XXVIмуниципальная научно-практическая конференция обучающихся

«Культура. Интеллект. Наука»

(муниципальный этап Краевого молодежного форума   
«Научно-технический потенциал Сибири»)

**Роботизированная двухколесная платформа**

Секция: «Информационные системы и технологии

в науке, технике и образовании»

проектно-исследовательская работа

***Выполнил:***

Пудовкин Илья Андреевич,

Анцыферов Георгий Андреевич

Гимназия №91, 8кл.

***Руководитель:***

Сиротинина И.В.,

учитель информатики

***Консультант:***

Пархачев С.Н.

инженер «АО ИСС»

Железногорск – 2018

Оглавление

[Введение 2](#_Toc507676088)

[Основное содержание 2](#_Toc507676089)

[1 этап. Просмотр тематической литературы и анализ существующих решений. 2](#_Toc507676090)

[2 этап. Разработка алгоритма движения робота. 3](#_Toc507676091)

[3 этап. Изучение особенности построения роботов на базе плат Arduino и их модулей. 3](#_Toc507676092)

[4 этап. Разработка схемы подключения датчиков и модулей. 5](#_Toc507676093)

[5 этап. Заказ деталей. 5](#_Toc507676094)

[6 этап. Сборка «Роботизированной двухколесной платформы» 6](#_Toc507676095)

[7 этап. Программирование микроконтроллера. 7](#_Toc507676096)

[8 этап. Тестирование сборки и анализ работы «Роботизированной двухколесной платформы». 7](#_Toc507676097)

[Заключение 8](#_Toc507676098)

[Список литературы 9](#_Toc507676099)

[Приложение 1.Характеристики модулей. 10](#_Toc507676100)

[Приложение 2. Блок-схема алгоритма. 11](#_Toc507676101)

[Приложение 3. Листинг программы 14](#_Toc507676102)

# Введение

На сегодняшний день существуют автоматизированные устройства, которые позволяют перемещаться по площади, огибая препятствия, но при этом у большинства из них недостатком является прохождение несколько раз по одной и той же точке. Если рассматривать вариант площади, не имеющей внутренних препятствий, то такой алгоритм перемещения влияет на время прохождения и энергозатратен. Возникает необходимость изменения алгоритма прохождения по оптимальной траектории.

# Основное содержание

Цель: Создать прототип «Роботизированной двухколесной платформы»с и разработать алгоритм ее поведения.

Задачи:

1. Анализ существующих решений.
2. Разработать алгоритм движения роботизированной платформы.
3. Изучить особенности построения роботов на базе плат Arduino и их модулей.
4. Изучить основы схемотехники.
5. Изучить особенности среды программирования роботов на базе платформы Arduino.
6. Разработка схемы подключения датчиков, модулей «Роботизированной двухколесной платформы».
7. Создать прототип «Роботизированной двухколесной платформы»
8. Проанализировать работу «Роботизированной двухколесной платформы»

.

Объект исследования: обеспечение автоматического прохождения площадей с прямыми углами.

Предмет исследования: роботизированная двухколесная платформа

Методы:

1. Анализ собранной информации;
2. Структурно-функциональный метод:
   1. изучение строения, структуры системного объекта;
   2. исследование его элементов и их функциональных характеристик;
   3. анализ изменения этих элементов и их функций;
3. Информационное и предметное моделирование;
4. Наблюдение;
5. Проверочные и воспроизводящие эксперименты.

### 1 этап. Просмотр тематической литературы и анализ существующих решений.

[Анализ](https://market-click2.yandex.ru/redir/GAkkM7lQwz62j9BQ6_qgZv4QadL_VGuxa0iz-lQkYBYW7AKYdCAmszLaLBZnxhKK_1KnUk4zsDCG8OG2i9p3bDT8IiEG3uI2GrJS-Fu0OV9r4lRv_bp1ZZ_TBx_DUtUMUQ6pAZFB1YT-gh3eCeNqTaXlKX0WBT98Lc0CSDU-cQ6xaF1OcLUXLo0DK9dOK6Ll2jzbOfyzPQhhQYBiJw5ONKrcwUaFrAcaQW4j_bff4olv2wjHwHuBDdGz7UXDxM9Zg5bfowXl8NPEZ76dUZ-dpPcdLNlzoz8P1qceFPQcgzLYYQRVGDWy3MWWgeYdCgtVBnYOVreCJtB_0GMrB9MRymSaqYdC4J9uy3-V0RBAQnnykuPGAM2vvP39Wah1x2UIR9nqQ4XiF0fycLBJu9i8xf5QRP28b-avu5ssLPkFVpSvXPQ91ia8DkjRpU1TLZGK1CZTy93osqdRoLoXsPmXDKLSheHNLF97v_gMODxwS3Oa1PhWAGAId7pvES7lksDmCOgE6rWd4-d5bh0f0THQh9YB_iVz7jldL1dOnSdLtaFxakO4o1FECwrfm73yHRKSQ_o5MeN-5RRR4SNnVxw6ks5N_EszAlaGfHh1jGsRsC7hOefn5fp2kisl6McrcfTC0A4W-KtkueShaFswSGU-N_aBMQ9Sap2CsqRo-tZ3RHIFFnnlcAhtgXF7N5VEfTsj8QOEfPtPdJC_dstwakv9aGl0RcnIXJ2giK-rGWxmOd9t0YTGspzhamBRPyUQWC5mYNaa_WvqlmVJwoZbWkSGoKuuvsas8xBSyx_tuVxJEEBAlA6iX_oJrF4eb-vJ98Sa1eRrt0FgnZQyhOhP1JzNeVvJF4ieTKKi?data=QVyKqSPyGQwwaFPWqjjgNlRHjqam9wmmWSPM7SH5g0wyPwqdiQgANrD4h663wALZR7QQVPzqpGnsEhprv3isZQmjBoQBVkqoAgcHfyor0apN-M7R1RLblAC_cw2wSXjE5sbML1mTO2f4Rgx4eujfhBJZn5urswwexBs06Ou1wHKkyS6nLVlO2dbdljRsDkApg5ys7a5PDWjjC6u6H5W4fVLL2L_OTD1OuL2vI29hwFEBUmm0wLSykKL-2Mbe4uIhjHd7FZ7UUAS2SM_g0Bv4eROdX_zdupe12LxbHZrqhuWJVo5-0-6WvgkOOEy6J1ND8QHHDI44JOQilUcplp0HWqDUBpKrMIWoU4qg_ZHH-7onawBraseLHNVhnGCLIiFgUgKcy5p_E2btwbhpBjoAkW8dlC1-BH1LQwEU9MtqJqT1_45xylupUTy1xuSxLDkjAUNhwgt7MNoWfq92cgnPB3lAzvXSmWvxZ897-gujIU78W7yTJnUf5HiOcMYekpa0GrHQYgUMZXsM5lwemlVIg794M-FweLz-mQZwhkL6Ogh-ElBw9I98mDHwwjJJEQM9TPtQ2Gbyqh6ETlOWWUr97ruIobXG2xDyagEEBwWxm3IvO1PO0OAEpQgo1gm5EapdRWxEoXD8Oz9YTs45-pq4qUotMxZHc9iMjfy69LOSSx4-2WetWmPc2rETG3n0ESDSbXFJS0MxT1Qqc94yG5ucLw%2C%2C&b64e=1&sign=c8a535045e56944bf377daffca7e50a5&keyno=1) рынка роботизированных платформ показал, что на сегодняшний день существуют готовые робототехнические платформы Lego, готовые решения, построенные на аппаратной платформе Arduino, отечественная разработка «IskraJS».

После изучения материала по данной теме [1],[2],[3],[4],[5],[6],[7] для основы роботизированной платформы была выбрана плата«Arduino NANO» (рис.1). Критерии выбора:

* 1. Приемлемая стоимость
  2. Количество pins.
  3. Компактность.
  4. Мощность и потребляемое питание.

### http://actrl.cz/blog/wp-content/uploads/nano_ch340_1.jpg2 этап. Разработка алгоритма движения робота.

Рис.1. Arduino NANO

Простыми траекториями прохождения площади с прямыми углами, не позволяющими несколько раз проходить по одной точке, может быть «Змейка» или «Спираль». Если площадь прохождения будет сложной прямоугольной формы (рис.2), то траектория «Спираль» не применима, поэтому была выбрана траектория «Змейка»

Робот начинает движение из нижнего левого угла в прямом направлении.

Ограничением движения справа может быть либо правая стенка, либо два препятствия, расположенные друг напротив друга около параллельных стенок.

Алгоритм движения.

1. Измеряем расстояние до препятствия впереди.
2. Если измеренное расстояние больше заданного (нет препятствия), то перемещаемся вперед.
3. Если измеренное расстояние меньше заданного (верхняя стенка), то робот осуществляет поворот вправо на 90°, снова проверяет наличие препятствия.
   1. если имеется препятствие, то отъезжает назад и поворачивает на 90°, иначе (препятствие отсутствует), сразу делает поворот на 90°, т.о. поворачивается относительно исходного направления движения на 180° .
4. Движется прямо вперед до следующего препятствия (нижняя стенка).
5. Если робот дважды встречает препятствие, вынуждающее его два раза отъехать назад за продвижение *верх — низ — верх* или *низ — верх — низ*, то алгоритм останавливается, иначе алгоритм повторяется (переход к пункту 1).

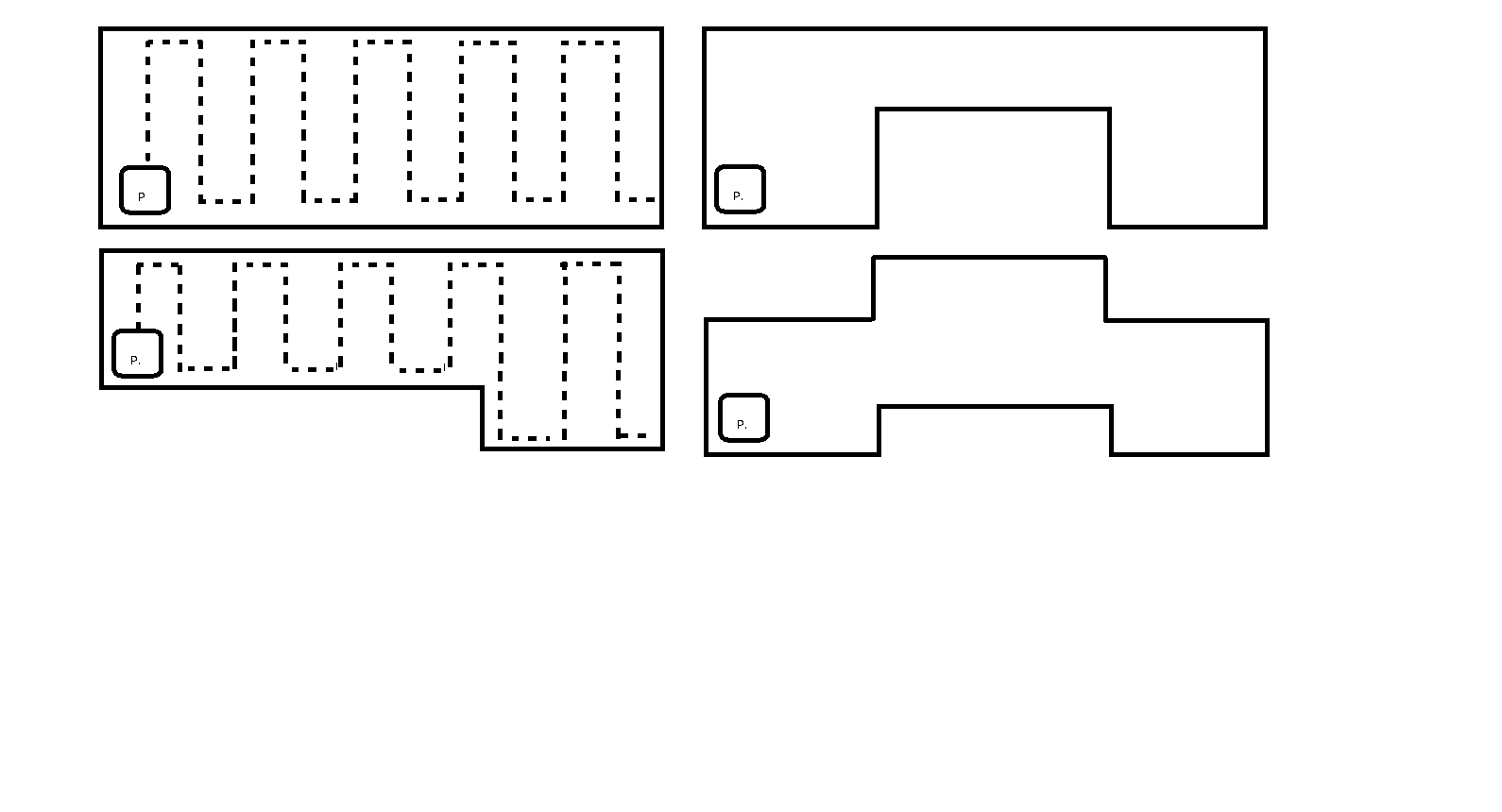


Рис.2. Карты площадей

### 3 этап. Изучение особенности построения роботов на базе плат Arduino и их модулей.

Построение роботов с использованием любой технологии подразумевает изучение принципов работы специальных микросхем, которые называются микроконтроллерами. Они предназначены для управления электронными устройствами и представляют собой однокристальный компьютер, способный выполнять простые задачи. Контроллер, являясь «уменьшенной копией» компьютера, содержит все присущие ему основные модули: центральный процессор, оперативную память, flash-память, внешние устройства. Рис. 3. Структура микроконтроллера

Плата Arduino Nano v3.0 (Приложение 1) выполнена на базе процессора с тактовой частотой 16 МГц, обладает памятью 32кБ, два из которых выделено под загрузчик, позволяющий прошивать Arduino с обычного компьютера через USB. Также имеется 2 кБ SRAM-памяти, которые используются для хранения временных данных (это оперативная память платформы) и 1кБ EEPROM-памяти для долговременного хранения данных (аналог жёсткого диска).

На платформе расположены 14 контактов, которые могут быть использованы для цифрового ввода и вывода. Какую роль исполняет каждый контакт, зависит от программы. Некоторые контакты обладают дополнительными ролями. Например, Serial 0-й и 1-й — используются для приёма и передачи данных по USB; LED 13-й — к этому контакту подключен встроенный в плату светодиод. Также имеется 6 контактов аналогового ввода и входной контакт Reset для сброса. [8]

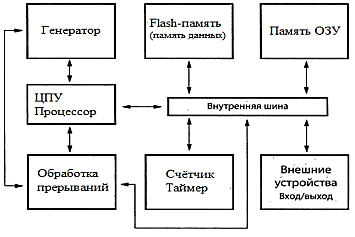
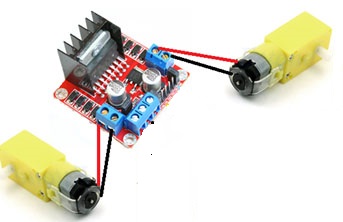


Рис.3Структура микроконтроллера

Выводы микроконтроллера являются слаботочными, поэтому ток мотора, при подключении его напрямую, выведет их из строя. Эту проблему решает драйвер. Он позволяет управлять скоростью и направлением вращения мотора[10]. Для обеспечения перемещения платформы выбраны двигатели постоянного тока, колеса. Для управления двигателями –драйвер L298N Рис.4

Рис.4. Драйвер L298N 1



Для принятия решения о направлении дальнейшего движения разрабатываемой платформы использовались ультразвуковые дальномеры HC-SRO4. Рис.5. [3]. (Приложение1)

Рис.5.Ультразвуковой ДальномерHC-SRO4.

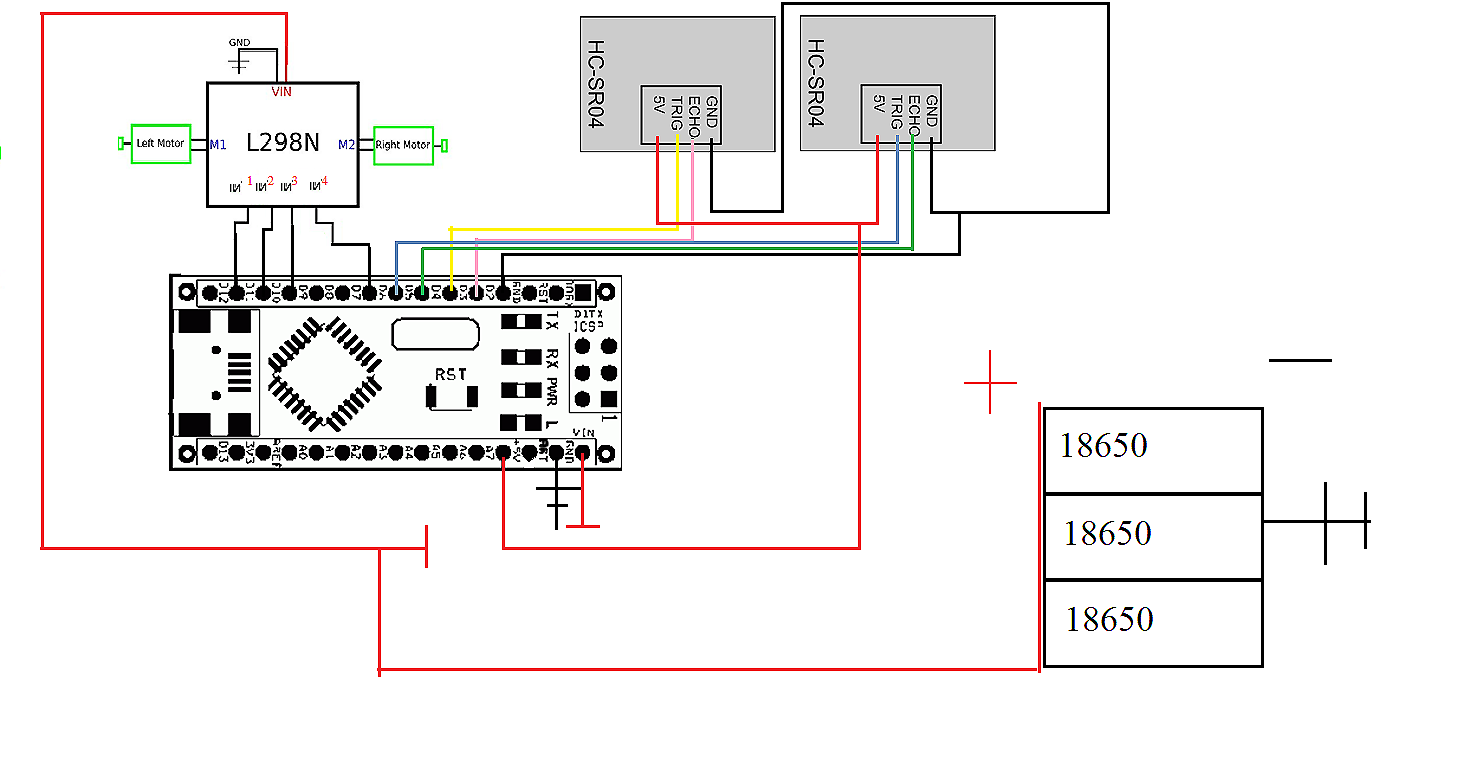
### 4 этап. Разработка схемы подключения датчиков и модулей.

Принцип действия схемы «Роботизированной двухколесной платформы». Рис.6.

Для считывания с первого датчика используется пин2, для передачи сигнала используется пин3. Для второго датчика используются пин4 и пин5 соответственно. Сначала данные считываются с 1-го датчика. Для этого на 1-ый датчик(4) arduino (1) посылает сигнал «0» в течении 2-х мксек, далее передается сигнал «1» в течении 20 мксек и посылается сигал «0», далее считываются сигналы с датчика следующим образом: если датчик в течение 100000 мксек не передал сигнал, то arduino (1) устанавливает время отраженного ультразвукового сигнала в ноль, иначе измеряется время того, сколько сигнал пребывал в состоянии «1», пока не переключился в «0».

Затем перезагружаем датчик на случай, если датчик «зависает». Для этого, переключается пин из состояния «получения сигнала» в состояние «передачи сигнала». После чего на этот пин передается сигнал «0» и держится в течение 10 млсек, далее пин переключается снова в состояние «получение сигнала».

Затем значение времени переводится из микросекунд в сантиметры делением на 58.

2-ой ультразвуковой датчик работает аналогично.

6

1

2

3

4

5

7

Рис.6. Схема «Роботизированной двухколесной платформы».

Цифровые пины arduino (1) D11, D10, D9 и D6 подключены к пинам IN1, IN2, IN3 и IN4 драйвера L298N (2)соответственно.

Направление вращения ротора двигателя управляется сигналами «1» или «0» на каждый пин. Для левого мотора (6), сигнал«1» на IN1 и сигнал «0» на IN2 обеспечит вращение по часовой стрелке,а сигналы «0» и «1» на теже пины − вращение против часовой стрелки. На пин IN 3 сигнал «1», на пин IN4 сигнал «0» −правый мотор (7) вращается по часовой стрелке, сигналы «0» и «1» на теже пины− вращение правого мотора против часовой стрелки.

Остановить вращение можно подачей сигнала «0» на указанные выше пины.

Питание системы осуществляется через литий-ионный аккумулятор (5) емкостью 6000mAh

### 5 этап. Заказ деталей.

Были заказаны следующие детали:

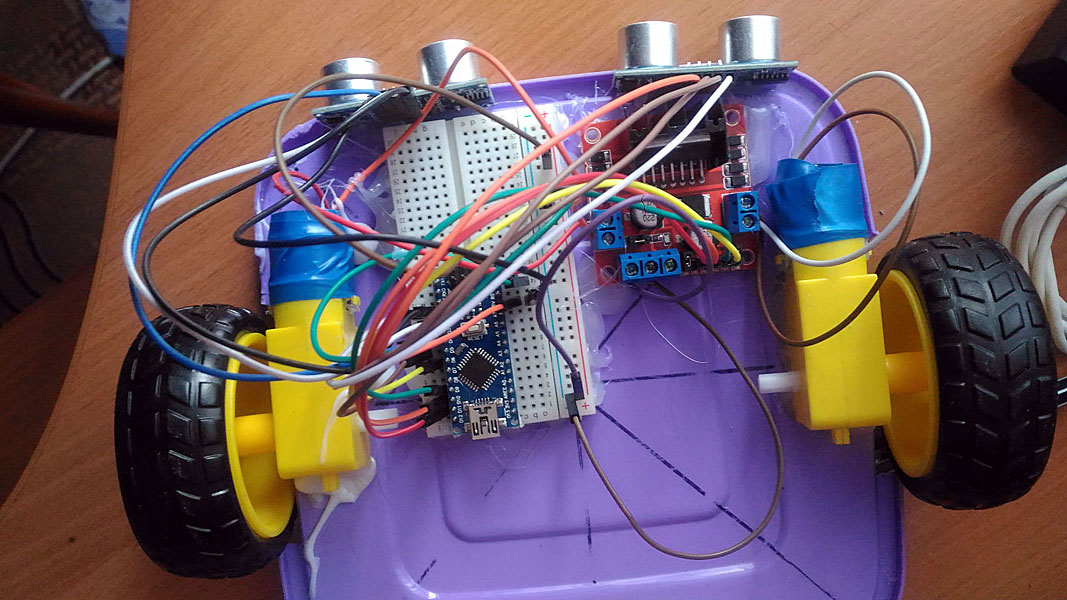
1. Arduino Nano v:3.0
2. 2 двигателя постоянного тока
3. 2ультразвуковых дальномера HC-SRO4
4. 1 драйвер управления двигателями (L298N)
5. Винты, гайки для крепления компонентов платформы между собой.

Состав платформы и характеристики модулей указаны в Приложении 1.

На приобретение деталей было истрачено 1500 руб.

### 6 этап. Сборка «Роботизированной двухколесной платформы»

При тестировании первого прототипа (Рис.7.) была выведена из строя плата ArduinoNano v:3.0 и заменена другой. Из-за неправильного подключения одного ультразвукового датчика, Arduino не получала от него сигнал, исправление в подключении датчика. В результате Arduino начала получать сигналы, но платформа перемещалась в произвольном направлении, не прекращая работу.

Рис.7. 1-ый прототип «Роботизированной двухколесной платформы»

Следующим шагом было − сборка «Роботизированной двухколесной платформы» на основе заготовки, вырезанной из листа фанеры с помощью электролобзика, имеющей крепежные отверстия. Рис.8. Прототип «Роботизированной двухколесной платформы»

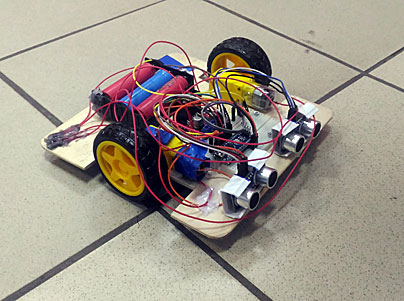


Рис.8. Прототип «Роботизированной двухколесной платформы»

Для управления платформой была написана новая прошивка платы Arduino.

### 7 этап. Программирование микроконтроллера.

Разработка приложений на базе плат Arduino осуществляется в специальной среде программирования Arduino IDE. Среда предназначена для написания и загрузки собственных программ в память микроконтроллера. Среда разработки Arduino состоит из редактора программного кода, области сообщений, окна вывода текста, панели инструментов и панели меню. Базовая структура программы для Arduino состоит из двух обязательных частей: функций setup() и loop(). Перед функцией setup() идет объявление переменных, подключение вспомогательных библиотек. Функция setup() запускается один раз после каждого включения питания или сброса платы. Она используется для инициализации переменных, установки режима работы портов и т. д. Функция loop() в бесконечном цикле последовательно исполняет описанные команды. Для взаимодействия с различными устройствами, для обеспечения ввода и вывода используются специализированные процедуры и функции [10].

Для того чтобы платформа работала по созданному алгоритму, микроконтроллер нужно было запрограммировать.

В программе использованы специализированные процедуры[**11**]:

voidstop\_rob() −остановка движения;

voidmove\_back(int left, int right, int t)−движение назад;

voidmove\_forvard(int left, int right, int t)−движение вперед;

voidturn\_right(int \_speed, int \_time)−поворот направо;

voidturn\_left(int \_speed, int \_time)−поворот налево;

функция long get\_duration(inttrigPin, intechoPin)−прием/передача сигнала от ультразвукового дальномера.

В рабочем цикле программы использованы следующие переменные:

f\_stop\_all− флаг сигнализирующий о необходимости перевести робота в режим пониженного энергопотребления.

stop\_rob− счетчик того, сколько раз робот встретил препятствие при движении "низ-верх-низ" или "верх-низ-верх"

f\_tern\_right−флаг в какую сторону нужно осуществить поворот. Если он имеет значение «true», то вправо, иначе влево.

moving−переменная, отвечающая за то, в какое состояние перевести программы, т.е. какое действие выполнить(остановить, поворот направо, поворот налево, движение вперед ).

Алгоритм работы. Блок-схема . Приложение 2

Листинг программ . Приложение 3

### 8 этап. Тестирование сборки и анализ работы «Роботизированной двухколесной платформы».

При тестировании системы обнаружилось:

1. Скорость движения платформы изначально была очень высокая и arduino не успевала опрашивать датчики. Снизив скорость движения до 20-30℅ от максимально возможной, была достигнута обратная связь с ультразвуковыми датчиками.
2. Платформа двигается не по прямой линии, т.к. моторы вращаются с разной скоростью, подобрать такие дешевые моторы с одинаковой скоростью вращения очень сложно. Для корректировки движения в программе менялись скорости вращения двигателей. Значение входных параметров процедур *move\_forvard* и *move\_back*.
3. Платформа поворачивает не на угол 90°. Для корректировки поворота в программе изменялись значения входных параметров процедур *turn\_left* и *turn\_right*.
4. Отражение ультразвукового сигнала происходит не от всех поверхностей. Например, при тестировании в комнате с линолеумом, заходящим на стены, датчики не срабатывали.
5. При тестировании в школьном холле выяснилась, что при недостаточно ровном покрытии, движение платформы отклоняется от прямого направления.
6. Препятствия могут иметь любой размер. Под препятствиями будем понимать «выступы» и «впадины». Для максимального обхода площади, оптимальный размер препятствий должен быть более 3-х размеров прототипа.
7. Время прохождения 1м2  -30 сек. В режиме работы зарядки аккумулятора хватит на 250 мин. = 4часа.

(4\*60\*60)/30=480м2 .

Вывод: зарядки аккумулятора хватит на обход площади 480м2 .

# Заключение

Разработан и собран прототип «Роботизированной двухколесной платформы», разработан алгоритм ее поведения.

Данная платформа может перемещаться на площади с прямоугольными углами по траектории «змейка». Вдоль стенок могут быть препятствия, не расположенные друг напротив друга. Для максимального обхода площади, оптимальный размер препятствий должен быть более 3-х размеров прототипа.

Платформа применима для многоцелевого использования. На ее основе можно сделать робота-уборщика, робота-пылесоса для спортивного зала, школьных коридоров и других помещений, не имеющих препятствий внутри пространства. К платформе можно подключить дополнительно несколько датчиков и использовать для обследования территории (освещенность, температура т.д.)

# Список литературы

1. Обзор робототехнических платформLego[Электронный ресурс].URL:.[**:https://kurets.ru/zanudnye-obzory/112-obzor-robototexnicheskix-platform-dlya-obrazovaniya**](URL:https://kurets.ru/zanudnye-obzory/112-obzor-robototexnicheskix-platform-dlya-obrazovaniya)(Дата обращения: 3.10.2017г.)
2. «Робоняша» [Электронный ресурс]. <URL:http://amperka.ru/product/robonyasha-iskra-js>  
   (Дата обращения: 3.11.2017г.)
3. Ультразвуковой дальномер HC-SR[Электронный ресурс]. 04. 04.12.17. URL: <http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:hc-sr04-ultrasonic-sensor-distance-module>(Дата обращения: 3.10.2017г.)
4. Блум Д. Изучаем Arduino. — СПб.: БХВ-Петербург, 2015.
5. Петин В. А. Проекты с использованием контроллера Arduino. — СПб.: БХВ-Петербург, 2014. Cоммер У. Программирование микроконтроллерных плат Arduino/Freeduino. — СПб.: БХВ-Петербург, 2012.
6. Амперка. Вики [Электронный ресурс]. URL: http://wiki.amperka.ru/ (Дата обращения: 3.10.2017г.)
7. Информационный портал RoboCraft [Электронный ресурс]. URL:http://robocraft.ru/page/summary (Дата обращения: 12.11.2017г.)
8. Страковский Д. А., Симаков Е. Е. Создание робота-гонщика на платформе Arduino // Юный ученый. — 2016. — №3. — С. 120-124. URL: http://yun.moluch.ru/archive/6/412/ (дата обращения: 26.10.2017)
9. Информационный портал Arduino.ru [Электронный ресурс]. URL:http://arduino.ru/ (Дата обращения: 14.11.2017г.)
10. Драйвер двигателей L298N[Электронный ресурс]. URL: <https://www.youtube.com/watch?v=NqZrzrYu9TM>(Дата обращения: 14.11.2017г.)
11. Программируем робот-пылесос на Arduino8.06.2016[Электронный ресурс**]**URL:<https://www.youtube.com/watch?v=xctFBCc3s40>

# Приложение 1.Характеристики модулей.

**Краткие характеристики ArduinoNano**

| **Микроконтроллер** | **Atmel ATmega168 или ATmega328** |
| --- | --- |
| Рабочее напряжение (логическая уровень) | 5 В |
| Входное напряжение (рекомендуемое) | 7-12 В |
| Входное напряжение (предельное) | 6-20 В |
| Цифровые Входы/Выходы | 14 (6 из которых могут использоваться как выходы ШИМ) |
| Аналоговые входы | 8 |
| Постоянный ток через вход/выход | 40 мА |
| Флеш-память | 16 Кб (ATmega168) или 32 Кб (ATmega328) при этом 2 Кб используются для загрузчика |
| ОЗУ | 1 Кб (ATmega168) или 2 Кб (ATmega328) |
| EEPROM | 512 байт (ATmega168) или 1 Кб (ATmega328) |
| Тактовая частота | 16 МГц |
| Размеры | 1.85 см x 4.2 см |

**Характеристика двигателя**

1. Рабочее напряжение: 3В ~12В
2. Максимальный крутящий момент = 800 см мин.(когда он составляет 3 в)

**Характеристики ультразвукового дальномераHC-SRO4**

Напряжение питания: 5V   
Ток покоя: < 2 мА   
Эффективный угол обзора: < 15 °   
Диапазон измерений: 2 см - 400 см   
Разрешение: 0,3 см

**Состав ««Роботизированной двухколесной платформы»**

 Корпус платформы робота;

 Основание платформы робота;

 Литий-ионный аккумулятор 12В 12Ач;

 2 колеса Ø450мм;

 2 двигателя постоянного тока (P=4Вт);

1 драйвер управления двигателями 5А 12 В;

2ультразвуковых дальномераHC-SRO4;

 плата «Arduino NANO»v: 3.0 с микроконтроллером ATmega328P;

 Винты, гайки для крепления компонентов робота между собой.

# Приложение 2. Блок-схема алгоритма.

нет

f\_turn\_right=true

moving=2

moving =3

да

да

cm1<25

или

cm2<25

движение вперед

нет

moving=0

cm1,cm2

нет

нет

да

f-stop-all=

true

Режим пониженного потребление энергии

да

начало

moving=0;

f-stop-all=false;

cm1=0;

cm2=0;

stop\_rob=0;

f\_tern\_right=true;

cm2=0

Пока вкл.Arduino

нет

да

нет

moving=2

Поворот вправо

cm1,cm2

cm1<25

или

cm2<25

поворот на 90°

вправо

moving=0

f\_stop\_rob=0

moving=4

движение назад

поворот на 90°

вправо

moving=0

f\_stop\_rob+=1

F\_stop\_rob>=1

да

f\_turn\_right=false

поворот на 90°

вправо

нет

да

Поворот влево

cm1,cm2

cm1<25

или

cm2<25

поворот на 90°

влево

moving=0

f\_stop\_rob=0

moving=4

движение назад

поворот на 90°

влево

moving=0

f\_stop\_rob+=1

F\_stop\_rob>=1

да

f\_turn\_right=true

поворот на 90°

влево

нет

нет

да

moving=3

Остановка

f-stop-all=true

нет

да

# Приложение 3. Листинг программы

// ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ

// переменные ультразвуковых дальномеров

#include <avr/sleep.h>

#define TURN\_RIGHT\_1 3

#define TURN\_LEFT\_2 4

#define TURN\_RIGHT\_2 5

#define TURN\_LEFT\_1 6

#define STOP\_ROBOT 30

#define FORWARD 10

#define BACKWARD 20

#define MAX\_LEN 25

int echoPin1 = 2;

int trigPin1 = 3;

int echoPin2 = 4;

int trigPin2 = 5;

//long duration1, duration2, cm1, cm2;

// переменные двигателей

int speed1\_1 = 6;

int speed1\_2 = 9;

int speed2\_1 = 10;

int speed2\_2 = 11;

int moving = FORWARD;

// переменные таймеров анти-залипания

int timer1 = 0;//для двиджения по высоте

int timer2 = 0;//для движения по длине

int timer3 = 0;//

int timer4 = 0;//

void stop\_rob()

{

analogWrite(speed1\_1, 0);//левое вперед

analogWrite(speed1\_2, 0);//левое назад

analogWrite(speed2\_1, 0);//парвое вперед

analogWrite(speed2\_2, 0);//правое наззад

}

void move\_back(int left, int right, int t)

{

analogWrite(speed1\_1, 0);//левое вперед

analogWrite(speed1\_2, left);//левое назад

analogWrite(speed2\_1, 0);//парвое вперед

analogWrite(speed2\_2, right);//правое наззад

delay(t);

}

void move\_forvard(int left, int right, int t)

{

analogWrite(speed1\_1, left);//левое вперед

analogWrite(speed1\_2, 0);//левое назад

analogWrite(speed2\_1, right);//парвое вперед

analogWrite(speed2\_2, 0);//правое наззад

delay(t);

}

void turn\_right(int \_speed, int \_time)

{

analogWrite(speed1\_1, \_speed);//левое вперед

analogWrite(speed1\_2, 0);//левое назад

analogWrite(speed2\_1, 0);//парвое вперед

analogWrite(speed2\_2, 0);//правое наззад

delay(\_time);

}

void turn\_left(int \_speed, int \_time)

{

analogWrite(speed1\_1, 0);//левое вперед

analogWrite(speed1\_2, 0);//левое назад

analogWrite(speed2\_1, \_speed);//парвое вперед

analogWrite(speed2\_2, 0);//правое наззад

delay(\_time);

}

long get\_duration(int trigPin, int echoPin)

{

//ПЕРВЫЙ ДАЛЬНОМЕР

//посылается и принимается звуковой сигнал

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(20);

digitalWrite(trigPin, LOW);

long duration = pulseIn(echoPin,HIGH,100000);

//эти 4 строчки "перезапускают" дальномер (если ваш дальномер так-же,

//как и мой, зависает при дальности препятствия свыше 3-х метров.)

pinMode(echoPin,OUTPUT);

digitalWrite(echoPin,LOW);

delay(10);

pinMode(echoPin,INPUT);

//если дальномер "залип" и показывает "0", то время сигнала выставляется 3 метра.

if (duration == 0)

{

duration=17400;

}

//перевод времени достижения сигнала в сантиметры

long cm = duration / 58;

return cm;

}

// SETUP программы (выполняется 1 раз, при старте)

void setup() {

//назначение портов

pinMode(trigPin1, OUTPUT);

pinMode(echoPin1, INPUT);

pinMode(trigPin2, OUTPUT);

pinMode(echoPin2, INPUT);

pinMode(speed1\_1, OUTPUT);

pinMode(speed1\_2, OUTPUT);

pinMode(speed2\_1, OUTPUT);

pinMode(speed2\_2, OUTPUT);

// pinMode(relay, OUTPUT);

//генерация рандомного числа со свободной ножки

randomSeed(analogRead(0));

//включение реле

//digitalWrite(relay, HIGH);

Serial.begin(9600);

}

bool f\_turn\_right = true;

int f\_stop\_rob=0;

bool stop\_all=false;

// РАБОЧИЙ ЦИКЛ ПРОГРАММЫ

void loop()

{

if(!stop\_all)

{

//ОПРЕДЕЛЕНИЕ ПРЕПЯТСТВИЙ

long cm1 = get\_duration(trigPin1, echoPin1);

Serial.print(cm1);

Serial.print("|");

//небольшая задержка, для предотвращения смешения сигналов со вторым дальномером

delay(50);

long cm2 = get\_duration(trigPin2, echoPin2);

Serial.println(cm2);

delay(50);

if(moving==FORWARD)

{

move\_forvard(70,65,20);

if (cm1 < MAX\_LEN || cm2 < MAX\_LEN)

{

if(f\_turn\_right)

{

moving = TURN\_RIGHT\_1;

}

else

{

moving = TURN\_LEFT\_1;

}

}

}

else if(moving==TURN\_RIGHT\_1)

{

turn\_right(70,700);

f\_turn\_right = false;

moving = TURN\_RIGHT\_2;

}

else if(moving==TURN\_RIGHT\_2)

{

if (cm1 < MAX\_LEN || cm2 < MAX\_LEN)

{

if(f\_stop\_rob>=1)

{

moving = STOP\_ROBOT;

}

else

{

move\_back(75,60,200);

f\_stop\_rob+=1;

}

}

else

{

f\_stop\_rob=0;

}

if(moving!=STOP\_ROBOT)

{

turn\_right(70,700);

moving = FORWARD;

}

}

else if(moving==TURN\_LEFT\_1)

{

turn\_left(70,700);

moving=TURN\_LEFT\_2;

f\_turn\_right = true;

}

else if(moving==TURN\_LEFT\_2)

{

if (cm1 < MAX\_LEN || cm2 < MAX\_LEN)

{

if(f\_stop\_rob>=1)

{

moving = STOP\_ROBOT;

}

else

{

move\_back(75,60,200);

f\_stop\_rob+=1;

}

}

else

{

f\_stop\_rob=0;

}

if(moving!=STOP\_ROBOT)

{

turn\_left(70,700);

moving = FORWARD;

}

}

else if(moving==STOP\_ROBOT)

{

stop\_rob();

stop\_all=true;

set\_sleep\_mode(SLEEP\_MODE\_PWR\_DOWN);

}

}

else

{

sleep\_enable();

sleep\_cpu();

}

}