C. Дисплей будет подключен к микроконтроллеру через

интерфейс I2C, поэтому нам потребуется дополнительная библиотека Wire для работы с этим интерфейсом.

По заданию платформа должна плавно передвигаться путем отслеживания черной линии. Для выполнения этой задачи мы будем использовать пропорционально интегрально дифференцирующее регулирование или сокращенно ПИД регулирование.

ПИД регулятор формирует управляющий сигнал, который является суммой трёх слагаемых: пропорционального, интегрального и дифференциального.

Пропорциональная составляющая P – отвечает за пропорциональное управление. Принцип действия в том, что выходной сигнал регулятора противодействует изменению регулируемой величины от некоторого заданного значения. Чем больше ошибка рассогласования, тем больше командное отклонение регулятора. Но недостатком пропорционального звена управления является то, что регулятор никогда не стабилизируется в в том значении, в котором он задан. Дальнейшее увеличение коэффициента всегда приводит к автоколебаниям. Для, того чтобы этого избежать, все используют интегральную и дифференциальные звенья. Интегральная составляющая I накапливает ошибку регулирования, что позволяет ПИД регулятору устранять статическую ошибку. Иными словами: интегральное звено постоянно вносит некоторое смещение и если система подвержена некоторыми постоянным ошибкам. А вот если же этих ошибок нет или они пренебрежительно малы, то интегральная составляющая сама будет вносить ошибку смещения. Самым главным недостатком интегрального звена управления является эффект насыщения интегратора. Дифференциальная составляющая D пропорциональна темпу изменения отклонения регулируемой величины и предназначена для противодействия отклонениям от целевого значения, которые прогнозируются в будущем. Примечательно то, что дифференциальная компонента устраняет затухающие колебания.

Дифференциальное регулирование особенно эффективно для процессов, которые имеют большие запаздывания. Недостатком дифференциального закона управления является его неустойчивость к воздействию шумов.

Рассмотрим ПИД регулирование на примере нашей платформы. Ошибкой `err` здесь будет являться разность между значениями двух аналоговых инфракрасных датчиков линии AL и AR, расположенных по обе стороны от линии. Если значения с датчиков одинаковы, то управляющее значение `u` равно нулю, и платформа должна ехать прямо. Если же значения различны, то `u` будет отлично от нуля, и следовательно платформе нужно будет повернуть в какую-то сторону. Для считывания значений с датчиков мы будем использовать функцию `analogRead()`. Для управления электродвигателями воспользуемся библиотечной функцией `prizm.setMotorPower(электродвигатель#, мощность)`, которая служит для регулирования уровня мощности и направления вращения электродвигателей. Параметр `мощность` определяет величину мощности, с которой следует вращать вал электродвигателя, и устанавливается в пределах от 0 до 100. Это процентный показатель, то есть «0» означает полное отключение, а «100» означает включение на полную мощность. Кроме того, для остановки электродвигателя есть два режима: остановка выбегом и остановка торможением. Если задана мощность «0», то электродвигатель остановится после выбега. Если задана мощность «125», то электродвигатель остановится после торможения. Для пропорционального переноса значения управляющего воздействия из текущего диапазона значений в диапазон значений мощности двигателя будем использовать функцию `map()`.

Для обнаружения перекрестков мы используем три цифровых инфракрасных датчика линии, расположенных так, как показано на рисунке 33. Для считывания значений с датчиков мы будем использовать функцию `digitalRead()`. Если все три датчика сработают одновременно, или одновременно сработает средний и правый датчик или средний и левый, то

перед нами будет метка или перекресток. Для создания такого условия будет использован условный оператор if.



Рисунок 33 – Расположение инфракрасных датчиков линии

Для создания маршрута платформы будем использовать массив, заполненный вручную определенными значениями. Далее при обнаружении перекрестка платформа будет обращаться к массиву, и смотреть, какое действие нужно совершить. Для выполнения этой задачи нам поможет оператор switch. Он сравнивает значение переменной массива со значением, определенным в операторах case. Когда найден оператор case, значение которого равно значению переменной, выполняется программный код в этом операторе. Вот так выглядит пример программной реализации оператора switch:

```

switch (var) {
  case 1:
    //выполняется, когда var равно 1
    break;
  case 2:
    //выполняется когда var равно 2
    break;
  default:
    // выполняется, если не выбрана ни одна альтернатива
    // default необязателен
}

```

Для решения задачи по проезду плавных перекрестков был разработан следующий алгоритм. Допустим нам нужно повернуть налево. Мы перестаем использовать правый аналоговый датчик линии для движения и дальше едем только на левом. Правый датчик в этот момент следит, пресек ли он прямую линию, по которой платформа двигалась до поворота. Если пересек, то дальнейшее движение опять осуществляется с помощью двух датчиков. Значение ошибки для нахождения управляющего воздействия рассчитывается немного иначе. Мы находим среднее значение между показаниями левого датчика на линии и вне линии, и при расчете ошибки из текущего показания датчика вычитаем найденное среднее значение.

Для остановки перед препятствием с помощью ультразвукового датчика воспользуемся библиотечной функцией `prizm.readSonicSensorCM()`. Данная функция считывает значения с датчика и преобразует их в значения расстояния, выраженные в сантиметрах. Далее с помощью оператора `if` создаем условие, при котором при определенном расстоянии платформа должна остановиться. В приложении Б приводится код разработанной программы.

Выводы по разделу

Подведем следующие выводы:

- на контроллере всего три аналоговых вывода, которые уже заняты, поэтому при дальнейшей модернизации платформы могут возникнуть трудности;
- для управления платформы с помощью ПИД регулятора решено не использовать интегральную составляющую, так как интегральное звено постоянно вносит некоторое смещение, если система подвержена некоторыми постоянным ошибкам, и соответственно, на прямолинейных участках трассы платформа вела бы себя некорректно;
- библиотека для программирования очень полезная вещь, позволяющая очень просто работать с датчиками и двигателями, управляемыми контроллером.

4. Конструкторско-экспериментальный раздел

4.1 Сборка устройства

В соответствии с принципиальной электрической схемой была произведена сборка платформы, установка всех датчиков и их подключение. Готовая собранная платформа представлена на рисунках 34-38.

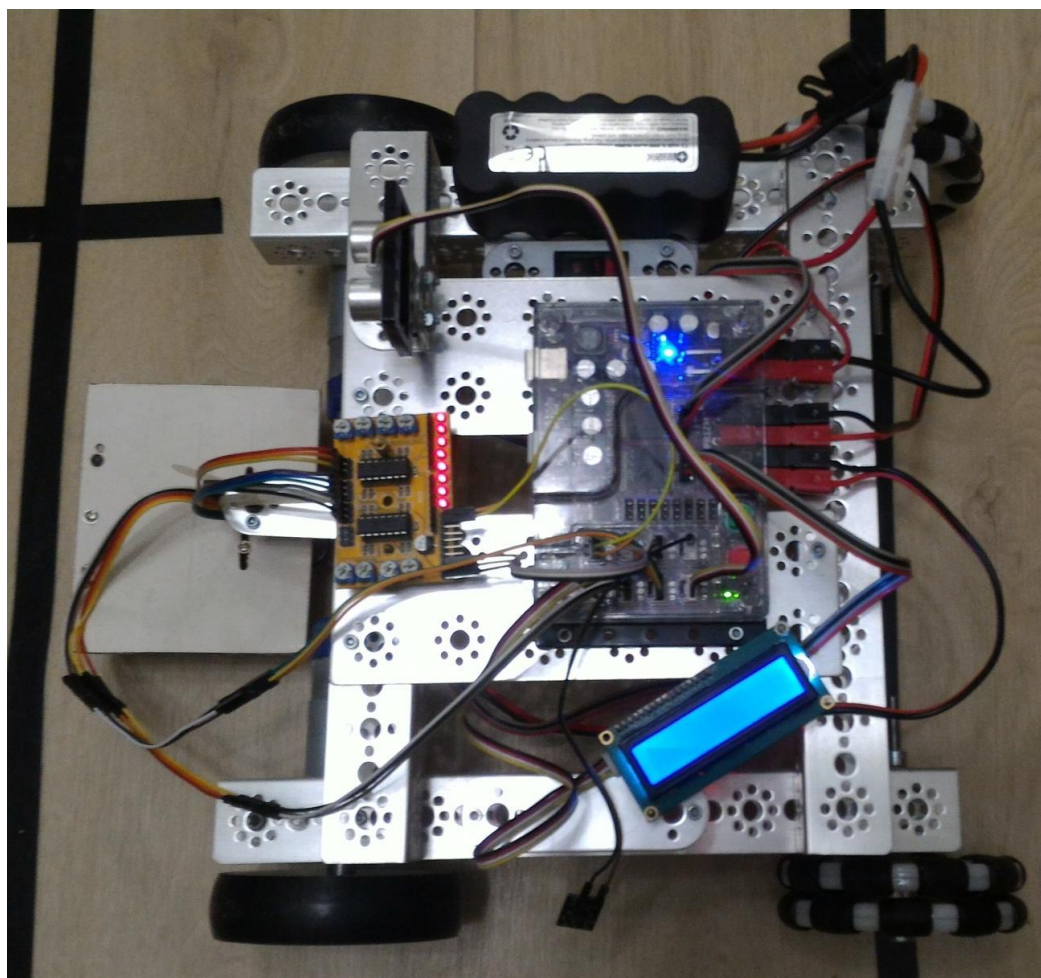


Рисунок 34 – Собранная платформа, вид сверху

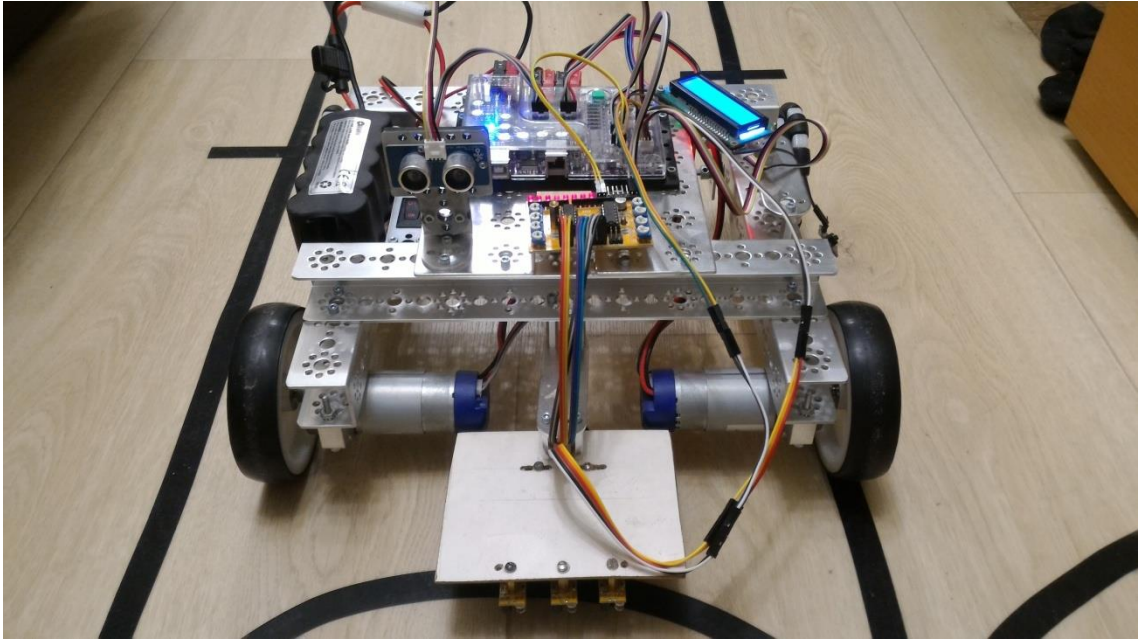


Рисунок 35 – Собранная платформа, вид спереди

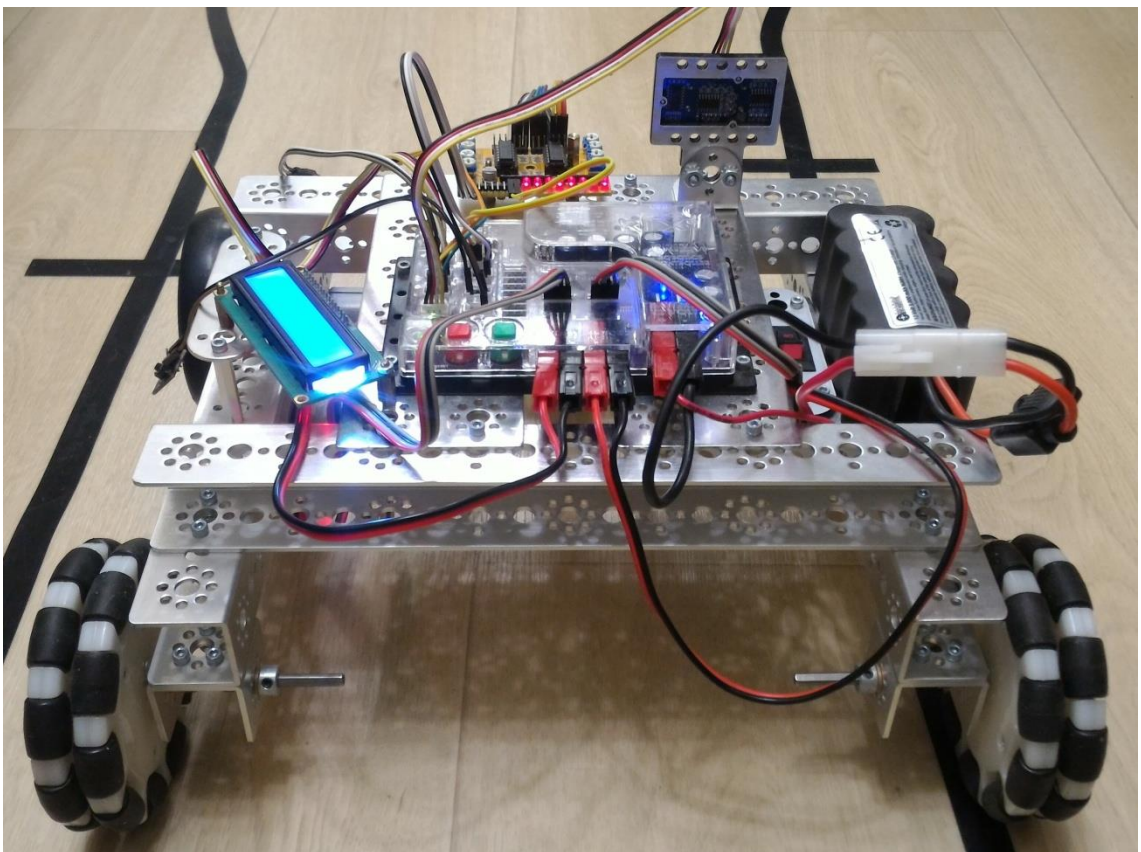


Рисунок 36 – Собранная платформа, вид сзади

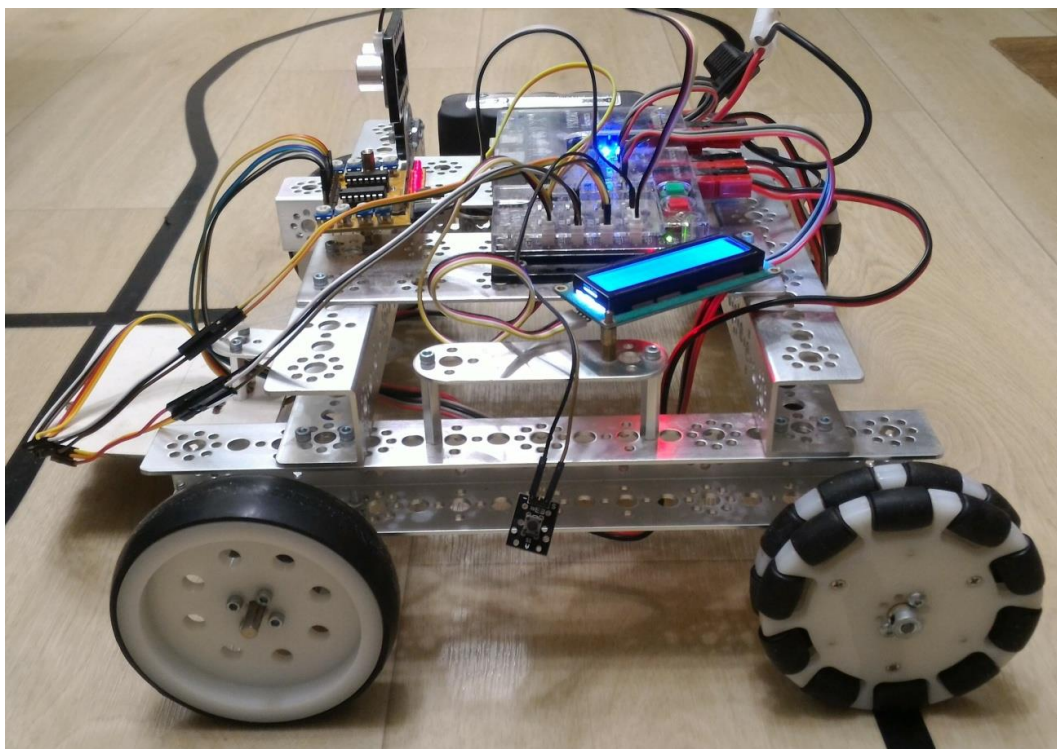


Рисунок 37 – Собранная платформа, вид сбоку

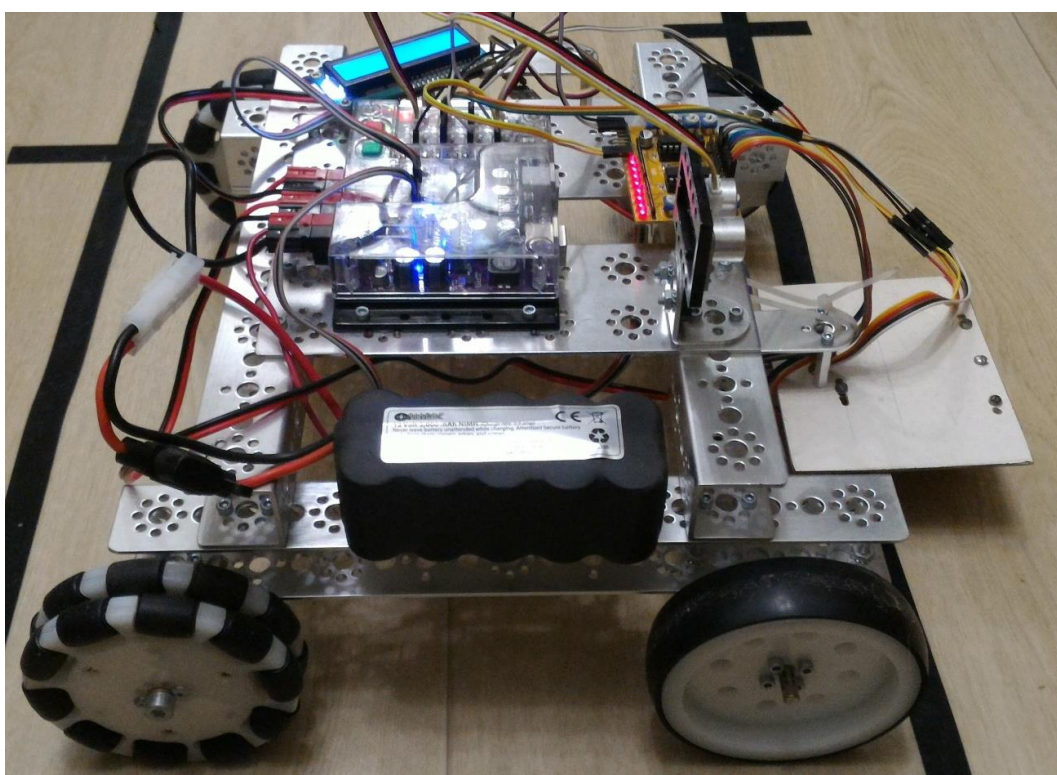


Рисунок 38 – Собранная платформа, вид сбоку

4.2 Отладка устройства и экспериментальные исследования

Первым делом нам нужно настроить ПИД регулятор, чтобы добиться плавного передвижения платформы. Существует несколько методов настройки, но мы будем использовать тот, который осуществляется с помощью последовательного подбора каждого из коэффициентов вручную. Для начала обнуляем все коэффициенты, ставим платформу на линию и запускаем. Платформа, скорее всего, будет рывками передвигаться по трассе. Путем увеличения пропорционального коэффициента, добиваемся стабильной езды платформы без выезда за пределы трассы. Далее по аналогии подбирают два других коэффициента, увеличивая и уменьшая их, для достижения плавного передвижения платформы.

После настройки ПИД регулятора, нужно настроить 3 цифровых датчика линии. Делается это с помощью подстроечных резисторов на восьмиканальном модуле МН. Располагаем платформу так, чтобы один из цифровых датчиков оказался над линией. На модуле МН есть индикаторные светодиоды, ориентируясь на которые можно настроить датчик. Крутим потенциометр до тех пор, пока светодиод не погаснет. Это будет означать, что датчик видит линию. После этого проделываем вышеупомянутые действия с двумя другими датчиками. Процесс регулировки показан на рисунке 39.

Данные датчики нужны для обнаружения перекрестков, поэтому нам так же необходимо отрегулировать расстояние между ними. Это делается для того чтобы при передвижении платформы на участках без перекрестков не было ложных срабатываний сразу двух датчиков. Если такое произойдет, то платформа посчитает, что перед ней перекресток. Для этого добавляем в программу блок кода, который будет следить за датчиками, и, в случае одновременного их срабатывания, выводить сообщение на дисплей. Далее ставим платформу на трассу, запускаем, и если произойдет одновременное

срабатывание, то увеличиваем расстояние между датчиками, и повторяем вышесказанные действия повторно.

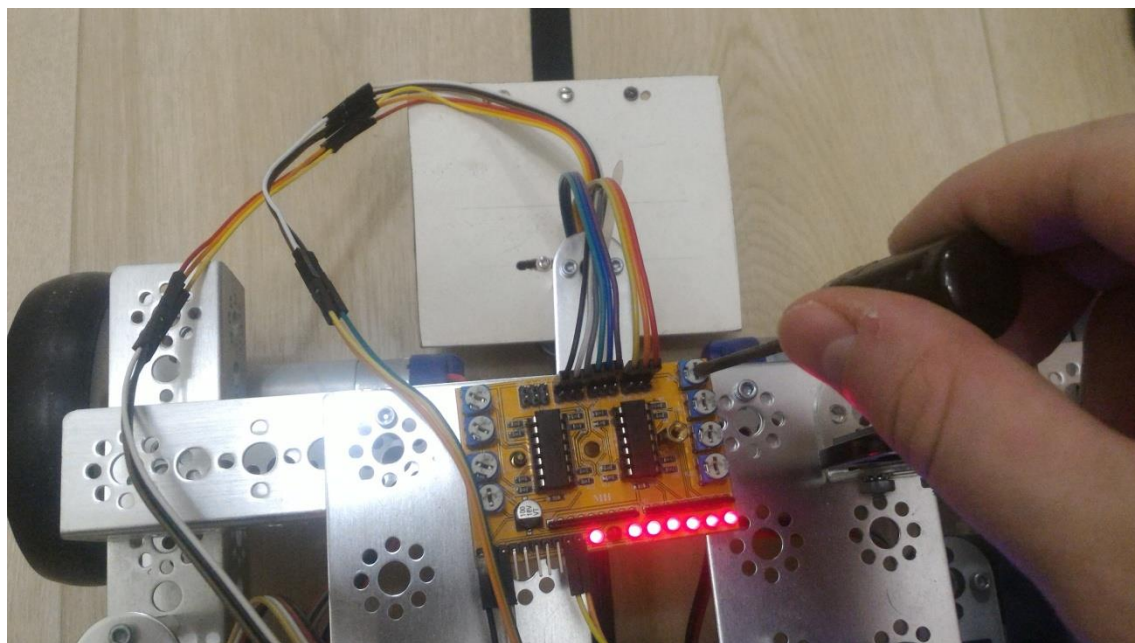


Рисунок 39 – Регулировка цифровых датчиков

Для полноценного испытания платформы была использована трасса, показанная на рисунке 40.

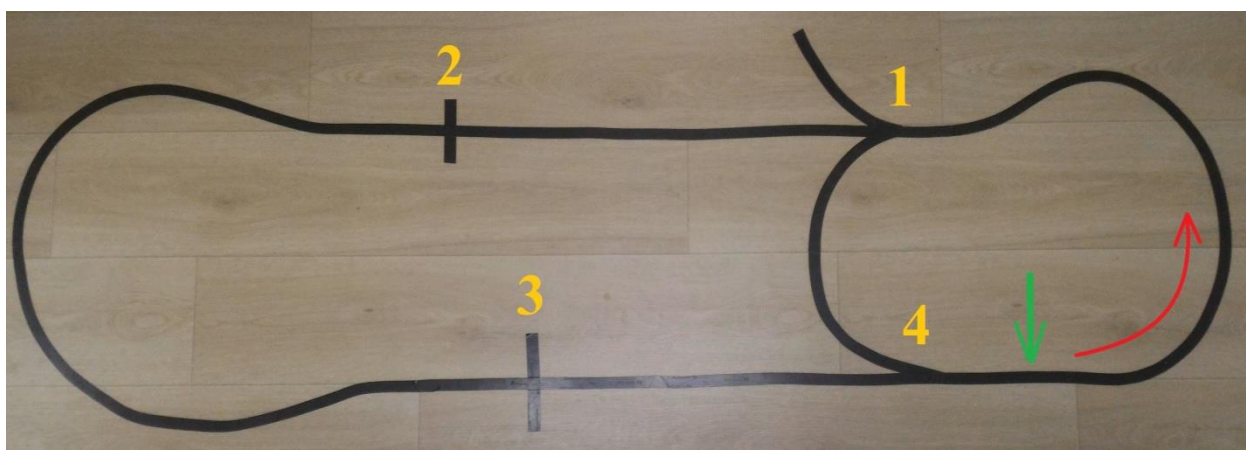


Рисунок 40 – Трасса для испытаний платформы

Цифрами на ней обозначены перекрестки с плавными поворотами и метки. Начало движения обозначено зеленой стрелкой, а направление после начала движения обозначено красной стрелкой.

Для прохождения данной трассы нам нужно вручную заполнить массив действий для каждого перекрестка. Программа написана таким образом, что платформа не различает метки и перекрестки, поэтому любое разветвление для нее – это перекресток.

Маршрут для движения будет следующий: первый перекресток проезжаем в прямом направлении; на втором перекрестке останавливаемся (это будет меткой для остановки) и ждем нажатия кнопки для продолжения движения; на третьем перекрестке останавливаемся (это будет еще одна метка для остановки) и ждем нажатия; на четвертом проезжаем прямо; далее на первом перекрестке поворачиваем налево; и на четвертом едем прямо. Далее платформа циклично будет ездить по этому маршруту.

В программе массив заполняется числами, которые соответствуют определенным действиям на перекрестках. В таблице 16 указаны действия на перекрестке и соответствующие им числа массива.

Таблица 16 – Действия на перекрестке

Действие на перекрестке	Соответствующее число массива
Прямо	0
Направо	1
Налево	2
Остановка	3

Еще раз кратко по порядку перечислим действия на перекрестках и соответствующие им числа:

- прямо (число 0);
- остановка (число 3);
- остановка (число 3);

- прямо (число 0);
- налево (число 2);
- прямо (число 0);

Так выглядит заполненный массив в программе для указанного выше маршрута:

```
int action[ ] = {0, 3, 3, 0, 2, 0}
```

Из-за особенностей программы, а так же для более корректного и плавного движения, на перекрестке с одним поворотом (например: перекресток 4) рекомендуется выбирать не проезд прямо, а поворот, противоположный повороту на перекрестке. Например: при передвижении платформы от третьего перекрестка к четвертому действие на четвертом перекрестке рекомендуется указать как поворот направо, а при передвижении от первого перекрестка к четвертому – поворот налево. Тогда окончательный вид массива будет такой:

```
int action[ ] = {0, 3, 3, 1, 2, 2}
```

В ходе испытаний было выявлено, что максимальная скорость платформы, при которой она работает корректно и не покидает трассу, составляет 6 см/с или 0.06 м/с. Это обуславливается тем, что при увеличении скорости цифровые датчики линии не успевают заметить перекресток.

При прохождении плавного поворота было выявлено, что минимальный радиус этого поворота составляет 15 см.

При следовании по маршруту платформа ни разу не сбилась, корректно прошла все перекрестки и метки.

При ходовых испытаниях под нагрузкой проблем не выявлено. Благодаря энкодерам электродвигателей контроллер поддерживает заданную скорость постоянной при любых нагрузках. Мощности аккумуляторной батареи для двигателей хватает с запасом.

Вывод по разделу

Подведем следующие выводы:

- задачи, поставленные в техническом задании, выполнены;
- найдена максимальная скорость передвижения платформы для корректного прохождения трассы;
- определен минимальный радиус плавного поворота, который платформа может корректно проехать;
- произведены испытания под нагрузкой, в ходе которых проблем не было выявлено.

Заключение

В ходе выполнения бакалаврской работы, была разработана автоматизированная мобильная платформа, которая способна передвигаться по трассе из черной линии, отслеживая ее инфракрасными датчиками линии. Так же платформа способна проезжать плавные повороты и перекрестки в соответствии с заданным ей маршрутом и останавливаться на специальных метках, при этом постоянно отслеживая пространство перед собой для предотвращения столкновения с неожиданным препятствием на трассе. В процессе выполнения была рассмотрена актуальность темы и существующие робототехнические решения в сфере складской логистики. Был произведен анализ исходных данных, который заключался в изучении технических характеристик робототехнического набора для создания дистанционно управляемых моделей серии TETRIX MAX и входящего в его комплектацию робототехнического контроллера TETRIX PRIZM. Разработана структурная схема, показывающая количество устройств в системе, а так же их взаимодействия. Был произведен подбор, сравнение и выбор необходимых компонентов, таких как: инфракрасные датчики линии, дисплей LCD 1602 и тактовая кнопка. Разработана принципиальная электрическая схема, а так же перечень элементов к ней. Разработан алгоритм работы платформы и его программная реализация, в которой была применена специализированная библиотека для удобной работы с микроконтроллером, двигателями и датчиками. Произведена сборка платформы, установка микроконтроллера, подключение всех компонентов по принципиальной электрической схеме, а также настройка и калибровка инфракрасных датчиков линии. Была произведена загрузка и отладка управляющей программы. Описан метод задания маршрута для платформы при помощи массива. Показаны изображения готовой платформы и трассы для испытаний. Представлена программа и описаны результаты экспериментальных исследований.

Список используемой литературы

1. Amazon. Алгоритмы работы самого крупного ритейлера в мире [Электронный ресурс]. URL: <https://habr.com/ru/company/pochtoy/blog/406783/> (дата обращения: 07.05.2018)
2. Робототехника Amazon Robotics [Электронный ресурс]. URL: <https://mentamore.com/robototехника/amazon-robotics.html> (дата обращения: 07.05.2018)
3. Ronavi-Robotics [Электронный ресурс]. URL: <https://ronavi-robotics.ru/> (дата обращения: 07.05.2018)
4. TETRIX MAX Programmable Robotics Set [Электронный ресурс]. URL: <https://www.pitsco.com/RU/Shop/Robotics/TETRIX-MAX-Programmable-Robotics-Set>
5. TETRIX PRIZME programming guide [Электронный ресурс]. URL: <https://asset.pitsco.com/sharedimages/resources/tetrix-prizm-programming-guide.pdf> (дата обращения: 07.05.2018)
6. Оптопара принцип работы [Электронный ресурс]. URL: <https://principraboty.ru/optopara-princip-raboty-optrony-princip-raboty/> (дата обращения: 07.05.2018)
7. Устройство и работа датчика линии [Электронный ресурс]. URL: <http://elektrik.info/device/1588-kak-ustroen-i-rabotaet-datchik-linii.html> (дата обращения: 07.05.2018)
8. KY-032 модуль с инфракрасным датчиком [Электронный ресурс]. URL: <https://umnyjdomik.ru/ky-032-modul-s-infrakrasnym-datchikom-obnaruzheniya-prepyatstvuj-dlya-arduino.html> (дата обращения: 07.05.2018)
9. Инфракрасный датчик FC-51 [Электронный ресурс]. URL: <https://robo02.ru/2018/03/22/infrared-sensor-of-distance-touch-fc-51/> (дата обращения: 07.05.2018)

10. Модуль датчика линии TCRT5000 [Электронный ресурс]. URL: <https://robot-kit.ru/3094/> (дата обращения: 07.05.2018)
11. Бесконтактный датчик YL-73 [Электронный ресурс]. URL: <http://arduino-kid.ru/collection/modules/YL-73> (дата обращения: 07.05.2018)
12. Подключение дисплея LCD 1602 к arduino по i2c / ИС [Электронный ресурс]. URL: <https://arduinomaster.ru/datchiki-arduino/lcd-i2c-arduino-displey-ekran/> (дата обращения: 07.05.2018)
13. Блок-схемы алгоритмов [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%BA-%D1%81%D1%85%D0%B5%D0%BC%D0%B0> (дата обращения: 07.05.2018)
14. Справочник языка Ардуино [Электронный ресурс]. URL: <http://arduino.ru/Reference> (дата обращения: 07.05.2018)
15. PID controller [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/PID_controller (дата обращения: 07.05.2018)
16. Warehouse Robots for Retail Automation [Электронный ресурс]. URL: <https://www.nanalyze.com/2018/08/warehouse-robots-retail-automation/> (дата обращения: 07.05.2018)
17. Types of autonomous mobile robots, and their warehouse use cases [Электронный ресурс]. URL: <https://www.supplychaindive.com/news/4-types-of-autonomous-mobile-robots-and-their-warehouse-use-cases/529548/> (дата обращения: 07.05.2018)
18. Каталог электронных компонентов [Электронный ресурс]. URL: <http://www.chip-dip.ru> (дата обращения: 17.05.2019)
19. Каталог электронных компонентов [Электронный ресурс]. URL: <https://www.impulsi.ru/> (дата обращения: 17.05.2019)
20. TETRIX PRIZM Robotics Controller [Электронный ресурс]. URL: <https://www.pitsco.com/RU/Shop/Robotics/TETRIX-PRIZM-Robotics-Controller> (дата обращения: 17.05.2019)

Приложение А

Перечень элементов к схеме принципиальной электрической

Поз.	Наименование	Кол.	Примечание
	Датчики		
DA1...DA5	Инфракрасный датчик линии	6	
DD3	Ультразвуковой датчик расстояния TETRIX	1	
M1...M2	Мотор TorqueNADO	2	
DD2	Четырехканальный модуль МН	1	
DD1	Контроллер TETRIX PRIZM	1	
HG1	Жидкокристаллический дисплей LCD-	1	
B	Аккумуляторная батарея TETRIX	1	

					20-110304.113/09.311 ПЗ					
Изм.	Лист	№ докум.	Подп.	Дата						
Разраб.	Пименов Е.Е.				Автоматизированная мобильная платформа Перечень элементов	Лит.	Лист	Листов		
Пров.	Позднов М. В.							1	1	
Н.контр.	Позднов М. В.					ТГУ гр. Элб-1601а				
УТВ.	Шевцов А. А.									

Приложение Б

Программа автоматизированной мобильной платформы

```
#include <GyverButton.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);
#include <PRIZM.h>
PRIZM prizm;

#define maxSpeed 21
#define L1 9 // левый крайний датчик
#define R1 2 // правый крайний датчик
int sensorWhite[2] = {40, 60}; // показания датчиков вне линии в порядке A0, A1...
int sensorBlack[2] = {690, 690}; // показания датчиков на линии в порядке A0, A1...

int action[] = {0, 3, 3, 1, 2, 2}; // массив для действия на перекрестке
int i = 0; //индекс для массива
int d = 0;
//действия на перекрестках
#define STATE_FORWARD 0
#define STATE_LEFT 2
#define STATE_RIGHT 1
#define STATE_PER 5
#define STATE_STOP 3
#define STATE_МЕТКА 4

GButton myButt(4, HIGH_PULL, NORM_OPEN);

// прототипы используемых функции
void runForward();
void steerRight();
void steerLeft();
void steerStop();
void steerPer();

//ПИД

// Пропорциональная
float up; // пропорциональная составляющая
float kp = 1.7; // коэффициент пропорциональной составляющей

// Дифференциальная
float ud; // дифференциальная составляющая
float dif = 0;
```

```
float kd = 60; // коэффициент дифференциальной составляющей
int oldErr; // значение ошибки, полученной на предыдущей итерации
unsigned long time = 0; // время получения текущей ошибки
unsigned long pastTime = 0; // время получения предыдущей ошибки
```

Продолжение приложения Б

```
unsigned long time1 = 0;
unsigned long lastTime1 = 0;
```

```
unsigned long time4 = 0;
unsigned long lastTime4 = 0;
```

```
int err; //разница показаний датчиков(ошибка)
int pause = 20; // время задержки
```

```
int flag;
int Left;
int Right;
int targetState = STATE_FORWARD;
```

```
68astime button_flag = 0;
68astime button;
```

```
68astime Dleft;
68astime Dright;
68astime Dfor;
```

```
int AL;
int AR;
int sonic;
int Lmap;
int Rmap;
float u;
int gost = 0;
int encod = 0;
int fkl = 0;
```

```
void setup()
{
```

```
pinMode(17, INPUT);
pinMode(2, INPUT);
pinMode(5, INPUT);
pinMode(3, INPUT);
pinMode(4, INPUT_PULLUP);
```

```
lcd.init();
lcd.backlight();
```

```

prizm.PrizmBegin();
prizm.setMotorInvert(1, 1);
Serial.begin(9600);
}

```

Продолжение приложения Б

```

void loop() {
  myButt.tick();

  sonic = prizm.readSonicSensorCM(17);
  Dleft = digitalRead(2);
  Dright = digitalRead(5);
  Dfor = digitalRead(3);
  button = !digitalRead(4);

  AL = analogRead(A1);
  AR = analogRead(A2);

  Lmap = map(AL, sensorWhite[0], sensorBlack[0], 0, 100);
  Rmap = map(AR, sensorWhite[1], sensorBlack[1], 0, 100);

  Lmap = constrain(Lmap, 0, 100);
  Rmap = constrain(Rmap, 0, 100);

  switch (gost)
  {
    case 0:

      err = Lmap - Rmap;
      up = kp * err;
      time = millis(); // текущее время
      ud = kd * (err - 69asti) / (time - 69astime); // de/dt
      69asti = err;
      69astime = time;
      u = up + ud; // управляющее воздействие
      u = constrain(u, -200, 200);
      Left = map(u, 0, 110, maxSpeed, 0);
      Left = constrain(Left, -maxSpeed, maxSpeed);
      Right = map(u, 0, -110, maxSpeed, 0);
      Right = constrain(Right, -maxSpeed, maxSpeed);
      break;

    case 1:

      err = 50 - Rmap;
      up = kp * err;
      time = millis(); // текущее время

```

```

ud = kd * (err - 70asti) / (time - 70astime); // de/dt
70asti = err;
70astime = time;
u = up + ud; // управляющее воздействие
u = constrain(u, -200, 200);

```

Продолжение приложения Б

```

Left = map(u, 0, 110, maxSpeed, 0);
Left = constrain(Left, -maxSpeed, maxSpeed);
Right = map(u, 0, -110, maxSpeed, 0);
Right = constrain(Right, -maxSpeed, maxSpeed);

```

```
break;
```

```
case 2:
```

```

err = Lmap - 50;
up = kp * err;
time = millis(); // текущее время
ud = kd * (err - 70asti) / (time - 70astime); // de/dt
70asti = err;
70astime = time;
u = up + ud; // управляющее воздействие
u = constrain(u, -200, 200);
Left = map(u, 0, 65, maxSpeed, 0);
Left = constrain(Left, -maxSpeed, maxSpeed);
Right = map(u, 0, -65, maxSpeed, 0);
Right = constrain(Right, -maxSpeed, maxSpeed);

```

```
break;
```

```
}
```

```
if ((Dleft && Dfor && flag == 0) || (Dright && Dfor && flag == 0) || (Dleft && Dfor && Dright && flag == 0))
```

```
{
```

```

flag = 4;
targetState = STATE_PER;

```

```
}
```

```
if (myButt.isPress() && flag == 4) {
```

```

lcd.clear();
targetState = STATE_FORWARD;
flag = 5;

```

```
}
```

```
if (i > 5) i = 0;
```

```
if (sonic < 30) {
```

```
prizm.setMotorPower(1, 125);
```

```
prizm.setMotorPower(2, 125);
```

```
} else {
```

```
    switch (targetState)
```

Продолжение приложения Б

```
    {
        case STATE_FORWARD:  runForward();    break;
        case STATE_LEFT:     steerLeft();     break;
        case STATE_RIGHT:    steerRight();    break;
        case STATE_STOP:     steerStop();     break;
        case STATE_PER:      steerPer();      break;
        case STATE_METKA:    steerStop();     break;
    }
}
}
```

```
void steerPer()
```

```
{
    switch (action[i])
    {
        case 0:  runForward();
                break;
        case 1:  steerRight();
                break;
        case 2:  steerLeft();
                break;
        case 3:  steerStop();
                break;
    }
}
```

```
void runForward()
```

```
{
    lcd.setCursor(0, 1);
    lcd.print("F");
    gost = 0;

    prizm.setMotorPower(1, Left);
    prizm.setMotorPower(2, Right);

    if (flag == 4 && (DLeft == 0 && DFor) && (DRight == 0 && DFor)) {
        targetState = STATE_FORWARD;
        flag = 0;
        i++;
    }
}
```

```

if (flag == 5) {
    targetState = STATE_FORWARD;
    flag = 0;
    i++;
}

```

Продолжение приложения Б

```

}
void steerRight()
{
    gost = 1;
    lcd.setCursor(0, 1);
    lcd.print("R");
    prizm.setMotorPower(1, Left);
    prizm.setMotorPower(2, Right);

    if (AL > 400 && d == 0) d++;

    if (AL < 100 && d == 1) {
        flag = 0;
        targetState = STATE_FORWARD;
        i++;
        d = 0;
    }
}
void steerLeft()
{
    gost = 2;
    lcd.setCursor(0, 1);
    lcd.print("L");
    prizm.setMotorPower(1, Left);
    prizm.setMotorPower(2, Right);
    fkl = 1;

    if (fkl == 1 && d == 0) {
        d++;
        prizm.resetEncoders();
    }

    encod = prizm.readEncoderDegrees(2);

    if (encod > 360 && d == 1) {
        flag = 0;
        targetState = STATE_FORWARD;
        i++;
        fkl = 0;
        prizm.resetEncoders();
        d = 0;
    }
}

```



```
void steerStop()
{
  prizm.setMotorPower(1, 0);
  prizm.setMotorPower(2, 0);
}
```

Продолжение приложения Б

```
lcd.setCursor(4, 0);
lcd.print("Please,");
lcd.setCursor(1, 1);
lcd.print("press button!");
}
```