

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)

11.03.04 Электроника и микроэлектроника

(код и наименование направления подготовки)

Электроника и робототехника

(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Система позиционирования автономной транспортной тележки

Обучающийся

И.А Жегалин

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., доцент, А. В. Прядилов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н., доцент, О.В. Лебединская

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Название бакалаврской работы: «Система позиционирования автономной транспортной тележки».

Данная бакалаврская работа состоит из пояснительной записки представленной на сорока трех страницах и включающей в себя: девятнадцать рисунков, одной таблицы, список источников из двадцати наименований, в том числе пять источников на иностранном языке и три страницы приложений, и изображений на шести листах формата А1.

Целью работы является создание системы позиционирования транспортной тележки на основе умной камеры PiXuCam 2 а также создание макета тележки.

Задачи стоящие перед этой работой:

- Обзор существующих решений.
- Разработка электрических схем и проведение необходимых расчётов.
- Выбор элементов схем.
- Написание программного обеспечения для системы управления.
- Сборка готового макета, загрузка в него программного кода.
- Проведение теста системы позиционирования.
- Сбор макета автономной тележки.

Результатом выполнения бакалаврской работы является система позиционирования, основанная на работе умной камеры PiXuCam 2.

Эта работа представляет интерес для широкого круга читателей. Это будет полезно, как для людей интересующихся робототехникой и электроникой, так и для людей заинтересованных работе умных камер и составления систем позиционирования. Изготовленный макет будет включён в перечень лабораторных стендов университета, используемых в процессе обучения.

ABSTRACT

The topic of the given graduation work is: « Autonomous transport cart positioning system ».

The graduation work consists of an explanatory note presented on forty-three pages and includes: nineteen figures, one table, a list of from twenty titles references including five foreign sources and three appendices, and the graphic part on six A1 sheets.

The aim of the work is to create a transport trolley positioning system based on the PixyCam 2 smart camera, as well as to create a cart layout.

Tasks facing this work:

- Overview of existing solutions.
- Development of electrical circuits and carrying out the necessary calculations.
- Selection of circuit elements.
- Writing software for the control system.
- Assembling a ready-made prototype, loading program code into it.
- Conducting a positioning system test.
- Collecting the layout of an autonomous cart.

The result of the bachelor's work is a positioning system based on the work of the PixyCam 2 smart camera.

This work is of interest to a wide range of readers. This will be useful both for people interested in robotics and electronics, and for people interested in the work of smart cameras and the creation of positioning systems. The manufactured model will be included in the list of laboratory stands of the university used in the learning process.

Содержание

Введение.....	5
1 Обзор и анализ автономных транспортных систем.....	7
1.1 Анализ основных видов складских мобильных роботов.....	7
1.2 Анализ систем позиционирования роботов.	8
1.2.1 Умные камеры.....	11
1.2.2 Программируемая система разработки	15
2 Разработка структурной схемы автономной транспортной тележки.....	17
2.1 Обзор и выбор компонентов транспортной тележки.....	17
2.1.1 Плата управления.....	17
2.1.2 Модуль с камерой PiXuCam2	20
2.1.3 Моторы.....	21
2.1.4 Драйвер моторов	23
2.2 Итоговая схема.....	25
3 Разработка алгоритма работы системы позиционирования автономной транспортной тележки	26
4 Программная часть	28
4.1 Первоначальная настройка камеры PiXuCam 2	28
1.2 Программная часть системы позиционирования	34
Заключение	42
Список используемых источников.....	43
Приложение А - Структурная схема автономной тележки	45
Приложение Б – Блок схема работы системы позиционирования автономной транспортной тележки.	46

Введение

В современном мире автономная техника становится все более востребованной. Так как в абсолютно любой сфере деятельности человека нашлось место роботам: наука, военное дело, медицина и прочее. Дома человеку помогают роботы - пылесосы, в воздухе уже давно летают БПЛА – беспилотные летательные аппараты. Роботы выполняют задания на дне океана или в жерле вулкана. Словом, нет ни одной сферы деятельности, где человеку не хотелось бы облегчить себе работу. Именно для этого люди и придумали роботов.

Не обошел стороной прогресс даже производство: фабрики, заводы, склады – это места, где также задачи выполняют роботизированные комплексы. К примеру, на АВТОВАЗе применяются роботы для сварки кузовов, штамповки и сборки деталей. Все это позволяет выпускать сравнительно больше автомобилей чем в то время, когда всеми этими делами занимался исключительно человек. Также роботов используют для работы на складах для, например, транспортировки грузов от одного манипулятора до другого. Такие мобильные транспортные тележки обладают машинным зрением или системой позиционирования, которая помогает роботизированной тележке ориентироваться в пространстве и выполнять поставленную задачу.[15]

Такие автономные транспортные тележки намного облегчают работу сотрудникам складов и фабрик. Благодаря роботам улучшаются логистические цепочки, происходит экономия за счет увеличения производительности и снижения затрат на оплату низкоквалифицированным работникам склада. [6]

Машинное зрение - это способность роботов «видеть» мир вокруг себя, используя камеры, датчики света и другие устройства. Современные роботы могут использовать машинное зрение для различных задач, в том числе для

определения препятствий, навигации, распознавания объектов и распознавания лиц.

Большинство современных роботов оснащены камерами, которые служат глазами. Эти камеры могут быть различного типа: от простых монохромных камер до продвинутых 3D-камер. Роботы также могут использовать другие датчики, такие как ультразвуковые датчики или лидары, чтобы получить дополнительную информацию о мире вокруг них.

Обладая машинным зрением, транспортные тележки могут самостоятельно определять положение в пространстве, выполнять заданную работу, перемещаться по складу и проходить заранее заданный маршрут, тем самым ускоряя и оптимизируя работу на производстве. Автоматизация складов подобными роботами имеет положительную тенденцию в мире, так как в перспективе это позволяет предприятию выходить на новый уровень производства и задействовать людей для более сложных задач.

1 Обзор и анализ автономных транспортных систем

1.1 Анализ основных видов складских мобильных роботов

В настоящее время существуют разные виды складских роботов. В зависимости от того, какую функцию на складе будет выполнять робот, различают несколько видов:

- паллетайзеры (штабеллеры) – это манипуляторы, предназначенные для переноса и складирования разных предметов на палеты для перемещения;

- робокары (буксировщики) – данные роботы используют для транспортировки грузов на разные расстояния по заранее заданной человеком программе;

- сортировщики – в основном данный тип роботов распространён в сфере онлайн заказов, так как эти роботы способны сформировать и отсортировать заказ;

- тележки – эти роботы предназначены для транспортировки грузов по помещению склада. Также более современные модели тележек способны самостоятельно разгружаться, расставлять грузы по стеллажам и контейнерам и осуществлять работы по погрузке и разгрузке товаров;

- дроны – если у склада высокие потолки, то некоторые компании используют автоматические дроны для обслуживания таких помещений.[7]

Также существует классификация роботов по степени автоматизации:

- автоматизированные управляемые транспортные средства (AGV – Automated Guided Vehicles) – данные транспортные средства перевозят товары, грузы и прочее по складу и производственным помещениям. Такие транспортные средства обычно двигаются по заранее обозначенному маршруту, который наносится, к примеру, магнитной лентой, полосами на полу, разнообразными датчиками. Также могут быть и физические направляющие, такие как желоба или рельсы;

— автоматизированные управляемые системы создания и поиска (AS/RS – Automated Storage and Retrieval Systems) – это компьютерная система, предназначенная для упрощения поиска и размещения товаров или грузов на складе. Данная система функционирует, как правило, в тандеме с системами управления складами (WMS – Warehouse Management System). Представляет из себя эта система краны или тележки, которые перемещаются на определенных фиксированных путях и используются в основном для ускорения выполнения заказов и обработки других материалов;

— автономные мобильные роботы (AMR -Automated Mobile Robot) – данный вид роботов может автономно перемещаться между препятствиями на территории производства или склада. Выполняя функцию контейнера для сборов заказов или грузов, их используют для подбора товаров. Эти роботы внутри содержат разные датчики , с помощью которых робот не только перемещаются между разными точками на складах или производстве, но и имеют полную программную навигацию по производству и складу. В отличие от AGV, этим роботам не нужны направляющие для перемещения.[12]

Еще на производстве могут быть роботы – манипуляторы, которые выполняют задачу по типу «поднять и переложить». Манипуляторы можно разместить на моторизированной базе для перемещения и манипулирования предметами. Такие манипуляторы используются для распределения предметов на складах и распределительных пунктах. Также эти роботы могут использоваться непосредственно людьми.

1.2 Анализ систем позиционирования роботов.

Для любых роботов необходима система навигации в среде, в которой робот находится. Навигация – это способность робота определить собственное расположение в пространстве опираясь на имеющуюся в программе систему отсчета, а после просчитывать путь к заданной цели и по маршруту. Для того, чтобы навигация робота начала работать, ему необходимо представление об

окружающей среде, то есть карта или любое другое средство передачи информации об окружающей среде. [14]

Навигация может состоять из трех основных частей[13]:

- определение положения в пространстве;
- прокладка пути;
- построение карты (интерпретация карты).

Также определенные системы навигации используют принцип одновременной локализации и отображения в пространстве (SLAM - Simultaneous Localization and Mapping) чтобы создавать трехмерные карты окружения, в которых находятся роботы. Она используется для определения местоположения и ориентации автономных роботов на неизвестной им местности, а также для обновления или дополнения уже известных карт окружающего пространства. На рисунке 1 показана карта помещения, составленная методом SLAM. Робот сканирует окружающее пространство при помощи сенсоров и постепенно строит карту местности. Информация о местоположении робота получается из одометрии, которая может быть получена различными способами. При выстраивании карты робот сверяется с ней и корректирует показания одометрии, если они не соответствуют показаниям карты.

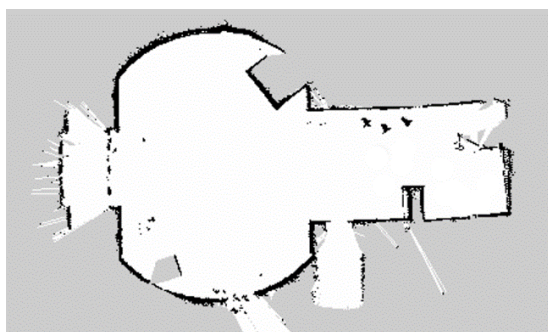


Рисунок 1 - Карта, составленная с помощью системы SLAM

Построение карты представляет из себя процесс сбора и интеграции информации с различных датчиков одного или нескольких видов. Выполняя

этот процесс, навигационная система робота пытается понять, как выглядит рабочее пространство. Основными аспектами, которыми орудует процесс, являются интерпретирование данных с сенсоров, а также представление данных об окружающей среде. Локализация – это процесс определения робота в пространстве. Разделить её можно на локальную и глобальную. Локальная отвечает за местоположение в реальном времени в пространстве, при заранее известной нулевой точке, т.е. при известных координатах начального местоположения робота. Глобальная же сможет ответить нам на вопрос, где находится робот, независимо от какого-то помещения, к примеру GPS трекинг. Таким образом можем сделать вывод, что SLAM – это два процесса, которые не могут быть решены независимо друг от друга. Прежде чем робот сможет ответить на вопрос о том, как выглядит окружающая среда (исходя из серии наблюдений), он должен знать, где эти наблюдения были сделаны. В то же время, трудно оценить текущее положение робота без карты. [11]

Но, как и любые системы, данная система не является идеальной. Сенсоры данной системы могут столкнуться с проблемой шумов, которая может возникнуть в результате плохих погодных условий, загрязнения окружающей среды или воздействия внешних факторов. Несмотря на то, что данная проблема не может быть решена на аппаратном уровне, ее можно устранить при помощи математического алгоритма - фильтра Калмана. Данный алгоритм учитывает измерения с различных типов датчиков, математическую модель движения и шумы.

Помимо таких сложных систем как SLAM существуют системы движения робота по линии. Эту систему в основном используют на складах и производствах, а также в образовательных целях. Существуют разные разновидности таких роботов, которые зависят от способа считывания линии, цвета линии, датчиков и сенсоров.

Рассмотрим робота, который работает на принципе инфракрасных датчиков и приемника. Механизм движения такого робота будет основан на принципах распространения света приема отраженного света обратно. К

примеру, свет, который попадает на белую линию с диода практически полностью отражается обратно. В свою очередь, если свет попадает на черную линию, то в таком случае свет будет поглощён. Данный робот, используя инфракрасный датчики (фотодиоды) и приемник для отраженного света, работает следующим образом: датчик излучил свет, после чего отраженный свет улавливает приемник, формируя на своем выходе какое-то напряжение. Это напряжение приходит на компаратор, который формирует это напряжение в определенный сигнал на выходе. Этот сигнал считывается микроконтроллером, который отправляет сигнал на модуль движения. Принцип работы такого робота довольно прост: датчики обнаруживают линию, после чего на микроконтроллер поступают соответствующие сигналы. Если датчики не обнаруживают линию, робот продолжает двигаться вперед, если же один из датчиков обнаруживает линию, то на микроконтроллер поступает сигнал, который заставляет робота повернуться. В случае же обнаружения линии обоими датчиками сразу робот останавливается.[8]

Данная система позиционирования робота в пространстве довольно примитивна, поэтому для бакалаврской работы было принято решение модернизировать систему умной камерой, чтобы минимизировать отклонения, погрешности датчиков и увеличить эффективность робота.

Ниже рассмотрим несколько видов умных камер для работы с системами позиционирования.

1.2.1 Умные камеры

Для того чтобы выбрать оптимальный вариант для проекта были рассмотрены несколько модулей с камерами.



Рисунок 2 - Модуль с камерой OV7670.

На рисунке 2 изображен модуль с камерой OV7670. Данный модуль представляет собой камеру с буфером типа FIFO (first in, first out – первым пришел, первым вышел). Этот модуль обеспечивает полномасштабное 8 бит изображение в окне. OV7670 умеет работать с разными форматами видео изображения, а также обеспечивает до 30 кадров в секунду. [3]

Основные характеристики:

- напряжение питания: 3,3 ВЖ;
- ток в активном режиме: 20 мА;
- ток в режиме сна: менее 20 мкА;
- уровень напряжения лог. 1 на контактах сигналов: в диапазоне 2,5 – 3 В;
- интерфейсы: SCCB (совместим с I2C) и параллельный 8 линий
- разрешающая способность матрицы: 0,3 Мп;
- максимальная: 640 x 480 px;
- минимальная 40 x 30 px;
- развертка по строкам;

- поддерживается масштабирование изображения;
- форматы передачи цвета: RGB565, RGB555, RGB444, YUV/YCbCr 4:2:2, GRB 4:2:2, Raw RGB Data;
- количество кадров в секунду: до 30;
- размер линзы: 1/6”;
- угол обзора: 25°;
- отношение сигнал/шум: 46 дБ;
- динамический диапазон: 52 дБ;
- методы автокоррекции: АЕС, АРС, АWB, АBF, АВLC;
- автокомпенсация помехи 50, 60 Гц;
- прогрессивная развертка;
- температура окружающего воздуха во время работы OV7670 300КР;
- рекомендуемая: от 0 до 50 °С;
- предельная: от –30 до 70 °С;
- размер модуля: 35 x 34 x 26 мм;
- вес: 12 г.



Рисунок 3 - Умная камера PixyCam2

Камера компьютерного зрения PixyCam2 представлена на рисунке 3. Данная камера была разработана для обнаружения различных объектов и/или позиционирования в окружающем пространстве. Данная камера была впервые запущена в 2013 году компанией Charmed Labs. Этот модуль машинного зрения совместим с самыми популярными на рынке микроконтроллерами, в том числе Arduino и Raspberry Pi. [16]

PixyCam2 обладает алгоритмами, с помощью которых камера способна обнаруживать разнообразные объекты по форме, цвету и положению в пространстве, обнаружение линий, пересечений и штрих кодов. Камера работает в режиме 60 кадров в секунду, что существенно увеличивает качество изображения.

Технические характеристики камеры PIXY2:

- MCU – Двухядерный NXP [LPC4330 Arm Cortex M4/M0](#) @ до 204 МГц с 264КБ RAM, 2МБ flash;
- датчик изображения – OnSemi (ранее Aptina) [MT9M114](#) разрешение 1296×976 с интегрированным процессором потока изображений;
- поле зрения объектива – 60 градусов по горизонтали, 40 градусов по вертикали;
- USB – 1x micro USB порт;
- Расширение
- I/O порт с последовательным UART, SPI, I2C, цифровой, аналоговый I/O;
- RC серво разъем для подключения до 2 сервоприводов;
- прочее – 2x белые светодиоды, 1x RGB светодиод, 1x кнопка;
- электропитание – 5В через USB вход, или от 6 до 10В через нерегулируемый вход;
- энергопотребление – 140 мА;
- размеры – 42 x 38 x 15 мм;
- вес – 10 грамм.

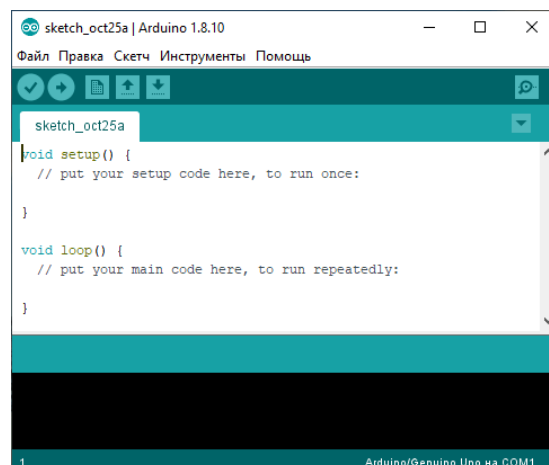
Исходя из темы бакалаврской работы был выбран модуль PiXuCam2, так как данный модуль обладает множественными функциями, такие как работа в режиме 60 кадров в секунду, режим работы с отслеживаем линий и штрихкодов. Также этот модуль был выбран, так как он имеет совместимость с микроконтроллером Arduino Uno.

1.2.2 Программируемая система разработки

Для бакалаврской работы была выбрана среда разработки электронных устройств Arduino. Плата микроконтроллера и среда разработки представлены на рисунке 4.



а)



б)

Рисунок 4 - Плата Arduino Uno – а, среда разработки – б

Данная среда была выбрана так как в нише разработки программно-аппаратных комплексов Arduino является наиболее доступным, простым в разработке и отлично поддерживаемым вариантом. Также из за небольшой стоимости эта плата и среда разработки стала идеальным решением для моей работы.

«Среда разработки Arduino состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста(консоли), панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino.»[2]

Выводы по первому разделу

Исходя из вышесказанного, для реализации автономной транспортной тележки с машинным зрением был выбран микроконтроллер Arduino Uno.

В качестве камеры для реализации системы позиционирования тележки была выбрана умная камера PiXuCam 2, которая обладает необходимыми возможностями для проектирования и запуска системы следования за линией.

2 Разработка структурной схемы автономной транспортной тележки

«Структурная схема — это совокупность элементарных звеньев объекта и связей между ними, один из видов графической модели. Под элементарным звеном подразумевается часть объекта, системы управления и т. д., которая реализует элементарную функцию».[5]

В структурной схеме необходимо показать связь между исполнительными устройствами и камерой посредством микроконтроллера. Структурная схема должна быть понятной и легко читаемой, а также выполнена простым способом.

2.1 Обзор и выбор компонентов транспортной тележки.

Данный раздел предназначен для выбора компонентов тележки. При выборе необходимо учитывать совместимость всех элементов, надежность соединения и работоспособность всех элементов в итоговой схеме. Другими словами, все элементы схемы должны работать и быть собраны как единый механизм, управление которым возмет на себя плата Arduino.

2.1.1 Плата управления

Исходя из прошлых разделов для бакалаврской работы была выбрана плата Arduino Uno.

Плата имеет габариты 69x53мм с выступающими USB и силовым разъёмом на 2 мм.

Данная плата управления, представленная на рисунке 5, имеет все необходимое для реализации проекта бакалаврской работы.

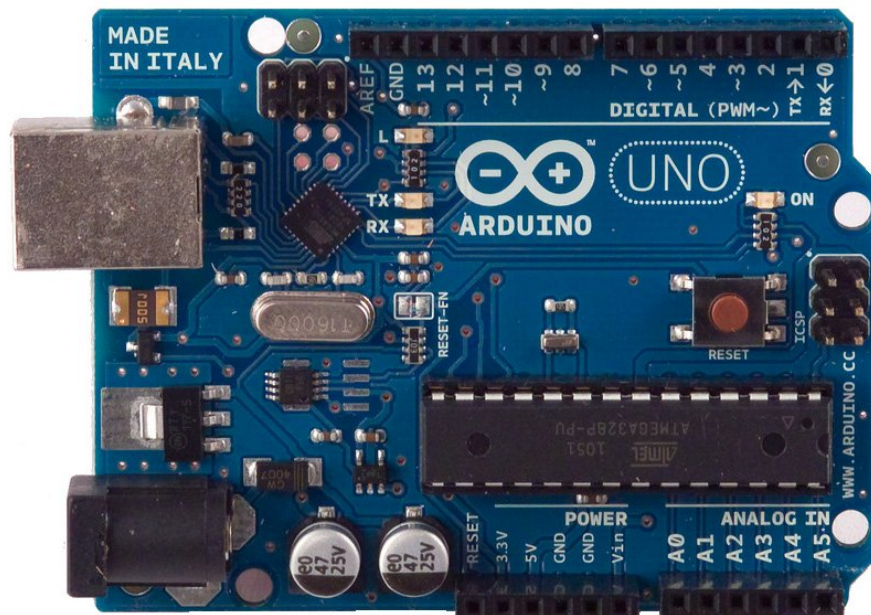


Рисунок 5 - Плата Arduino Uno

Ниже приведены основные характеристики микроконтроллера:

- микроконтроллер: ATmega328P;
- ядро: 8-битный AVR;
- тактовая частота: 16 МГц;
- flash-память: 32 КБ;
- RAM-память: 2 КБ;
- EEPROM-память: 1 КБ;
- пины ввода-вывода: 20;
- пины с прерыванием: 2;
- пины с АЦП: 6;
- разрядность АЦП: 10 бит;
- пины с ШИМ: 6;
- разрядность ШИМ: 8 бит;

- аппаратные интерфейсы: 1× UART, 1× I²C, 1× SPI;
- напряжение логических уровней: 5 В;
- Входное напряжение питания;
- через USB: 5 В;
- через DC-разъём или пин Vin: 7,5–12 В;
- максимальный выходной ток пина 3V3: 150 мА;
- максимальный выходной ток пина 5V: 1 А;
- размеры: 69×53 мм.

Также на плате имеется разъем ICSP(in-circuit serial programming), распиновка которого представлена на рисунке 6. Этот разъем повторяет пины 5V, GND, RESET, MOSI, MISO, SCK. Его обычно используют для загрузки программ в микроконтроллер на плате Arduino.[9] Но в нашем случае через данный разъем будет подключаться камера PiXuCam2 напрямую к плате.

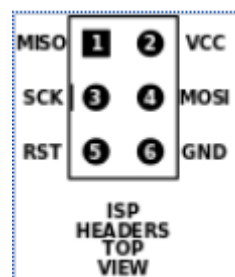


Рисунок 6 - Распиновка разъема ICSP

2.1.2 Модуль с камерой PiXuCam2



Рисунок 7 - Камера PiXu2

PiXu2 — это компактная камера для распознавания цветов и форм, которая использует технологию компьютерного зрения. Она может быть использована в различных проектах, связанных с робототехникой, автоматизацией и машинным зрением.[4] Основные характеристики камеры PiXu2:

- разрешение: 320x200 пикселей;
- частота кадров: до 60 кадров в секунду;
- дальность обнаружения объектов: до 7 метров;
- интерфейсы: UART, SPI, I2C;
- поддержка нескольких цветных фильтров
- встроенный процессор Atmel SAMD21 Cortex-M0+;
- размеры: 54x54x35 мм/

Камера PiXu2, представленная на рисунке 7, может быть использована для распознавания объектов, следования за линиями, распознавания лиц и других задач, связанных с компьютерным зрением. Она может быть использована как самостоятельное устройство или в сочетании с другими устройствами, такими как Arduino, Raspberry Pi и другие микроконтроллеры.

Камера PiXu2 использует технологию компьютерного зрения для распознавания цветов и форм объектов. Она может быть настроена на распознавание нескольких цветовых фильтров, что позволяет ей выделять объекты определенного цвета на фоне других объектов. Когда камера обнаруживает объект, она отправляет информацию о его координатах и размере по выбранному интерфейсу (UART, SPI, I2C) на подключенный микроконтроллер. Это позволяет контроллеру принимать решения на основе информации, полученной от камеры. Камера PiXu2 также имеет встроенный процессор Atmel SAMD21 Cortex-M0+, который может использоваться для выполнения простых задач обработки изображений на камере, что может снизить нагрузку на подключенный микроконтроллер.[1]

Поскольку данная камера имеет неоспоримые преимущества в удобстве и простоте написания кода, данный вариант стал идеальным для бакалаврской работы.

2.1.3 Моторы

Так как в данной бакалаврской работе разрабатывается система позиционирования, для движения тележки необходимы моторы с энкодером. Энкодер будет служить для более точной корректировки маршрута и положения вала электродвигателя. После подбора и изучения нескольких типов двигателей был сделан выбор в пользу электродвигателей с энкодером компании Pitsco TETRIX MAX TorqueNADO Motor. Внешний вид моторов представлен на рисунке 8.



Рисунок 8 - Мотор с энкодером Pitsco TETRIX MAX
TorqueNADO Motor

В таблице 1 приведены основные параметры данного двигателя[20]:

Таблица 1 – Основные параметры мотора с энкодером Pitsco TETRIX
MAX TorqueNADO Motor

Общая длина	134.1 мм
Максимальный диаметр	37 мм
Материал корпуса	Сталь с пластиковым корпусом энкодера
Вес мотора	324 г.
Номинальное напряжение	12 В
Ток остановки	8.7 А
Тип мотора	Щеточный постоянного тока
Электрическое соединение	Силовые полюса для двигателя, 4-контактный разъем энкодера
Длина провода	500 мм
Напряжение Энкодера	3.3...5 В
Тип энкодера	Эффект Холла

Данные двигатели отвечают всем требованиям, следовательно двигатели были выбраны для реализации подвижной части автономной тележки.

2.1.4 Драйвер моторов

Для того, чтобы привести в движение двигатели, необходимо в структурную схему автономной тележки добавить устройство под названием драйвер (от англ. driver – водитель). Это устройство выполняет важную роль в управлении транспортной тележкой. Для бакалаврской работы был выбран драйвер L298N. На рисунке 9 представлены основные элементы драйвера L298N.

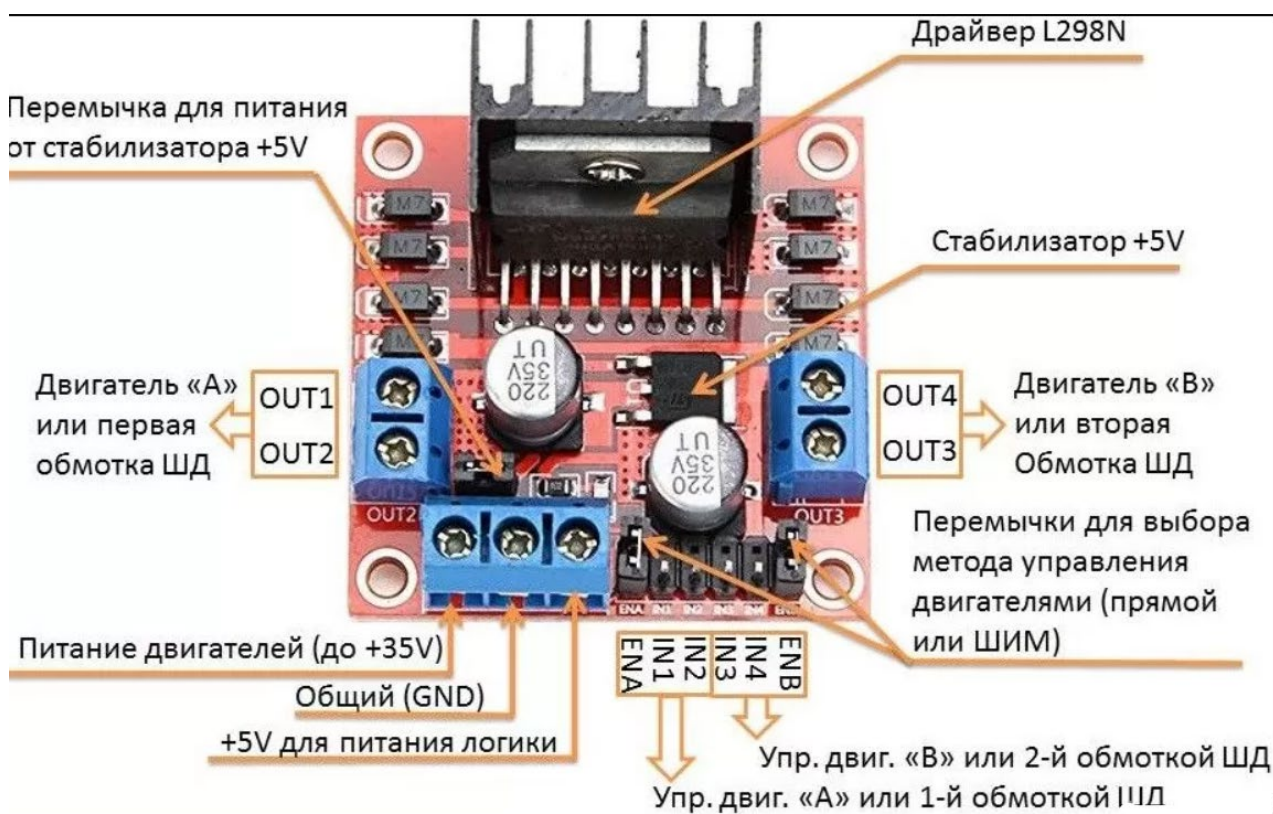


Рисунок 9- Драйвер L298N

- OUT1 и OUT2 – разъёмы для подключения первого двигателя;
- OUT3 и OUT4 – разъёмы для подключения второго двигателя;
- VSS – вход для питания двигателей;
- GND – общий провод (необходимо соединить с платой микроконтроллера);
- Vs – вход для питания логики +5V. Через этот вход происходит питание микросхемы L298N.;
- IN1, IN2 – контакты управления первым двигателем;
- IN3, IN4 – контакты управления вторым двигателем;
- ENA, ENB – контакты для активации/деактивации первого и второго двигателей или соответствующих обмоток ШД.

Драйвер L298N в основном используется для управления двигателями, работающими на постоянном токе. К этому драйверу можно подключить один биполярный или одновременно два щеточных двигателя постоянного тока. Благодаря драйверу имеется возможность изменять направление и вращения двигателей постоянного тока. Для этого на командные входы подаются соответствующие сигналы, там самым осуществляя управление двигателями.[10]

Основные параметры драйвера L298N:

- напряжение питания логики: 5V;
- потребляемый логикой ток: 36mA;
- напряжение питания моторов: от 5V до 35V;
- рабочий ток драйвера: 2 А;
- пиковый ток драйвера 3 А;
- максимальная мощность: 20 Вт (при температуре 75о С);
- диапазон рабочих температур: -25оС...+135оС;
- размеры модуля: 43.5 x 43.2 x 29.4 мм.

2.2 Итоговая схема

После выбора всех элементов схемы в конечном итоге структурно в проекте содержатся следующие элементы:

- плата Arduino Uno;
- модуль с камерой PixyCam2;
- драйвер моторов L298N;
- моторы с энкодером Pitsco TETRIX MAX TorqueNADO Motor;
- блок батарей;
- кнопка включения.

Все элементы схемы соединены между собой связями на электрическом и также на программном уровне. Камера отслеживает линию, передавая сигнал на плату Arduino. После этого микроконтроллер обрабатывает сигнал согласно коду и отправляет его на драйвер двигателей. Драйвер принимает сигнал и после этого распределяет его согласно двигателям, благодаря чему автономная тележка начинает свое движение согласно маршруту.

Структурная схема автономной тележки расположена в Приложении А

Выводы по второму разделу

В ходе работы над структурной схемой автономной транспортной тележки были изучены основные составляющие системы передвижения и позиционирования тележки.

Были рассмотрены и выбраны основные детали, такие как плата управления, модуль с камерой, моторы и драйвер моторов. Были изучены и представлены основные параметры данных электрических устройств и сопоставлены значения, необходимы для функционирования системы позиционирования.

После изучения и выбора основных деталей транспортной тележки была составлена структурная схема с помощью программного обеспечения КОМПАС 3D.

3 Разработка алгоритма работы системы позиционирования автономной транспортной тележки

Разработку алгоритма системы позиционирования автономной транспортной тележки стоит начать с обозначения того, что должна уметь делать система позиционирования как таковая. Система должна иметь следующий функционал:

- определение линии, по которой будет проводится движение тележки;
- прокладывание пути вдоль этой линии;
- определение поворотов и перекрестков;
- следование по заложенному в программе пути.

По ходу движения тележка должна двигаться с необходимой скоростью, чтобы не пропускать повороты и определять перекрестки.

При запуске тележки включается камера, которая начинает считывание линии, используя настройки которые были заранее заданы на камеру с помощью программы РихуМон. Далее камера должна определить линию движения и передать сигнал (в виде векторов следования) на плату микроконтроллера, который в свою очередь с помощью кода будет обрабатывать полученный сигнал. Далее с платы микроконтроллера на драйвер управления двигателями отправляется обработанный сигнал для начала движения платформы. После этого двигатели начинают движение. Камера отслеживает линию вектора и передает информацию на управляющую плату. Составленный алгоритм работы системы позиционирования автономной транспортной тележки представлен на рисунке 10.

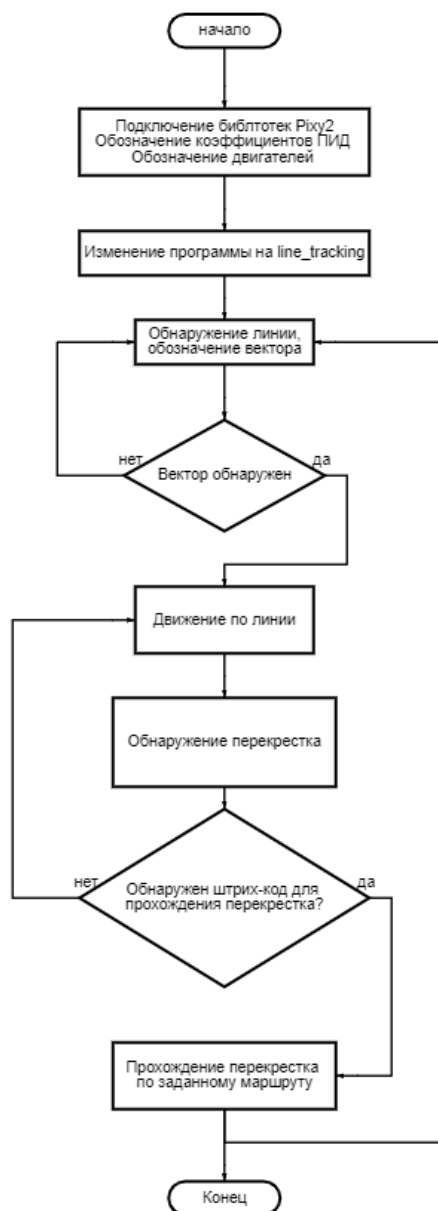


Рисунок 10 – Алгоритм работы системы позиционирования автономной транспортной тележки

Выводы по третьему разделу

В данном разделе был разработан алгоритм работы автономной транспортной тележки. Была составлена блок-схема работы тележки, которая в качестве системы позиционирования использует камеру PíxyCam2. Блок-схема с алгоритмом работы системы управления транспортной тележки расположена в приложении Б.

4 Программная часть

Программирование системы транспортной тележки на основе принципа отслеживания линии следует начать с настройки работы камеры PixyCam 2. На сайте разработчика данного модуля есть подробная инструкция, с помощью которой можно сделать первые шаги в создании программы и настройке камеры.

4.1 Первоначальная настройка камеры PixyCam 2

Для первоначальной настройки камеры прежде всего необходимо скачать с оригинального сайта и установить программу PixyMon.[19] Данная программа нужна для визуализации и вывода изображения на монитор с камеры. Камера подключается напрямую к компьютеру через идущий в комплекте кабеля посредством USB-порта. При подключении камеры программа автоматически распознает подключенное устройство и выводит изображение в окно монитора, представленное на рисунке 11

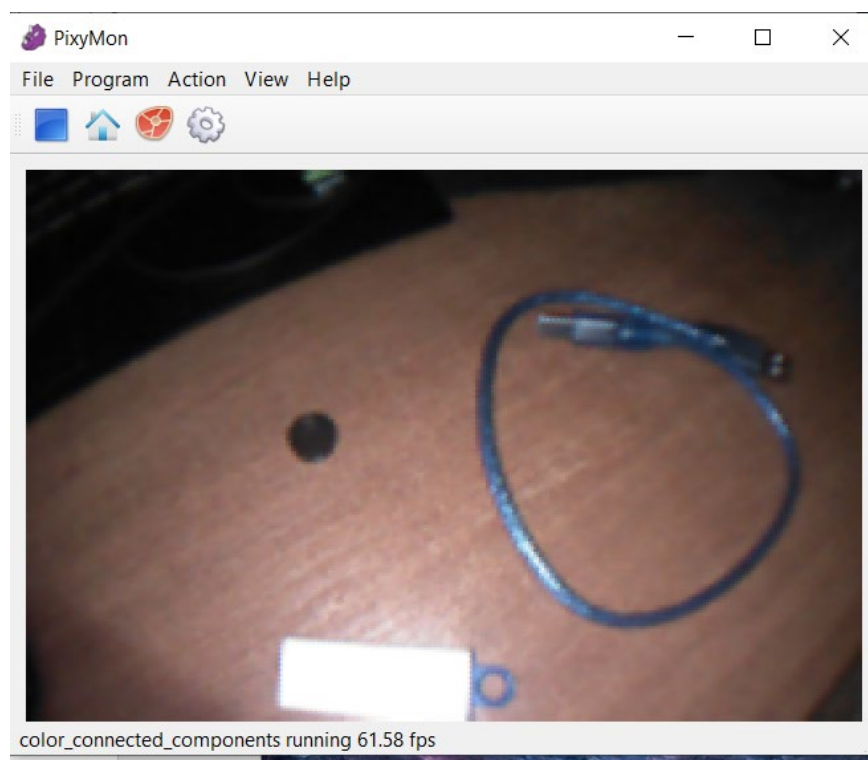


Рисунок 11 - Интерфейс программы PixyMon

После проверки работоспособности камеры можно приступать к настройке режима отслеживания линий. Для этого следует зайти в меню Program и выбрать режим line_tracking. Путь к меню представлен на рисунке 12. Прделав данные действия, цвет изображения на мониторе изменится на черно-белый и появятся вектора, по которым камера будет отслеживать линию и передавать сигналы по состоянию на микроконтроллер. Работа камеры PixyCam 2 в режиме line_tracking представлена на рисунке 13. [18]

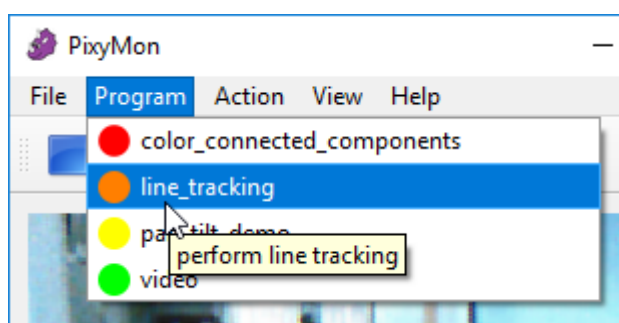


Рисунок 12 - Переключение режимов в PixyMon

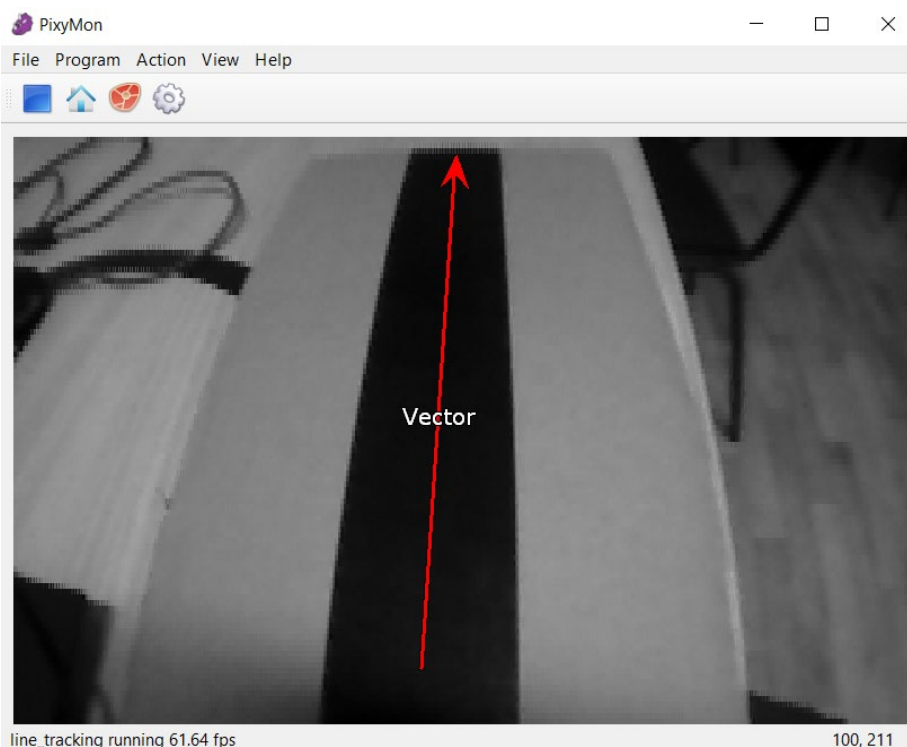


Рисунок 13 -Работа камеры PixyCam 2 в режиме line_tracking

При желании также можно проверить работоспособность светодиода, который расположен на плате модуля рядом с камерой. Он служит для подсветки окружающей среды и помощи камере в случае необходимости более точного отслеживания объектов. Для этого нужно войти в подменю Action и выбрать функцию Toggle Lamp. Путь к меню представлен на рисунке 12.

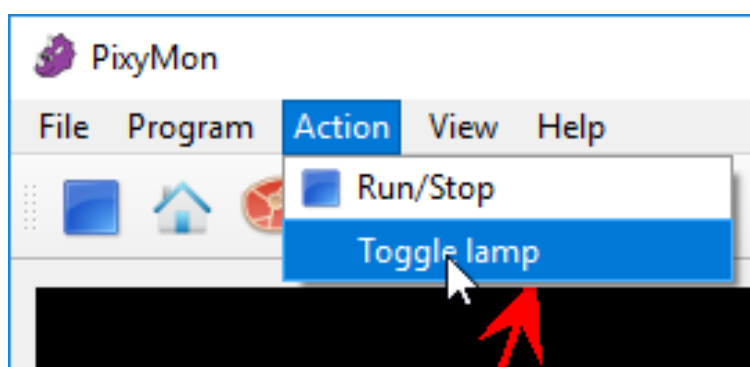


Рисунок 14 - Проверка работоспособности подсветки в программе PixyMon

Далее после проверки работоспособности основных функций следует приступить к более тонкой настройке. Необходимо настроить такие параметры как минимальная толщина линии, по которой будет ориентироваться тележка, начальный угол поворота, фильтры перекрестков и пересечений. Путем тестовых запусков были установлены следующие параметры, которые представлены на рисунке 15 и удовлетворяют необходимым требованиям для работы.

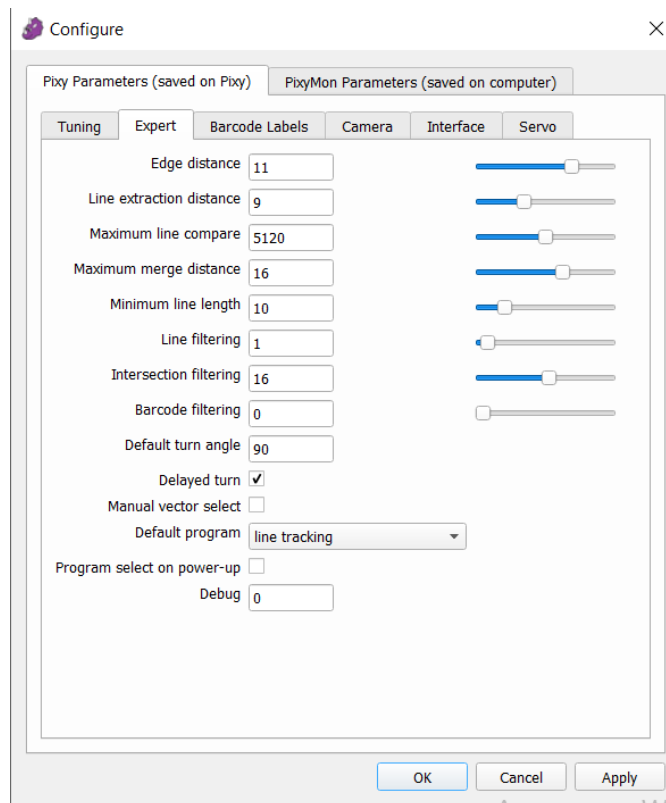


Рисунок 15 - Параметры камеры PixyCam 2

После настройки всех необходимых параметров приступаем к подключению модуля PixyCam 2 к микроконтроллеру Arduino Uno. Демонстрация подключения камеры к плате Arduino Uno продемонстрировано на рисунке 16. Подключение происходит напрямую к микроконтроллеру через ICSP-пины, находящиеся на плате отдельно.

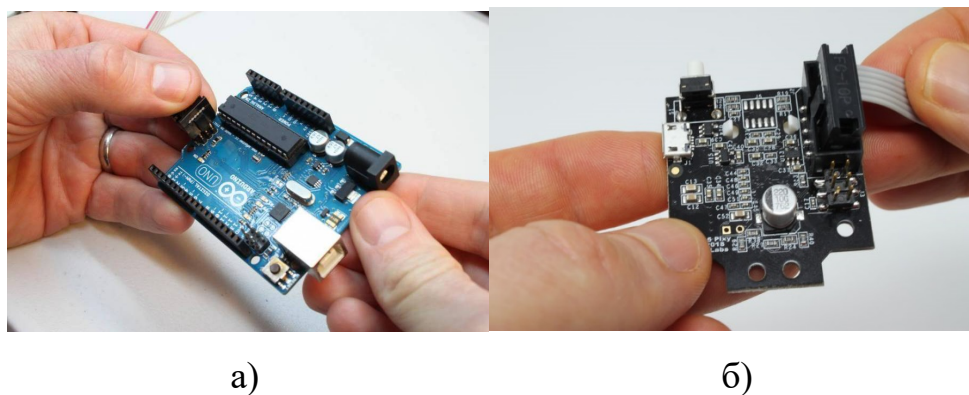


Рисунок 16 - Подключение шлейфа к плате Arduino Uno – а;
Подключение шлейфа к модулю PixyCam 2 – б.

Создатели PixyCam2 учли то, что камеру будут использовать в том числе и в проектах на Arduino и написали код для быстрого старта проверки работоспособности камеры. Для этого первоначально необходимо скачать библиотеки для работы с камерой PixyCam2 под названием «arduino_pixy2-x.y.z.zip». После того, как библиотеки будут загружены в среду разработки Arduino, для быстрой проверки необходимо пройти по пути Файл – Примеры – Pixy2 – line_hello_world. Путь к примеру кода изображен на рисунке 17.

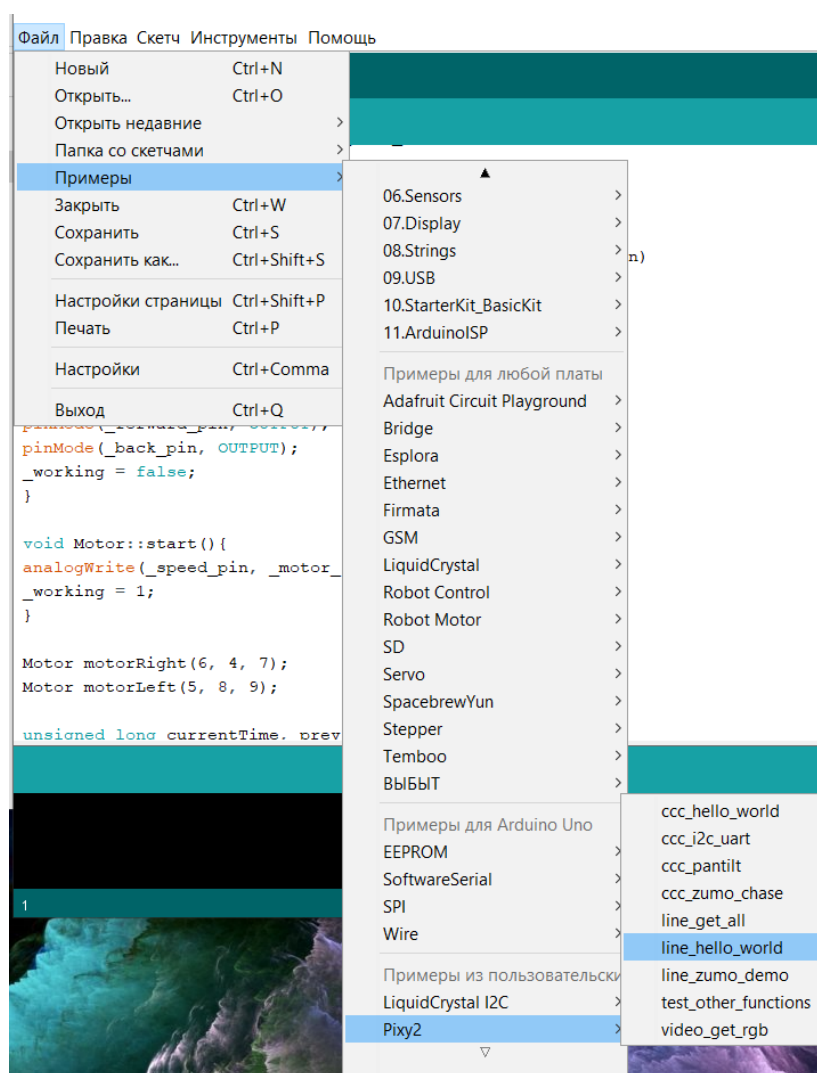


Рисунок 17 - Путь до проверочного скетча

Выбрав скетч, изображённый на рисунке 18, появится простая программа, которая проверит работоспособность камеры и связь между Arduino Uno и PixyCam2. Скетч этой программы станет основой для будущей программы системы позиционирования.



```
sketch_feb06a
#include <Pixy2.h>

Pixy2 pixy;

void setup()
{
  Serial.begin(115200);
  Serial.print("Starting...\n");

  pixy.init();
  // change to the line_tracking program. Note, changeProg can use partial strings, so for example,
  // you can change to the line_tracking program by calling changeProg("line") instead of the whole
  // string changeProg("line_tracking")
  Serial.println(pixy.changeProg("line"));
}

void loop()
{
  int8_t i;
  char buf[128];

  pixy.line.getMainFeatures();

  if (pixy.line.numVectors)
    pixy.line.vectors->print();

  if (pixy.line.numIntersections)
    pixy.line.intersections->print();

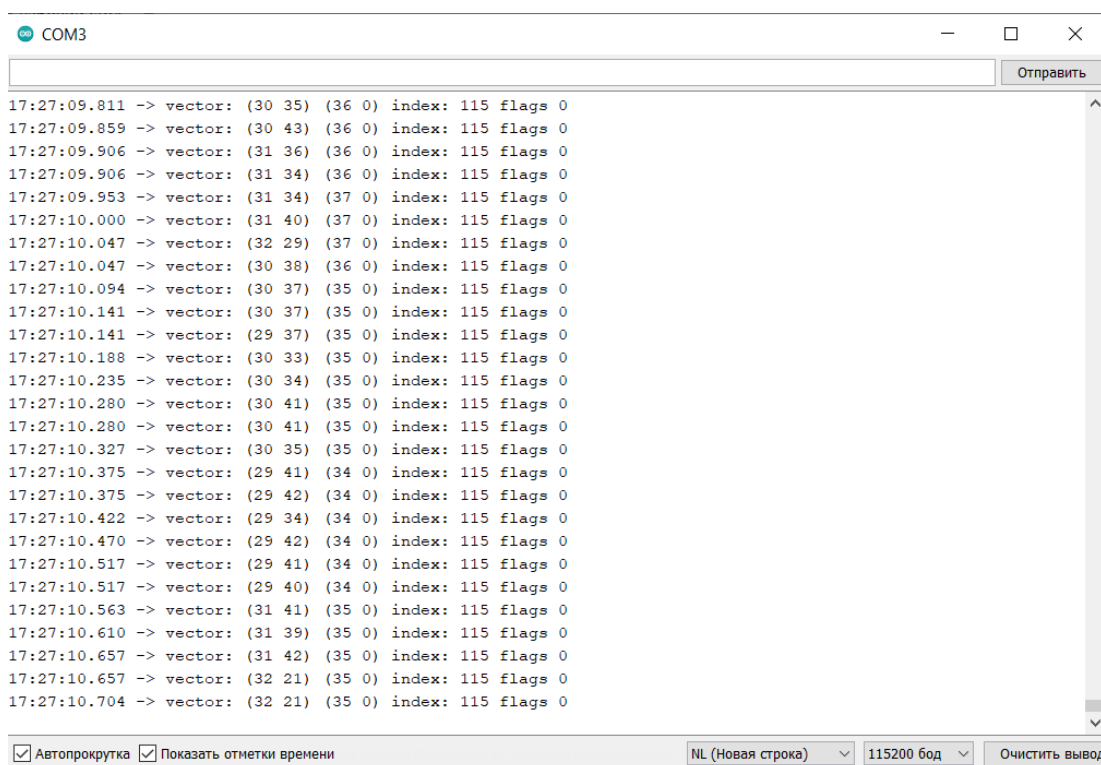
  if (pixy.line.barcodes)
    pixy.line.barcodes->print();
}

Скетч использует 7566 байт (23%) памяти устройства. Всего доступно 32256 байт.
Глобальные переменные используют 468 байт (22%) динамической памяти, оставляя 1580 байт для локальных пере
"Парам
32 Arduino Uno на COM3
```

Рисунок 18 - Скетч программы проверки работоспособности камеры PixyCam2

После загрузки данного скетча на микроконтроллер открывается монитор порта и проверяется работа камеры. На мониторе порта должно

отображаться следующая информация: координаты вектора, индексы и флаги.
Монитор порта при работающей программе представлен на рисунке 19.



```
COM3
17:27:09.811 -> vector: (30 35) (36 0) index: 115 flags 0
17:27:09.859 -> vector: (30 43) (36 0) index: 115 flags 0
17:27:09.906 -> vector: (31 36) (36 0) index: 115 flags 0
17:27:09.906 -> vector: (31 34) (36 0) index: 115 flags 0
17:27:09.953 -> vector: (31 34) (37 0) index: 115 flags 0
17:27:10.000 -> vector: (31 40) (37 0) index: 115 flags 0
17:27:10.047 -> vector: (32 29) (37 0) index: 115 flags 0
17:27:10.047 -> vector: (30 38) (36 0) index: 115 flags 0
17:27:10.094 -> vector: (30 37) (35 0) index: 115 flags 0
17:27:10.141 -> vector: (30 37) (35 0) index: 115 flags 0
17:27:10.141 -> vector: (29 37) (35 0) index: 115 flags 0
17:27:10.188 -> vector: (30 33) (35 0) index: 115 flags 0
17:27:10.235 -> vector: (30 34) (35 0) index: 115 flags 0
17:27:10.280 -> vector: (30 41) (35 0) index: 115 flags 0
17:27:10.280 -> vector: (30 41) (35 0) index: 115 flags 0
17:27:10.327 -> vector: (30 35) (35 0) index: 115 flags 0
17:27:10.375 -> vector: (29 41) (34 0) index: 115 flags 0
17:27:10.375 -> vector: (29 42) (34 0) index: 115 flags 0
17:27:10.422 -> vector: (29 34) (34 0) index: 115 flags 0
17:27:10.470 -> vector: (29 42) (34 0) index: 115 flags 0
17:27:10.517 -> vector: (29 41) (34 0) index: 115 flags 0
17:27:10.517 -> vector: (29 40) (34 0) index: 115 flags 0
17:27:10.563 -> vector: (31 41) (35 0) index: 115 flags 0
17:27:10.610 -> vector: (31 39) (35 0) index: 115 flags 0
17:27:10.657 -> vector: (31 42) (35 0) index: 115 flags 0
17:27:10.657 -> vector: (32 21) (35 0) index: 115 flags 0
17:27:10.704 -> vector: (32 21) (35 0) index: 115 flags 0
```

Автопрокрутка Показать отметки времени NL (Новая строка) 115200 бод Очистить вывод

Рисунок 19 - Монитор порта при проверке работоспособности.

В случае отсутствия этой информации следует проверить правильность настроек камеры в программе РіхuМоп, проверить правильность подключения самой камеры к микроконтроллеру.

После проверки работоспособности камеры можно приступать к написанию кода системы позиционирования транспортной тележки.

1.2 Программная часть системы позиционирования

Систему позиционирования транспортной тележки программируют по частям, начиная с типов устройств управления заканчивая программированием паттернов движения по маршруту. [17]

Код, основанный на алгоритме, который был описан в пункте 3, приведен ниже с комментариями.

```

#include <Pixy2.h> // Подключение библиотеки Pixy2
Pixy2 pixy;
double kp = 0.65; // Обозначение коэффициентов PID
double ki = 0.01;
double kd = 0.02;
#define RIGHT_FWD 6 // Обозначение пинов для работы двигателей
#define RIGHT_REV 7
#define LEFT_FWD 5
#define LEFT_REV 9
#define RIGHT_PWM 4
#define LEFT_PWM 8
unsigned long currentTime, previousTime;
double elapsedTime;
double error;
double lastError;
double input, output, setPoint;
double cumError, rateError;
double OUT_PID, Input_PID, temp_PID;
bool check_dir=false, stop_motor=false;
int time_mili=0;
int c=0, d=0, a=0, b=0;
void setup()
{
  pinMode(RIGHT_FWD, OUTPUT);
  pinMode(RIGHT_REV, OUTPUT);
  pinMode(LEFT_FWD, OUTPUT);
  pinMode(LEFT_REV, OUTPUT);
  Serial.begin(115200);
  Serial.print("Starting...\n");
  setPoint = 42;
  pixy.init();
  Serial.println(pixy.changeProg("line")); // Изменение программы
камеры на line_tracking.
}
void loop()
{
  int8_t i;
  char buf[128];

```

```

if(!stop_motor){
  pixy.line.getMainFeatures();
  if(check_dir)
  {
    delay(1000);
    if(pixy.line.numVectors)           //Обозначение векторов движения
    {
      a=pixy.line.vectors->m_x0;
      b= pixy.line.vectors->m_x1;
    }
    LEFT_FORWARD();
    RIGHT_FORWARD();
    if(c==1)
    {
      analogWrite(RIGHT_PWM,110);
      analogWrite(LEFT_PWM,70);
      delay(2000);
      analogWrite(RIGHT_PWM,0);
      analogWrite(LEFT_PWM,0);
      pixy.line.getMainFeatures();
      if (pixy.line.numVectors)
      {
        a=pixy.line.vectors->m_x0;
        b= pixy.line.vectors->m_x1;
      }
      time_mili=millis();
      while(((a+b)/2>=40))
      {
        Serial.print("ab");
        Serial.println((a+b)/2);
        Serial.print("a");
        Serial.println(a);
        Serial.print("b");
        Serial.println(b);
        //delay(10000000);
        analogWrite(RIGHT_PWM,110);
        analogWrite(LEFT_PWM,70);
        int time_temp=millis();
        if((time_temp-time_mili)>=5000)
        {
          analogWrite(RIGHT_PWM,110);
          analogWrite(LEFT_PWM,50);
        }
        delay(1000);
      }
    }
  }
}

```

```

    analogWrite(RIGHT_PWM,0);
    analogWrite(LEFT_PWM,0);
    pixy.line.getMainFeatures();
    if (pixy.line.numVectors)
    {
        a=pixy.line.vectors->m_x0;
        b= pixy.line.vectors->m_x1;
    }

}
//time_mili=millis();

check_dir=false;
c=0;

}
if(c==7)
{

    analogWrite(RIGHT_PWM,70);
    analogWrite(LEFT_PWM,110);
    delay(1000);
    analogWrite(RIGHT_PWM,0);
    analogWrite(LEFT_PWM,0);
    pixy.line.getMainFeatures();
    if (pixy.line.numVectors)
    {
        a=pixy.line.vectors->m_x0;
        b= pixy.line.vectors->m_x1;
    }
    time_mili=millis();
    while(((a+b)/2<=20))
    {
        if((time_mili-5000)>=0)
        {
            analogWrite(LEFT_PWM,150);
        }
        Serial.print("ab");
        Serial.println((a+b)/2);
        Serial.print("a");
        Serial.println(a);
        Serial.print("b");
        Serial.println(b);
    }
}

```

```

    //delay(10000000);
    analogWrite(RIGHT_PWM,70);
    analogWrite(LEFT_PWM,110);

    delay(1000);

    analogWrite(RIGHT_PWM,0);
    analogWrite(LEFT_PWM,0);
    pixy.line.getMainFeatures();
    if (pixy.line.numVectors)
    {
        a=pixy.line.vectors->m_x0;
        b= pixy.line.vectors->m_x1;
    }

}
check_dir=false;
c=0;

}

}
if (pixy.line.barcodes){ // Обозначение штрих-кодов

//pixy.line.barcodes->print();
c=pixy.line.barcodes->m_code;

Serial.println(c);
if(c==1|c==7) {
    analogWrite(RIGHT_PWM,100);
    analogWrite(LEFT_PWM,100);
    delay(1500);
    analogWrite(RIGHT_PWM,0);
    analogWrite(LEFT_PWM,0);
    check_dir=true;
}
if(c==3)
{
    analogWrite(RIGHT_PWM,100);
    analogWrite(LEFT_PWM,100);
    delay(6000);
}
if(c==5)
{

```

```

    STOP();
    stop_motor=true;
    check_dir=true;
  }
}
if(!check_dir)
{
  if (pixy.line.numVectors)
  {
    pixy.line.vectors->print();
    a=pixy.line.vectors->m_x0;
    b= pixy.line.vectors->m_x1;
    //Serial.println(a);
    //Serial.println(b);
    Input_PID=(a+b)/2;
    Serial.println(Input_PID);
    temp_PID=computePID(Input_PID);
    OUT_PID=constrain(temp_PID,-150,150);
    Serial.println(error);
    Serial.println(OUT_PID);
    if(-5<=error && error<=5)
    {
      //cumError=0;
      LEFT_FORWARD();
      RIGHT_FORWARD();
      analogWrite(RIGHT_PWM,100);
      analogWrite(LEFT_PWM,100);
      //Serial.println(123);
    }
    else if(error>5)
    {

      LEFT_FORWARD();
      RIGHT_FORWARD();
      analogWrite(RIGHT_PWM,OUT_PID);
      analogWrite(LEFT_PWM,75);

    }
    else if(error<-5)
    {
      LEFT_FORWARD();
      RIGHT_FORWARD();

      OUT_PID*=-1;
    }
  }
}

```

```

    analogWrite(RIGHT_PWM,75);
    analogWrite(LEFT_PWM,OUT_PID);

}

}
}
if (pixy.line.numIntersections)
{

    pixy.line.intersections->print();

    Serial.println(pixy.line.intersections->m_n);
    Serial.println(c);
    // analogWrite(RIGHT_PWM,0);
    // analogWrite(LEFT_PWM,0);
    // delay(100000);
}

}
}
double computePID(double inp){
    currentTime = millis();           //get current time
    elapsedTime = (double)(currentTime - previousTime); //время,
прошедшее с момента предыдущего вычисления

    error = setPoint - inp;           // определение ошибки
    cumError += error * elapsedTime;  // вычисление интеграла
    rateError = (error - lastError)/elapsedTime; // вычисление
производной

    double out = kp*error + ki*cumError + kd*rateError; //ПИД
ВЫХОД

    lastError = error;                //запомнить ошибку
    previousTime = currentTime;       //запомнить время на
данный момент

    return out;                       //функция возвращается к выводу
ПИД
}
void LEFT_FORWARD()
{
    digitalWrite(LEFT_FWD,HIGH);

```



```

    digitalWrite(LEFT_REV,LOW);
}
void RIGHT_FORWARD()
{
    digitalWrite(RIGHT_FWD,HIGH);
    digitalWrite(RIGHT_REV,LOW);
}
void STOP()
{
    digitalWrite(RIGHT_FWD,LOW);
    digitalWrite(RIGHT_REV,LOW);
    digitalWrite(LEFT_FWD,LOW);
    digitalWrite(LEFT_REV,LOW);
    analogWrite(RIGHT_PWM,0);
    analogWrite(LEFT_PWM,0);
}

```

Выводы по четвертому разделу

В данном разделе была проведена первоначальная настройка камеры PixyCam2 и проверка работоспособности данной камеры. Далее была проведена проверка работы камеры с микроконтроллером Arduino Uno и загружен код для старта разработки системы позиционирования автономной транспортной тележки.

После всех проверок работоспособности был написан код и загружен на плату Arduino Uno. Были использованы библиотеки, которые были специально разработаны создателями камеры PixyCam2 для более удобного написания кода. Были откалиброваны двигатели с энкодерами для более плавного прохождения маршрута.

Заключение

В современном мире уделяется все больше внимания автоматизации и робототехнике. Ведь роботизированные системы могут оказать помощь человеку в разных сферах деятельности. В данном проекте была разработана система позиционирования для автоматизированной транспортной тележки на основе модуля умной камеры PiXuCam 2. Реализована данная система была на базе программного обеспечения Arduino. Благодаря распространенности и доступности данной аппаратно-программной среды проект получилось реализовать.

В качестве основы для системы позиционирования была выбрана умная камера PiXuCam 2. Данная камера способна работать в нескольких режимах, в том числе и в режиме отслеживания линий, который был необходим в бакалаврской работе.

Также была разработана структурная схема, в которой содержится необходимая информация про связь между разными элементами транспортной тележки. Данная схема была разработана с использованием программного обеспечения КОМПАС 3Д

В ходе работы был разработан алгоритм движения тележки с использованием принципа отслеживания линий. На основе алгоритма был написан код, который был загружен в плату Arduino.

Разработанную систему позиционирования для транспортной тележки можно использовать как для работы на производстве, так и в образовательных целях.

Данная система имеет потенциал модернизации и улучшения, что в свою очередь позволит расширить и улучшить систему позиционирования в будущем

Список используемых источников

1. Arduino Uno: распиновка, схема подключения и программирование [Электронный ресурс]: Статья. – режим доступа <http://wiki.amperka.ru/products:arduino-uno>
2. Arduino: плата Arduino и её окружение [Электронный ресурс]: Статья. – режим доступа: http://arduino.ru/Arduino_environment
3. OV7670 модуль VGA камеры [Электронный ресурс]: статья – режим доступа <https://radioprogram.ru/shop/merch/58>
4. Pixy 2 – анонс системы машинного зрения для Arduino [Электронный ресурс]: Статья. – режим доступа <https://voltiq.ru/pixy-2-image-sensor-arduino/>
5. Wikipedia: Структурная схема [Электронный ресурс]: Статья. – режим доступа <https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%>
6. Автоматизация складов с помощью роботов [Электронный ресурс]: статья – режим доступа <https://top3dshop.ru/blog/warehouse-and-logistics-robots-review.html>
7. Автоматизированные роботы для складских работ [Электронный ресурс]: статья – режим доступа <https://vektor.ru/blog/robot-pogruzchik.html#tipy-mobilnyh-robotov-dlya-sklada>
8. Движущийся вдоль линии робот на Arduino [Электронный ресурс]: статья – режим доступа <https://microkontroller.ru/arduino-projects/dvizhushhij-sya-vdol-lyinii-robot-na-arduino/>
9. Для чего нужен разъем I2C на плате ардуино UNO? [Электронный ресурс]: Статья. – режим доступа <http://arduino.zl3p.com/faq/i2c>
10. Драйвер двигателя L298N Arduino [Электронный ресурс]: Статья. – режим доступа <https://3d-diy.ru/wiki/arduino-moduli/drayver-dvigatelya-l298n/>

11. Исследование методов SLAM для навигации мобильного робота внутри помещений [Электронный ресурс]: статья – режим доступа <https://habr.com/ru/articles/560856/>
12. Исследование рынка автономных роботов в сфере хранения [Электронный ресурс]: статья – режим доступа <https://electricalschool.info/robot/2557-issledovanie-rynka-robotov-dlya-skladov.html>
13. Методы визуальной навигации мобильного робота и построения картографических моделей внешней среды/Ж. С. Першина , С. Я. Каздорф, А. В. Лопота – АВТОМЕТРИЯ. 2019. Т. 55 с.92-93
14. Навигация мобильных роботов [Электронный ресурс]: статья – режим доступа <https://www.itweek.ru/themes/detail.php?ID=66917>
15. Сенюшкин, Н. С. Автономная система позиционирования в составе управления наземной многоцелевой подвижной платформой / Н. С. Сенюшкин, А. В. Суханов, В. В. Парамонов. — Текст : непосредственный // Молодой ученый. — 2011. — № 1 (24). — С. 44-46
16. Introducing Pixy2 [Электронный ресурс]: статья – режим доступа <https://pixycam.com/pixy2/>
17. Line tracking API [Электронный ресурс]: Статья. – режим доступа https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:line_api
18. Line tracking quickstart [Электронный ресурс]: Статья. – режим доступа https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:line_quickstart
19. Pixy2 Downloads [Электронный ресурс]: Статья. – режим доступа <https://pixycam.com/downloads-pixy2>
20. TETRIX® MAX TorqueNADO® Motor Specifications [Электронный ресурс]: .документация – режим доступа https://asset.pitsco.com/sharedimages/resources/torquenado_dcmotorspecs.pdf

Приложение Б

Блок схема работы системы позиционирования автономной транспортной тележки.

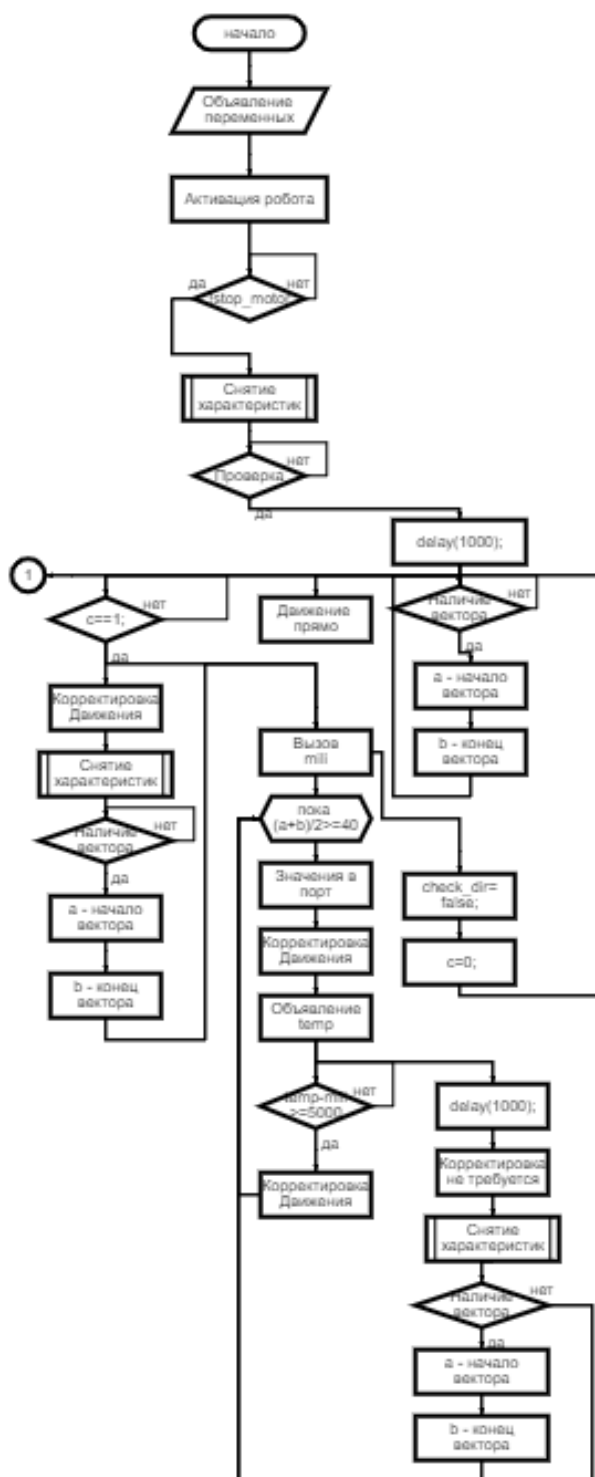


Рисунок Б.1 – Блок-схема работы системы позиционирования автономной транспортной тележки

Продолжение приложения Б

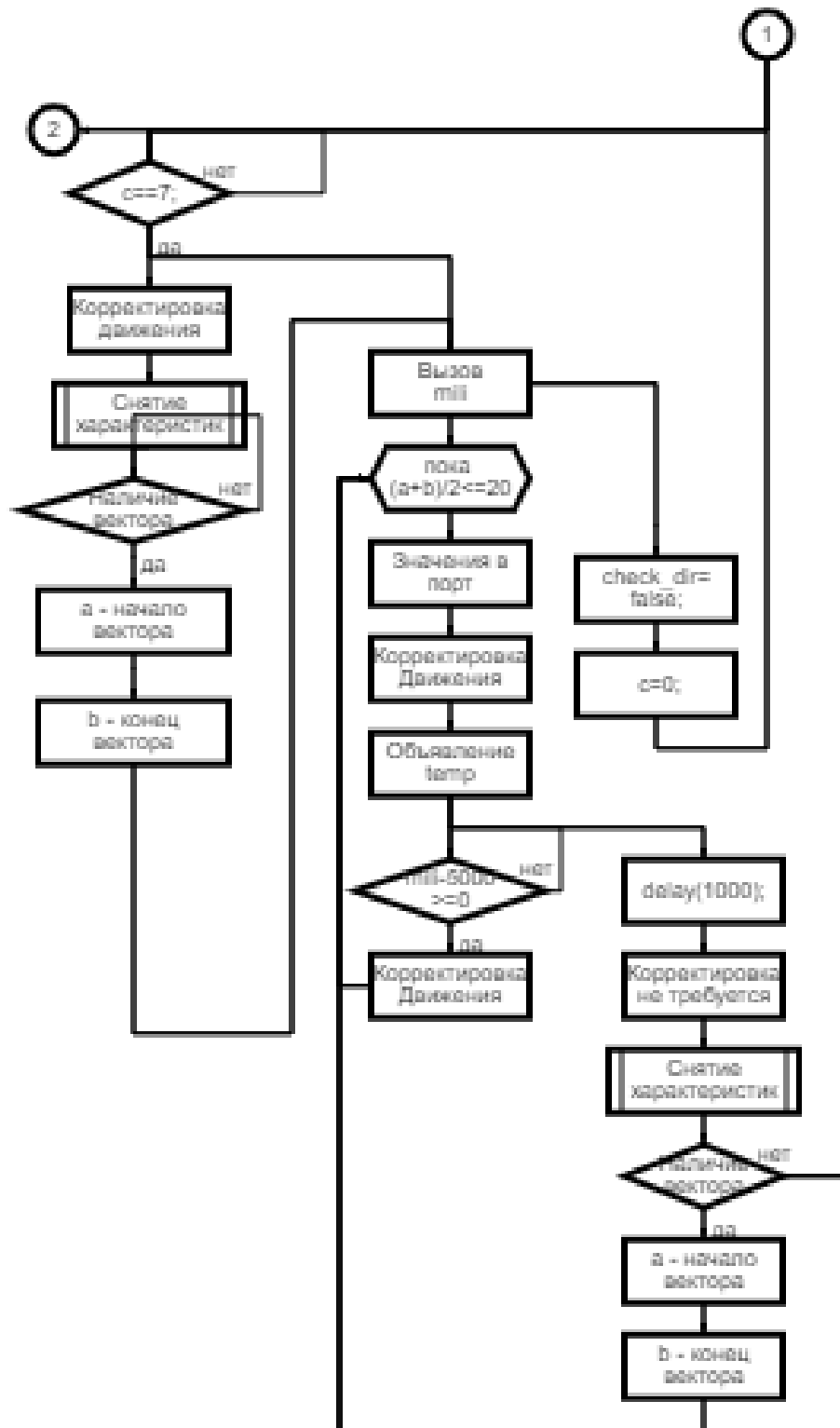


Рисунок Б.2 – Блок-схема работы системы позиционирования автономной транспортной тележки.

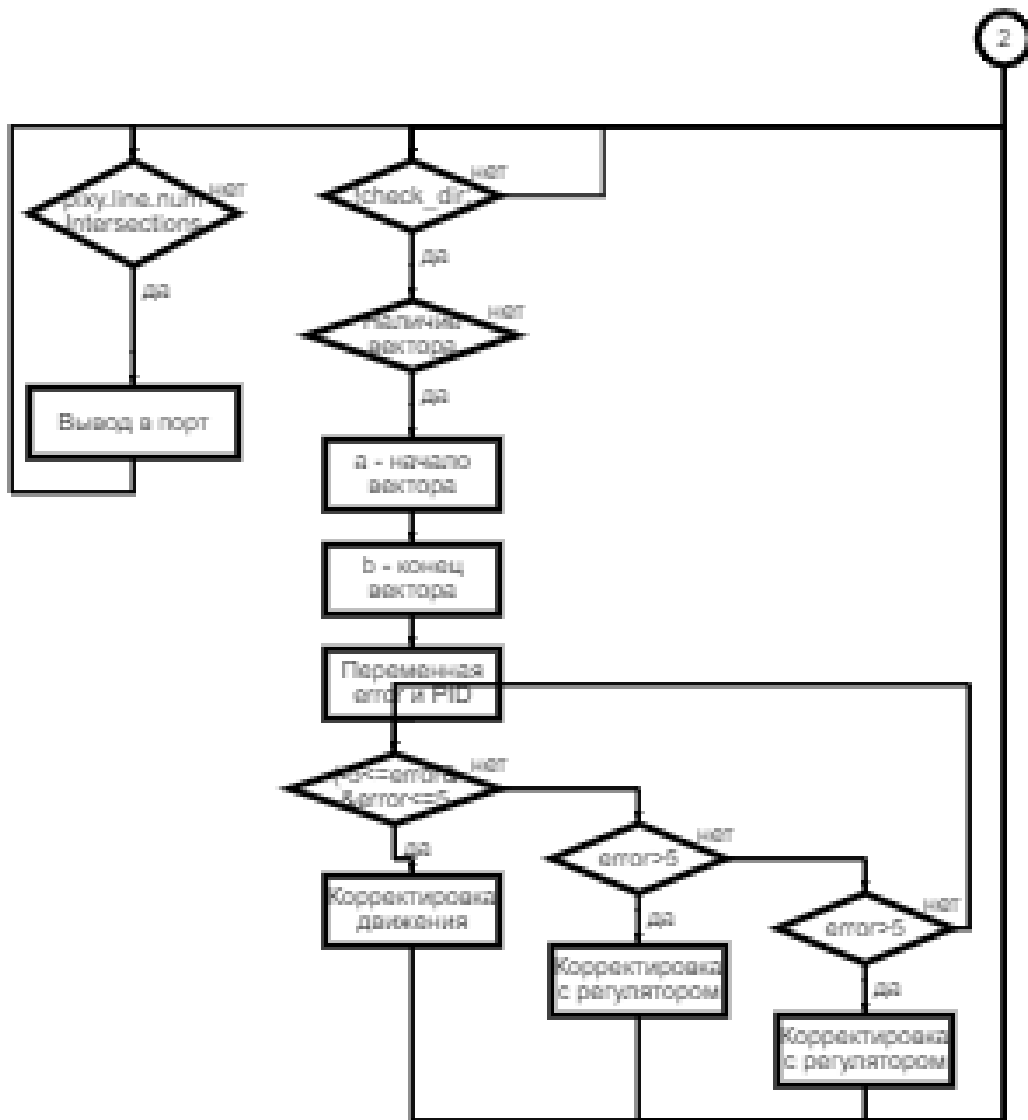


Рисунок Б.3 - Блок-схема работы системы позиционирования автономной транспортной тележки.