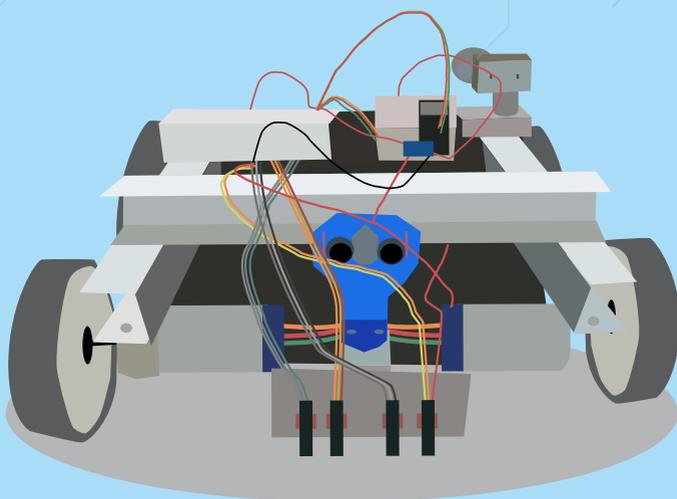


Министерство науки и высшего образования Российской Федерации
Тольяттинский государственный университет

Е.С. Глибин, А.В. Прядилов

МОБИЛЬНАЯ РОБОТОТЕХНИКА

Лабораторный практикум



© ФГБОУ ВО «Тольяттинский
государственный университет», 2023

ISBN 978-5-8259-1323-0

УДК 004.31(075.8)

ББК 32.973.26-04

Рецензенты:

д-р техн. наук, профессор, зав. кафедрой «Сервис технических и технологических систем» Поволжского государственного университета сервиса *Б.М. Горшков*;

д-р техн. наук, профессор кафедры «Промышленная электроника» Тольяттинского государственного университета *В.П. Певчев*.

Глибин, Е.С. Мобильная робототехника: лабораторный практикум / Е.С. Глибин, А.В. Прядилов. – Тольятти : Изд-во ТГУ, 2023. – 1 оптический диск. – ISBN 978-5-8259-1323-0.

Лабораторный практикум содержит методические указания по выполнению лабораторных работ по дисциплине «Мобильная робототехника».

Предназначен для студентов очной, заочной и дистанционной форм обучения (в том числе с использованием дистанционных образовательных технологий) направления обучения бакалавров 11.03.04 «Электроника и наноэлектроника» (профили «Промышленная электроника» и «Электроника и робототехника»).

Текстовое электронное издание.

Рекомендовано к изданию научно-методическим советом Тольяттинского государственного университета.

Минимальные системные требования: IBM PC-совместимый компьютер: Windows XP/Vista/7/8/10; PIII 500 МГц или эквивалент; 128 Мб ОЗУ; SVGA; CD-ROM; Adobe Acrobat Reader.

© Глибин Е.С., Прядилов А.В., 2023

© ФГБОУ ВО «Тольяттинский
государственный университет», 2023

Редактор *Т.М. Воропанова*
Технический редактор *Н.П. Крюкова*
Компьютерная верстка: *Л.В. Сызганцева*
Художественное оформление,
компьютерное проектирование: *Г.В. Карасева*

При оформлении пособия использовано
изображение от Freerik

Дата подписания к использованию 07.03.2023.

Объем издания 2,2 Мб.

Комплектация издания: компакт-диск, первичная упаковка.

Заказ № 1-80-21.

Издательство Тольяттинского государственного университета
445020, г. Тольятти, ул. Белорусская, 14,
тел. 8 (8482) 44-91-47, www.tltsu.ru

Содержание

Введение	5
Лабораторная работа 1. Сборка мобильной платформы	7
Лабораторная работа 2. Движение платформы без датчиков	8
Лабораторная работа 3. Движение с использованием релейного алгоритма	16
Лабораторная работа 4. Движение с использованием пропорционального алгоритма	25
Лабораторная работа 5. Движение с использованием ПИД-управления	27
Лабораторная работа 6. Детектирование и прохождение перекрестков	28
Лабораторная работа 7. Прохождение маршрута с перекрестками	29
Лабораторная работа 8. Система предупреждения столкновений	30
Приложение А	31
Приложение Б	34
Приложение В	37

Введение

В курсе «Мобильная робототехника» предполагается создание, программирование и отладка мобильных платформ с микроконтроллерным управлением. Мобильные платформы собираются на основе конструктора TETRIX PRIZM. Сначала собирается его каркас из металлического конструктора, а затем программируется аппаратная часть в виде микроконтроллера на базе Arduino (PRIZM ROBOTICS CONTROLLER). Основной задачей платформы является движение по черной линии, нанесенной на полу. Детектирование линии осуществляется посредством оптических датчиков линии (YL-73). Движение тележки реализуется за счет двух электродвигателей, управляемых через драйвер сигналом с микроконтроллера.

В ходе отладки и программирования предполагается решение следующих проблем:

- неравномерность передвижения по черной линии, рывки и дерганья, приводящие к съезду с неё, что связано с недостаточной калибровкой датчиков линии по восприятию черного и белого цветов;
- подбор составляющих ПИД (пропорционально-интегрально-дифференциального регулирования) для обеспечения должной плавности хода на прямолинейных участках и поворотах;
- задание маршрута прохождения по перекресткам по необходимой траектории движения по черной линии, плавно огибая все участки заданной траектории и проходя по углу в 90 градусов.

Цель дисциплины: формирование профессиональных компетенций, необходимых для разработки и эксплуатации мобильных платформ.

Задачи дисциплины: познакомиться с основными видами шасси мобильных роботов и их приводами, а также научиться разрабатывать программное обеспечение для управления ими.

Цель лабораторного практикума: научиться программировать и отлаживать движение мобильных платформ.

Задачи лабораторного практикума:

1. Собрать мобильную платформу.
2. Научиться передвигать мобильную платформу без сигнала с датчиков.

3. Научиться управлять движением платформы по релейному алгоритму.
4. Научиться управлять движением платформы по пропорциональному алгоритму.
5. Научиться управлять движением платформы по ПИД-алгоритму.
6. Научиться детектировать перекрестки.
7. Научиться проходить перекрестки в заданном направлении.
8. Научиться использовать систему предупреждения столкновений.

Требования по оформлению отчета приведены в файле «Правила оформления ЛР.doc», расположенном на Яндекс Диске https://yadi.sk/d/Fs-9ts_VInrE3Q в папке «Ко всем предметам».

Бланк титульного листа отчета приведен в прил. В.

Требования по технике безопасности при выполнении лабораторной работы

1. Перед началом выполнения лабораторной работы необходимо получить допуск у преподавателя.
2. Запрещается включать питание мобильной тележки до проверки схемы преподавателем.
3. Запрещается самостоятельно устанавливать на персональные компьютеры (или удалять) программное обеспечение.
4. Необходимо бережно относиться к используемому оборудованию, не дергать за провода, не допускать ударов или падения мобильной платформы с высоты.
5. Перед началом выполнения цикла лабораторных работ необходимо пройти инструктаж по технике безопасности и расписаться в журнале.
6. В случае возникновения нестандартной ситуации (появление дыма, хлопков и т. д.) необходимо обесточить оборудование и сообщить преподавателю.

После выполнения лабораторного практикума студенты должны:

- *знать* основные алгоритмы управления движением мобильной платформы;
- *уметь* программировать мобильную платформу на движение в соответствии с заданием;
- *иметь навык* практической сборки и программирования мобильных платформ.

Лабораторная работа 1

Сборка мобильной платформы

Цель работы: собрать и протестировать мобильную платформу.

План работы

1. Получить у преподавателя сборочные комплектующие.
2. Произвести сборку и тестирование мобильной платформы.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Arduino IDE.

2. Для сборки выдается один из трех вариантов тележки:

- Платформа из набора конструктора TETRIX PRIZM с контроллером PRIZM (оболочка программирования, аналогичная Arduino IDE). Внешний вид, схема и перечень элементов см. в прил. А.

- Платформа из набора конструктора TETRIX PRIZM с контроллером Arduino IDE и драйвером двигателей. Внешний вид, схема и перечень элементов см. в прил. Б.

- Платформа, собираемая из набора «Амперка». В базовой комплектации платформы только 2 датчика, что недостаточно для прохождения перекрестков. Поэтому данная платформа будет использоваться только для лабораторных работ 1–5 и 8.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Схема электрическая принципиальная собираемой платформы.
4. Фотография собранной платформы.
5. Выводы, включающие сравнение полученных результатов с исходным заданием на лабораторную работу.

Лабораторная работа 2

Движение платформы без датчиков

Цель работы: освоить управление движением платформы.

План работы

1. Разобраться, как заставить левое/правое колесо платформы вращаться вперед/назад с заданными скоростями.
2. Составить БСА (блок-схему алгоритма) программы.
3. Написать и протестировать управляющую программу.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Ардуино.
2. Мобильная платформа, шнур USB.

Задание

После включения тележка должна выполнить действия (длительность каждого действия 1 секунда). Информация о текущих скоростях выводится в монитор порта:

1. Левое колесо вращается вперед со скоростью 50 %.
2. Левое колесо вращается вперед со скоростью 100 %.
3. Левое колесо останавливается.
4. Левое колесо вращается назад со скоростью 50 %.
5. Левое колесо останавливается.
6. Правое колесо вращается вперед со скоростью 50 %.
7. Правое колесо вращается вперед со скоростью 100 %.
8. Правое колесо останавливается.
9. Правое колесо вращается назад со скоростью 50 %.
10. Правое колесо останавливается.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Управляющая программа с комментариями.

4. Описание работы платформы.
5. Выводы, включающие сравнение полученных результатов с теоретическими предпосылками и исходным заданием на лабораторную работу.

Методические рекомендации

Колеса вращаются моторами, представляющими собой двигатели постоянного тока. Направление вращения определяется полярностью поданного на них постоянного напряжения. Для регулировки скорости вращения используется режим ШИМ (широтно-импульсного модулирования), при котором напряжение подается не непрерывно, а чередуются отрезки времени с поданным напряжением питания и нулевым напряжением (рис. 2.1). В Ардуино ШИМ может быть сформирован на выводах, помеченных знаком «~» при помощи команды `analogWrite`.

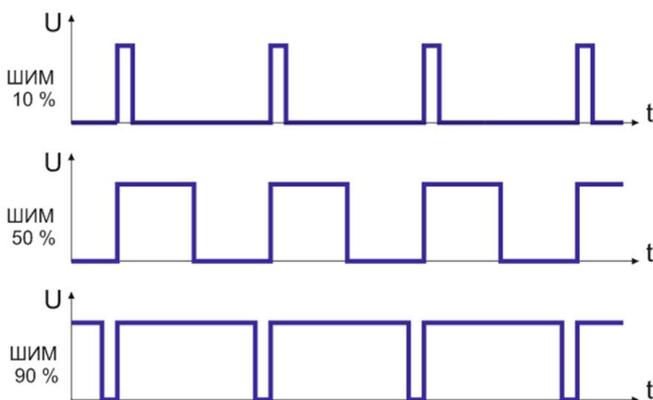


Рис. 2.1. ШИМ

Питание на двигатели подают через драйверы, являющиеся усилителями по мощности, так как мощность выходов Ардуино недостаточна для непосредственного подключения двигателей. Примером такого драйвера является драйвер, встроенный в контроллер **PRIZM** или драйвер на основе силовой микросхемы L298N, которую рассмотрим подробнее (рис. 2.2).

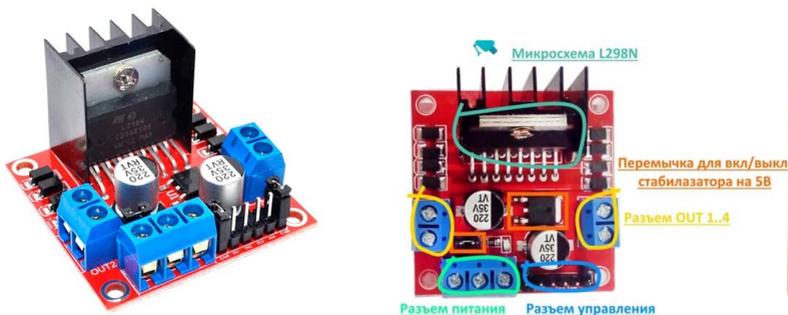


Рис. 2.2. Драйвер L298N
 (<https://robotchip.ru/obzor-drayvera-motora-na-l298n/>)

Технические параметры L298N:

Напряжение питания логики модуля: 5 В.

Потребляемый ток встроенной логики: 36 мА.

Напряжение питания драйвера: 5–35 В.

Рабочий ток драйвера: 2 А (пиковый ток 3 А).

Габариты: 43,5×43,2×29,4 мм.

Основной чип модуля — микросхема L298N, состоящая из двух Н-мостов (H-Bridge), один для выхода А, второй для выхода В. Н-мост широко используется в электронике и служит для изменения вращения двигателем, схема Н-моста содержит четыре транзистора (ключа) с двигателем в центре, образуя Н-подобную компоновку. Принцип работы прост, при одновременном закрытии двух отдельных транзисторов изменяется полярность напряжения, приложенного к двигателю. Это позволяет изменять направление вращения двигателя. На рис. 2.3 показана работа Н-мостовой схемы.

Для управления скоростью двигателя постоянного тока используется метод PWM (широтно-импульсной модуляции).

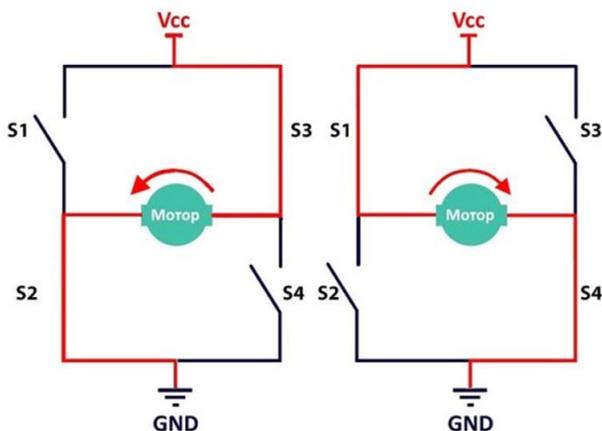


Рис. 2.3. Работа H-моста

(<https://robo-wiki.ru/electronics/component-parts/motors-and-drivers/l298n/>)

Модуль L298N (рис. 2.4) содержит разъем для подключения питания, ряд переключек для настройки модуля, два выхода А и В и разъем управления, которые регулируют скорость и направление вращения; назначение каждого представлено ниже:

- ▶ **Вывод Vss** – питание двигателей, от 5 до 35 В.
- ▶ **Вывод GND** – общий вывод заземления.
- ▶ **Вывод Vs** – питание для логической схемы.
- ▶ **Переключка ENA** – используется для управления скоростью двигателя А.
- ▶ **Выходы IN1 и IN2** – используются для управления направлением вращения двигателя А.
- ▶ **Выходы IN3 и IN4** – используются для управления направлением вращения двигателя В.
- ▶ **Переключка ENB** – используется для управления скоростью двигателя В.
- ▶ **Выходы OUT1 и OUT2** – разъем для двигателя А.
- ▶ **Выходы OUT3 и OUT4** – разъем для двигателя В.

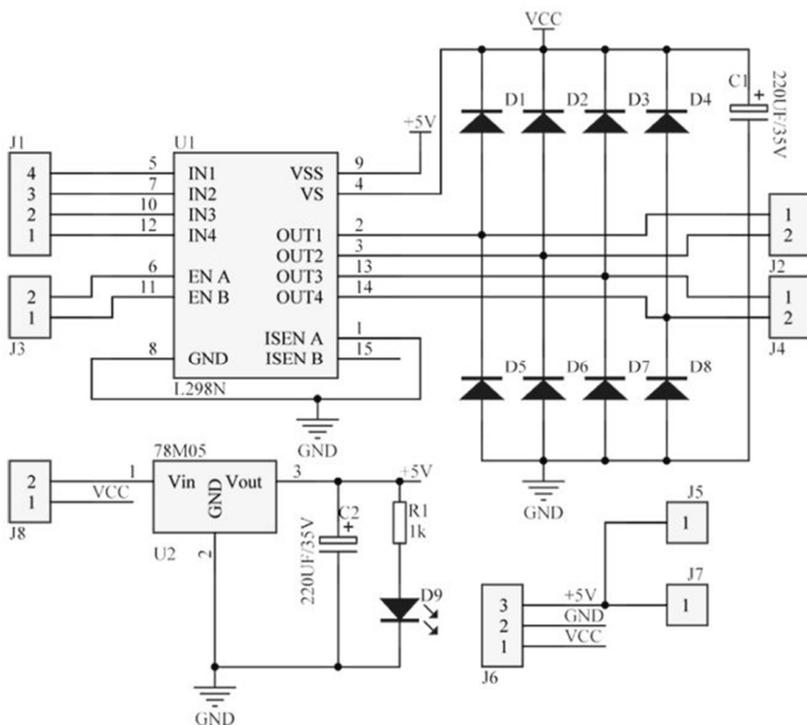


Рис. 2.4. Схема модуля L298N
<https://www.freepng.ru/png-z8zacc/>

Питание модуля

Питание модуля L298N осуществляется через трехконтактный разъем:

- ▶ **Vs** – источник питания двигателей, 3–35 В.
- ▶ **GND** – земля.
- ▶ **Vss** – источник питания модуля, 4,5–5,5 В.

Фактически у модуля L298N есть два контакта питания, а именно Vss и Vs. От Vs питаются двигатели с допустимым напряжением от 5 до 35 В, а от Vss питается логическая схема модуля – от 4,5 до 5,5 В. На плате установлен встроенный стабилизатор напряжения на 5 В (78M05), его можно включить или отключить с помощью переключки. Когда переключка установлена, стабилизатор включен и питает логику модуля (Vss) от источника питания двигателя (Vs). При включенном стабилизаторе вход Vss работает как выход и обеспечивает

5 В с током 0,5 А. Когда переключатель убран, стабилизатор отключен, и необходимо отдельно подключить питание 5 В на вход Vss.

Падение напряжения L298N

Падение напряжения драйвера L298N составляет около 2 В, это связано с внутренним падением напряжения в транзисторах в цепи H-моста. Таким образом, если мы подключим 12 В к источнику питания двигателя, то двигатели получают напряжение около 10 В. Это означает, что двигатель на 12 В не будет работать с максимальной скоростью. Для получения максимальной скорости напряжение, поданное на драйвер, должно быть выше на 2 В, чем напряжение двигателя. Учитывая падение напряжения на 2 В, если вы используете двигатели 5 В, необходимо обеспечить питание 7 В. Если у вас 12-ваттные двигатели, то напряжение питания вашего двигателя должно составлять 14 В.

Управление скоростью

Разъемы управления скоростью ENA и ENB используются для включения и выключения управления скоростью двигателей. Когда переключатель установлен, двигатель вращается с максимальной скоростью. Если необходимо управлять скоростью двигателей, необходимо убрать переключатель и подключить выводы к контактам с поддержкой PWM на Arduino.

Подключение

Первым делом необходимо подключить источник питания 12 В к двигателям; в примере используются распространенные двигатели постоянного тока, рассчитанные на 3–12 В (применяемые в робототехнике) (рис. 2.5). Учитывая внутреннее падение напряжения на микросхеме L298N, двигатели получают 10 В и будут вращаться не в полную силу. Далее, нужно подключить 5 В на логическую схему L298N, для этого надо воспользоваться встроенным стабилизатором напряжения, который работает от источника питания двигателя, поэтому переключатель EN должна быть установлена. Теперь осталось подключить управляющие провода ENA, IN1, IN2, IN3, IN4 и ENB к шести цифровым выводам Arduino 9, 8, 7, 5, 4 и 3. Обратите внимание, что выводы Arduino 9 и 3 поддерживают ШИМ. Теперь подключаем двигатели — один к клемме А (OUT1 & OUT2), а другой к клемме В (OUT3 & OUT4). Схема подключения приведена ниже.

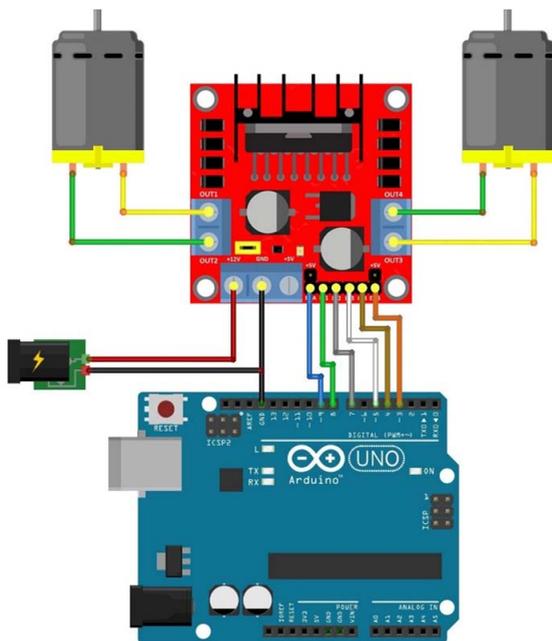


Рис. 2.5. Подключение модуля L298N
(<https://robotchip.ru/obzor-drayvera-motora-na-l298n/>)

Пример скетча для управления двигателем

```
// Двигатель А
int enA = 9;
int in1 = 8;
int in2 = 7;
// Двигатель В
int enB = 3;
int in3 = 5;
int in4 = 4;

void setup()
{
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
```

```
pinMode(in3, OUTPUT);  
pinMode(in4, OUTPUT);  
digitalWrite(in1, LOW);  
digitalWrite(in2, LOW);  
digitalWrite(in3, LOW);  
digitalWrite(in4, LOW);  
}
```

```
void loop()  
{  
  // Установка двигателя А и В на максимальную скорость (0–255)  
  analogWrite(enA, 255);  
  analogWrite(enB, 255);  
  // Вращение двигателем А и В вперед  
  digitalWrite(in1, HIGH);  
  digitalWrite(in2, LOW);  
  digitalWrite(in3, HIGH);  
  digitalWrite(in4, LOW);  
  delay(2000);  
  // Вращение двигателем А и В назад  
  digitalWrite(in1, LOW);  
  digitalWrite(in2, HIGH);  
  digitalWrite(in3, LOW);  
  digitalWrite(in4, HIGH);  
  delay(2000);  
  // Отключение мотора А и В  
  digitalWrite(in1, LOW);  
  digitalWrite(in2, LOW);  
  digitalWrite(in3, LOW);  
  digitalWrite(in4, LOW);  
}
```

Лабораторная работа 3

Движение с использованием релейного алгоритма

Цель работы: запрограммировать автоматическое движение платформы по линии с использованием релейного алгоритма.

План работы

1. Изучить теоретический материал.
2. Написать управляющую программу.
3. Загрузить управляющую программу в платформу и протестировать ее работу. При необходимости подкорректировать программу.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Arduino IDE.
2. Мобильная платформа, шнур USB.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Управляющая программа с комментариями.
4. Описание работы платформы.
5. Выводы, включающие сравнение полученных результатов с теоретическими предпосылками и исходным заданием на лабораторную работу.

Задачи

1. Разобраться с получением сигналов с датчиков линии для распознавания черной линии на белом фоне. Информация с датчиков получается в логическом виде (1 или 0). Вывести информацию с датчиков в монитор порта.

2. Написать программу движения платформы по черной линии с использованием релейного алгоритма. Добиться стабильного движения по линии. Черная линия имеет плавные изгибы, перекрестки отсутствуют.

Для получения информации о положении тележки служит нанесенная на светлом полу черная полоса и датчики линии. Датчик ли-

нии состоит из излучающего инфракрасного светодиода, свящего на пол, и принимающего фотодиода (рис. 3.1). Величина выходного напряжения Out зависит от степени светлости пола. Чем светлее пол под датчиком, тем напряжение меньше.

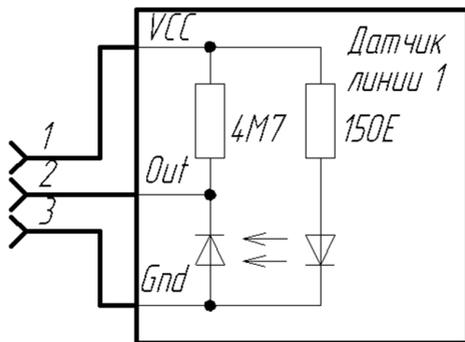


Рис. 3.1. Схема оптического датчика (автор А.В. Прядилов)

Для построения релейного алгоритма управления сигнал о цвете пола можно получать в логическом виде (1 или 0), сравнивая выходной сигнал датчика линии с уставкой, равной среднему значению между показаниями на светлом полу и черной линии. Иногда это делают, подавая сигнал с датчика линии сразу на компаратор и только с него заводя в микроконтроллер. Подстройка уставки осуществляется вращением потенциометра. В этом случае если датчик находится над светлым полом, то на выходе Out получится сигнал низкого уровня, а если над черной полосой, то высокого. Можно обойтись и без отдельного микроконтроллера, подав сигнал с датчика на аналоговый вход (АЦП) Ардуино и сравнивая с уставкой (задаваемой программно) в численном виде. Величина уставки подбирается экспериментально.

Следует учитывать, что внешнее освещение (особенно солнечное) может повлиять на величины выходных сигналов с датчика линии. Поэтому необходимо защитить датчик от внешнего освещения. Второй особенностью является свойство некоторых черных липких лент бликовать (т. е. вести себя подобно зеркалу, значительно отражая свет в одном направлении). Поэтому датчик линии дол-

жен смотреть не вертикально вниз, а под углом, чтобы бликующий отраженный свет не попадал в фотоприемник.

Краткие теоретические сведения

Для движения мобильного робота по линии может использоваться схема, показанная на рис. 3.2.

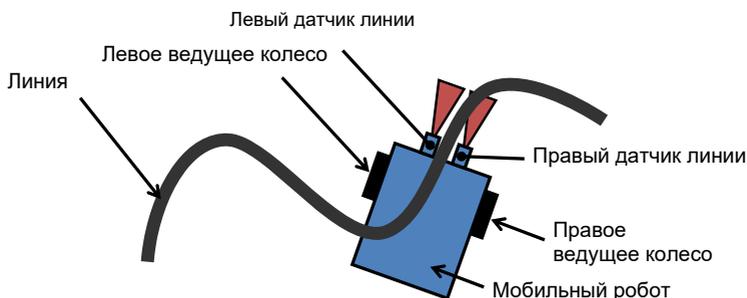


Рис. 3.2. Схема движения мобильного робота по линии
(автор Е.С. Глибин)

Робот включает два колеса и два датчика по одному слева и справа от линии, вдоль которой должен двигаться робот. Обычно датчики линии могут работать в цифровом и аналоговом режиме, как показано на рис. 3.3.

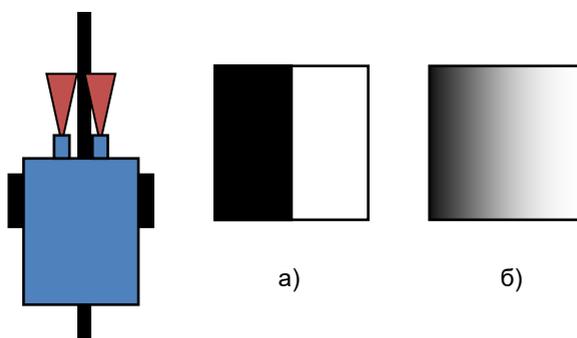


Рис. 3.3. Распознавание линии с помощью оптического датчика:
а — как видит линию человек и распознает цифровой датчик;
б — как «видит» линию аналоговый датчик
(автор Е.С. Глибин)

Цифровой датчик используется в релейном способе управления движением, аналоговый необходим для ПИД-регулирования движением робота.

Рассмотрим более простой в реализации релейный способ управления. Каждый такт работы программы выполняются следующие действия:

1. Считываются показания датчиков. Предположим, что датчик выдает логическую единицу, если фиксирует линию, другими словами, находится над участком чёрного цвета. Если датчик находится над участком белого цвета, то выдает на выходе логической 0.

2. В зависимости от показаний датчиков робот поворачивается влево, вправо, двигается вперед или останавливается, как показано в таблице.

Реакции системы управления на показания датчиков

Показания левого датчика	Показания правого датчика	Реакция
0	0	Движение вперед
0	1	Поворот вправо
1	0	Поворот влево
1	1	Остановка

3. Движение вперед осуществляется путем подачи напряжения с помощью драйвера на оба двигателя. Поворот вправо осуществляется путем включения левого двигателя в направлении движения вперед, а правого двигателя — в направлении движения назад. Кажется, что достаточно включить один двигатель и убрать напряжение с другого, чтобы выполнить поворот, однако двигатель без питания будет продолжать свободное вращение, и робот, скорее всего, съедет с трассы на небольшом повороте. Поэтому чтобы остановить двигатель, необходимо на него подать обратное напряжение.

При движении на небольших скоростях или при прохождении плавных поворотов релейный алгоритм работает хорошо, однако робот постоянно «дергается» около линии. Дело в том, что у нас только три варианта движения: два поворота и прямо, причем на постоянной скорости. Например, робот двигается вперед, а затем левый датчик

резко фиксирует линию, система управления подает команду поворота налево. Возможно несколько неудачных вариантов:

1. Робот уже проехал линию за счет инерции, несмотря на команду поворота влево.

2. Поворот влево осуществляется на максимальной скорости, что почти сразу потребует поворота вправо, если участок линии прямой. В результате движение на прямом участке в основном будет состоять из поворотов влево-вправо.

Для решения этих проблем можно использовать следующий подход. Будем использовать аналоговый датчик (рис. 3.3, б). Если левый датчик фиксирует белый цвет, а правый – серый цвет, значит, робот приближается к линии справа и необходимо выполнить поворот направо, как и в релейном способе. Но теперь мы будем регулировать скорость поворота; если фиксируется серый цвет правым датчиком, значит, мы только начинаем приближаться к линии и поворот можно сделать медленный, а если черный, то пора включать максимальную скорость поворота. То есть вместо управления «включить/остановить двигатель» можно использовать плавное регулирование скоростей в зависимости от того, как близко аналоговый датчик находится к линии. Как это сделать, рассматривает теория автоматического управления.

На рис. 3.4 показана схема системы автоматического управления.

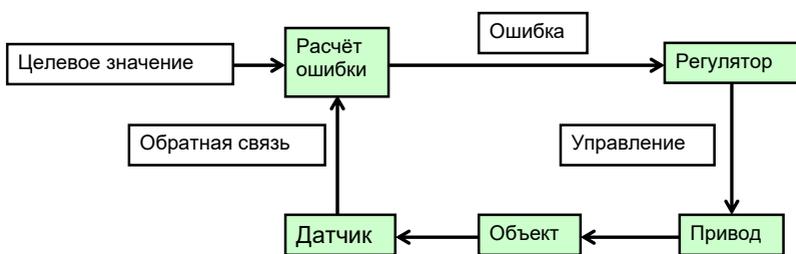


Рис. 3.4. Схема системы автоматического управления (автор Е.С. Глибин)

Система состоит из следующих элементов:

1. Объект – наш мобильный робот, а точнее, его положение относительно линии. Именно положением мы управляем, когда подразумеваем управление роботом, движущимся по линии.

2. Регулятор — то, что принимает решение об управлении объектом, в нашем случае это программа, работающая в управляющем микроконтроллере.

3. Привод — часть системы, осуществляющая управляющее воздействие на объект управления. В нашем случае это моторы.

4. Датчики.

5. Блок расчета ошибки вычисляет, насколько реальный регулируемый параметр (он получается с помощью датчиков) отличается от заданного параметра (задается пользователем, в нашем случае целевое значение — центр линии). К примеру, в системе управления нагревателем на этот блок поступает измеренная датчиками реальная температура и заданная пользователем желаемая.

Существенным отличием от предыдущего способа является необходимость вычисления входного параметра для блока расчета ошибки. То есть вместо принятия решения в зависимости от показания левого и правого датчика решение будет зависеть от одного параметра — отклонения робота от линии, вычисленного по показаниям датчиков. Отклонение может измеряться, например, в миллиметрах или просто в условных единицах.

Например, датчик выдает сигнал в 200 условных единиц, если находится над белым участком, и 1000 — если над чёрным участком. Тогда в простейшем случае использования двух датчиков можно вычислить среднее, взяв показания левого датчика со знаком минус. Если робот находится строго над линией, оба датчика выдают одинаковые значения и среднее значение равно 0. Если робот находится левым датчиком над линией, а правым над белым участком, то положение в условных единицах будет равно $(-1000 + 200) / 2 = -400$. Если наоборот, то $+400$. Более точно вычислить отклонение от линии можно, если использовать линейку из 4–8 датчиков вместо двух. Очевидно, что функция перевода показаний 8 датчиков в один численный параметр будет в этом случае немного сложнее.

Следующим этапом будет вычисление ошибки $e(t)$, в нашем случае мы хотим, чтобы отклонение было равно нулю, и ошибка будет равна преобразованным показаниям датчиков. В случае системы нагрева ошибка $e(t)$ равна разнице измеренной температуры и заданной пользователем.

Теперь регулятор должен выработать такие управляющие воздействия, чтобы ошибка начала уменьшаться, стремиться к нулю. Популярными являются ПИД-регуляторы (пропорциональный, интегральный, дифференциальный).

Управляющее воздействие в ПИД-регуляторе вычисляется по формуле

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}. \quad (3.1)$$

Подобно преобразованию показаний датчиков в одно число управляющий параметр $u(t)$ необходимо преобразовать в конечные воздействия на два мотора. Например, принимаем, что значение параметра $u = -1$, и это будет означать резкий поворот влево, $u = +1$ – резкий поворот вправо, $u = 0$ – движение вперед. Промежуточные значения позволяют регулировать скорость поворота, изменяя соотношения напряжений на моторах. Условный код управления может выглядеть следующим образом:

```
int error = calculateError(); // вычисляем ошибку отклонения робота
float u = PID(error); // вычисляем управляющее воздействие
moveMotorL(u); // управление левым мотором
moveMotorR(u); // управление правым мотором
```

Управляющее воздействие $u(t)$ является суммой трех слагаемых – P , I , D , степень влияния которых на конечное значение определяется коэффициентами K_p , K_i , K_d . Значения коэффициентов подбираются в конкретной системе автоматического управления, выбор их нетривиален, а влияние они оказывают существенное. Если один или два коэффициента равны 0, то получается частный случай ПИД-регулятора, например, П-регулятор, И-регулятор, ПИ-регулятор, ПД-регулятор.

Условный код П-регулятора выглядит следующим образом:

```
const float Kp = 0.2f; // пропорциональный коэффициент П-регулятора
int error = calculateError(); // вычисляем ошибку отклонения робота
float u = Kp * error;
moveMotorL(u); // управление левым мотором
moveMotorR(u); // управление правым мотором
```

```

Условный код ПИД-регулятора выглядит следующим образом:
const float Kp = 0.2f; // пропорциональный коэффициент ПИД-регулятора
const float Ki = 0.01f; // интегральный коэффициент
const float Kd = 0.02f; // дифференциальный коэффициент
float SUM = 0; // глобальная переменная для интегральной составляющей
float prevError = 0; // предыдущее значение ошибки
...
void controlRobot()
{
int error = calculateError(); // вычисляем ошибку отклонения робота
float u = Kp * error;

/* вычисление интегральной составляющей */
SUM += Ki * error;

if (SUM < -1)
    SUM = -1;
if (SUM > 1)
    SUM = 1;

u += SUM;

/* вычисление дифференциальной составляющей */
u += Kd * (error - prevError);
prevError = error;

moveMotorL(u); // управление левым мотором
moveMotorR(u); // управление правым мотором
}

```

В указанном коде функция controlRobot() должна вызываться с максимально постоянной частотой, иначе значения интегральной и дифференциальной составляющих будут некорректны – обратите внимание на наличие времени в формуле (3.1) и отсутствие его в коде, по сути, dt принимается константной для упрощения кода.

Обычно порядок подбора коэффициентов робота следующий:

1. На низкой скорости подбираем только K_p таким образом, чтобы на самых крутых поворотах робот проходил так, чтобы линия была максимально близко к датчикам. K_i , K_d равны нулю на этом этапе.

2. Повышаем скорость, подбираем значение K_d — движение робота должно быть плавное, без инерции. Если инерция присутствует, необходимо понизить пропорциональный коэффициент K_p .

3. В конце подбирают коэффициент K_i , уменьшая отклонение робота от линии, параллельно немного уменьшаются уже настроенные коэффициенты.

Лабораторная работа 4

Движение с использованием пропорционального алгоритма

Цель работы: запрограммировать автоматическое движение платформы по линии с использованием пропорционального алгоритма.

План работы

1. Изучить теоретический материал.
2. Написать управляющую программу.
3. Загрузить управляющую программу в платформу и протестировать ее работу. При необходимости подкорректировать программу.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Arduino IDE.
2. Мобильная платформа, шнур USB.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Управляющая программа с комментариями.
4. Описание работы платформы.
5. Выводы, включающие сравнение полученных результатов с теоретическими предпосылками и исходным заданием на лабораторную работу.

Задачи

1. Разобраться с получением аналогового сигнала с датчиков линии для распознавания позиции платформы относительно черной линии. Разработать процедуру калибровки датчиков и нормализации получаемых показаний. Вывести информацию с датчиков в монитор порта.

2. Написать программу движения платформы по черной линии с использованием пропорционального алгоритма. Добиться стабильного движения по линии. Черная линия имеет плавные изгибы, перекрестки отсутствуют.

Для построения пропорционального алгоритма управления необходимо использовать датчик линии в аналоговом режиме. Напряжение на выходе датчика постепенно изменяется в процессе внесения датчика на черную линию. Скорость поворота (разница в скоростях или в заполнении ШИМ) должна быть пропорциональна разнице сигналов с левого и правого датчиков линии.

Лабораторная работа 5

Движение с использованием ПИД-управления

Цель работы: запрограммировать автоматическое движение платформы по линии с использованием релейного алгоритма.

План работы

1. Изучить теоретический материал.
2. Написать управляющую программу.
3. Загрузить управляющую программу в платформу и протестировать ее работу. При необходимости подкорректировать программу.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Arduino IDE.
2. Мобильная платформа, шнур USB.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Управляющая программа с комментариями.
4. Описание работы платформы.
5. Выводы, включающие сравнение полученных результатов с исходным заданием на лабораторную работу.

Задачи

1. Разобраться с ПИД-регулированием.
2. Написать программу движения платформы по черной линии с использованием ПИД-регулятора. Добиться стабильного движения по линии. Выбрать коэффициенты ПИД-регулятора. Черная линия имеет плавные изгибы, перекрестки отсутствуют.

Лабораторная работа 6

Детектирование и прохождение перекрестков

Цель работы: запрограммировать автоматическое движение платформы по перекресткам с возможностью движения прямо или поворота.

План работы

6. Изучить теоретический материал.
1. Написать управляющую программу.
2. Загрузить управляющую программу в платформу и протестировать ее работу. При необходимости подкорректировать программу.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Arduino IDE.
2. Мобильная платформа, шнур USB.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Управляющая программа с комментариями.
4. Описание работы платформы.
5. Выводы, включающие сравнение полученных результатов с теоретическими предпосылками и исходным заданием на лабораторную работу.

Задача

Написать программу движения платформы по черной линии с использованием любого из регуляторов. Черная линия имеет плавные изгибы и крестообразные перекрестки (пересечения линий под углом 90 градусов). При движении платформа должна распознавать момент въезда на перекресток. Написать процедуры:

- а) прохождения перекрестков прямо;
- б) поворотов направо/налево;
- в) остановки;
- г) разворота на перекрестке на 180 градусов.

Лабораторная работа 7

Прохождение маршрута с перекрестками

Цель работы: запрограммировать автоматическое движение платформы по линии с использованием релейного алгоритма.

План работы

1. Изучить теоретический материал.
2. Написать управляющую программу.
3. Загрузить управляющую программу в платформу и протестировать ее работу. При необходимости подкорректировать программу.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Arduino IDE.
2. Мобильная платформа, шнур USB.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Управляющая программа с комментариями.
4. Описание работы платформы.
5. Выводы, включающие сравнение полученных результатов с теоретическими предпосылками и исходным заданием на лабораторную работу.

Задачи

1. Написать программу движения платформы по черной линии с использованием любого из регуляторов. Черная линия имеет плавные изгибы и крестообразные перекрестки (пересечения линий под углом 90 градусов). Маршрут должен быть пройден по заранее известной траектории. При въезде на перекресток платформа в заданной последовательности выполняет одно из действий:
 - а) прохождение перекрестков прямо;
 - б) повороты направо/налево;
 - в) остановки;
 - г) разворот на перекрестке на 180 градусов.

Лабораторная работа 8

Система предупреждения столкновений

Цель работы: дополнить программы движения системой предупреждения столкновений.

План работы

1. Изучить теоретический материал.
2. Написать управляющую программу.
3. Загрузить управляющую программу в платформу и протестировать ее работу. При необходимости подкорректировать программу.

Оборудование и материалы

1. Персональный компьютер с установленным программным обеспечением Arduino IDE.
2. Мобильная платформа, шнур USB.

Требования к содержанию отчета

1. Титульный лист.
2. Цель работы.
3. Управляющая программа с комментариями.
4. Описание работы платформы.
5. Выводы, включающие сравнение полученных результатов с теоретическими предпосылками и исходным заданием на лабораторную работу.

Задачи

1. Написать подпрограмму работы с дальномером.
2. Дополнить программу движения по линии системой предупреждения столкновений: если перед платформой находится препятствие — платформа должна остановиться и ожидать исчезновения препятствия.

*Платформа из набора конструктора TETRIX PRIZM
с контроллером PRIZM*

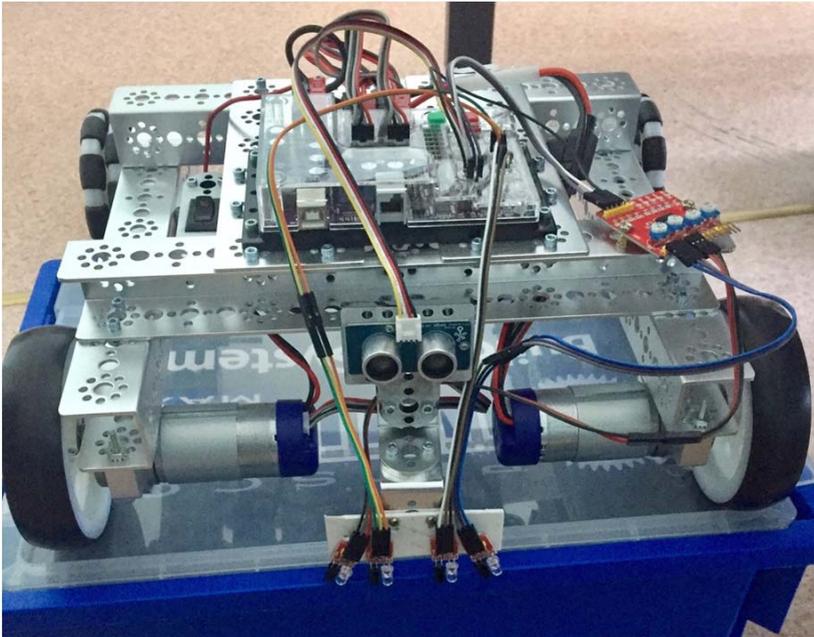


Рис. А.1. Внешний вид платформы из набора конструктора TETRIX PRIZM с контроллером PRIZM (фото из лаборатории института машиностроения)

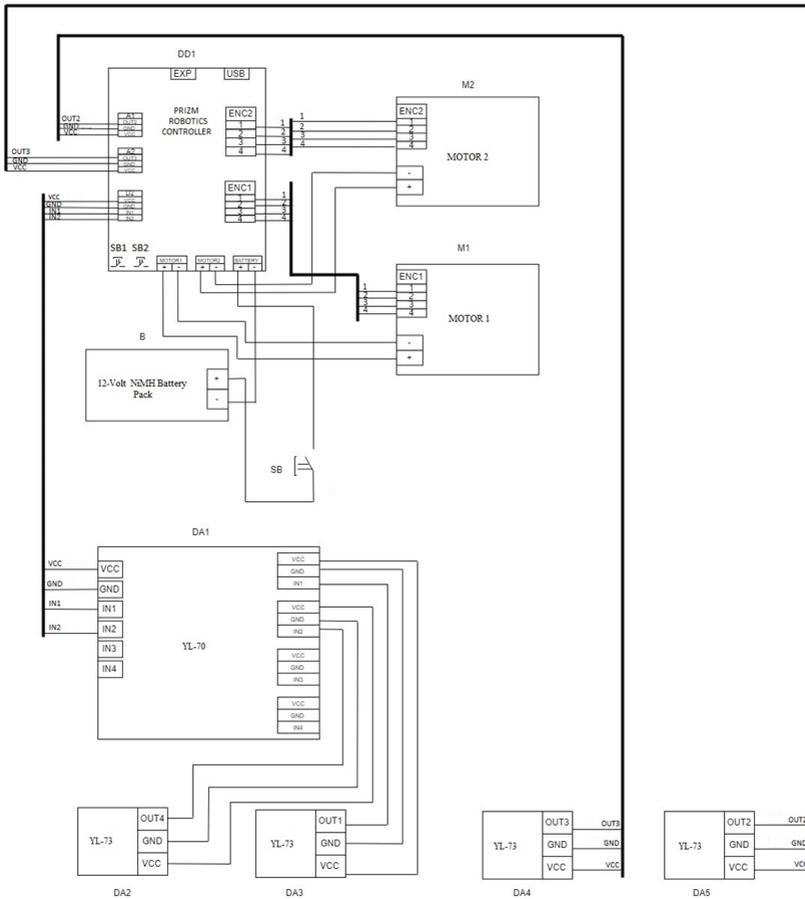


Рис. А.2. Схема электрическая принципиальная платформы из набора конструктора TETRIX PRIZM с контроллером PRIZM (автор А.В. Прядилов)

		<i>Обозначение</i>	<i>Наименование</i>	<i>Кол</i>	<i>Примечание</i>
<i>Лейб. помечен.</i>			<i>Микроконтроллер</i>		
		<i>001</i>	<i>АТmega328P с установленным загрузчиком операционной системы Arduino UnoR1</i>	<i>1</i>	
			<i>Датчики</i>		
		<i>DA2-DA4</i>	<i>Инфракрасный датчик препятствия Y1-73</i>	<i>4</i>	
<i>Способ. №</i>		<i>DA1</i>	<i>Датчик опт.-ий 4-канальный для Arduino Y1-70</i>	<i>1</i>	
			<i>Моторы</i>		
		<i>M1.2</i>	<i>Электродвигатель постоянного тока 12 В, частота вращения 152 об/мин, крутящий момент 2,26Нм</i>	<i>2</i>	
			<i>Кнопки</i>		
<i>Лейб. и дата</i>		<i>SB</i>	<i>Выключатель аккумуляторной батареи контроллера PRIZM</i>	<i>1</i>	
		<i>SB1</i>	<i>Пусковая кнопка</i>	<i>1</i>	
		<i>SB2</i>	<i>Кнопка сброса параметров/остановки</i>	<i>1</i>	
<i>Инд. № дробл.</i>			<i>Батарея</i>		
		<i>B</i>	<i>Перезаряжаемая никель-металлогидридная аккумуляторная батарея TETRIX на 12 В</i>	<i>1</i>	
<i>Инд. №</i>					

Рис. А.3. Перечень элементов для платформы из набора конструктора TETRIX PRIZM с контроллером PRIZM (автор А.В. Прядилов)

*Платформа из набора конструктора TETRIX PRIZM
с контроллером Ардуино и драйвером двигателей*

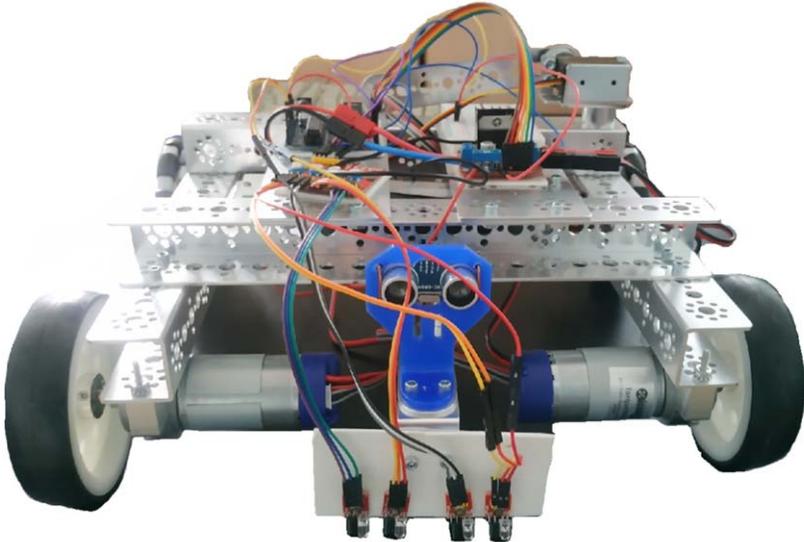


Рис. Б.1. Внешний вид платформы из набора конструктора TETRIX PRIZM с контроллером Ардуино и драйвером двигателей (фото А.В. Прядилова)

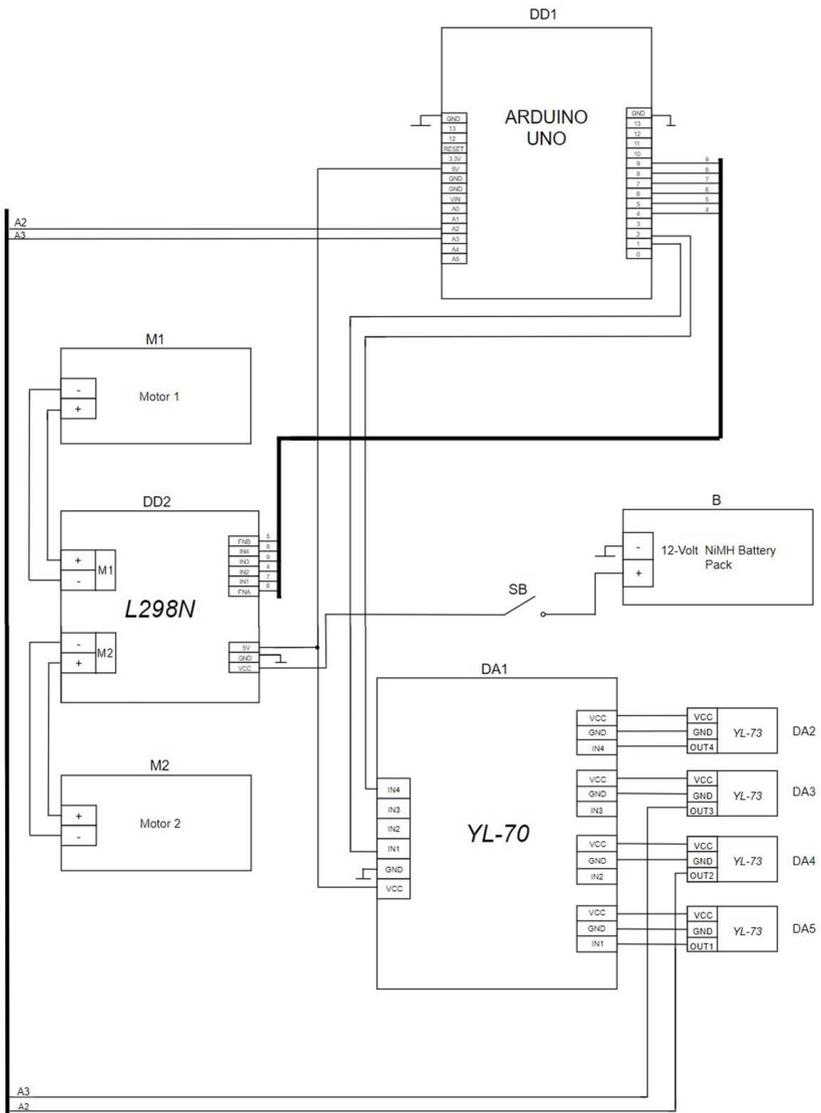


Рис. Б.2. Схема электрическая принципиальная платформы из набора конструктора TETRIX PRIZM с контроллером Ардуино и драйвером двигателей (автор А.В. Прядилов)

Форм.	Зона	Поз.	Обозначение	Наименование	Кол.	Примеч.
				Микроконтроллерная плата		
			DD1	Arduino UNO R3	1	
				Датчики		
			DA2-DA5	Инфракрасный датчик препятствия YL-73	4	
				Напряжение питания: 3,3–5 В		
				Расстояние обнаружения препятствий: 0,12-10 см		
				Эффективный угол обнаружения препятствий:		
			DA1	Датчик оптический 4-х канальный YL-70		
				Моторы		
			M 1,2	Электродвигатель постоянного тока 12 В	2	
				Частота вращения: 152 об/мин		
				Крутящий момент: 2,26 Нм		
				Драйвер		
			DD2	Драйвер двигателей L298N	1	
				Напряжение питания двигателей: до 35 В		
				Рабочий ток (на каждый канал): 2 А		
				Периодический ток (80%-вкл, 20%-выкл): 2,5 А		
				Кратковременный ток: 3 А		
				Батарея	1	
			B	Перезаряжаемая никель-металлогидридная		
				Аккумуляторная батарея TETRIX на 12 В		
				Кнопки		
			SB	Пусковая кнопка	1	

Рис. Б.3. Перечень элементов для платформы из набора конструктора TETRIX PRIZM с контроллером Ардуино и драйвером двигателей (автор А.В. Прядилов)

Бланк титульного листа

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Тольяттинский государственный университет»

Институт машиностроения

Кафедра «Промышленная электроника»

Отчет по лабораторной работе № __

Название лабораторной работы

по дисциплине: Название дисциплины

Исполнитель: Ф. И. О.

Бригада: №

Группа:

Руководитель: Ф. И. О.

Тольятти 2022