

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ  
РОБОТОМ-ПОГРУЗЧИКОМ НА ОСНОВЕ КОНТРОЛЛЕРА ARDUINO**

Выпускная квалификационная работа  
обучающегося по направлению подготовки 02.04.01 Математика и  
компьютерные науки очной формы обучения, группы 07001531  
Горбачева Артёма Юрьевича

Научный руководитель  
к.т.н., доцент  
Чашин Ю.Г.

Рецензент  
доцент кафедры  
информационно-  
телекоммуникационных систем  
и технологий НИУ «БелГУ»,  
кандидат технических наук  
Заливин А.Н.

БЕЛГОРОД 2017

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	6
<b>1.1 Введение в предметную область .....</b>	<b>6</b>
<b>1.2 Анализ существующих методов перемещения и способы их         реализации.....</b>	<b>9</b>
ГЛАВА 2. РАЗРАБОТКА ЭЛЕМЕНТОВ ПРОЕКТА.....	22
<b>2.1 Выбор программных и аппаратных компонентов .....</b>	<b>22</b>
<b>2.1.1 Выбор аппаратной вычислительной платформы.....</b>	<b>22</b>
<b>2.1.2 Выбор двигателей .....</b>	<b>28</b>
2.1.2 Выбор необходимых модулей .....	34
2.1.2 Выбор среды программирования .....	38
2.1.2 Выбор среды управления системой .....	39
<b>2.2 Разработка алгоритмов.....</b>	<b>41</b>
ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОЕКТА .....	46
<b>3.1 Реализация аппаратной части системы управления .....</b>	<b>46</b>
<b>3.2 Реализация скетча.....</b>	<b>53</b>
<b>3.3 Реализация управления с мобильной платформы.....</b>	<b>58</b>
ГЛАВА 4. АПРОБАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ.....	61
ЗАКЛЮЧЕНИЕ .....	63
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	66
ПРИЛОЖЕНИЕ .....	69

## ВВЕДЕНИЕ

Сейчас мы живем в мире, где большинство повседневных задач упрощены или автоматизированы и с каждым годом эта тенденция возрастает.

В обиход современного человека плотно вошли технологии удаленного, автоматизированного управления. Все чаще стало появляться выражение "автоматизированная система управления" в научных, промышленных и бытовых отраслях. Что же понимать под выражением "автоматизированная система управления"? Есть много определений для этого понятия: «Автоматизированная система управления — это устройство, которое воспринимает, мыслит и действует». Около десяти лет назад добавилась к этим трем характеристикам способность к коммуникации, и теперь многие говорят, что система управления «воспринимает, мыслит, действует и коммуницирует». Хорошо это или плохо, но такое определение заставляет считать современную бытовую технику (стиральные машины и так далее) автоматизированными системами управления.

Возможно, для снятия противоречий стоит использовать подходящие прилагательные: мобильная система управления, сельскохозяйственная система управления и так далее. Ученые давно видели перспективу создания «умной» техники, которая могла бы «воспринимать, думать и действовать».

В настоящее время промышленные системы управления плотно закрепились в отраслях машиностроения, приборостроения и других отраслях, где необходима точность сборки и высокая производительность труда. Мобильные системы управления находят все более широкое применение для выполнения различных задач в условиях, когда присутствие человека в зоне их работы или невозможно по соображениям безопасности, либо же нежелательно из-за ограничения производительности обслуживаемого технологического оборудования, так как часть процессов

происходит автоматически, а остальной можно управлять удаленно, что делает ее актуальной для изучения и совершенствования.

**Целью** данной работы является разработка автоматизированной системы управления роботом-погрузчиком на основе контроллера Arduino.

Система предусматривает управление роботом-погрузчиком как в ручном режиме, так и в автономном (без помощи человека). Суть разработки заключается в движении робота-погрузчика по складским помещениям без участия человека, распределении грузов и выбора траектории.

**Задачи**, которые были поставлены в рамках магистерской диссертации:

- анализ методов и средств автоматизации складского учета;
- формирование требований к системе управления;
- формулировка требований к разрабатываемой специализированной микроконтроллерной платформе;
- выбор аппаратных средств;
- разработка и реализация проекта на платформе Arduino под управлением Android;
- апробация проекта.

Решения о том, как и куда будет двигаться система платформа под управлением, то есть планирование перемещения робота-погрузчика является важнейшей проблемой функционирования автономных робототехнических систем и одной из наиболее активно исследуемых областей современного научно-практического знания.

Механически, робот-погрузчик выглядит, как платформа с двумя или четырьмя электродвигателями и несколькими датчиками. В таких системах прослеживается динамическое взаимодействие, которое не позволяет рассматривать автоматизированную систему управления как совокупность множества маленьких (равных по значимости) автономных систем. Эти особенности робототехнических устройств приводят к необходимости построения иерархических структур систем управления. С точки зрения управляющих систем возникает проблема реального времени, которая

заключается в том, что вычислительная техника, даже в наше время, не успевает справиться с поставленными задачами и зачастую сдерживает темп движения. Эту проблему решает грамотно поставленная задача, и правильно написанный программный код. Программирование ведётся на основе расчётов по математической модели робота, т.е. применяется так называемое аналитическое программирование. Задание для курсового проектирования заключается в разработке автоматизированной системы управления, включающей в себя исполнительную систему и систему управления, выбор компонентов этих систем.

Работа состоит из введения, четырех глав и заключения. В первой главе рассмотрена предметная область и требования, предъявляемые к разрабатываемой системе управления. Вторая глава разработка элементов проекта, выбору аппаратных и программных компонентов, разработке алгоритмов. В третьей главе описана реализация проекта, реализация физической части, написание программного кода и разработка мобильного приложения под управлением Android. В заключительной главе происходит тестирование и отладка системы.

## ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

### 1.1 Введение в предметную область

Основным инструментом любого складского помещения является порядок и последовательность. Для усовершенствования процесса складского управления была разработана автоматизированная система управления.

Представим огромное складское помещение, в котором есть несколько категорий товаров. Каждая категория находится в отдельной части склада. Целью нашего робота будет являться доставка товара на склад в необходимую часть склада согласно категории. Товары будут двигаться на тележке, которая управляется микроконтроллером, передвигается с помощью коллекторных двигателей и ориентируется в пространстве с помощью датчиков. Движение тележки производится по ранее установленной траектории.

Прежде чем затрагивать тему "робота - погрузчика", устройства и строения системы управления, стоит обратиться к необходимости создания и необходимости существования роботов.

Основу современного робота составляет совокупность исполнительных механизмов и устройств с системами привода, позволяющих взаимодействовать со средой при выполнении той или иной поставленной задачи. Привод - совокупность устройств, предназначенных для приведения в действие машин и механизмов. Функциональные возможности робота, его универсальность и соответствие характеристикам определяется в первую очередь разнообразием действий которые может выполнить робот.

Список перечня операций для выполнения, которых предназначен робот, бывает разным, в зависимости от этого списка различаются роботы специальные, специализированные и универсальные. Специальные роботы предназначены для выполнения одной конкретной задачи.

Специализированные роботы могут выполнять несколько однотипных операций. Универсальные роботы могут выполнять различные основные и вспомогательные операции в пределах их технических возможностей. Увеличение степени универсальности робота расширяет область его возможных применений, но одновременно неизбежно сопровождается недоиспользованием этих возможностей на каждой конкретной операции, а также удорожанием робота. Оптимальные в этом отношении являются специальные роботы, но с другой стороны это предельно сужает их рынок, а, следовательно, и объём производства. [1]

Манипуляционный робот - автоматическая машина (стационарная или передвижная), состоящая из исполнительного устройства в виде манипулятора, имеющего несколько степеней подвижности, и устройства программного управления, которая служит для выполнения в производственном процессе двигательных и управляющих функций. Такие роботы производятся в напольном, подвесном и порталном исполнениях. Получили наибольшее распространение в машиностроительных и приборостроительных отраслях.

Мобильный робот - это автоматическая машина, в которой имеется движущееся шасси с автоматически управляемыми приводами.

Мобильные роботы могут быть колёсными, шагающими и гусеничными (существуют также ползающие, плавающие и летающие мобильные робототехнические системы) в зависимости от рода и условий эксплуатации. [2]

Классификация роботов по показателям, определяющих их конструкцию. К таким показателям относятся:

1. тип приводов;
2. грузоподъёмность;
3. количество манипуляторов;
4. тип и параметры их рабочей зоны;
5. подвижность и способ размещения;

б. использование по назначению.

Приводы, которые используются в манипуляторах и системах передвижения роботов, могут быть электрическими, гидравлическими и пневматическими. Часто их применяют в комбинации.

Грузоподъемность робота - это грузоподъемность его манипуляторов, а для транспортного робота ещё и его шасси.

Количество манипуляторов у роботов в большинстве случаев ограничено одним. Однако в зависимости от назначения существуют конструкции роботов без манипуляторов и с 2, 3 и совсем редко 4 манипуляторами. Тип и параметры рабочей зоны манипулятора определяют область окружающего робота пространства, в пределах которой он может выполнять манипуляции, не передвигаясь, т.е. при неподвижном основании.

Подвижность робота определяется наличием или отсутствием у робота системы передвижения. В первом случае роботы называют мобильными, а во втором стационарными.

По способу размещения стационарные и мобильные роботы бывают напольными, подвесными и встраиваемые в другое оборудование.

Использование робота по назначению зависит от внешних условий, в которых он должен функционировать. Различают исполнение нормальное, пылезащитное, теплозащитное, влагозащитное, взрывобезопасное и т.д.

Классификация автоматических роботов по способу управления:

1. программные;
2. адаптивные;
3. интеллектуальные.

Программное устройство управления. Такие устройства функционируют по заранее заданной программе, в основном предназначены для решения однообразных задач в неизменных условиях окружения.

Адаптивное устройство управления. В таких системах управление осуществляется исходя из полученной информации о текущем состоянии внешней среды и самого робота, получаемой в процессе управления от



сенсорных устройств (решают типовые задачи, и адаптируются под условия функционирования программно, исходя из данных полученных от информационной системы).

Интеллектуальное устройство управления. Эти системы для адаптации и выполнения других функций робота используют методы искусственного интеллекта. [3]

Устройства управления могут быть индивидуальными, входящими в состав каждого робота и групповыми, управляющими несколькими роботами. Конструктивно индивидуальные устройства управления выполняют обычно отдельно от механической части робота, значительно реже в общем корпусе. Мобильные роботы обычно имеют устройство управления состоящее из 2-х частей - бортовой и входящей в состав пульта оператора (или в дополнение к нему).

Подавляющее большинство роботов имеет электронные устройства управления, выполненные на микроконтроллере. Однако существуют и неэлектрические устройства управления роботов, часто реализуемых на пневматике и предназначенных для применения в особых взрывоопасных и пожароопасных условиях, также в условиях повышенных температур. [1]

## **1.2 Анализ существующих методов перемещения и способы их реализации**

Переходя к сравнению систем управления, стоит определить основные критерии и классификации. Применения для двигающейся платформы бессчетное множество. Такие системы могут применяться для сортировки грузов, в военной промышленности (роботы - саперы, машины в разведке), роботы - сыщики, которые участвуют в спасательных работах, роботы в космосе - исследуют грунт и рельеф. Сейчас набирают популярность роботы - курьеры и почтальоны. Так, например, Starship Technologies помогли компании Domino разработать новую систему роботизированной доставки.

Шестиколесные роботы ориентируются на улицах, могут объехать небольшие препятствия и определить кратчайший маршрут до места назначения. Они могут развивать скорость до 16 км/ч, но если вокруг много пешеходов, то в интересах безопасности приспособляются к скорости движения людей[4].

Несмотря на самые различные способы применения роботизированной техники, их объединяет одно - способ перемещения в пространстве.

Можно выделить четыре принципиально различных типа мобильных роботов - наземные, воздухоплавающие, водоплавающие, подземные и космические.

Наземные мобильные роботы обычно подразделяются на три больших класса: колесные наземные мобильные роботы, шагающие наземные мобильные роботы и гибридные наземные мобильные роботы. Помимо этих трех наиболее многочисленных классов мобильных роботов существует большое количество специализированных мобильных роботов, ориентированных на ограниченное применение. К их числу относятся рельсовые роботы, адсорбционные роботы (способные передвигаться по крутым участкам, цепляясь за поверхность с помощью вакуумных присосок), роботы на магнитной или воздушной подушке, а также не попадающие ни в одну из перечисленных групп ползающие роботы.

Среди множества разнообразных типов мобильных роботов в настоящее время наибольший практический интерес вызывают колесные наземные мобильные роботы. Предложено большое количество принципов классификации колесных наземных мобильных роботов: Если воспользоваться классификацией по способу управления работой колес, то можно выделить следующие три группы колесных роботов:

1. автомобильная группа (поворот осуществляется только за счет передних колес);
2. группа с произвольным независимым управлением поворотом каждого колеса влево или вправо (например, кресло-каталка);

3. группа роботов, способных перемещаться во всевозможных направлениях.

Большинство применяемых на практике колесных мобильных роботов относится ко второй группе, т.е. данный метод управления оказывается наиболее важным. Что касается роботов, колеса которых могут поворачиваться в любую сторону, то они пока находятся на стадии экспериментальных исследований и опытных испытаний. Среди шагающих мобильных наземных роботов практический интерес представляют конструкции с разным числом конечностей - от многоногих шагающих аппаратов, напоминающих сороконожку, до роботов с 8, 6, 5, 4, 3 и 2 конечностями. Кроме того, в исследовательских центрах и научных лабораториях изучаются принципы создания безногих мобильных аппаратов, способных перемещаться подобно змеям или морским моллюскам. На практике потребность в шагающих аппаратах возникает в связи с необходимостью использования роботов для передвижения по местности с большим количеством препятствий или неровностей, а также так, где от него требуется умение взбираться и спускаться по ступенькам обычной лестницы.

Рассмотрим подробнее группу с произвольным независимым управлением поворотом каждого колеса влево или вправо.

Главной задачей и основным критерием сравнения существующих систем является способ перемещения из точки А в точку Б (Рис. 1.1)

Для автономного универсального робота проблемы ориентации и перемещения в окружающем пространстве являются одними из ключевых. Робот должен не имитировать разумное поведение, а по-настоящему быть разумным. Он должен осознавать - где он находится, каковы особенности данного места и как из точки "А", при необходимости, можно попасть в точку "Б" за минимальное время или покрыв минимальное расстояние. Наличие продвинутой системы ориентации, позволит роботу планировать свои действия по перемещению в пространстве, просчитывать оптимальный маршрут движения, собирать и использовать информацию об окружающих

местность препятствиях, что в конечном итоге, позволит ему выполнять полезную функциональную нагрузку.



Рис. 1.1. Основная задача любого мобильного робота.

Для достижения этой цели существует несколько способов перемещения, которые подчиняются таким критериям как:

- сложность;
- скорость;
- стабильность;
- стоимость реализации;
- автономность;

Попробуем разобраться с особенностями поведения мобильного робота в изначально неизвестной для него статической среде (следующий шаг – в условиях динамично меняющейся обстановки).

Мы определили основные способы перемещения:

#### 1. Лидар (Рис. 1.2).

Принцип действия лидара прост. Объект (поверхность) освещается коротким световым импульсом, и измеряется время, через которое сигнал вернется к источнику. Когда вы светите фонариком на объект (поверхность), то вы видите свет, отраженный от объекта и вернувшийся на вашу сетчатку. Свет распространяется очень быстро - около 300000 километров в секунду, или 0,3 метра за наносекунду, поэтому включение света кажется

мгновенным. Однако это не так. Свет возвращается с некоторой задержкой, которая зависит от расстояния до объекта. Расстояние, которое прошел фотон на пути до объекта и обратно, можно рассчитать по формуле:  $\text{расстояние} = (\text{скорость света} \times \text{время пролета}) / 2$

Оборудование, необходимое для измерения этого малого промежутка времени, должно работать чрезвычайно быстро. Это стало возможным только с достижениями в современной электронике и вычислительной техники. Лидар запускает быстрые короткие импульсы лазерного излучения на объект (поверхность) с частотой до 150000 импульсов в секунду. Датчик на приборе измеряет промежуток времени, необходимый для возврата импульса. Свет движется с постоянной и известной скоростью, поэтому лидар может вычислить расстояние между ним и цели с высокой точностью.

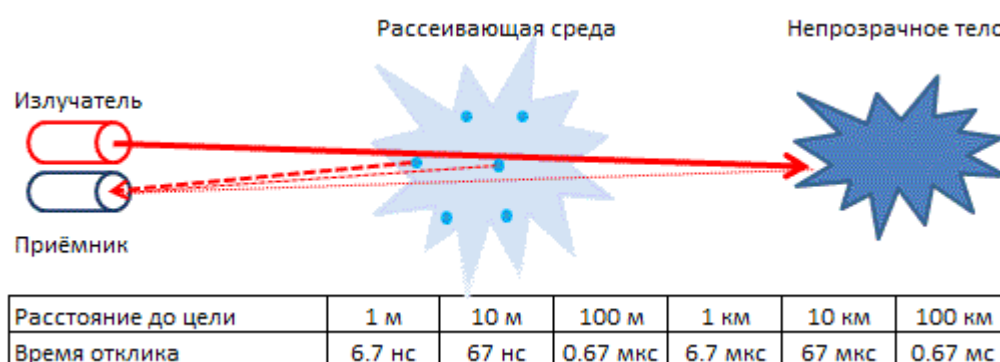


Рис. 1.2. Лидар

Существует два типа лидарных методов измерения: прямой метод измерения (также известный как некогерентный), и когерентное детектирование. Когерентные системы лучше всего подходят для доплеровских или фазочувствительных измерений и, как правило, используют оптическое гетеродинное детектирование. Это позволяет им работать при гораздо меньшей мощности, но за при этом конструкция фотоприемной схемы намного сложнее. Существуют две основные категории импульсных лидаров: микроимпульсные и высокоэнергетические системы. Микроимпульсные лидары работают на более мощной компьютерной

технике с большими вычислительными возможностями. Эти лазеры меньшей мощности и классифицируются как "безопасные для глаз", что позволяет использовать их практически без особых мер предосторожности. Лидары с большой энергией импульса в основном применяются для исследования атмосферы, где они часто используются для измерения различных параметров атмосферы, таких как высота, наслоение и плотность облаков, свойства частиц облака, температуру, давление, ветер, влажность и концентрацию газов в атмосфере[5].

2. RTLS (сокр. от англ. Real-time Locating Systems — система позиционирования в режиме реального времени) — автоматизированная система, обеспечивающая идентификацию, определение координат, отображение на плане местонахождения контролируемых объектов в пределах территории, охваченной необходимой инфраструктурой (Рис. 1.3). RTLS накапливает, обрабатывает и хранит информацию о местонахождении и перемещениях людей, предметов, мобильных механизмов и транспортных средств с целью мониторинга технологических и бизнес-процессов, сигнализации об отклонениях от регламентов, а также с целью ретроспективного анализа тех или иных процессов и ситуаций.

Система позиционирования в режиме реального времени (Real Time Location Services, RTLS) — относительно новая сфера применения технологии WLAN. RTLS является автоматизированной системой, которая обеспечивает идентификацию, определение координат, отображение на плане местонахождения контролируемых объектов в пределах территории, охваченной необходимой инфраструктурой. Она способна накапливать, обрабатывать и хранить информацию о местонахождении и перемещении людей, предметов, мобильных механизмов и транспортных средств. Также эта система позволяет просигнализировать об отклонениях движения объектов от заданных параметров. К основным характеристикам RTLS можно отнести:

- точность позиционирования — точность определения координат контролируемого объекта, от нескольких десятков метров до нескольких сантиметров;

- достоверность позиционирования — помехи и многолучевые затухания в реальных условиях в значительной мере влияют на точность позиционирования, поэтому говоря о точности позиционирования RTLS обычно указывают и вероятностную характеристику достоверности;

- периодичность опроса.

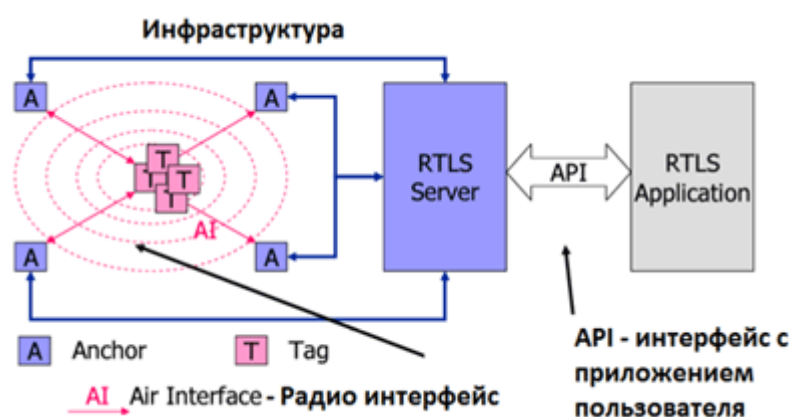


Рис. 1.3. Система позиционирования в режиме реального времени (Real Time Location Services, RTLS)

Функционал решения системы RTLS включает элементы аналитики. Например, функция «Черный ящик» — дает возможность наглядно и объективно анализировать в ретроспективе ситуации и процессы, используя накопленную информацию о перемещениях помеченных людей или объектов, многократно воспроизводить процесс в различном масштабе и с разной скоростью, чтобы изучить все тонкости.

Основные используемые для позиционирования группы технологий — это:

- радиочастотные технологии;
- спутниковые технологии навигации (GPS, ГЛОНАСС);

- технологии локального позиционирования (инфракрасные и ультразвуковые);

- радиочастотные метки – RFID.

В RTLS используется сочетание стандартных технологий. Применяется технология CSS в соответствии со стандартом ISO24730-5 (линейно-частотная модуляция).

Объекты, контролируемые системой, снабжаются метками-тегами RTLS с уникальным идентификационным номером. Основными элементами инфраструктуры являются анкера. Они закрепляются в точках с известными координатами, относительно которых осуществляется позиционирование. Анкерами называются неподвижные беспроводные базовые станции. Каждый анкер выполняет, по меньшей мере, две функции:

- обеспечивает измерение расстояний до меток;
- выполняет функции узла беспроводной сети.

Получается, каждый анкер должен иметь два радио интерфейса:

- CSS, служит для измерения расстояний и управления метками;
- ZigBee, служит для создания ячеистой беспроводной сети;
- Сеть ZigBee включает несколько типов устройств (базовых станций);
- координаторы;
- маршрутизаторы;
- конечные устройства.

Координатор имеет право запускать сеть и управлять ею. Также в его функции входит задавать настройки во время подключения устройства к сети и отвечать за ключи безопасности. Он является центром управления всей сети, так сказать доверительным центром.

Маршрутизатор присоединяется к координатору или к другим маршрутизаторам, он выполняет функцию поддержки дочерних устройств. Также он расширяет область покрытия сети, осуществляет маршрутизацию пакетов по сети. Он должен быть готов к передаче данных в любой момент времени. Самое важное в работе устройства, это способность его



восстанавливать маршруты в случаях перегрузки сети или отказа какого-либо устройства.

Конечное устройство не может транслировать пакеты и осуществлять маршрутизацию, но оно может принимать и отправлять сообщения. Они подключаются к маршрутизатору или к координатору и не имеют возможности поддерживать дочерние устройства. Таким образом, анкеры образуют сегменты самоорганизующейся самовосстанавливающейся беспроводной ячеистой системы именно при включении. Ячеистая структура обеспечивает повышенную живучесть сети, то есть система способна продолжать работу при выходе из строя любого элемента. После установки анкеров появляется возможность отслеживать перемещения объекта прямо на мониторе оператора, также сохранять историю для последующего анализа.

Периодически метка посылает блинки, короткие широковещательные пакеты, которые принимаются анкерами. По этим блинкам можно понять, какие из анкеров видят метку в данный момент. Используются самые передовые математические алгоритмы для расчета координат меток. Контроль местонахождения объекта ведется по особым параметрам, они задаются с автоматизированного рабочего места оператора системы. Проблемой является то, что системой позиционирования во времени используется преимущественно частотный диапазон 2,4 ГГц. Стремительно развивающийся интернет и существующие системы (от микроволновых печей до Bluetooth систем) вызывают перенасыщение и излишнюю интерференцию.

Отсюда, как следствие, происходят частые различные сбои. Многие утверждают, что «Cisco» уж каким-нибудь образом решит эту проблему, да только нужно было бы отнестись к данной проблеме с достаточной серьезностью.

Система RTLS открывает новые перспективы автоматизации во многих приложениях. К сожалению, ни одна из существующих и разрабатываемых в настоящее время технологий позиционирования по отдельности не в

состоянии обеспечить охват услугами всего разнообразного и разнородного окружения (например, одновременно открытых мест и закрытых помещений) с необходимой высокой точностью. Только если использовать систему в комплексе уже с действующими технологиями. Внедрение такого перспективного новшества позволило бы лучше удовлетворить потребности клиентов, и повысить их конкурентоспособность и доходы.

При совместном использовании данной системы RTLS и системы интеллектуального видеонаблюдения появляется возможность сравнивать данные, предоставляемые системой по идентификации и позиционировании «своих» объектов, с данными видеонаблюдения.[6]

### 3. Машинное зрение . (Рис. 1.4)

Машинное (техническое, компьютерное) зрение — один из наиболее перспективных методов автоматизации действий с применением компьютерных технологий и робототехники. В самом общем виде системы машинного зрения подразумевают преобразование данных, поступающих с устройств захвата изображения, с выполнением дальнейших операций на основе этих данных. В настоящее время машинное зрение наиболее востребовано в медицине и биотехнологиях, военной отрасли, автомобильной промышленности. Во всяком случае, в этих областях уже есть четко сформулированные задачи для машинного зрения, решением которых занимаются ведущие исследовательские центры и робототехнические компании. При этом возможное поле применения машинного зрения, по-настоящему, огромно — эти технологии могут быть использованы практически во всех сферах жизнедеятельности.

В частности, в медицине обработка полученных при помощи компьютерного зрения изображений и видеоданных может применяться для более точной постановки диагнозов, а в промышленности использование машинного зрения позволяет значительно снизить себестоимость продукции за счет частичного или полного отказа от выполнения ручных операций, таким образом, сводя к минимуму воздействие «человеческого фактора».

Система машинного зрения включает следующие основные компоненты:

- подсистему формирования изображений;
- вычислительную систему;
- алгоритмы анализа изображений.

Развитие машинного зрения интенсивно идет по всем трем перечисленным компонентам. Однако, в условиях, когда качественные устройства захвата изображений становятся вполне доступными для их не только промышленного или военного использования, но и бытового, а вычислительные мощности позволяют совершать миллиарды операций в секунду, давая тем самым возможность их использования в системах реального времени, критичным становится построение эффективных алгоритмов для анализа получаемых изображений [7].

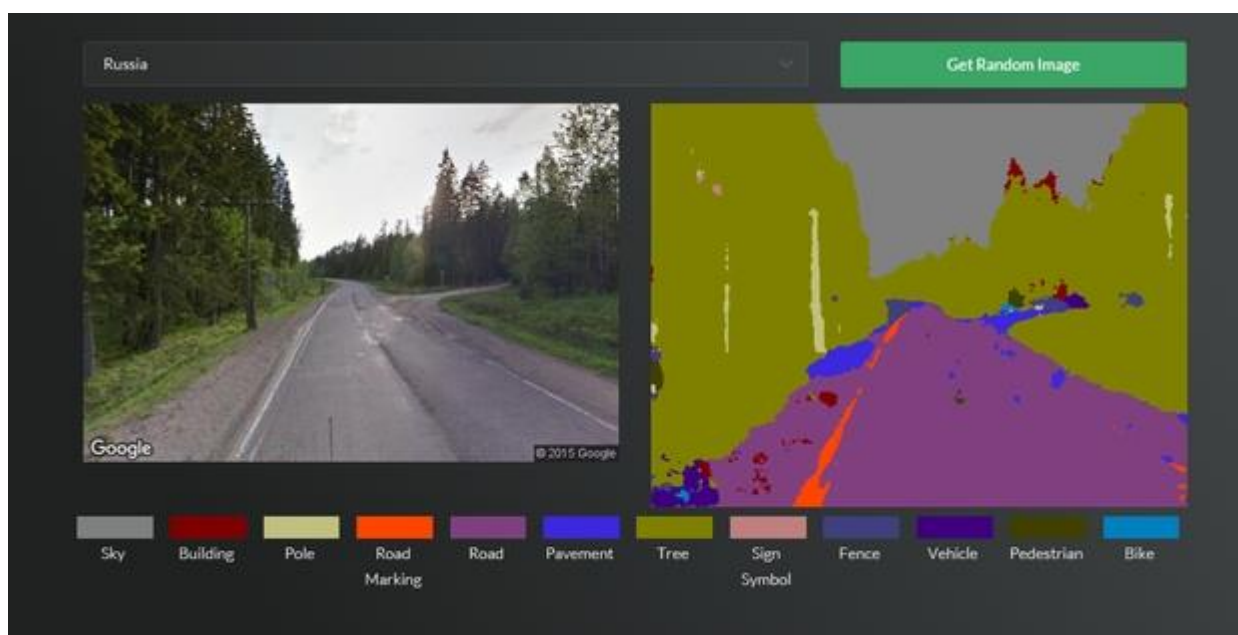


Рис. 1.4. Машинное зрение.

Одной из наиболее сложных задач в машинном зрении при применении его в робототехнике является распознавание, включающее в себя обнаружение и идентификацию объектов. В частности, решение этой задачи

позволяют выполнить оценку положения объекта относительно устройства захвата изображения.

Перспективным направлением развития машинного зрения является получение и дальнейшая обработка данных с использованием пассивных устройств захвата изображения, к которым относится камера. В отличие от применения различных датчиков, такой метод вплотную приближает работу системы машинного зрения (особенно, при использовании стереозрения) к той, что использует человек для решения зрительных задач. Другими словами, на входе и робототехническая система, и человек получают одну и ту же зрительную информацию. Разнятся только алгоритмизация и вычислительные мощности, которые используются для ее преобразования и интерпретации. Но и в этом направлении уже сделаны весомые шаги, дающие решать различные задачи в самом общем виде.

Такой подход позволяет эффективно внедрять робототехнические системы, использующие машинное зрение, непосредственно в обычную для человека среду. А это дает возможность расширить сферы применения машинного зрения, выводя их далеко за рамки жестко детерминированных сред в промышленности и постепенно захватывая все новые области его использования в быту.

Именно на таком направлении развития машинного зрения, основанном на использовании камер, специализируется наша компания. Среди наших партнеров, в сотрудничестве с которыми мы ведем наши разработки, — компании из области медицины, создания беспилотных летательных аппаратов и другие. Один из первых успешных опытов, показывающих возможности машинного зрения, был продемонстрирован нами еще в 2001 году. Тогда, совместно с немецкой компанией Neugorpower, на основе стандартной модели конструктора Lego и простой веб-камеры мы сделали робота, собирающего из девяти случайным образом разбросанных по столу кубиков логотип компании. Комплекс демонстрировался на

международной выставке в Вене, а мы были приглашены для участия в передаче Computer Club на западногерманском канале WDR.

#### 4. Движение по линии (Рис. 1.5).

Движение по линии осуществляется благодаря фото-датчикам. Существует множество подходов для решения задачи следования по линии. Выбор одного из них зависит от конкретной конструкции робота, от количества сенсоров, их расположения относительно колёс и друг друга.

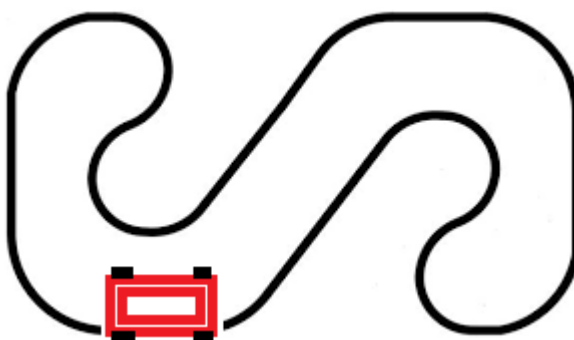


Рис. 1.5. Движение по линии

За счет того, что освещенность измеряется в ИК-диапазоне, существенным является то, что линия должна быть физической. Линия обнаруживается за счет того, что свет от светлой поверхности свет отражается значительно лучше, чем от черной. Измерения происходят следующим образом: выбирается средняя из 16 измерений разности между освещенностью при включенном и выключенном светодиоде. Если эта разность меньше порогового значения (которое пользователь может задать), то считается, что датчик находится над черной линией. Одно измерение освещенности происходит приблизительно за 60–100 микросекунд. На получение данных со всех трех датчиков уйдет не более чем  $100 \cdot 16 \cdot 3 = 4.8$  миллисекунды. Это позволяет достаточно часто опрашивать датчики.

## ГЛАВА 2. РАЗРАБОТКА ЭЛЕМЕНТОВ ПРОЕКТА

### 2.1 Выбор программных и аппаратных компонентов

На современном рынке представлено разнообразие программных и аппаратных компонентов. Сердцем нашего проекта служит аппаратная вычислительная платформа. Выбор платформы напрямую зависит от требований проекта. Различают 2 основных бренда микроконтроллерных печатных плат: Arduino и Raspberry PI. Так же придется выбирать и в остальном:

- типы двигателей;
- типы датчиков;

В зависимости от выбранной платформы будет зависеть и выбор среды программирования. Так же огромную роль играет управляющее воздействие. Так мы должны определиться, что послужит системой, которая будет управлять роботом. Выбор невелик:

- ПК;
- мобильная платформа;
- специальное программируемое оборудование;

#### 2.1.1 Выбор аппаратной вычислительной платформы

Как мы уже сказали сердцем нашего проекта будет аппаратная вычислительная платформа. Микроконтроллер является важной частью мобильного робота. Используя микроконтроллер в качестве ключевого вычислительного и логического устройства управляющей системы мобильного робота, возникает потребность в высокой частоте работы, и больших объемах оперативной памяти этого микроконтроллера.

Существуют 2 основных бренда микроконтроллерных печатных плат:

- Arduino;
- Raspberry Pi.

В каких случаях лучше выбрать Arduino? Это делается, если основная задача – считывать данные сенсоров, менять значения на двигателе или других устройствах. Учитывая требования Arduino к электропитанию и простоту обслуживания этой системы, устройство вполне можно эксплуатировать не выключая, при этом почти не вмешиваясь в его работу.

Когда лучше остановиться на Raspberry Pi? Это делается при решении таких задач, которые было бы логично выполнять на персональном компьютере. Raspberry Pi упрощает управление потоком операций в разных ситуациях: если вы подключаетесь к Интернету для считывания или записи данных, воспроизводите какую-либо медиа-информацию или подключаетесь к внешнему дисплею. [12]

Для нашего проекта идеально подходит аппаратная вычислительная платформа Arduino. Но их существует огромное множество. Рассмотрим подробнее.

Arduino — аппаратная вычислительная платформа, состоящая из двух основных компонентов: плата ввода-вывода и среда разработки на языке Processing/Wiring. Arduino удобна для разработки электронных устройств как для новичков, так и для профессионалов. Эта платформа пользуется огромной популярностью во всем мире из-за простого языка программирования, открытой архитектуры и программного кода. Особенность данной платформы является то что она программируется без использования программаторов через USB. С помощью Arduino компьютер может выйти за рамки виртуального мира в физический, благодаря множеству датчиков которые можно подключить к плате. Датчики могут получать информацию об окружающей среде, а также управлять различными исполнительными устройствами. Может и взаимодействовать с программным обеспечением на компьютере (например, Flash, Processing, MaxMSP).

Плата Arduino состоит из микроконтроллера и элементов обвязки для программирования и интеграции с другими схемами. На многих платах так же имеется линейный стабилизатор напряжения. Тактирование осуществляется на частоте 16 или 8 МГц кварцевым резонатором (прибор, в котором пьезокристаллический эффект и явление механического резонанса используется для построения высокочастотного (свойство колебательной системы, определяющее полосу резонанса и показывающее, во сколько раз запасы энергии в системе больше, чем потери энергии за один период колебаний) резонансного элемента электронной схемы. В микроконтроллер предварительно прошивается загрузчик (программа отвечающая за загрузку исполнительных файлов и запуск новых процессов) BootLoader, поэтому внешний программатор не нужен.

Микроконтроллеры "ATMEGA8" и "ATMEGA168" (Рис. 2.1) являются основой для печатных плат марки "Ардуино".

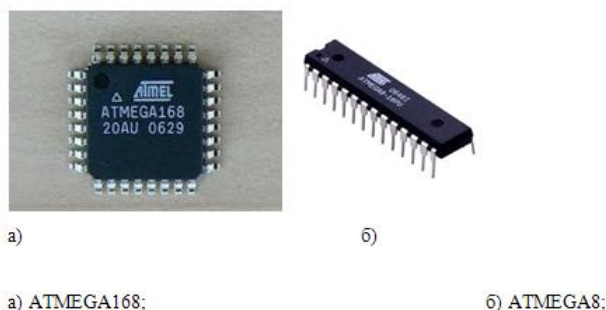


Рис. 2.1. Микроконтроллеры

Плата Arduino содержит инвертирующую схему для конвертирования уровней сигналов RS-232 (Recommended Standard 232, физический уровень для асинхронного интерфейса) в уровни ТТЛ (Транзисторно-транзисторная логика-разновидность цифровых логических микросхем, построенных на основе биполярных транзисторов (трёхэлектродный полупроводниковый прибор) и резисторов.), и наоборот. Интегрированная среда разработки Arduino — это кроссплатформенное приложение на Java, включающая в себя



редактор кода, компилятор и модуль передачи прошивки в плату. Язык программирования используемый для Arduino очень похож на СИ++, дополненный некоторыми библиотеками. Обработка программ осуществляется с помощью препроцессора, а компилируется с помощью AVR-GCC. [19]

Аппаратные средства имеют возможность расширения так как принципиальные схемы выложены в свободный доступ. Микроконтроллеры "АТМЕГА8" и "АТМЕГА168" являются основой для печатных плат марки "Ардуино" (Рис. 2.2).

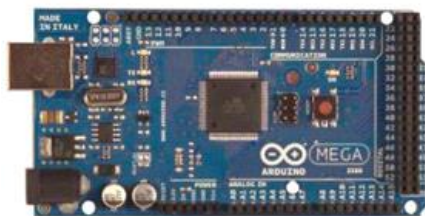


Рис. 2.2. Микроконтроллерная печатная плата торговой марки Arduino

Схемы микроконтроллерных печатных плат, марки "Ардуино", выпускаются с лицензией, а значит, опытные инженеры имеют возможность создания собственных версий таких устройств, расширяя и дополняя их. Даже обычные пользователи могут разработать опытные образцы с целью экономии средств. Топологии данных печатных плат находятся в свободном доступе.

Существует несколько плат этой торговой марки:

- Arduino Uno (Рисунок 2.3). На платформе расположены 14 контактов (pins), которые могут быть использованы для цифрового ввода и вывода. Какую роль исполняет каждый контакт, зависит от программы. Все они работают с напряжением 5 В, и рассчитаны на ток до 40 мА. Также каждый контакт имеет встроенный, но отключённый по умолчанию резистор на 20 - 50 кОм. Некоторые контакты обладают дополнительными ролями: Serial: 0-й и 1-й. Используются для приёма и передачи данных по USB. Внешнее

прерывание: 2-й и 3-й. Эти контакты могут быть настроены так, что они будут провоцировать вызов заданной функции при изменении входного сигнала. PWM: 3-й, 5-й, 6-й, 9-й, 10-й и 11-й. Могут являться выходами с широтно-импульсной модуляцией (pulse-width modulation) с 256 градациями. LED: 13-й. К этому контакту подключен встроенный в плату светодиод.



Рис. 2.3. Arduino Uno

Если на контакт выводится 5 В, светодиод загорается; при нуле — светодиод гаснет. Помимо контактов цифрового ввода/вывода на Arduino имеется 6 контактов аналогового ввода, каждый из которых предоставляет разрешение в 1024 градации. Кроме этого на плате имеется входной контакт Reset. Его установка в логический ноль приводит к сбросу процессора. Это аналог кнопки Reset обычного компьютера[8].

- Arduino Mega 2560 (Рис. 2.4). это расширенная версия Arduino Uno. Платформа выполнена на базе более продвинутого чипа ATmega2560, имеет больше контактов и большее количество аппаратных serial-портов для взаимодействия с компьютером и другими устройствами. Это самая последняя модель: Arduino Mega 2560 Rev3, пришедшая на смену предыдущим Mega 2560 и Mega 1280.

Платформа выполнена таким образом, чтобы быть максимально совместимой со своими младшими собратьями и модулями расширения (shields).

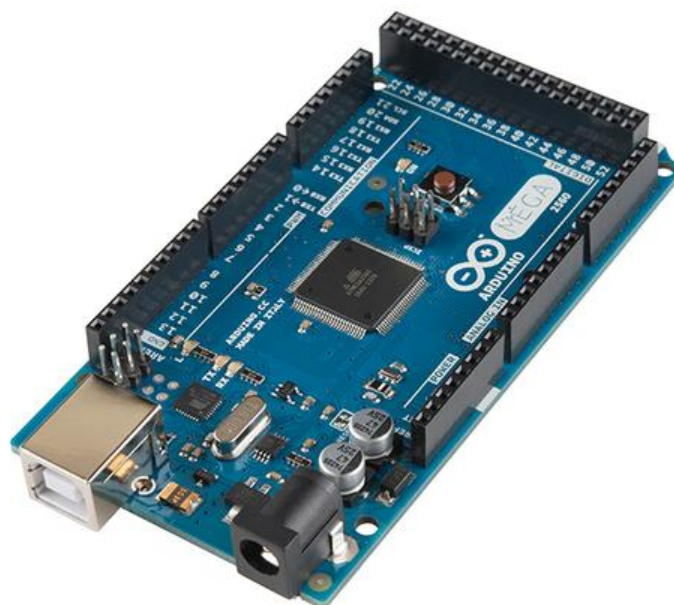


Рис. 2.4. Arduino Mega 2560

Левая часть платы по конфигурации контактов идентична Arduino Uno, как по расположению, так и по назначению. Это означает, что Arduino Mega 2560 может просто подменить Arduino Uno, если её возможностей перестало хватать. Исключение составляют 20-й и 21-й контакты, которые предназначены для коммуникации с другими устройствами по протоколу I<sup>2</sup>C. В базовой модели они совмещены с 4-м и 5-контактами. Питание, распределение напряжений, защита USB и принципы взаимодействия аналогичны базовой модели.

- Arduino Mini (Рис. 2.5). Arduino Mini — это Arduino Uno, исполненный на компактной плате. Он лишён USB порта и разъёма для питания. Это последняя, 5-я версия Arduino Mini, где в качестве чипа используется тот же ATmega328p. Arduino Mini предназначен для применения в местах, где занимаемое пространство критично. Взаимодействие Для прошивки микроконтроллера, в виду отсутствия

собственного USB-порта, можно использовать отдельный USB-Serial Converter. На время соединения портов конвертера с serial-портом платформы, взаимодействие осуществляется так же, как и с обычным Arduino.

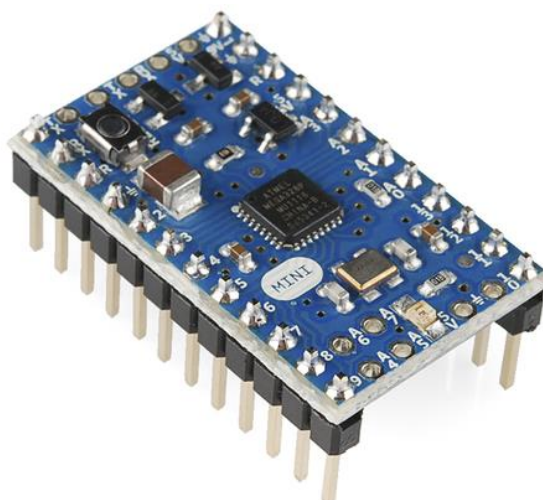


Рис. 2.5. Arduino Mini

В качестве управляющей системы была выбрана Микроконтроллерная печатная плата "Arduino Uno" которая обладает необходимым количеством вход/выходов и необходимой оперативной памятью и частотой работы.

### 2.1.2 Выбор двигателей

Проектируя мобильного робота необходимо понимать, что он предполагает в своей конструкции двигатели (моторы), которые будут вращать и (или) передвигать части конструкции или всю конструкцию.

Двигатель (мотор) - устройство преобразующее какой-либо вид энергии в механическую. Двигатели подразделяются на:

- а) электродвигатели;
- б) двигатели внутреннего сгорания;
- в) паровые двигатели.

Разные конструкции мобильных роботов отдают предпочтение разным видам двигателей. Технологически целесообразно рассматривать электрические двигатели постоянного тока, преобразующие электрическую энергию в механическое вращение, так как они позволяют упростить программный контроль, схему и конструкцию мобильного робота. Электродвигатели постоянного тока в свою очередь делятся на:

- а) коллекторные двигатели постоянного/переменного тока;
- б) бесколлекторные двигатели постоянного/переменного тока;
- в) шаговые двигатели;
- г) сервоприводы аналоговые/цифровые.

#### 1. Коллекторный электродвигатель .

Коллекторный электродвигатель - электрическая машина, в которой датчиком положения ротора и переключателем тока в обмотках является одно и то же устройство - щеточно-коллекторный узел (Рис. 2.6).



Рис. 2.6. Коллекторный электродвигатель

Основным достоинством коллекторных двигателей постоянного тока (ДПТ) является возможность регулирования частоты вращения в широком диапазоне, линейность механической и, в большинстве случаев, регулировочной характеристики, большой пусковой момент, высокое быстродействие, малая масса и объем на единицу полезной мощности и более высокий КПД по сравнению с двигателями переменного тока той же мощности. Недостатком коллекторных двигателей постоянного тока является наличие щеточно-коллекторного узла, что ограничивает их долговечность и

является источником радиопомех. Вследствие искрения на скользящем контакте эти двигатели не пригодны для эксплуатации во взрывоопасных средах. [9]

По функциональному назначению коллекторные двигатели постоянного тока подразделяются на силовые и управляемые. В свою очередь, силовые электродвигатели выполняются со стабилизацией и без стабилизации частоты вращения. Двигатели с центробежно-вибрационными регуляторами частоты вращения имеют стабильность в пределах  $\pm (2-5)\%$ . Точность стабилизации частоты вращения двигателей с электронными регуляторами зависит от принятой системы стабилизации. Статическая система стабилизации обеспечивает стабильность частоты вращения до  $\pm 0,5\%$ , а статическая система - с точностью, определяемой стабильностью частоты эталонного источника.

Центробежный регулятор - механизм, реализующий отрицательную обратную связь для регулировки скорости вращения в машинах разнообразных принципов действия и назначения.

Важным функциональным свойством двигателей является быстродействие, которое определяется в основном конструктивным исполнением и видом возбуждения. Двигатели с зубцовым якорем имеют постоянную времени 30-100 мс, с полым якорем 15-20 мс, с гладким и печатным якорями 5-10 мс. Для регулирования частоты вращения двигателей с параллельным возбуждением и с возбуждением от постоянных магнитов применяется в основном якорное управление, т.е. изменение напряжения питания якорной цепи, при этом механическая характеристика, не изменяя жесткости, смещается параллельно своему положению при  $U = U_n 0M..$  Практические пределы регулирования частоты вращения двигателей постоянного тока составляют от 1:5 до 1:20. [9]

## 2. Бесколлекторные двигатели постоянного тока.

Конструктивно бесколлекторный двигатель состоит из ротора с постоянными магнитами и статора с обмотками. Обращая внимание на то, что в коллекторном двигателе наоборот, обмотки находятся на роторе. Поэтому, далее в тексте ротор - магниты, статор - обмотки. Для управления двигателем применяется электронный регулятор (Рис. 2.7).

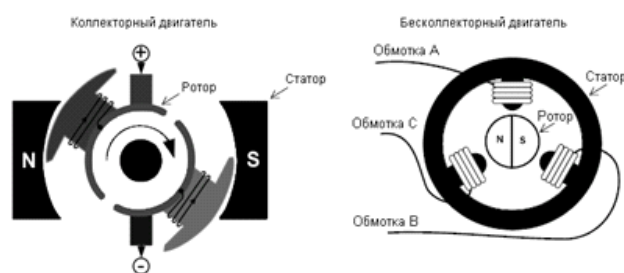


Рис. 2.7. Конструкция коллекторного и бесколлекторного двигателей

Бесколлекторный двигатель не имеет коллектора, а его функции выполняет электроника. Из конструкции двигателя удаляется довольно сложный, требующий обслуживания тяжелый и искрящий узел - коллектор. Конструкция двигателя существенно упрощается. Двигатель получается легче и компактнее. Значительно уменьшаются потери на коммутацию, поскольку контакты коллектора и щетки заменяются электронными ключами. В итоге получаем электродвигатель с наилучшими показателями КПД и показателем мощности на килограмм собственного веса, с наиболее широким диапазоном изменения скорости вращения. [13]

На практике бесколлекторные двигатели греются меньше, чем их коллекторные аналоги. Переносят большую нагрузку по моменту. Применение мощных неодимовых магнитов сделали бесколлекторные двигатели еще более компактными. Конструкция бесколлекторного двигателя позволяет эксплуатировать его в воде и агрессивных средах (разумеется, только двигатель, подвергать регулятор воздействию влаги нельзя так как это приводит к короткому замыканию, для профилактики короткого замыкания



регулятор нужно будет герметезировать, что сопровождается с большими затратами ресурсов). Бесколлекторные двигатели практически не создают радиопомех.

Единственным недостатком двигателей такого типа считают сложный дорогостоящий электронный блок управления (регулятор). Однако, если необходимо управлять оборотами двигателя, без электроники никак не обойтись. Если вам не надо управлять оборотами бесколлекторного двигателя, без электронного блока управления все равно не обойтись.

### 3. Шаговые двигатели

Шаговый электродвигатель - это синхронный бесщёточный электродвигатель с несколькими обмотками (Рис. 2.8), в котором ток, подаваемый в одну из обмоток статора, вызывает фиксацию ротора. Последовательная активация обмоток двигателя вызывает дискретные угловые перемещения (шаги) ротора.[18]

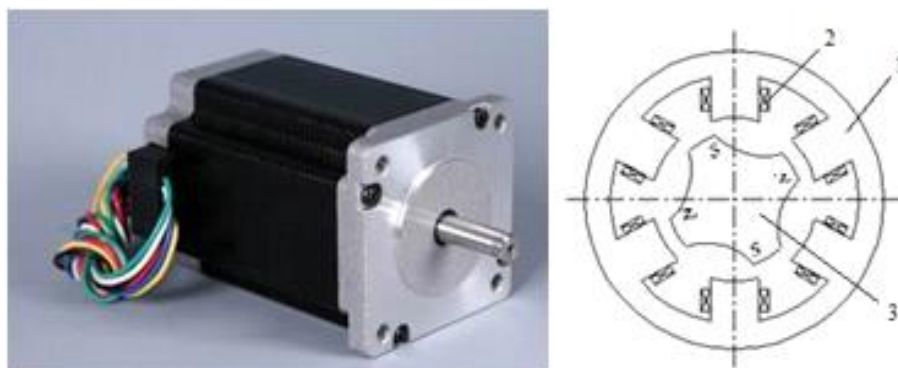


Рис. 2.8. Шаговый двигатель. а) статор, б) обмотки возбуждения, в) ротор

Конструктивно шаговые электродвигатели состоят из статора (Рисунок 2.8 а), на котором расположены обмотки возбуждения (Рис. 2.8. б), и ротора (Рис. 2.8 в), выполненного из магнито-мягкого или из магнито-твёрдого материала. Шаговые двигатели с магнитным ротором позволяют получать большой крутящий момент и обеспечивают фиксацию ротора при обесточенных обмотках.



Шаговые электродвигатели применяются в приводах машин и механизмов, работающих в старт-стопном режиме, или в приводах непрерывного движения, где управляющее воздействие задаётся последовательностью электрических импульсов, например, в станках с ЧПУ. В отличие от сервоприводов, шаговые приводы позволяют получать точное позиционирование без использования обратной связи от датчиков углового положения. Шаговые двигатели с постоянными магнитами могут использоваться в качестве датчиков угла поворота благодаря возникновению ЭДС на обмотках при вращении ротора.

#### 4. Сервоприводы.

Сервопривод - это электромотор с редуктором, управляемый через отрицательную обратную связь, которая точно управляет параметрами движения. (Рис. 2.9)

Сервоприводы различаются по размерам. И хотя официальной классификации не существует, производители давно придерживаются нескольких размеров с общепринятым расположением крепёжных элементов.

Бывают ещё так называемые сервоприводы "специального вида" с габаритами, не попадающими в данную классификацию, однако процент таких сервоприводов весьма мал.



Рис. 2.9. Сервопривод

Сервоприводы бывают аналоговые и цифровые. Внешне они ничем не отличаются: электромоторы, редукторы, потенциометры у них одинаковые, различаются они лишь внутренней управляющей электроникой. Вместо

специальной микросхемы аналогового сервопривода у цифрового аналога можно заметить на плате микропроцессор, который принимает импульсы, анализирует их и управляет мотором. Таким образом, в физическом исполнении отличие лишь в способе обработки импульсов и управлении мотором.

Судя по классификации и обозначенной во введении задачи, материалом корпуса может быть любой материал используемый для конструкций твёрдотельных роботов. Для экономии денег и создания малобаритного лёгкого робота хорошо подойдёт пластик. В качестве основного двигателя мобильного робота хорошо подойдёт простой в управлении, достаточно мощный и имеющий большой КПД - коллекторный электродвигатель постоянного тока. [14]

### **2.1.3 Выбор необходимых модулей.**

Для того, чтобы реализовать проект, необходимо определить минимальный состав модулей. Главной задачей нашей платформы является движение по линии. Поэтому необходимо выбрать датчики, которые будут определять нахождение линии.

Чтобы двигаться по линии нам необходимо 2 датчика. Каждый будет считывать уровень освещенности и микроконтроллер после обработки сигнала даст понять какого цвета линия и есть ли она.

Датчики бывают двух типов:

- цифровые;
- аналоговые.

Микроконтроллерная платформа Arduino способна работать со всеми типами датчиков, поэтому остается решить какой из них больше подходит для поставленной задачи.

**Цифровой датчик** линии позволяет определять цвет поверхности около него (Рис. 2.10). Выходом является простой бинарный цифровой

сигнал: логический 0 или 1 в зависимости от цвета, который он видит перед собой. Единица — чёрный или пустота, ноль — не чёрный. Основной фотоэлемент работает в инфракрасном спектре. Дополнительно на сенсоре установлен светодиод, который загорается когда поверхность под датчиком светлая. Это удобно для диагностики и настройки. [16]



Рис. 2.10. Цифровой датчик линии

Переменный резистор, установленный на сенсоре, позволит регулировать чувствительность сенсора в широких пределах. Это позволяет откалибровать датчик под конструкцию, материал покрытия и различные условия внешнего освещения. Сенсор может быть закреплён на днище мобильной платформы, чтобы заставить робота не выезжать за пределы территории обозначенной контуром, или чтобы он следовал за нарисованной линией. Датчик подключается к управляющей электронике через 3 провода.

**Аналоговый датчик** линии способен не только отличать чёрную поверхность от белой. В отличие от цифрового датчика линии, он способен отличить оттенки серого (Рис. 2.11). Это даёт нам возможность точно контролировать процесс перехода границы от чёрного к белому и наоборот, ведь результатом является усреднённое значение небольшого пятна под фотоэлементом. Основной фотоэлемент работает в инфракрасном спектре. Подключение Выходным результатом работы сенсора является аналоговый

сигнал. Чем светлее поверхность под сенсором — тем меньше его выходное напряжение. Датчик подключается к управляющей электронике через 3 провода. [17]



Рис. 2.11. Аналоговый датчик линии

Датчик освещает поверхность направленным ИК светодиодом (длина световой волны 940нм). Световой поток отражается от поверхности и попадает на кремниевый NPN фото-транзистор, где преобразуется в электрический сигнал. Так как используется фото-транзистор NPN типа (обратной проводимости), с нагрузкой на коллекторе, то уровень электрического сигнала на выходе «S», обратно-пропорционален отраженному от поверхности свету и прямо пропорционален удалению от отражающей поверхности. Чем светлее отражающая поверхность, тем меньше уровень сигнала на выходе «S». Чем дальше отражающая поверхность, тем выше уровень сигнала на выходе «S» (Рис. 2.12).

Из графика видно, что если датчик линии находится в 5 мм от белой отражающей поверхности, то уровень на выходе «S» будет равен  $\sim 0,3\text{В}$ . Если цвет поверхности изменится с белого на чёрный, то уровень на выходе «S» увеличится с  $\sim 0,3\text{В}$  до  $\sim 4,6\text{В}$ . Если цвет поверхности не изменился

(остался белым), а датчик удалился с 5 мм, до 20 мм, то уровень на выходе «S» увеличится до ~ 4,3В.

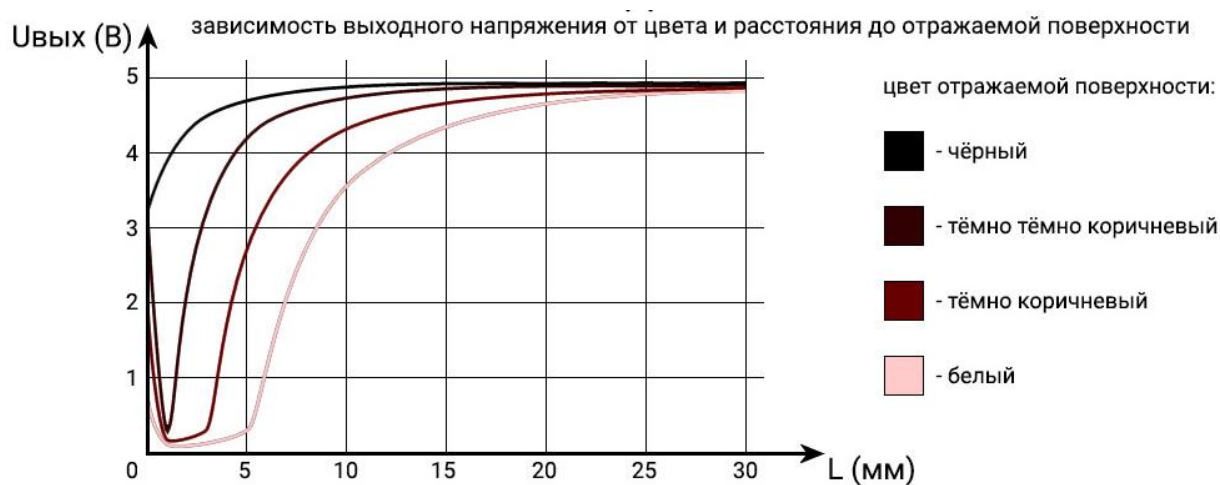


Рис. 2.12. Зависимость выходного напряжения от цвета и расстояния до отражаемой поверхности

Система управления должна управляться. Это необходимо для выбора конечной точки следования робота, корректировки маршрута, других плановых и внеплановых вмешательств в устройство.



Рис. 2.13. Bluetooth датчик HC-05

Так принципиально различают 2 способа подключения к устройству:

- проводной (подключение устройства к ПК USB кабелем);

- беспроводной (подключение устройства к ПК либо мобильному устройству посредством технологии Bluetooth/ Wi-fi).

Так как мы имеем динамически подвижное устройство, то подключать необходимо беспроводной технологией. Для своих задач идеально подходит технология Bluetooth.

**Bluetooth датчик** выступает в роли последовательного порта, через который передаются и принимаются данные (Рисунок 2.13). То есть, используя окно серийного монитора в Arduino IDE или Bluetooth приложение на вашем компьютере или телефоне, вы можете контролировать и управлять вашим проектом. Например, в качестве терминала можно использовать Teraterm. Модуль производит обмен данными с помощью контактов tx/rx.[15]

#### 2.1.4 Выбор среды программирования

Программы для Arduino пишутся на обычном C++, дополненным простыми и понятными функциями для управления вводом/выводом на контактах. Для удобства работы с Arduino существует бесплатная официальная среда программирования «Arduino IDE», работающая под Windows, Mac OS и Linux.

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 (Рис. 2.14).

Программа, написанная в среде Arduino, называется скетч. Скетч пишется в текстовом редакторе, имеющем инструменты вырезки/вставки, поиска/замены текста. Во время сохранения и экспорта проекта в области сообщений появляются пояснения, также могут отображаться возникшие ошибки. Окно вывода текста(консоль) показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию. Кнопки

панели инструментов позволяют проверить и записать программу, создать, открыть и сохранить скетч, открыть мониторинг последовательной шины.

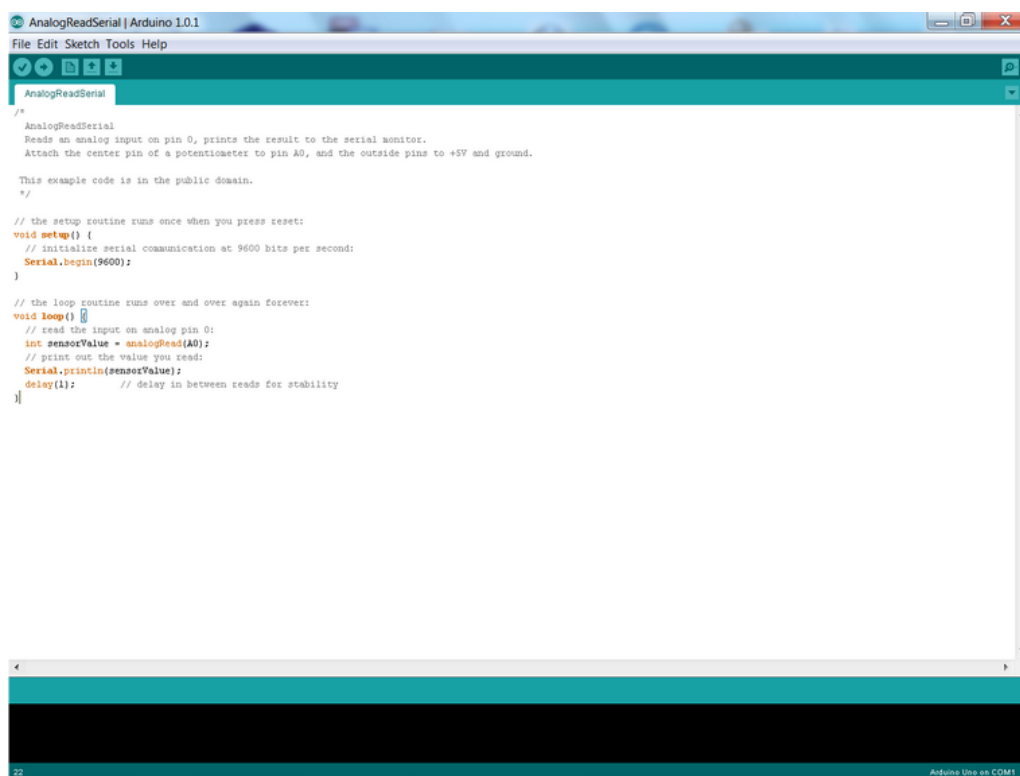


Рис. 2.14. Среда разработки Arduino IDE

Фактически программы пишутся на C/C++, а компилируются и собираются с помощью широко известного `avr-gcc`. Все особенности сводятся к тому, что имеется набор библиотек, включающий в себя некоторые функции (вроде `pinMode`) и объекты (вроде `Serial`), а при компиляции Вашей программы среда разработки создает временный `.cpp` файл, в который кроме Вашего кода включается еще несколько строчек, и полученный результат отправляется компилятору а затем линковщику с нужными параметрами.

### 2.1.5 Выбор среды управления роботом

Управляться наш робот будет при помощи мобильного устройства на платформе Android.

Android — операционная система для смартфонов, интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google, телевизоров и других устройств. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Изначально разрабатывалась компанией Android, Inc., которую затем купила Google. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать библиотеки и компоненты приложений, написанные на Си и других языках. В 86 % смартфонов, проданных во втором квартале 2014 года, была установлена операционная система Android [10].

По широте возможностей платформа Android не уступает операционным системам настольных ПК. Это многоуровневая среда на основе ядра Linux с богатыми функциональными возможностями. В подсистему пользовательского интерфейса входят:

- окна;
- представления;
- виджеты для отображения общих элементов.

Android обладает широким спектром возможностей подключения, охватывающим Wi-Fi, Bluetooth и протоколы передачи данных через сотовую сеть (GPRS, EDGE, 3G и др.).

Для разработки приложения мы будем использовать MIT App Inventor 2. MIT App Inventor - это среда программирования, разработанная учеными из Массачусетского Технологического Института (MIT). Предназначена она для разработки приложений для мобильных устройств (смартфонов и планшетных компьютеров), работающих на операционной системе Android (Рис. 2.15).

App Inventor ориентирован не на десктопное использование, а предназначен для создания приложений под мобильное устройство — смартфон или планшет с ОС Android. Он умеет, например, «понимать»



данные акселерометра мобильного гаджета, управлять встроенной камерой, видит, как ориентирован телефон в пространстве и многое другое. App Inventor — это полностью облачное приложение. Чтобы начать программировать на нём, нужен только интернет и браузер. Переход на страницу языка осуществляется по этой ссылке. Интерфейс на английском и на русском языках. Интерфейс языка программирования MIT App Inventor состоит из двух основных частей — дизайнера и редактора блоков[11]

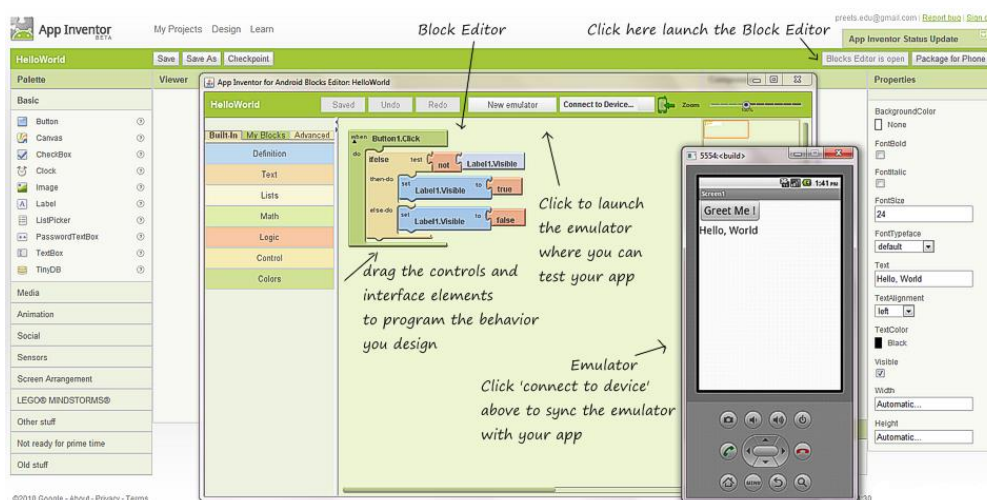


Рис. 2.15. App Inventor

В дизайнера конструируется наше приложение из элементов — экранов, кнопок, ячеек, изображений, звуков. В редакторе блоков программируется поведение этих элементов.

## 2.2 Разработка алгоритма

Существует несколько реализаций такого подхода. Зависят они напрямую от количества датчиков.

Здесь представлены варианты движения с одним и двумя датчикам линии. В этой работе использовались 3 датчика для того, чтобы распознать перекрестки и развилки.

Есть несколько алгоритмов для разного набора датчиков:

а) Движение платформы по линии с одним датчиком (Рис. 2.16.)



Рис. 2.16. Движение платформы по линии с одним датчиком

Робот будет двигаться не строго по черной линии, а по ее границе, подворачивая то влево, то вправо и постепенно перемещаясь вперед. Если датчик видит черный цвет, то робот поворачивает в одну сторону, если белый — в другую. [20]

б) Движение платформы по линии с двумя датчиками (Рис. 2.17)

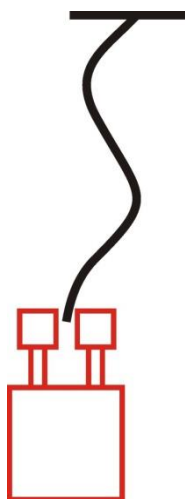


Рис. 2.17. Движение платформы по линии с двумя датчиками

Если оба датчика видят белый цвет — двигаемся вперед, если один из датчиков видит белый, а другой черный — поворачиваем в сторону черного, если оба датчика видят черный цвет — мы на перекрестке (например,

остановимся). Для реализации алгоритма потребуется отслеживать показания обоих датчиков, и только после этого задавать движение роботу. [21]

Лучше всего передать алгоритм данной системы получится в блок-схеме. В блок схеме 1 представлено движение без учета линий, то есть управление непосредственно командами (Рис. 2.18)

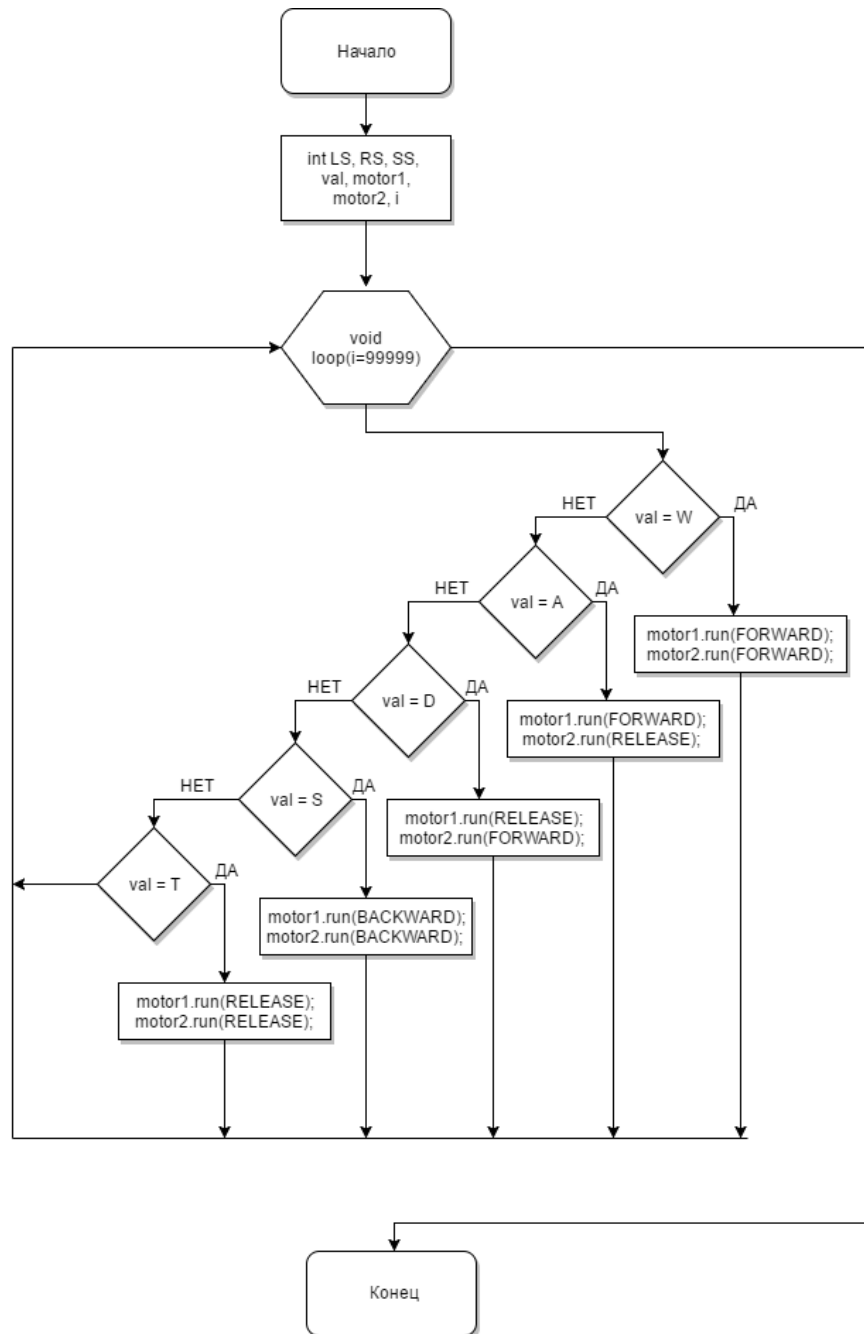


Рис. 2.18. Блок-схема управления без учета линий

Третий датчик устанавливается справа в середине всей системы. В этих местах будут находиться черные квадраты.



- возможность запуска платформы в промышленных масштабах;
- мобильность платформы;
- возможность внедрения платформы в уже созданные машины.

Все эти причины дают право нам приступать к разработке, а в последующем - модернизировать свой проект. Ведь такую платформу можно дополнять и совершенствовать множество раз.

## ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОЕКТА

Для реализации проекта понадобится реализовать аппаратную часть системы управления, реализовать скетч, реализовать управление системы.

Чтобы реализовать аппаратную часть системы управления необходимо определить минимальный набор модулей, двигателей и датчиков. Организовать необходимую площадку для размещения всех комплектующих платформы. Затем подсоединить к необходимым пинам в микроконтроллеры все выходы и входы. Так же огромную роль играет распределение энергопотребление и возможность питания от батареи. Тем самым система управления станет автономной.

Для реализации скетча необходимо ознакомиться с особенностями среды программирования, использовать необходимые библиотеки и реализовать алгоритм.

Управляться весь механизм будет с помощью мобильного приложения на платформе Android. Для этого создадим приложение и загрузим его в смартфон.

### 3.1 Реализация аппаратной части системы управления

Аппаратная часть предусматривает прежде всего сборку механизмов и физическую составляющую будущего проекта.

Организовать необходимую площадку на которой разместятся мотор-редукторы, микроконтроллер, плата управляющая мотор-редукторами, датчики освещенности и необходимыми позициями, bluetooth модуль, источник питания. Для этого идеально подходит трехколесная платформа Arduino для создания робота автомобиля (Рис. 3.1) В ее состав входит сама площадка, 2 мотор редуктора с колесами (Рис. 3.2), отсек для четырех пальчиковых батареек, который в итоге пришлось заменить на 2 параллельно

соединенные кроны. Все это соединяем согласно прилагающейся к набору инструкцией.

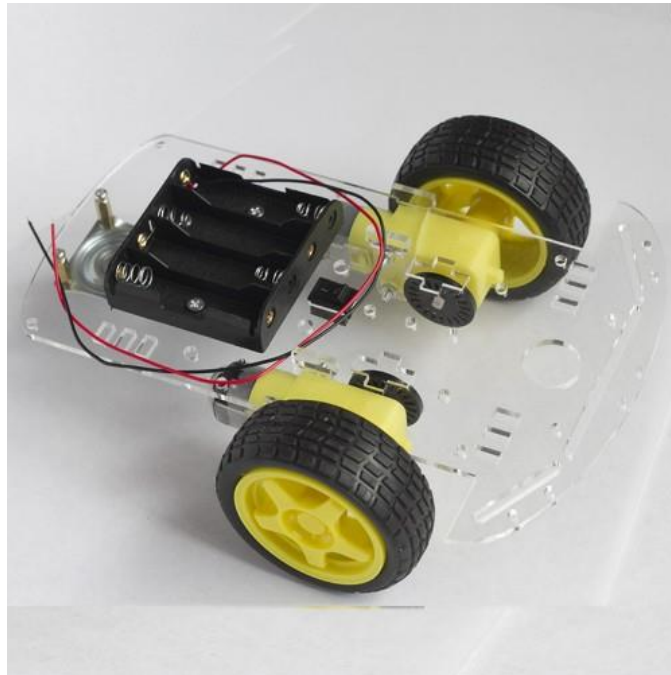


Рис. 3.1. Трехколесная платформа Arduino для создания робота



Рис. 3.2. Мотор-редктор на 6-12В с установленным колесом

Определить минимальный набор модулей и датчиков:

1. Датчики освещенности.

Для этого проекта были выбраны цифровые датчики освещенности производителя RobotDyn с тремя выводами под пины (питание +, питание -,

сигнал) (Рис. 3.3). Потребовалось 3 датчика: два из них необходимы для считывания самой линии, по которой происходит движение, 1 нужен для определения перекрестков и стоп-линий (Рис. 3.4).

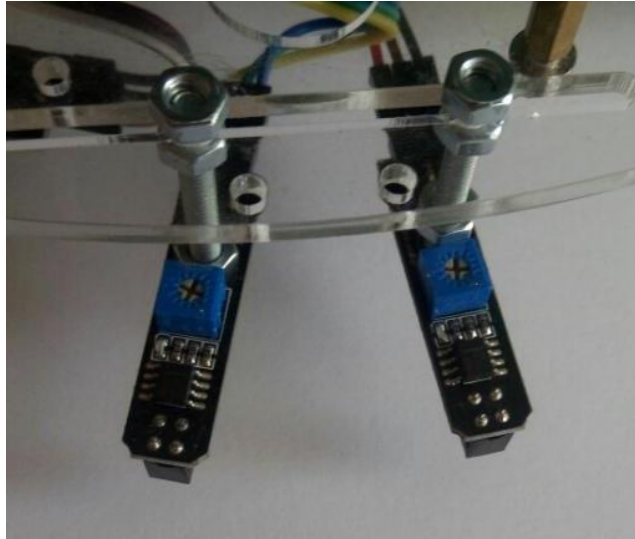


Рис. 3.3. Датчики освещенности для считывания линии движения

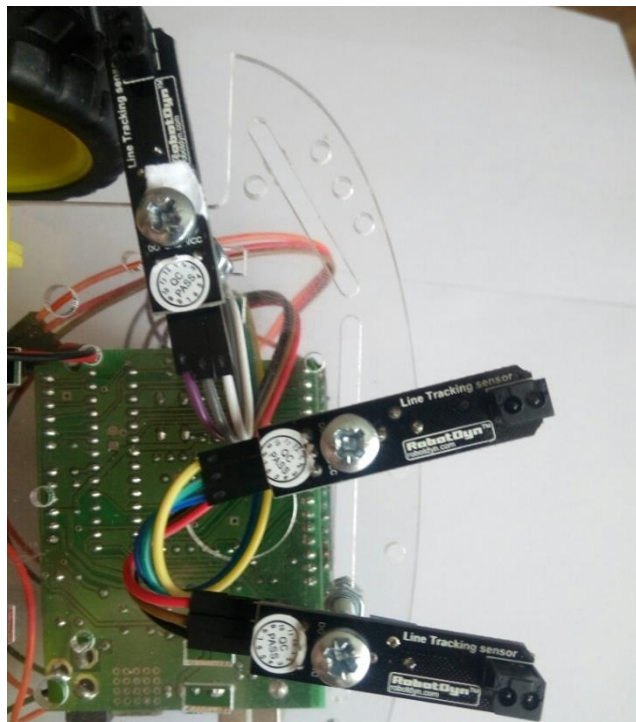


Рис. 3.4. Расположение датчиков освещенности на платформе

Датчики имеют резистор, с помощью которого регулируется чувствительность света.



## 2. Модуль bluetooth

Модуль bluetooth, с помощью которого реализуется получения сигнала с мобильного устройства (Рис. 3.5). Bluetooth модуль имеет несколько входов и выходов. В этом проекте используются только входы для получения сигнала и последующей его передачи микроконтроллеру.

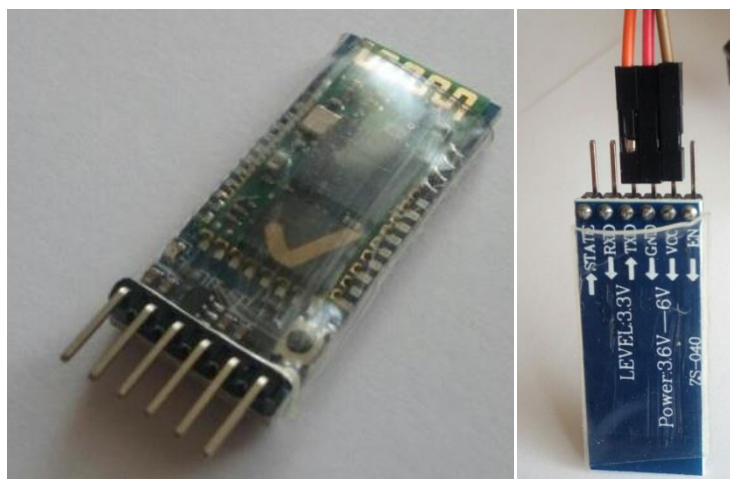


Рис. 3.5. Bluetooth модуль и его подключение

Модуль bluetooth работает от 3,6 до 6В. Модель данного модуля HC-05. Так же она имеет кнопку перезагрузки.

## 3. Motor Shield.

Данная плата позволяет подключить одновременно к Arduino до четырех коллекторных DC двигателей, либо до двух биполярных шаговых двигателей (Рис. 3.6). К тому же на плате имеются разъемы для подключения двух сервоприводов. На борту данного шилда имеется две микросхемы L293D. L позволяет управлять слаботочными двигателями с током потребления до 600 мА на канал. На двух пятипиновых клеммниках можно насчитать 4 разъема для подключения двигателей (M1, M2, M3, M4), центральные выводы на пятипиновых клеммниках соединены с землей и служат для удобства при подключении пятипроводных шаговых двигателей. Использование двух микросхем L293D позволяет одновременно подключить 4 моторчика постоянного тока либо 2 шаговых мотора, либо два моторчика и

шаговый. Также, на плату выведены 2 разъема для подключения сервоприводов.

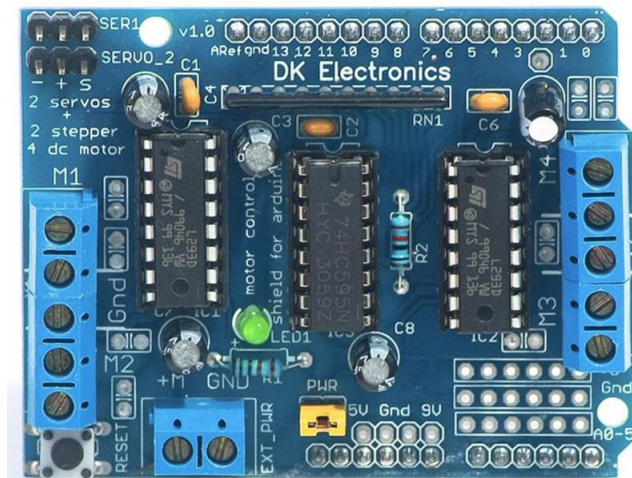


Рис. 3.6. Motor Shield для подключения к Arduino несколько мотор-редукторов.

Питание силовой части производится либо от внешнего клеммника либо замыканием джампер (Рис. 3.7). При замкнутом джампере напряжение для объединенного питания должно лежать в пределах от 6 до 12 Вольт

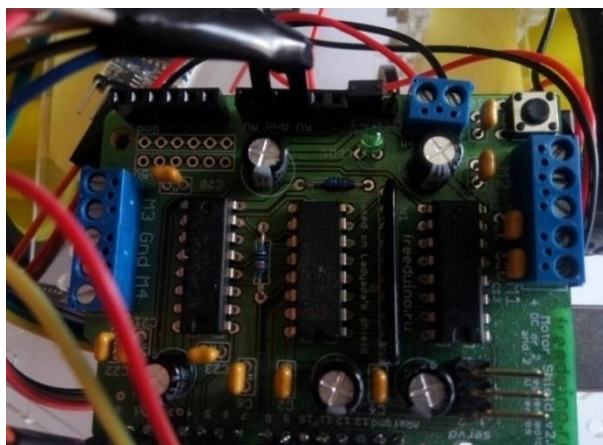


Рис. 3.7. Подключение датчиков и модулей к Motor Shield.

Сама плата дублирует цифровые и аналоговые порты для подключения периферии. В данном случае датчики освещенности и модуль bluetooth.

#### 4. Arduino UNO.

Сам микроконтроллер размещается по центру передвижной площадки. Джемпер на плате включен в режим внешнего питания (Рис. 3.8), а джемпер на Motor Shield включен в режим питания от микроконтроллера.

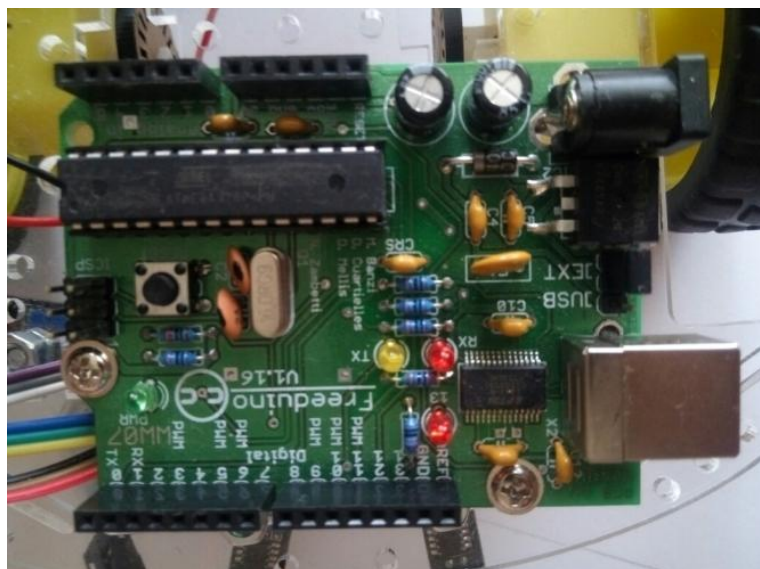


Рис. 3.8. Положение джемпера в режиме внешнего питания.

Ток будет получен от двух параллельно соединенных батареек типа Крона (Рис. 3.9). Тем самым мы добьемся повышенную емкость при постоянном напряжении.

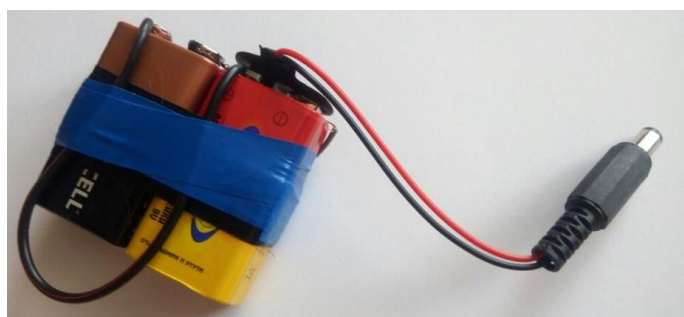


Рис. 3.9. Питание системы движения

Модуль bluetooth подключен к 1 пину цифрового входа на плате, датчики освещенности подключены к 3,5,6 пинам соответственно (Рисунок 3.10).

Питание на всех модулях и датчиках общее. Параллельно соединены клеммы (+) и (-). Подключены к питанию 5В и Ground соответственно (Рис.3.11).

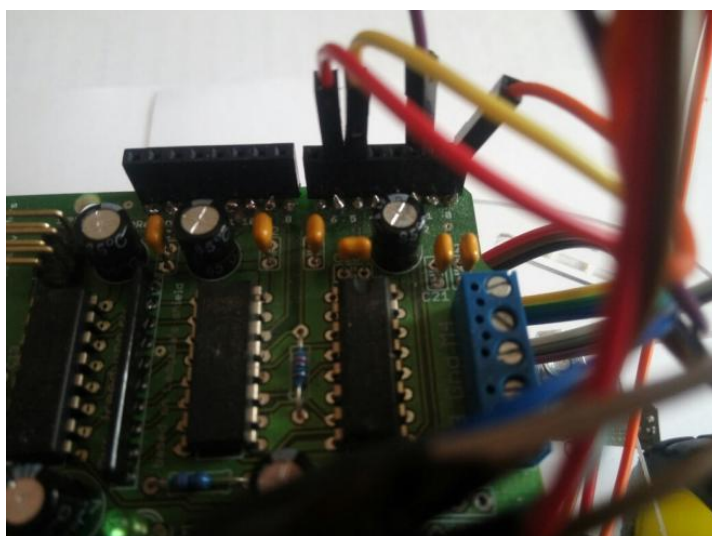


Рис. 3.10. Подключения для считывания с датчиков и модулей

В конечном итоге платформа имеет законченный вид для демонстрации (Рис. 3.12).



Рис. 3.11. Питание модулей и датчиков.

В этом варианте исполнения есть все необходимое для первого представления о микроконтроллере и способе его применения в таком варианте, как движение по линии.



Для реализации движения назад не будет хватать некоторых компонентов. А именно: двух мотор редукторов и такого же набора датчиков освещенности с противоположенной стороны платформы.

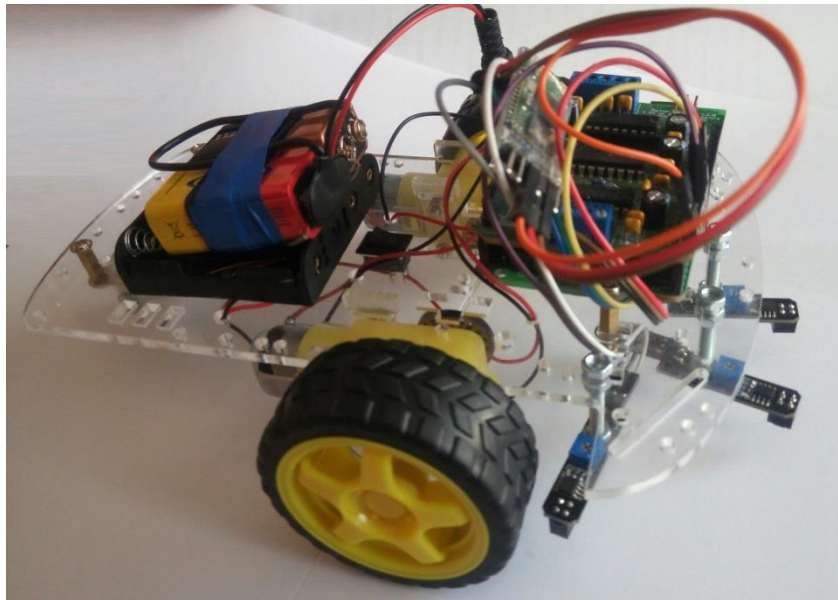


Рис. 3.12. Модель дл демонстрации в сборе.

### 3.2 Реализация скетча

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 (Рис. 3.13). Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino. Скетч - это программа, написанная для платформы Arduino и имеющая определенную структуру.

Скетч обязательно содержит 2 функции: функцию setup и функцию loop. Прошивка Arduino при включении вызывает функцию setup. Функция setup вызывается лишь раз, при каждом запуске платы. Это место идеально подходит для инициализации (задания начальных значений) переменных, установки режимов пинов (ввод/вывод), задания соответствия подключенных датчиков/сервоприводов/прочего с пинами. После выполнения функции setup

идет циклический вызов функции loop (т.е. сразу после выхода из функции setup, выполняется функция loop, после выхода из неё, она же вызывается снова). Процесс продолжается пока питание не будет отключено.

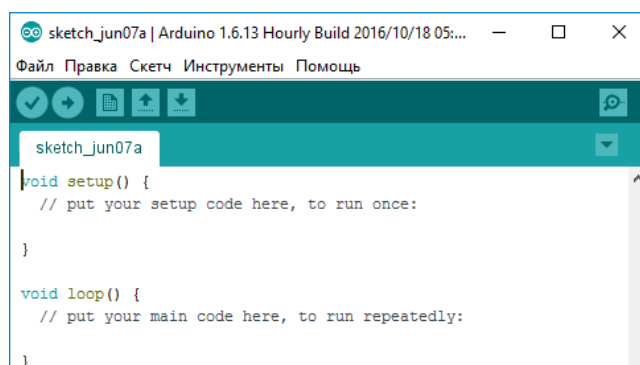


Рис. 3.13 Программная среда разработки.

Чтобы реализовать полученный ранее алгоритм, необходимо воспользоваться библиотекой "AFMotor", которой управляется Motor Shield (машинная часть проекта). До выполнения функций необходимо объявить библиотеки и переменные. После объявления входных параметров необходимо указать пин, через который приходят или выходят данные. Мотор редукторы должны иметь позицию подключения и значение ШИМ (Листинг 3.1).

Листинг 3.1

```
#include <AFMotor.h>
#define Left 6 // левый датчик
#define Right 5 // правый датчик
#define Stopp 2 // стоп датчик
int val;
AF_DCMotor motor2(2, MOTOR12_8KHZ); // объект motor №2, ШИМ 64 кГц
AF_DCMotor motor1(1, MOTOR12_8KHZ); // объект motor №1, ШИМ 64 кГц
```

В функции setup() необходимо задать скорость передачи по последовательному порту и скорость вращения двигателей (Листинг 3.2).

Значение вращения двигателей меняется от 100 до 255, где 100 - это полная остановка, а 255 максимальные обороты мотор-редуктора.

### Листинг 3.2

```
void setup()
{
  Serial.begin(9600); // устанавливаем скорость передачи по посл. порту 9600
  motor2.setSpeed(200); // устанавливаем скорость вращения 100/255
  motor1.setSpeed(200); // устанавливаем скорость вращения 100/255
}
```

В функции loop() объявляем переменные, в которые будет приходить информация с датчиков (Листинг 3.3).

### Листинг 3.3.

```
void loop()
{
  int LS = digitalRead(Left);
  int RS = digitalRead(Right);
  int SS = digitalRead(Stopp);
}
```

Так как система управления будет двигаться не только по линии, но и работать в ручном режиме, необходимо задать условия. Значения, которые будут приходить с мобильного устройства полностью управляют платформой (Листинг 3.4). Стоит отметить некоторые значения для мотор-редукторов:

1. motor.run(FORWARD) - вращение вперед
2. motor.run(BACKWARD) - вращение назад
3. motor.run(RELEASE) - остановка вращения

Так же отдельно в каждом условии можно задать скорость вращения двигателей.

Листинг 3.4.

```

motor1.run(FORWARD); // вращение вперед
motor2.run(FORWARD); } // вращение вперед
if ( val == 'S'){ // При нажатии клавиши "S"
    motor1.run(BACKWARD); //Движение назад
    motor2.run(BACKWARD); } //Движение назад
if ( val == 'D'){ // При нажатии клавиши "D"
    motor2.run(FORWARD); // вращение вперед
    motor1.run(RELEASE); } //стоп машина
if ( val == 'A'){ // При нажатии клавиши "A"
    motor2.run(RELEASE); //стоп машина
    motor1.run(FORWARD); } // вращение вперед
// При отпускании клавиш в программе в порт шлется "T"
if ( val == 'T') { // При нажатии клавиши "T"
    motor1.run(RELEASE); //стоп машина
    motor2.run(RELEASE); } } // стоп машина

```

После необходимо запрограммировать платформу, чтобы она умела распознавать линию. Для этого переведем описанный ранее алгоритм в программный код. С датчиков освещенности приходит сигнал либо 1, либо 0, поэтому будем останавливать тот мотор со стороны которого датчик прислал 1. Так же не стоит забывать о третьем датчике, который служит для распознавания перекрестка - в каждом условии необходимо учитывать его значение. Начало движения контролируют сигналы с мобильного устройства. Для движения по разным траекториям был реализован разный алгоритм на разных кнопках. Так при нажатии на желтую кнопку платформа поедет влево на первой развилке, а при нажатии на зеленую поедет вправо (Листинг 3.5). В



листинге рассмотрим только одну траекторию, остальные будут находиться в Приложении.

Листинг 3.5.

```

if(val == 'G'){ //При нажатии на зеленую кнопку
    if(LS==0 && RS==0 && SS==0) { // Движение вперед
        motor1.run(FORWARD); // вращение вперед
        motor2.run(FORWARD); } // вращение вперед
    if(LS==0 && RS==0 && SS==1) { // Движение вперед
        motor1.run(FORWARD); // вращение вперед
        motor2.run(FORWARD); } // вращение вперед
    if( LS==1 && RS==0 && SS==0) { // Поворот направо
        motor1.run(FORWARD); // вращение вперед
        motor2.run(RELEASE); } //стоп машина
    if( LS==1 && RS==0 && SS==1) { // Поворот направо
        motor1.run(FORWARD); // вращение вперед
        motor2.run(RELEASE); } //стоп машина
    if(LS==0 && RS==1 && SS==0) { // Поворот налево
        motor1.run(RELEASE); //стоп машина
        motor2.run(FORWARD); }// вращение вперед
    if(LS==0 && RS==1 && SS==1) { // Поворот налево
        motor1.run(RELEASE); //стоп машина
        motor2.run(FORWARD); }// вращение вперед
    if (LS==1 && RS==1 && SS==1) { //Стоп
        motor1.run(RELEASE); //стоп машина
        motor2.run(RELEASE); } //стоп машина
    if (LS==1 && RS==1 && SS==0) { //Развилка
        motor2.run(RELEASE); //стоп машина
        motor1.run(FORWARD); motor1.setSpeed(200); } // вращение вперед

```

### 3.3 Реализация управления с мобильной платформы

Сигналы, приходящие через bluetooth адаптер, регулируют движение платформы. Управляться система управления будет при помощи мобильного устройства на платформе Android.

По широте возможностей платформа Android не уступает операционным системам настольных ПК. Это многоуровневая среда на основе ядра Linux с богатыми функциональными возможностями. В подсистему пользовательского интерфейса входят:

- окна;
- представления;
- виджеты для отображения общих элементов.

Для разработки приложения выбран MIT App Inventor 2.

MIT App Inventor 2 имеет два основных окна интерфейса для разработки.

В окне Designer разрабатывается внешний вид нашего приложения и создаются кнопки, которые, собственно, и будут посылать сигналы на платформу (Рис. 3.14).

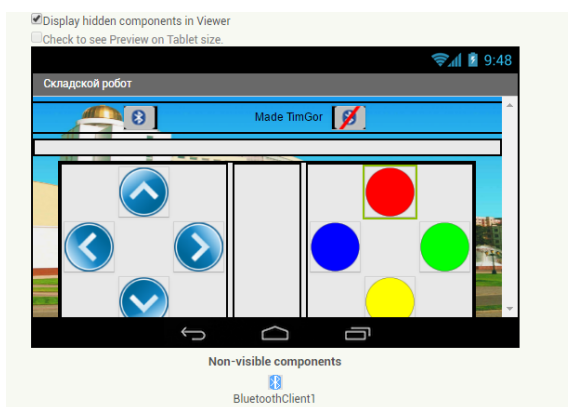


Рис. 3.14. Внешний вид приложения.

Слева находится бокс-окно, в котором расположены кнопки формы, текст-боксы (Рисунок 3.15). Справа находится окно компонентов, где мы присваиваем кнопкам, формам, bluetooth клиенту значение (Рис. 3.16).

В вкладке Properties выставляются параметры такие как (размер, цвет, изображение) и присваиваются значение.

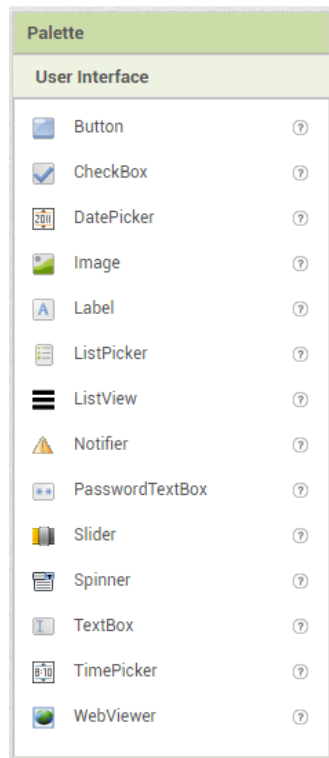


Рис. 3.15. Вкладка виджетов.

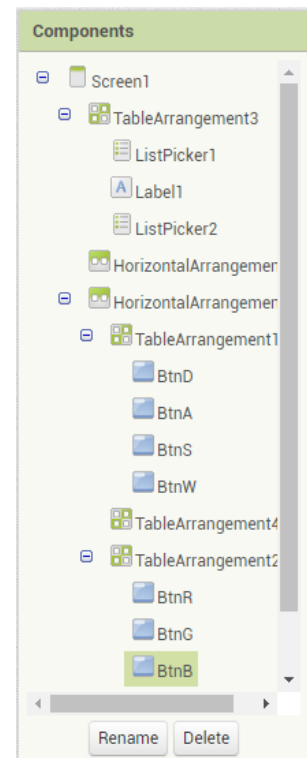


Рис. 3.16. Вкладка компонентов.

Второе окно разработки - blocks, в котором программируются кнопки, модули, формы из окна Designer (Рис. 3.17).

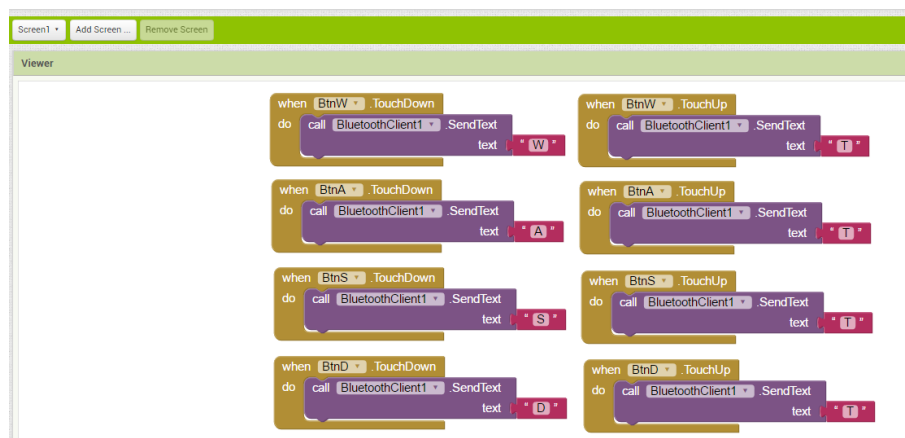


Рис. 3.17. Окно разработки Designer

Программирование состоит из форм в виде пазл, что облегчает процесс программирования.

Создаем условия:

1. Если нажата кнопка "W", то на устройство шлетя движение вперед.
2. Если нажата кнопка "A", то на устройство шлетя движение влево.
3. Если нажата кнопка "S", то на устройство шлетя движение вправо.
4. Если нажата кнопка "D", то на устройство шлетя движение назад.
5. Если все эти кнопки не нажаты, то на устройство шлетя "T" - остановка всех двигателей

После создания приложение необходимо откомпилировать и скачать на компьютер. Для этого переходим в вкладку Build-App(save .apk to my computer) (Рис. 3.18). После этого приложение компилируются и сохраняется на компьютер.

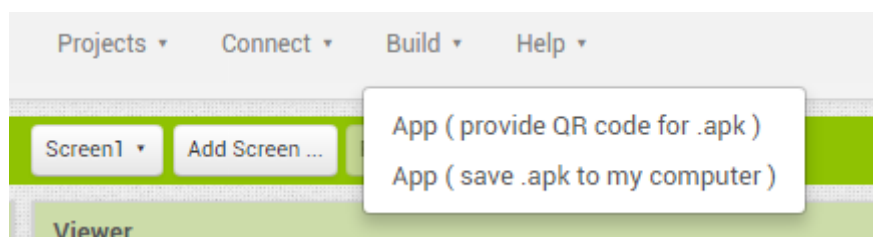


Рис. 3.18. Компиляция и создание приложения

После загрузки приложения в телефон устанавливаем его и соединяемся с платформой. После этого устройство готов и можно приступать к апробации.

## ГЛАВА 4. АПРОБАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ

Разработанная в рамках магистерской диссертации система управления изначально предназначалась для использования в любых системах транспортировки и доставки складских помещений. Структура системы настолько гибкая, что использовать такую систему можно в любом производстве и в любой службе.

Достигается это с помощью универсального кода и возможности модернизации системы. В процессе внедрения системы одной из особенностей стала настройка датчиков распознавания линий.

На каждом датчике присутствует надстроечный резистор, который регулирует чувствительность попадания света в принимающий сенсор. Так для данной системы эмпирическим методом было выявлено его значение. Значение надстроечного резистора на всех датчиках было выставлено на 70%.

При таком значении резистора с помощью тестовых запусков найдена идеальная высота от нижней части датчика до поверхности по которой движется платформа. Это значение - 7-8мм.

Важным моментом стала сама линия по которой двигается платформа. Она должна быть темного или черного цвета. Более того цвет должен полностью поглощать излучение от датчиков. Эмпирическим методом было выявлено, что подходит черная диэлектрическая лента и линия напечатанная на лазерном принтере.

Следующим шагом стало определение ширины ленты и расстояние между датчиками. Эти параметры зависят друг от друга. С помощью многих тестов было определено: ширина линии, по которой движется платформа - 19 мм., расстояние между датчиками - 35мм. Все эти данные верны и удобны в

использовании при следующих габаритах платформы: ширина (колесная база) - 115 мм., длина - 125 мм.

Одним из важных параметров стал расчет энергопотребления платформы. Идеально для такой платформы подойдут перезаряжаемые аккумуляторы на 9 В с емкостью 3000-5000 mAh. В демонстрационной версии же использовались две батарейки типа Крона на 9В, соединенные параллельно.

## ЗАКЛЮЧЕНИЕ

В результате выполнения магистерской диссертации была создана автоматизированная система управления роботом погрузчиком на платформе Android под управлением с мобильного устройства. Для этого была разработана физическая модель робота-погрузчика, программный код (скетч), который загружается в устройство и мобильное приложение на мобильное устройство на платформе Android.

Следует отметить, что в процессе выполнения данной работы была выполнена поставленная цель: "Разработка автоматизированной системы управления роботом-погрузчиком на основе контроллера Arduino "

Для достижения цели магистерской диссертации выполнены поставленные задачи:

- анализ методов и средств автоматизации складского учета;
- формирование требований к системе управления;
- формулировка требований к разрабатываемой специализированной микроконтроллерной платформе;
- выбор аппаратных средств;
- разработка и реализация проекта на платформе Arduino под управлением Android;
- апробация проекта.

Особенностью данной работы стала ее разработка. Разрабатывался проект с изучением и использованием сразу нескольких систем.

Во-первых, большое внимание уделялось выбору аппаратной платформы. Был проведен сравнительный анализ существующих микропроцессорных платформ. Выбор пал на микропроцессорную платформу Arduino. Основными критериями выбора считались:

- низкая стоимость;
- кроссплатформенность;

- простая и понятная среда программирования;
- аппаратные средства с возможностью расширения и открытыми принципиальными схемами.

Во-вторых, были изучены методы основной задачи, от которой принципиально зависела разработка - движение платформы по местности. Выбором стало движение системы по черной линии с возможностью поворотов и нахождения траектории. Преимущества этого метода:

- стабильный алгоритм;
- низкая стоимость необходимых компонентов (модулей и датчиков);
- эффективность реализации;
- устойчивость к всевозможным факторам;
- низкое энергопотребление.

В-третьих, был проведен анализ существующих систем для управления данной физической модели. Для этого было выбрано управление с мобильных устройств с операционной системой Android. Преимущества выбора:

- широкое распространение устройств;
- интеграция с Google;
- большой выбор средств для разработки приложений;
- стоимость устройств;

Как можно заметить в магистерской диссертации удалось упростить процесс существования складских помещений за счет внедрения систем управления на платформе Arduino. Это введение значительно снизит потребность в ресурсах и повысит производительность работы склада. С помощью данной системы управления увеличится скорость доставки, правильность сортировки.

Конечно, данную систему управления можно модернизировать. К ее усовершенствованию можно отнести:

- установку еще двух мотор-редукторов;



- установку такого же набора датчиков распознавания линий с противоположной стороны;

- установку датчиков расстояний;

- установку камер видеонаблюдения;

- установку электромагнитной площадки для транспортировки грузов.

Все эти введения помогут еще повысить производительность системы и не потребуют больших денежных ресурсов.

Подводя итоги, можно сказать, что данная система управления удачно в себе сочетает удобство и автономность использования с функциональными возможностями.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. А.И. Корендясев, Б.Л. Саламандра, Л.И. Тывес; "Теоретические основы робототехники." В 2 кн. Институт машиноведения им. А.А. Благонравова Р.А.Н. - М.: Наука, 2006г, Кн. 1, 468с.
2. А.И. Корендясев, Б.Л. Саламандра, Л.И. Тывес; "Теоретические основы робототехники." В 2 кн. / Институт машиноведения им. А.А. Благонравова Р.А.Н. - М.: Наука, 2006г, Кн. 2, 269с.
3. Е.И. Юревич; "Основы робототехники" 2-е издание перераб. и доп. СПб: БХВ - Петербург, 2005г, 205с.
4. Domino и роботы-курьеры [Интернет портал] URL: <https://www.retail.ru/news/142570/> (дата обращения: 10.02.2017)
5. Как работает лидар? [Интернет портал] URL: [http://www.laserportal.ru/content\\_979](http://www.laserportal.ru/content_979) (дата обращения: 17.03.2017)
6. ТЕХНОЛОГИЯ ПОЗИЦИОНИРОВАНИЯ В РЕАЛЬНОМ ВРЕМЕНИ, ТЕХНОЛОГИЯ RTLS. [Интернет портал] URL: <https://nauchforum.ru/studconf/tech/iv/1232> (дата обращения: 17.09.2016)
7. Машинное зрение [Интернет портал] URL: <http://robodem.ru/machinevision> (дата обращения: 20.09.2016)
8. Arduino Uno [Интернет портал] URL: <http://amperka.ru/product/arduino-uno> (дата обращения: 20.01.2017)
9. Щеточно-коллекторный узел [Интернет портал] URL: <http://www.ngpedia.ru/id536040p2.html> (дата обращения: 20.02.2017)
10. Android [Интернет портал] URL: <https://ru.wikipedia.org/wiki/Android> (дата обращения: 08.04.2017)

11. MIT App Inventor — каждый может создать мобильное приложение [Интернет портал] URL: <https://newtonew.com/app/mit-app-iventor-intro> (дата обращения: 08.04.2017)
12. Raspberry Pi [Интернет портал] URL: [https://ru.wikipedia.org/wiki/Raspberry\\_Pi](https://ru.wikipedia.org/wiki/Raspberry_Pi) (дата обращения: 10.04.2017)
13. Бесколлекторные двигатели постоянного тока. [Интернет портал] URL: <http://www.avislab.com/blog/brushless01/> (дата обращения: 10.03.2017)
14. Сервоприводы [Интернет портал] URL: <http://wiki.amperka.ru/робототехника:сервоприводы> (дата обращения: 16.03.2017)
15. Bluetooth модуль HC-06 подключение к Arduino. Управление устройствами с телефона. [Интернет портал] URL: <https://lesson.iarduino.ru/page/bluetooth-modul-hc-06-podklyuchenie-k-arduino-upravlenie-ustroystvami-s-telefona/> (дата обращения: 26.03.2017)
16. Датчик линии цифровой [Интернет портал] URL: <http://amperka.ru/product/digital-line-sensor> (дата обращения: 27.03.2017)
17. Датчик линии аналоговый [Интернет портал] URL: <http://amperka.ru/product/analog-line-sensor> (дата обращения: 27.03.2017)
18. Как работают шаговые двигатели [Интернет портал] URL: <http://robotosha.ru/electronics/how-stepper-motors-work.html> (дата обращения: 25.01.2017)
19. RS-232 [Интернет портал] URL: <https://www.arduino.cc/en/Tutorial/ArduinoSoftwareRS232> (дата обращения: 30.01.2017)

20. Движение по линии на одном датчике цвета [Интернет портал] URL:  
[http://studrobots.ru /движение-по-линии-на-одном-датчике-цвете/](http://studrobots.ru/движение-по-линии-на-одном-датчике-цвете/) (дата  
обращения: 18.02.2017)
21. Движение по линии с двумя датчиками [Интернет портал] URL:  
<http://studrobots.ru/движение-по-линии-с-двумя-датчиками-2/> (дата  
обращения: 30.01.2017)

## ПРИЛОЖЕНИЕ

### Листинг программы

```
#include <AFMotor.h>

#define Left 6    // левый датчик
#define Right 5  // правый датчик
#define Stopp 2  // стоп датчик

int val;

AF_DCMotor motor2(2, MOTOR12_8KHZ); // создаем объект motor №2,
ШИМ 64 кГц
AF_DCMotor motor1(1, MOTOR12_8KHZ); // создаем объект motor №1,
ШИМ 64 кГц

void setup()
{
  Serial.begin(9600); // устанавливаем скорость передачи по посл. порту 9600
  бод
  motor2.setSpeed(200); // устанавливаем скорость вращения 100/255
  motor1.setSpeed(200); // устанавливаем скорость вращения 200/255
}

void loop()
{
  int LS = digitalRead(Left);
  int RS = digitalRead(Right);
```

```
int SS = digitalRead(Stopp);  
if (Serial.available())  
{  
    val = Serial.read();  
  
    // Задаём движение вперёд  
    if (val == 'W') // При нажатии клавиши "W"  
    {  
        motor1.run(FORWARD); // вращение вперед  
        motor2.run(FORWARD); // вращение вперед  
    }  
  
    // Задаём движение назад  
    if ( val == 'S')  
    {  
        motor1.run(BACKWARD); //Движение назад  
        motor2.run(BACKWARD); //Движение назад  
    }  
  
    // Задаём движение вправо  
    if ( val == 'D')  
    {  
        motor2.run(FORWARD); // вращение вперед  
        motor1.run(RELEASE); //стоп машина  
    }  
}
```

```
// Задаём движение влево
if ( val == 'A')
{
    motor2.run(RELEASE); //стоп машина
    motor1.run(FORWARD); // вращение вперед
}

// Стоп режим
// При отпускании клавиш в программе в порт шлется "T"
if ( val == 'T') // При нажатии клавиши "T"
{
    // Выводы ENABLE притянуты к минусу, моторы не работают
    motor1.run(RELEASE); //стоп машина
    motor2.run(RELEASE); // стоп машина
}
}

// Задаём движение по линии 1 склад НАЛЕВО

// Задаём движение по линии 1 склад

if(val == 'G')
{
    if(LS==0 && RS==0 && SS==0) // Движение вперед
```

```
{  
    motor1.run(FORWARD); // вращение вперед  
    motor2.run(FORWARD); // вращение вперед  
}  
if(LS==0 && RS==0 && SS==1) // Движение вперед  
{  
    motor1.run(FORWARD); // вращение вперед  
    motor2.run(FORWARD); // вращение вперед  
}  
  
if( LS==1 && RS==0 && SS==0) // Поворот направо  
{  
    motor1.run(FORWARD); // вращение вперед  
    motor2.run(RELEASE); //стоп машина  
}  
if( LS==1 && RS==0 && SS==1) // Поворот направо  
{  
    motor1.run(FORWARD); // вращение вперед  
    motor2.run(RELEASE); //стоп машина  
}  
  
if(LS==0 && RS==1 && SS==0) // Поворот налево  
{  
    motor1.run(RELEASE); //стоп машина
```



```

    motor2.run(FORWARD); // вращение вперед
}
if(LS==0 && RS==1 && SS==1) // Поворот налево
{
    motor1.run(RELEASE); //стоп машина
    motor2.run(FORWARD); // вращение вперед
}
if (LS==1 && RS==1 && SS==1) //Стоп
{
    motor1.run(RELEASE); //стоп машина
    motor2.run(RELEASE); //стоп машина
}
if (LS==1 && RS==1 && SS==0) //Развилка
{
    motor2.run(RELEASE); //стоп машина
    motor1.run(FORWARD); // вращение вперед
    motor1.setSpeed(200); // устанавливаем скорость
вращения 200/255
}}

```

// Задаём движение по линии 2 склад НАПРАВО

// Задаём движение по линии 2 склад

```
if(val == 'Y')
{
    if(LS==0 && RS==0 && SS==0) // Движение вперед
    {
        motor1.run(FORWARD); // вращение вперед
        motor2.run(FORWARD); // вращение вперед
    }
    if(LS==0 && RS==0 && SS==1) // Движение вперед
    {
        motor1.run(FORWARD); // вращение вперед
        motor2.run(FORWARD); // вращение вперед
    }

    if( LS==1 && RS==0 && SS==0) // Поворот направо
    {
        motor1.run(FORWARD); // вращение вперед
        motor2.run(RELEASE); //стоп машина
    }
    if( LS==1 && RS==0 && SS==1) // Поворот направо
    {
        motor1.run(FORWARD); // вращение вперед
        motor2.run(RELEASE); //стоп машина
    }
}
```

```
if(LS==0 && RS==1 && SS==0) // Поворот налево
{
    motor1.run(RELEASE); //стоп машина
    motor2.run(FORWARD); // вращение вперед
}
if(LS==0 && RS==1 && SS==1) // Поворот налево
{
    motor1.run(RELEASE); //стоп машина
    motor2.run(FORWARD); // вращение вперед
}
if (LS==1 && RS==1 && SS==1) //Стоп
{
    motor1.run(RELEASE); //стоп машина
    motor2.run(RELEASE); //стоп машина
}
if (LS==1 && RS==1 && SS==0) //Развилка
{
    motor1.run(RELEASE); //стоп машина
    motor2.run(FORWARD); // вращение вперед
    motor2.setSpeed(200); // устанавливаем скорость
вращения 200/255
}}
```

}