

## Учебное пособие к набору TETRIX® "Соревнование в коробке"

Автор текста: Аарон Локк.

Консультанты по контенту: Пол Аттли, Памела Сайферс и Тим Лэнкфорд.

Авторы моделей и рендеров *SolidWorks® Compose™* и *KeyShot®*: Тим Лэнкфорд, Брайан Эккелберри и Джейсон Редд.

Компьютерная вёрстка: Тодд Макджордж

©2018 Pitsco, Inc., 915 E. Jefferson, Pittsburg, KS 66762

Авторские права защищены. Изделие и сопутствующая документация защищены авторским правом и распространяются по лицензиям, ограничивающим их использование, копирование и распространение. Запрещено воспроизводить какую-либо часть данного изделия или сопутствующей документации какими-либо способами без предварительного письменного разрешения со стороны корпорации Pitsco.

Все прочие наименования продукции, упомянутые в данном документе, могут оказаться товарными знаками соответствующих собственников.

# Содержание

<b>О пособиях</b> .....	2
-------------------------	---

## **Робототехнические соревнования — Введение**

Виды робототехнических соревнований .....	3–4
Конструктивные варианты роботов для соревнований .....	4
Советы по проведению соревнований и главные факторы успеха .....	5
Стратегии создания роботов для соревнований.....	5–9

## **Раздел № 1 — Техническое конструирование для защиты окружающей среды**

Отправные положения.....	10
Упражнение № 1 — Проектирование и постройка прототипа.....	11–12
Упражнение № 2 — Испытание и анализ .....	13–14
Упражнение № 3 — Изменение и улучшение проекта .....	15–17
Соревнование № 1 — Очистка мусорной свалки .....	18–19

## **Раздел № 2 — Восполнение потребностей промышленности**

Отправные положения.....	20–23
Упражнение № 4 — Системы навигации .....	24–26
Упражнение № 5 — Подъёмные системы .....	27–28
Упражнение № 6 — Захваты и ковши .....	29–31
Соревнование № 2 — Складские роботы .....	32–36

## **Раздел № 3 — Автономные спасательные роботы**

Отправные положения.....	37–40
Упражнение № 7 — Внесение изменений .....	41–42
Упражнение № 8 — Сочетание датчиков (повторение) .....	43–47
Упражнение № 9 — В какую сторону? .....	48–57
Упражнение № 10 — Всего-то гончий пёс .....	58–65
Соревнование № 3 — Спасательные роботы.....	66–70

## **Приложение**

Электромонтажная схема аппаратуры дистанционного управления .....	71
Электромонтажная схема контроллера PRIZM.....	72
Включение контроллера PRIZM .....	73
Предусмотренные для робота варианты управления и навигации .....	74–79
Робототехнические подъёмные механизмы.....	80–82
Робототехнические хватные механизмы.....	83–84
Толковый словарь .....	85–86

## О пособиях

Назначение пособия — дать вам начальное представление о робототехнических соревнованиях и опыт, который поможет успешно проектировать и конструировать роботов для соревнований. Раздел "Робототехнические соревнования — Введение" содержит много информации о видах робототехнических соревнований, советы для достижения успеха в соревнованиях и описание приёмов, которые при прохождении любого раздела пособия помогут проектировать, конструировать, программировать и испытывать роботов, предназначенных для соревнований. Следующая за введением часть пособия разбита на три раздела. У каждого раздела своя, свойственная только ему, тема, а каждый последующий раздел строится на основе предыдущего.

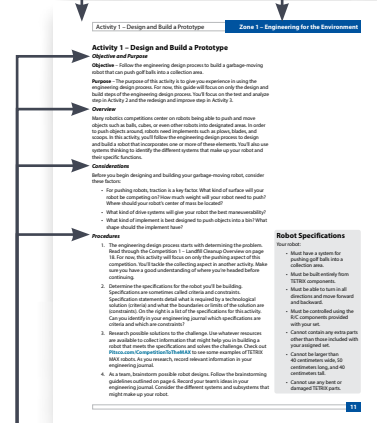
У каждого раздела три основные части:

- **Отправные положения** — Каждый раздел начинается с отправных положений, имеющих отношение к упражнениям и соревнованию, имеющимся в этом разделе. Обязательно прочтите отправные положения, прежде чем погрузиться в упражнения.
- **Упражнения** — В каждом разделе есть 3–4 упражнения, которые помогут вам подготовиться к соревнованию, ждущему в конце раздела. Упражнения следует выполнять по порядку. В ходе упражнений вам предстоит самим проектировать и конструировать роботов — никаких конкретных инструкций по сборке не будет. Чтобы придумать и смастерить роботов, отвечающих заданным техническим условиям, придётся положиться на собственные творческие способности и прежний опыт роботостроения. В каждом упражнении четыре части:
  - Задача и цель
  - Обзор
  - Требуемые учёта обстоятельства
  - Методика и порядок действий
- **Соревнование** — Каждый раздел завершается соревнованием. В случае выполнения упражнений раздела, подводящих к соревнованию, можно считать, что пригодный для соревнований робот уже почти у вас в кармане. Каждое соревнование делится на четыре главных части:
  - Обзор
  - Соревновательная арена
  - Правила и учёт баллов
  - Дополнительные сведения

Завершающий раздел пособия — приложение. Изучение роботов разных систем будет сопровождаться ссылками на разные разделы приложения.

В тексте пособия могут встретиться незнакомые слова, помеченные поджирением. Определения слов даны в толковом словаре в конце пособия.

Раздел      Упражнение



Каждое упражнение делится на четыре главных части:

- Задача и цель
- Обзор
- Требуемые учёта обстоятельства
- Методика и порядок действий

## Робототехнические соревнования — Введение

По всему миру набирают популярность робототехнические соревнования: от небольших состязаний на школьном уроке, в которых можно бросить вызов своим товарищам, до всемирных турниров и сражений боевых роботов, транслируемых в телеэфире. Робототехнические соревнования — серьёзное мероприятие по нескольким причинам. Во-первых, они дают представление о важных понятиях из области точных и естественных наук, технологии, машиностроения. Во-вторых, робототехнические соревнования помогают становлению полезных профессиональных навыков и умений. Вот лишь некоторые из тех, что будут формироваться у вас по мере прохождения этого пособия или участия в робототехнических соревнованиях.

- Творческое и новаторское мышление
- Решение задач и критическое мышление
- Сотрудничество и коллективная работа
- Умение руководить
- Зона ответственности
- Производительность
- Инициативность
- Обмен информацией

Хотя знания и навыки в области техники, естественных и точных наук важны, главная причина популярности робототехнических соревнований — их увлекательность. Мы настоятельно призываем вас превратить изучение этого пособия по соревнованиям в забаву!

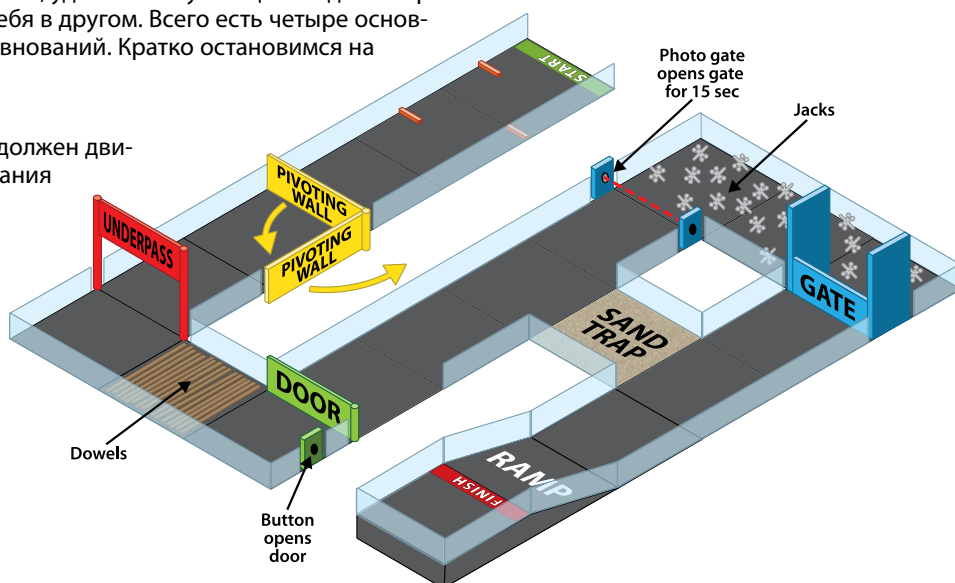
Начнём с главного: прежде чем открыть пособие, надо иметь хотя бы начальное представление о том, как собирать роботов серии TETRIX® MAX и программировать их на выполнение заданий, используя робототехнический контроллер PRIZM®. Наилучший способ набраться такого опыта — выполнить упражнения и конкурсные задания из *Руководства по сборке управляемых моделей и Руководства по программированию контроллера PRIZM*, прилагаемых к набору конструктора TETRIX MAX. Если вы два этих руководства ещё не освоили, очень рекомендуем сделать это перед дальнейшей работой с данным пособием.

### Виды робототехнических соревнований

Робототехнические соревнования бывают разного масштаба, вида и формы. Преследуют разные цели, решают разные задачи. Успех зависит от того, удастся ли построить робота, в точности соответствующего условиям конкретного соревнования. Иными словами, робот, удачно выступающий в одном соревновании, может неважно проявить себя в другом. Всего есть четыре основных вида робототехнических соревнований. Кратко остановимся на каждом из них.

#### Соревнования на трассе

В соревнованиях на трассе робот должен двигаться по маршруту, выполняя задания и объезжая по пути препятствия. Побеждает обычно тот, кто за отведённое время продвинется дальше других или пройдёт трассу быстрее всех. В этом виде соревнований главные слагаемые успеха – управление движением, скорость, манёвренность и преодоление препятствий.



### Соревнования на арене

Соревнования на арене входят в число наиболее популярных робототехнических соревнований. Их проводят на огороженной игровой площадке. Робот движется по игровой площадке и выполняет задания: собирает предметы, оставляет предметы в зачётных местах, нажимает кнопки, включает реле, бросает предметы в цель. За выполнение этих заданий начисляются баллы. Как правило, чем труднее задание, тем больше баллов оно приносит. Обычно роботы выступают на арене поочередно в течение отведённого времени, набирая как можно больше баллов. При всей важности скорости и манёвренности, в соревнованиях на арене главную роль играют приспособления и орудия робота.

### Соревнования один на один

Соревнования один на один похожи на соревнования на арене, но разница в том, что здесь одновременно соревнуются друг с другом два робота. В турнирах обычно друг другу противостоят два робота — победитель переходит в следующий круг. С учётом размера арены соревнования один на один могут быть очень захватывающими. Роботу нужно не только набрать как можно больше баллов, но при этом и маневрировать вблизи другого робота. Из-за этого дополнительного фактора важными становятся прочность и износостойкость, чтобы при столкновении робот не сломался.

### Соревнования объединённых команд

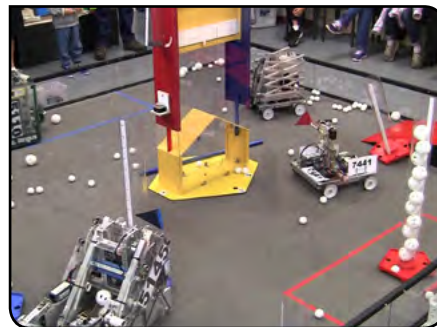
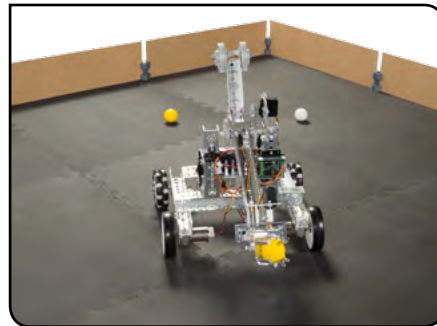
Соревнования объединённых команд входят в число наиболее захватывающих и окупающих затраченные усилия состязаний на арене. Как и соревнования один на один, соревнования объединённых команд обычно проводятся на арене. Но вместо двух роботов в состязании противостоят друг другу команды роботов. Помимо слаженности и сотрудничества в соревнованиях объединённых команд нередко требуется много внимания уделять стратегии. Чтобы набрать как можно больше баллов, роботам и командам нужно действовать сообща.

### Конструктивные варианты роботов для соревнований

Как правило, робототехнические соревнования устраивают для роботов двух конструктивных вариантов: управляемых человеком и **автономных**. В некоторые соревнования даже включают элементы обоих с целью более полной проверки робота и его создателей.

В соревнованиях роботов, управляемых человеком, используется технология беспроводного дистанционного управления (Д/У) и технология управления по каналу Bluetooth: оператор, а иногда и несколько операторов управляют роботом с безопасного расстояния. Для таких соревнований нужны хорошие проектно-конструкторские навыки и умения в области роботостроения. Но не менее важны и такие свойства человека, как навыки вождения, скорость реакции и пространственная ориентация. Как и в любой другой игре или спорте, соревнования управляемых человеком роботов требуют, чтобы соревнующиеся целеустремленно оттачивали своё мастерство.

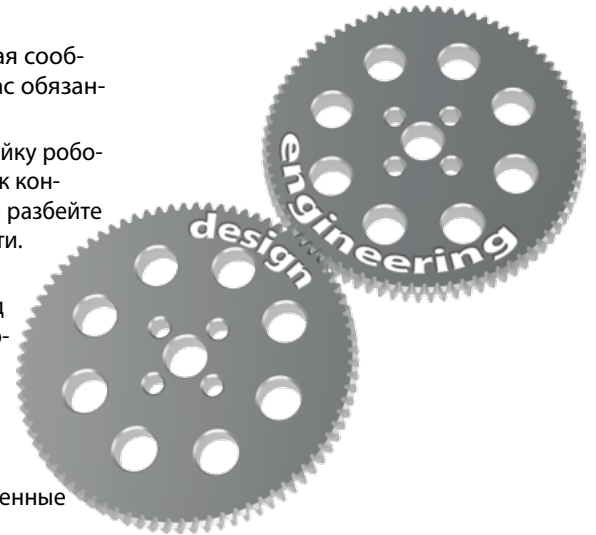
Из собственно соревнований автономных роботов человеческая составляющая исключается. Но тут, чтобы заранее запрограммировать робота на выполнение заданий, не обойтись без творчества и изобретательности. Когда начинается соревнование, запускается исполнение управляющей программы и робот соревнуется с соперниками самостоятельно, повинаясь заложенной в него программе. Для успеха в таких соревнованиях нужны хорошие проектно-конструкторские навыки и умения в области роботостроения. Но для программирования/составления управляющего кода необходимо также, чтобы соревнующиеся хорошо умели решать задачи, логически мыслить и даже общаться. Ведь программирование — это общение с компьютером на понятном ему языке.



## Советы по проведению соревнований и главные факторы успеха

Для достижения успеха в робототехнических соревнованиях есть несколько главных правил. В том числе:

- Извлекать уроки из поражений — если что-то не получилось, не отчаивайтесь. Человеку свойственно ошибаться, вот и всё. Великие команды, соревнуясь, извлекают уроки из неудач и перестраиваются.
- Действуйте профессионально — делайте всё, что умеете, как можно лучше, качественно и честно.
- Ведите себя уважительно — будьте великодушны, учтивы и вежливы по отношению к окружающим: товарищам по команде, судьям соревнований и другим командам.
- Сотрудничайте с другими людьми — добивайтесь целей, работая сообща. Делайте свою часть работы, выполняйте возложенные на вас обязанности и при необходимости помогайте друг другу.
- Планируйте — хотя, возможно, так хочется сразу взяться за постройку робота, не спеша обдумайте план действий. Составьте перечень методик конструирования, нарисуйте эскизы проектов, напишите **псевдокод** и разбейте общее конкурсное задание на более удобные для выполнения части.
- Разработайте долгосрочные планы — Не всегда создание самого большого, самого тяжёлого, самого крепкого робота решает исход дела — иногда побеждают команды, которые по-настоящему хорошо выполнили какую-то отдельную часть конкурсного задания.
- Будьте изобретательны — не бойтесь проверять в деле новые идеи. Случается, что наилучшими решениями оказываются те, которые никто прежде не пытался осуществить.
- Пользуйтесь техническим журналом — ведите точные и своевременные записи.
- Воспринимайте это как забаву — сказано уже достаточно.



## Стратегии создания роботов для соревнований

Проектируя и конструируя роботов для соревнований можно руководствоваться несколькими стратегиями. Упомянутые здесь стратегии проектирования не исключают друг друга, то есть вам не надо выбирать между ними. Наоборот, обычно они дополняют друг друга, если используются одновременно. Перед тем как окунуться в соревнование, найдите время и прочтите, что написано об этих стратегиях.

### Алгоритм проектирования технических конструкций

Многие технологи, инженеры и проектировщики роботов прибегают к алгоритму проектирования технических конструкций, чтобы разработать решения к общемировым техническим задачам. И точно такой же алгоритм часто используют команды, участвующие в робототехнических соревнованиях, если хотят обеспечить себе успех при проектировании и конструировании своего робота. У алгоритма проектирования технических конструкций много вариантов, но во всех них соблюдаются одни и те же основные принципы и закономерности. Рассмотрим вкратце каждый этап этого алгоритма.

#### Определение задачи или возможности

Наша способность творить — одно из многих свойств человека, отличающее его от других живых существ. Мы находим неординарные способы восполнения нужд и потребностей, решения задач и выхода из тупиков. Но прежде чем решить задачу или разрешить проблему, нужно её уяснить. Обычно для ясного понимания и формулировки решаемой задачи требуется поставить правильные вопросы и исследовать тему. Иногда помогает письменная формулировка задачи, в которой ясно изложена проблема или задача, требующая разрешения.

### Определение технических условий

После чёткой постановки задачи или задания нам понадобятся технические условия — они дают понять, какого рода решения приемлемы. **Технические условия** часто подразделяются на критерии и ограничения. **Критерии** — это желательные следствия применения решения. Они констатируют, что должно произойти от реализации решения, и насколько хорошо это должно быть. С другой стороны, **ограничения** представляют собой рамки или пределы, нередко обусловленные имеющимися в распоряжении ресурсами. Обычно мы ограничены во времени, материалах, капитале, людях, инструментах и энергии. Наличие чёткого и окончательного перечня критериев и ограничений поможет определить, осуществимо ли решение.

### Научные исследования

Исследования улучшают понимание задачи или задания. К типичным методам исследования относятся штудирование книг, статей и тематически связанных источников в интернете; постановка научных опытов; опросы населения; изучение исторических событий или проблем, имеющих отношение к рассматриваемой задаче.

### Проведение мозгового штурма для поиска возможных решений

**Идеация** — это процесс порождения идей. Мозговой штурм, вероятно, наиболее типичный метод идеации. Мозговой штурм преследует целью выявление как можно большего числа решений. Мозговой штурм часто удаётся лучше всего при групповом или коллективном обсуждении возникающих идей, разнообразных и непохожих друг на друга. Организуя мозговой штурм в команде, придерживайтесь следующих принципов:

- Поощряйте выдвижение творческих, грандиозных и диких идей, даже если поначалу они кажутся смешотворными. Будьте находчивы, пользуйтесь заданными техническими условиями и имеющимися материалами. Подобное мышление нередко приводит к наилучшим решениям, потому что до них не додумались другие люди.
- Не теряйте сосредоточенности на проблеме или задаче. Позаботьтесь о том, чтобы все члены команды ясно понимали задачу.
- Осуждать других людей или их предложения нельзя. Порой выдвижение творческих или диких идей бывает рискованным. Люди не должны опасаться выступать с такими идеями из-за страха осуждения.
- Отталкивайтесь от чужих предложений. У кого-то может возникнуть некая идея, которая, стоит её высказать, порождает новую мысль у окружающих.
- Записывайте все идеи и решения по мере их упоминания. Их учёт в виде списка пригодится, когда придёт пора оценивать решения.
- Поощряйте к участию и высказыванию своего мнения всех членов команды. Даже у самых незаметных, самых сдержанных учеников нередко бывают великие идеи. Таким надо лишь немного помочь с их выражением.

### Формулирование технического решения, которое предстоит осуществить

После мозгового штурма время приступить к формулировке какого-то решения. Иногда в алгоритме проектирования технических конструкций это называют этапом формирования технического проекта. Ещё раз просмотрите список идей, созданный на этапе мозгового штурма, и найдите одну идею или сочетание идей, из которых получится отдельное решение. Остановив свой выбор на том или ином варианте, помните о диктуемых задачей критериях и ограничениях. Подробно распишите своё решение от начала до конца, подготовив письменные описания, разработав методики сборки, составив перечни нужных материалов и набросав эскизы. Эскизы могут быть простыми зарисовками, сделанными от руки и передающими основные идеи проекта, либо более сложными чертежами, выполненными при помощи программного приложения для автоматизированного проектирования и черчения (CADD), дающими подробное представление о выбранном решении.

Вот был бы у меня мозг.





### Создание прототипа или модели технического решения

Когда конструктивное решение придумано и продумано, создаётся прототип или модель. Именно тут идеи обретают материальную форму. Прототипы и модели могут быть натурными или уменьшенными вариантами выбранного конструктивного решения — это зависит от задачи. Цель прототипа или модели — получить некий осязаемый объект, пригодный для испытаний, помогающих понять, насколько хорошо решение соответствует критериям и ограничениям.

### Испытание и анализ технического решения

При испытании прототипа (опытного образца) или модели возможны самые разные способы оценки. Одни испытания могут быть простыми и охватывать лишь небольшой набор функций решения. Другие испытания могут быть весьма сложными; с их помощью определяют, в какой степени решение соответствует техническим условиям задачи или конкурсного задания. В ходе типовых испытаний оцениваются функциональность, долговечность, эргономичность, эстетичность, безопасность и, разумеется, соответствие критериям и ограничениям, продиктованным задачей. По ходу испытаний вашего опытного образца нужно будет собрать как качественные, так и количественные данные. Качественные данные представляют собой данные наблюдений, которые не поддаются точному измерению. Количественные данные можно измерить. Обычно с данными удобнее работать, когда они сведены в схемы, таблицы и графики. Они помогают анализировать данные и определять тенденции и закономерности. Они также помогают выводить заключения и доводить результаты испытаний до сведения других людей.

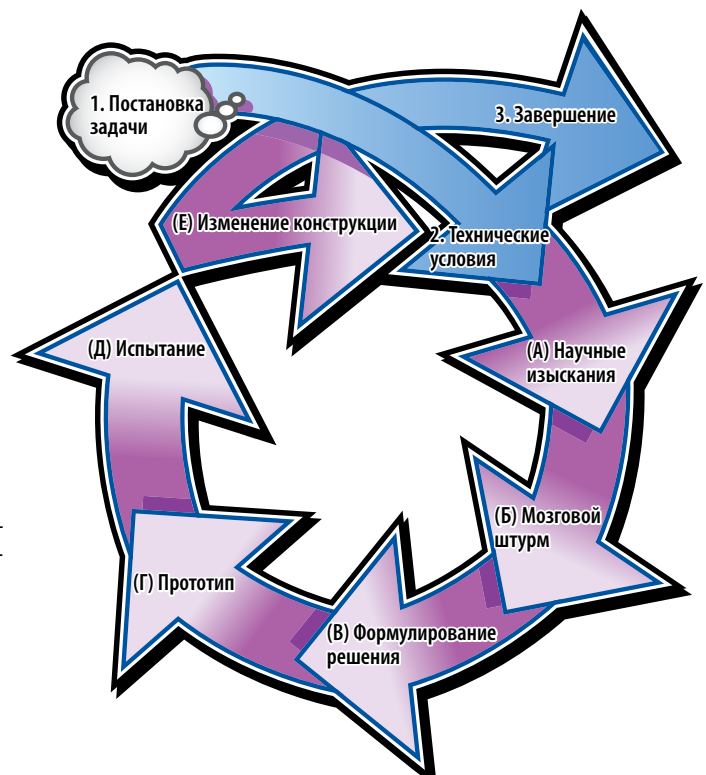
### Повторное проектирование или совершенствование технического решения

По завершении испытаний необходимо принять решение.

1. Можно, например, повторить процесс. Цель здесь — переделать найденное решение, внося изменения с учётом собранных данных и полученной информации. Можно вернуться к любому этапу в этом цикле — выбор зависит от результатов испытаний и того, что для вас в решении главное. Часто бывает так, что для дальнейшего улучшения конструктивного решения требуется неоднократное повторение действий.
2. Ещё один вариант действий: можно признать решение годным, отказавшись от дальнейшего его улучшения. Наступит время, когда вам скорее всего придётся избрать этот вариант, потому что бесконечно улучшать решение нельзя: ресурсы будут исчерпаны.
3. Третий вариант: прекратить разработку, не найдя удачного решения. Иногда при создании технической конструкции просто не хватает времени или денег на дальнейшие поиски жизнеспособного решения.

### На этом проектирование завершается

Когда создание изделия или решения закончено, о нём необходимо сообщить нужным людям. Затем наступает черёд внедрения или использования решения. Если изделие представляет собой новое, оригинальное изобретение, можно подать заявку на патент. Патент защищает ваши идеи и разработки от копирования и продажи кем-то другим. На этапе реализации проектного решения вы, возможно, столкнётесь с новыми затруднениями или задачами, которые не проявились во время испытаний. После этого можно вернуться к нерешённым вопросам и задачам и воспользоваться алгоритмом проектирования технических конструкций.



### Системный подход

В ходе разработки технических конструкций проектировщики роботов, инженеры и технологи нередко прибегают к так называемому системному подходу.

**Системный подход** — способ рассмотрения вещей в их взаимосвязи внутри целого. При таком подходе рассматриваются отношения между частями системы, а не сами её части.

**Системы** можно сравнить с кирпичиками, из которых складывается техническое устройство. Каждая система выполняет предназначенное ей действие, но, взаимодействуя между собой, все системы служат общей цели. Менее крупные системы вложены в более крупные технические, природные или общественные системы. Так, иммунная система в вашем организме отражает нападения болезнетворных микробов. В автомобиле система охлаждения отводит от двигателя тепло, чтобы он не перегрелся. В роботе электрическая цепь снабжает сервоприводы и электродвигатели энергией, чтобы робот мог выполнить отведённые ему задачи. К прочим типичным для роботов системам относятся шасси, система обработки команд, ходовая система и приводной механизм, манипуляторы, ловушки и датчики.

В составе всех систем есть три главных группы: входные элементы, процессы и выходные элементы. В группу входных элементов входят люди, информация, материалы, время, энергия, деньги и орудия труда. Входные элементы попадают в систему и подвергаются изменениям в результате применения процессов (то есть алгоритмов обработки). К процессам относятся производственные и управленческие действия, ведущие к появлению технического решения или изделия. Процессы превращают входные элементы в выходные элементы. Выходные элементы представляют собой результаты работы системы и включают в себя окончательные варианты изделий или решений, неиспользованные ресурсы, отходы, а также положительные и отрицательные последствия.



Иногда в составе систем бывают системы поменьше — их называют подсистемами. Представьте себе электрическую цепь внутри робота. В этой системе много подсистем: аккумуляторные батареи, контроллеры, переключатели, провода и т. п. В каждой из этих подсистем есть составные части определённого назначения. Батарейки и аккумуляторы служат источником электричества, контроллеры командуют электродвигателями и сервоприводами, переключатели регулируют подачу электрического тока в цепи, а провода проводят электричество в другие системы и подсистемы робота. Эти подсистемы сообща работают в системе верхнего уровня, которая снабжает робота энергией. Часто выходной элемент одной подсистемы становится входным элементом другой.

Итак, системы зачастую состоят из подсистем. Иногда подсистемы состоят из других подсистем, ещё менее крупных. Системный подход даёт понимание того, как создать и соединить эти системы и подсистемы так, чтобы они работали слаженно. При системном подходе нередко приходится в чём-то ограничить одну систему, чтобы улучшить работу другой. Это называют **компромиссом**. Чтобы определить наилучший вариант компромисса для взаимодействия систем и подсистем и достижения поставленной цели или решения задачи, необходим большой объём логических рассуждений и творческих находок.

Используя системный подход при проектировании и постройке робота, следуйте таким простым советам:

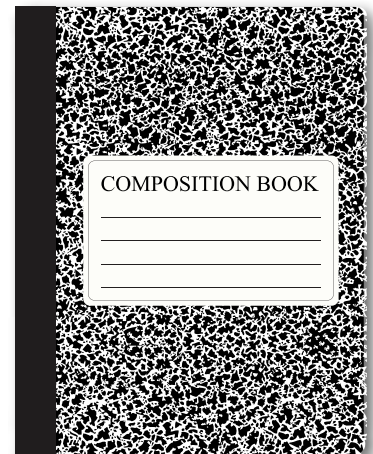
- Стремитесь к простоте. Избегайте чрезмерного усложнения системы в техническом или ином плане.
- Определите для каждой системы входные элементы, процессы и желаемые выходные элементы.
- Найдите способы улучшить подсистемы, которые улучшат общую работу системы.
- По возможности предельно уменьшите число деталей и узлов. Это тем более важно при ограниченном числе имеющихся составных частей. Ищите возможность совместного использования составных частей системами.
- Найдите способы сделать составные части многоцелевыми. Например, помимо снабжения робота энергией, аккумуляторная батарея могла бы служить противовесом или помогать распределению массы на ведущие колёса.
- Обеспечьте пропорциональность своих инженерных компромиссов. Так, возможно, придётся поступиться скоростью ради улучшения управляемости или манёвренности.
- Придумывайте такие системы, которые удобно собирать, разбирать и прикреплять к другим системам.

### **Технический журнал**

От многих конструкторов требуют тщательно записывать их идеи, придуманные конструкции, решения, процессы, ход и результаты испытаний, а также вносимые в них изменения. Хотя порой это кажется нерациональной тратой времени, упорядочение мыслей и аккуратное ведение записей в конечном итоге окупится.

Вот перечень того, что стоило бы включить в технический журнал:

- Ежедневная ведомость выполненных работ
- Перечни проектных критериев и ограничений
- Исследовательские заметки
- Результаты мозгового штурма или иных форм идеации
- Принятые технические решения
- Списки имеющихся в распоряжении ресурсов
- Эскизы, планы и подробные чертежи
- Процессы, требующие завершения, и способы их завершения
- Выполненные расчеты, имеющие отношение к проекту
- Методики и результаты испытаний
- Анализ данных и заключения
- Документация, в которой отражены изменения, внесённые при повторении этапов проектирования технических конструкций



# Раздел № 1 — Техническое конструирование для защиты окружающей среды

## Отправные положения

Техника окружает нас повсюду. Для донесения и получения информации мы используем смартфоны, для развлечения — экраны и телевизоры, передающие изображение высокой чёткости, у нас есть автомобили, которым почти не нужен водитель, и персональные компьютеры, более мощные, чем те, с помощью которых люди добрались до Луны. Но что происходит, когда наша техника перестаёт работать? Мы ремонтируем её или просто покупаем что-то взамен?

Мы, как общество, выбрасываем всё больше и больше мусора. Когда что-то перестаёт работать, мы выбрасываем старое и покупаем новое. Ведь, учитывая темпы развития техники, мы нередко считаем, что нам нужны самые современные и самые лучшие устройства. Но, избавляясь от старых, сломанных приборов и заменяя их новыми, мы создаём новую проблему для общества. Электронные отходы или, иначе говоря, электронный утиль зачастую содержат ядовитые химикаты, способные серьёзно навредить окружающей среде и людям. Попадая на мусорные полигоны, свалки металлолома и свалки бытовых отходов, электронный утиль может превратить их в места скопления ядовитых отходов.

Но так же, как технология способна породить проблемы для окружающей среды, она может дать и их решения. Для начала можно придумать, как сбросить природные ресурсы, наладив их многократное использование, восстановление и переработку с целью вторичного применения. Можно разработать технологию слежения за здоровьем окружающей среды и получения ценных сведений о нём с целью принятия более обоснованных решений. Ещё мы можем изобретать чистые технологии, ослабляющие отрицательное воздействие других вредных технологий.

## Цель соревнования

В этом разделе главным будет проектирование и конструирование роботизированной техники, способной заезжать в опасные зоны (мусорные свалки и захоронения ядовитых отходов) и безопасно сортировать мусор, отделяя ядовитые отходы от пригодных для переработки. Вникая в аспекты дистанционного управления моделями для робототехнических соревнований, вы выполните три упражнения, которые подведут вас непосредственно к соревнованию, припасённому в конце раздела. Для выполнения этих упражнений вам понадобится алгоритм проектирования технических конструкций и системный подход. До начала упражнения № 1, пожалуй, стоит прочесть всё о соревновании по очистке мусорной свалки, начав со страницы 18.

Для этого первого соревнования понадобится робот, способный толкать перед собой одни предметы, собирая при этом другие. Никаких объяснений ожидаемого порядка сборки не будет. Вместо этого, в пособии изложены простейшие действия и сведения, опираясь на которые вы строите собственного робота, реализуя собственные творческие замыслы и находки. Вы уже должны были создать нескольких роботов, осваивая *Руководство по сборке управляемых моделей* и *Руководство по программированию контроллера PRIZM PRIZM*, прилагаемых к вашему робототехническому набору TETRIX MAX. Выполнять упражнения в этом разделе придётся, обращаясь к знаниям и опыту, почерпнутым при создании прежних роботов.



## Упражнение № 1 — Проектирование и постройка прототипа

### Задача и цель

**Задача** — Следуя алгоритму проектирования технических конструкций, создайте мусороуборочного робота, способного толкать мячи для гольфа на накопительный участок.

**Цель** — Узнать, как проектировать технические конструкции по алгоритму. Пока что в пособии будут рассматриваться только этапы проектирования и конструирования из всего алгоритма проектирования технических конструкций. В упражнении № 2 вы займётесь испытанием и анализом, а в упражнении № 3 — рассмотрением и улучшением придуманной конструкции.

### Обзор

На многих робототехнических соревнованиях основное внимание уделяется способности роботов, толкая разные предметы (шарики, кубики или даже других роботов), передвигать их в заданное место. Чтобы перемещать предметы по поверхности, роботам нужны орудия труда: плуги, отвалы и ковши. В ходе этого упражнения вы, следуя алгоритму проектирования технических конструкций, проектируете и конструируете робота, в состав которого входит не меньше одного из указанных элементов. Вам также понадобится системный подход, чтобы увидеть разные системы, входящие в состав вашего робота, и свойственные этим системам функции.

### Требуемые учёта обстоятельства

Перед проектированием и конструированием мусороуборочного робота примите во внимание такие факторы:

- Для роботов-толкатей сила сцепления с опорной поверхностью играет главную роль. На какого рода поверхности предстоит соревноваться вашему роботу? Насколько тяжёлый груз нужно будет толкать вашему роботу? Где должен находиться центр масс вашего робота?
- Какого рода приводные системы сделают вашего робота наиболее манёвренным?
- Какого рода рабочий орган лучше придумать, чтобы заталкивать им предметы в мусорный бак? Какая форма должна быть у этого рабочего органа?

### Методика и порядок действий

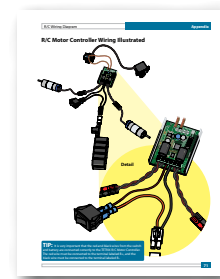
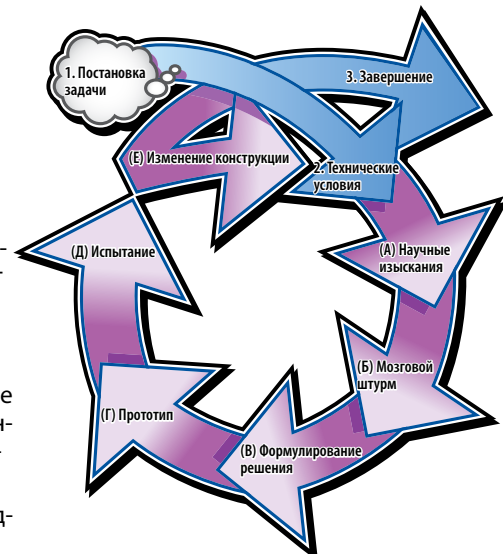
1. Проектирование технических конструкций начинается с формулирования задачи. Прочтите от начала до конца обзорную часть главы "Соревнование № 1 — Очистка мусорной свалки" на странице 18. Пока что упражнение сосредоточено только на таком аспекте этого соревнования, как толкание предметов. Над собиранием предметов вы будете работать в другом упражнении. Вы должны хорошо понимать, куда двигаться, прежде чем приступите к делу.
2. Определите, по каким техническим условиям будете строить робота. Технические условия иногда называют критериями и ограничениями. В техническом задании (описании технических условий) излагаются требования к техническому решению (критерии) и устанавливаются границы или пределы для этого решения (ограничения). Справа — перечень технических условий для этого упражнения. Сумеете ли вы в своём техническом журнале указать, где в технических условиях критерии, а где ограничения?
3. Исследуйте возможные решения этой задачи. Используйте любые имеющиеся ресурсы, чтобы собрать информацию, полезную для постройки робота, соответствующего техническим условиям, и для выполнения задания. Загляните на сайт [Pitsco.com/CompetitionToTheMAX](http://Pitsco.com/CompetitionToTheMAX) и ознакомьтесь с примерами роботов серии TETRIX MAX. По ходу исследования записывайте всё, что относится к делу, в свой технический журнал.

### Технические характеристики робота

Ваш робот:

- Должен иметь систему толкания мячей для гольфа на накопительный участок.
- Должен быть собран только из деталей конструктора TETRIX.
- Должен уметь разворачиваться во всех направлениях и двигаться передним и задним ходом.
- Должен управляться радиоаппаратурой, входящей в ваш набор.
- В его составе не должно быть никаких дополнительных деталей, кроме тех, что входят в предоставленный вам набор.
- Его габариты не должны превышать 40 см в ширину, 50 см в длину и 40 см в высоту.
- В нём нельзя использовать никакие погнутые или испорченные детали TETRIX.

4. Всей командой обсудите все вероятные и невероятные варианты конструкции робота, которые придут вам на ум. Следуйте правилам мозгового штурма, изложенным на странице 6. Запишите выдвинутые вашей командой идеи в свой технический журнал. Обдумайте разные системы и подсистемы, которые можно ввести в состав вашего робота.
5. Просмотрите записи идей, возникших у вашей команды во время мозгового штурма, и отберите для робота те из них, которые, как вы считаете, больше всего соответствуют проектным критериям и ограничениям. Нередко бывает так, что наилучшее решение задачи даёт сочетание идей, возникших во время мозгового штурма. Остановив выбор на некой идее, действуйте в следующем порядке, разрабатывая конструкцию робота в своём техническом журнале:
  - A. Определите и составьте список систем и подсистем, которые войдут в состав вашего мусороуборочного робота. Письменно опишите, как каждая система должна работать и взаимодействовать с другими системами робота.
  - B. Нарисуйте эскизы робота, чтобы было легче мысленно представить, как будут взаимодействовать его системы.
  - C. Запишите методику и порядок конструирования своего робота. В каком порядке следует собирать робота?
  - D. Определите и составьте список основных материалов, которые понадобятся. Не стоит указывать в списке каждый винт и скобу, вместо этого обдумайте, какими будут основные структурные, навигационные и функциональные элементы вашего робота.
6. Теперь, наконец, пора конструировать прототип мусороуборочного робота. Сконструируйте робота по созданному проекту. В ходе сборки нередко бывает необходимо внести изменения в первоначальный план, поскольку вскрываются недостатки проекта. В ходе сборки обновляйте проект робота в техническом журнале. Видоизменяя проект и прототип робота, помните о критериях и ограничениях.
7. Оснастите робота всей нужной электропроводкой. За справкой по электромонтажу для своего робота обращайтесь в приложение к этому пособию.
8. Включите робота и контроллер. Если не поймёте, как подсоединить контроллер к роботу, см. *Руководство по сборке управляемых моделей*, прилагаемое к вашему конструктору TETRIX. Вы должны уметь управлять своим роботом. В следующем упражнении вы испытаете робота тщательнее. Пока что просто убедитесь, что робот работает. Когда закончите, выключите робота и контроллер.



Страница 71



**Совет:** Проследите, чтобы для следующего упражнения аккумуляторная батарея робота была полностью заряжена.

## Упражнение № 2 — Испытание и анализ

### Задача и цель

**Задача** — Продолжить работу по алгоритму проектирования технических конструкций: испытывать и анализировать результаты испытания своего мусороуборочного робота.

**Цель** — Продолжить проектирование технических конструкций по алгоритму. Вам предстоит испытать своего робота, чтобы узнать, как он покажет себя в предстоящем соревновании. Проверка эксплуатационных характеристик или эксплуатационные испытания — крайне важный этап в любом робототехническом соревновании. Ни в коем случае нельзя выходить соревноваться, если робот не прошёл испытания.

### Обзор

В ходе предыдущего упражнения вы придумали и собрали мусороуборочного робота, способного толкать мячи для гольфа на накопительный участок. Но одно лишь обладание роботом не означает, что вы подготовлены к соревнованию. Важно испытать вашего робота и проанализировать, как он работает. Первый рабочий образец редко бывает наилучшим решением технической задачи. Испытания выявляют недостатки конструктивного образца, указывают на сильные и слабые стороны конструктивного решения и подводят к новым идеям, направленным на его улучшение. В этом упражнении вы продолжите создавать техническую конструкцию: будете испытывать и анализировать поведение своего робота на соревновательной арене. После сбора данных вы должны будете изменить конструкцию своего робота так, чтобы его рабочие характеристики улучшились.

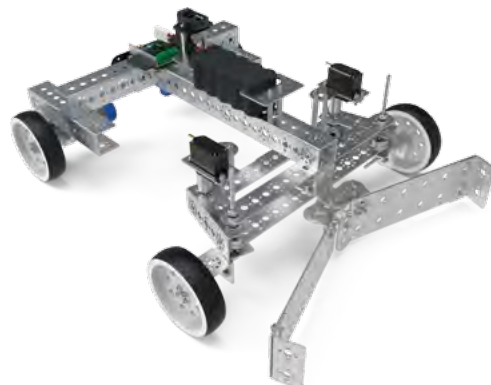
### Требующие учёта обстоятельства

Перед испытанием и анализом мусороуборочного робота примите во внимание такие факторы:

- Как выглядит успех?
- Какого рода качественные (наблюдательные) данные вы можете собрать?
- Какого рода количественные (измеримые) данные вы можете собрать?
- Обдумайте каждую систему и подсистему вашего робота. Как эти системы взаимодействуют? Влияет ли одна система на другую?

### Методика и порядок действий

1. Во-первых, вам надо разработать испытание для своего робота. Оно может быть настолько простым или сложным, насколько вам захочется. Запишите порядок и методику испытаний в свой технический журнал. Помните: пока что главное в этом упражнении — способность робота затолкать мячи для гольфа на накопительный участок.
2. Решите, какого рода данные надо собрать. Впоследствии, проводя испытание, надо будет записать собранные данные. Зачастую полезно сводить данные в схемы и таблицы. При необходимости создайте эти схемы и таблицы в своём техническом журнале.
3. Проследите за тем, чтобы ваш мусороуборочный робот был готов к испытанию. Перед включением робота и контроллера аккумуляторная батарея должна быть заряжена, а контроллер должен быть подсоединён к приёмнику.
4. Есть два варианта испытания мусороуборочного робота. Для мелких функциональных испытаний робота подойдёт рабочая станция или другой испытательный участок, назначенный учителем. Для серьёзных эксплуатационных испытаний придётся вывести робота на соревновательную арену.
  - A. Для испытания за пределами арены:
    - i. Проследите, чтобы на испытательном участке не было препятствий и помех.
    - ii. Поставьте робота на пол в отведённом для испытания участке.



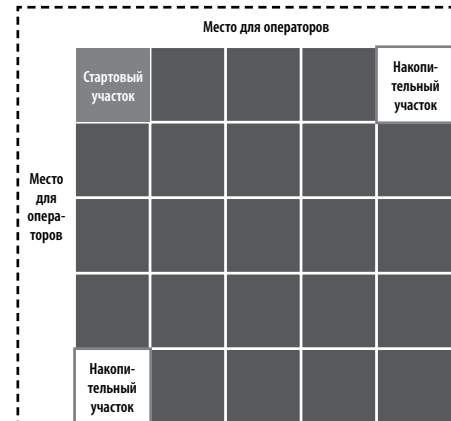
**Совет:** Обычно считается правильным начинать испытывать системы робота по отдельности, одну за другой. После успешного испытания каждой отдельной системы можно перейти к испытанию взаимодействия систем.

**Совет:** У вас будет возможность испытать своих роботов как внутри, так и снаружи соревновательной арены.

- iii. Включите робота и контроллер.
- iv. Проведите испытание согласно методике, которую вы описали в техническом журнале. Собранные во время испытания данные запишите в свой технический журнал.
- v. По завершении испытания выключите робота и контроллер.

В. Для испытания в пределах арены:

- i. Для испытания робота на арене вам, возможно, придётся подождать своей очереди — это зависит размера вашего класса и числа учеников в нём. Когда подойдёт ваша очередь, у вас будет три минуты на проведение испытаний на соревновательной арене.
- ii. Подготовьте арену к испытаниям. Если нужно, разбросайте по арене мячи для гольфа.
- iii. Поставьте на арену робота. Включите робота и контроллер.
- iv. Проведите испытание согласно методике, которую вы описали в техническом журнале. Собранные во время испытания данные запишите в свой технический журнал.
- v. По истечении ваших трёх минут на соревновательной арене выключите робота и контроллер. Уберите робота с арены.
- vi. Уберитесь на соревновательной арене и подготовьте её для следующей группы.



- 5. По завершении испытания и сбора некоторых данных наступает черёд анализа этих данных. Что данные говорят о вашем роботе? Какие заключения можно вывести? Наблюдается ли в ваших данных какие-либо тенденции или закономерности? Указывают ли данные на то, что робот ведёт себя, как и ожидалось? Запишите ход и результаты анализа в свой технический журнал.
- 6. Следующий этап в проектировании технических конструкций — использование собранных данных и сведений для изменения или улучшения конструкции вашего робота. Для этого придётся вернуться к предыдущему этапу цикла и пройти цикл ещё раз. В зависимости от результатов испытания может возникнуть необходимость сделать шаг назад и провести дополнительные изыскания, чтобы определить, какое решение лучше. Для разрешения проблем, обнаруженных при испытаниях, возможно, понадобится собрать новые идеи методом мозгового штурма. Либо можно просто слегка подправить проект. В любом случае для улучшения проекта своего мусороуборочного робота повторите этапы проектирования технических конструкций. Запишите все внесённые в проект изменения в свой технический журнал.
- 7. Далее, внесите изменения, отражающие изменения в проекте, в прототип робота. Видоизменяя робота, помните о критериях и ограничениях.
- 8. Теперь вы готовы провести ещё одно испытание робота. Продолжайте испытывать, анализировать, дорабатывать и улучшать своего робота согласно этим методикам. Продолжайте повторять эти этапы, внося улучшения, пока учитель не скажет, что можно переходить к следующему упражнению.



**Совет:** Помните, что неудача нередко бывает частью процесса технического конструирования. Благодаря неудаче мы узнаём, что не работает и чего не следует делать. Испытывая и дорабатывая своего робота не бойтесь оплошать.



## Упражнение № 3 — Изменение и улучшение проекта

### Задача и цель

**Задача** — Следуя алгоритму проектирования технических конструкций, оснастите своего робота ловушкой.

**Цель** — Повторить этапы проектирования технической конструкции и оснащения робота, участвующего в соревновании, системой сбора предметов.

### Обзор

При проектировании и конструировании робота для соревнования зачастую бывает полезно поочерёдно уделять внимание какой-то одной системе. Так вы сможете довести эту систему до оптимальных рабочих показателей, прежде чем перейти к следующей системе. Но по ходу проектирования, сборки, испытания и анализа некой системы важно также помнить о том, как с ней будут взаимодействовать другие системы робота.

К этому времени вы уже должны будете спроектировать и сконструировать мусороуборочного робота, умеющего передвигать предметы, толкая их перед собой. Но в соревновании по очистке мусорной свалки суметь затолкнуть ядовитые отходы, изображаемые мячами для гольфа, на накопительные участки — лишь половина дела. Вторая половина — собрать пригодный для переработки мусор, который изображают шарики для настольного тенниса. Для этого роботу понадобится ловушка — система для сбора и накопления предметов. В ходе этого упражнения вы вновь повторите этапы проектирования технических конструкций, чтобы спроектировать, собрать и испытать ловушку для шариков от настольного тенниса.

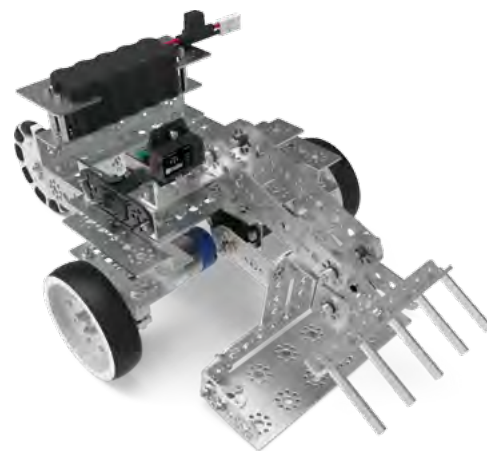
### Требующие учёта обстоятельства

Перед проектированием и конструированием ловушки примите во внимание такие факторы:

- Ловушки бывают разные. Есть пассивные, то есть им не нужны электродвигатели, а есть активные, в которых электродвигатели вращают конвейеры, двигают уборочные рукояти и вращают уборочные колёса.
- Загляните на сайт [Pitsco.com/CompetitionToTheMAX](https://Pitsco.com/CompetitionToTheMAX) и ознакомьтесь с примерами роботов серии TETRIX MAX с разными видами ловушек.
- Активные ловушки могут представлять собой довольно сложные системы с несколькими подсистемами. Когда будете разрабатывать проект, держите в уме число имеющихся сервоприводов и электродвигателей. Проектируя активную ловушку, можно создать эффективную систему подбора шариков для настольного тенниса, но при этом ухудшить способность робота выполнять повороты и маневрировать.

### Методика и порядок действий

1. Внося изменения в проект мусороуборочного робота с целью оснастить его ловушкой, следуйте алгоритму проектирования технических конструкций. Начните с критического анализа новых технических условий для этого конкурсного задания.
2. Исследуйте возможные решения и разные виды ловушек. Для сбора информации воспользуйтесь всеми подручными ресурсами. Не забудьте, что можно зайти на сайт [Pitsco.com/CompetitionToTheMAX](https://Pitsco.com/CompetitionToTheMAX) и посмотреть на примеры роботов из конструктора TETRIX MAX, сделанных для соревнований. По ходу исследования записывайте всё, что относится к делу, в свой технический журнал.
3. всей командой придумывайте разнообразные конструкции ловушек, которыми можно оснастить робота. Обдумайте как активные, так и пассивные конструкции ловушек. Следуйте правилам мозгового штурма, изложенным на странице 6. Запишите выдвинутые вашей командой идеи в свой технический журнал.
4. Из этого списка выберите идею или набор идей, которые кажутся вам подходящими для робота. Принимая решения, помните о критериях и ограничениях задания.
5. Спланируйте действия по оснащению робота придуманной ловушкой.



Заберите для переработки.

Передвиньте толкаем на накопительный участок.

## Технические характеристики робота

Ваш робот:

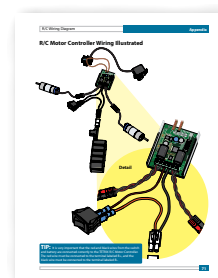
- Должен иметь систему для толкания мячей для гольфа на накопительный участок.
- Должен иметь устройство для сбора шариков для настольного тенниса.
- Должен быть собран только из деталей конструктора TETRIX.
- Должен уметь разворачиваться во всех направлениях и двигаться передним и задним ходом.
- Должен управляться радиоаппаратурой, входящей в ваш набор.
- В его составе не должно быть никаких дополнительных деталей, кроме тех, что входят в предоставленный вам набор.
- Его габариты не должны превышать 40 см в ширину, 50 см в длину и 40 см в высоту.
- В нём нельзя использовать никакие погнутые или испорченные детали TETRIX.

Спроектируйте эту новую конструктивную особенность робота в своём техническом журнале. Для начала нарисуйте эскизы — это поможет наглядно представить, как система ловушки будет взаимодействовать с другими системами вашего робота. Опишите методику установки ловушки на вашего робота. Составьте перечень главных составных частей, которые понадобятся для оснащения робота ловушкой.

6. Теперь пора собрать ловушку по придуманной вами конструктивной схеме. По ходу конструирования вносите в проект и конструкцию необходимые изменения. В ходе сборки обновляйте проект робота в техническом журнале. Видоизменяя проект и прототип робота, помните о критериях и ограничениях.
7. Если понадобится, оснастите своего робота всей нужной электропроводкой. За справкой по электромонтажу для своего робота обращайтесь в приложение к этому пособию.
8. Включите робота и контроллер. Если не поймёте, как подсоединить контроллер к роботу, см. *Руководство по сборке управляемых моделей*, прилагаемое к вашему конструктору TETRIX. Вы должны уметь управлять своим роботом.
9. Теперь всё готово для испытания робота. Начните с разработки испытания, как делали это в предыдущем упражнении. Запишите порядок и методику испытаний в свой технический журнал. Решите, какого рода данные надо собрать. При необходимости создайте в своём техническом журнале схемы и таблицы, чтобы упорядочить данные.
10. Есть два метода испытаний робота. Для мелких функциональных испытаний робота подойдёт рабочая станция или другой испытательный участок, назначенный учителем. Для серьёзных эксплуатационных испытаний придётся вывести робота на соревновательную арену.

А. Для испытания за пределами арены:

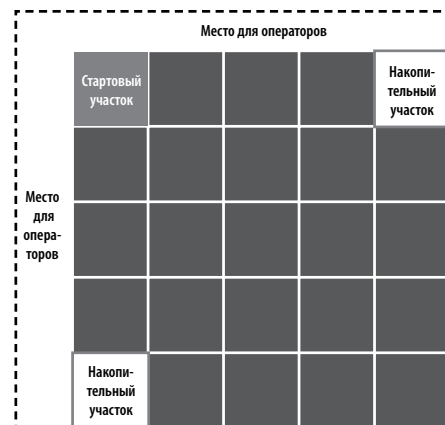
- i. Проследите, чтобы на испытательном участке не было препятствий и помех.
- ii. Поставьте робота на пол в отведённом для испытания участке.
- iii. Включите робота и контроллер.
- iv. Проведите испытание согласно методике, описанной в вашем техническом журнале. Собранные во время испытания данные запишите в свой технический журнал.
- v. По завершении испытания выключите робота и контроллер.



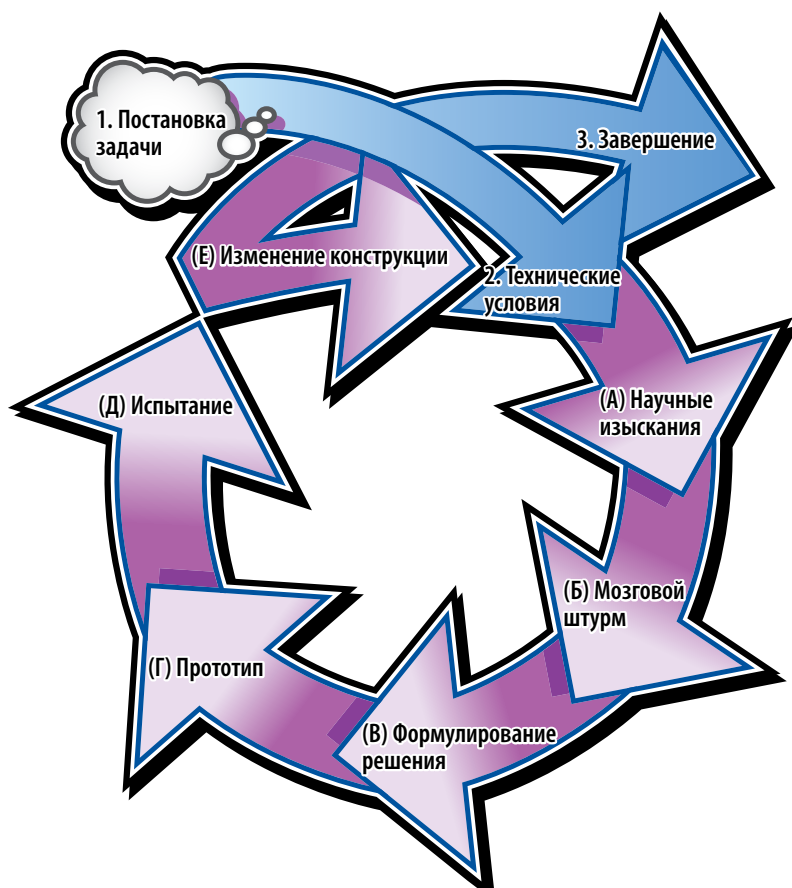
Страница 71

В. Для испытания в пределах арены:

- i. Для испытания робота на арене вам, возможно, придётся подождать своей очереди — это зависит размера вашего класса и числа учеников в нём. Когда подойдёт ваша очередь, у вас будет три минуты на проведение испытаний на соревновательной арене.
- ii. Подготовьте арену к испытаниям. Если нужно, разбросайте по арене мячи для гольфа и шарики для настольного тенниса.
- iii. Поставьте на арену робота. Включите робота и контроллер.
- iv. Проведите испытание согласно методике, которую вы описали в техническом журнале. Собранные во время испытания данные запишите в свой технический журнал.
- v. По истечении ваших трёх минут на арене выключите робота и контроллер. Уберите робота с арены.
- vi. Уберитесь на соревновательной арене и подготовьте её для следующей группы.



11. Проанализируйте данные собранные в ходе испытаний. Что они говорят о рабочих показателях робота? Сделайте все возможные для вас выводы и запишите ход и результаты анализа в технический журнал.
12. Продолжите проектирование технической конструкции, внося изменения и улучшения в придуманную конструктивную схему своего робота. Повторите все этапы цикла, чтобы определить, спроектировать и сконструировать робота улучшенной конструкции. Улучшая робота, помните о критериях и ограничениях конкурсного задания.



## Соревнование № 1 — Очистка мусорной свалки

### Обзор

У компании "Пуп земли" есть большая свалка хозяйственных отходов. Раньше работавшие здесь люди плохо следили за тем, какой именно мусор привозили на свалку. Теперь их задача — исправить эту ошибку, а именно: переместить часть токсичных материалов отдельно на особые накопительные участки, продолжая собирать пригодные для переработки материалы. Из-за ядовитости мусора и газов, выделяемых свалкой, людям заниматься этим опасно. "Пуп земли" хочет, чтобы вы создали робота-мусорщика, который будет перемещать токсичный мусор в специальную сборную ёмкость, попутно собирая мусор, пригодный для переработки.

Во время этого соревнования мячи для гольфа будут изображать токсичный мусор, а шарики для настольного тенниса — мусор, пригодный для переработки. Вам надо будет придумать и собрать робота, который толкает мячи для гольфа на накопительный участок, при этом собирая шарики для настольного тенниса в свою внутреннюю ловушку.

Отделяя токсичный мусор от годного для переработки, ваш робот должен избежать загрязнения. Загрязнение происходит тогда, когда предназначенный для переработки материал перемещают на участок с токсичными отходами, то есть, когда шарик для настольного тенниса попадает на накопительный участок или мяч для гольфа оказывается в ловушке робота.

Это не прямое противостояние, поэтому каждый раз на территории свалки (на соревновательной арене) будет находиться только один робот. Каждой команде даётся 3 минуты, за которые она должна набрать как можно больше баллов. Баллы подсчитываются по истечении 3-минутного периода и записываются в официальную таблицу подсчёта баллов. Засчитанный мяч должен остаться на зачётном участке до истечения трёх минут — только тогда за него начисляются баллы.

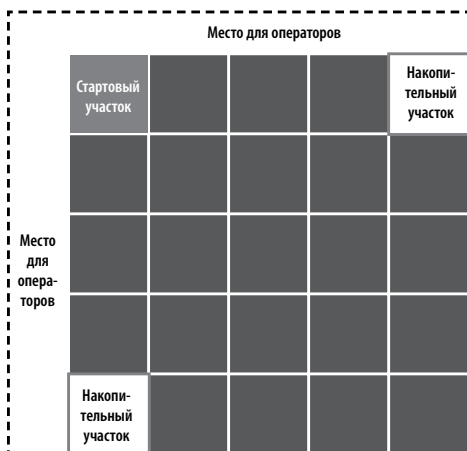
### Соревновательная арена

Соревновательная арена в виде мусорной свалки может иметь разные размеры — это зависит от условий в классе. Она должна представлять собой квадратную или прямоугольную площадку, огороженную по краям стенками. Один угол арены послужит стартовым участком. Ещё два угла станут участками сбора и накопления мячей для гольфа.

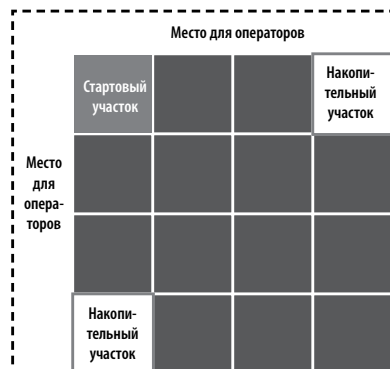
С двух сторон у края арены необходимо оставить место для операторов. Именно там вы будете стоять, управляя своим роботом во время соревнования. За образец соревновательной арены возьмите эти два схематичных вида сверху.



Соревновательное поле 3 x 3 м



Соревновательное поле 2,44 x 2,44 м



## **Правила и учёт баллов**

### **Правила соревнований**

- Командам разрешается использовать только детали из их конструктора TETRIX. Никакие дополнительные детали и узлы не допускаются.
- Гнуть или портить детали из конструктора TETRIX запрещено.
- Робот не должен быть больше 40 см в ширину, 50 см в длину и 40 см в высоту.
- Робот должен начать движение с назначенного стартового участка.
- Каждой команде даётся 3 минуты, за которые она должна набрать как можно больше баллов.
- По истечении заданного времени робот должен замереть. Любое намеренное движение робота по вине оператора по истечении 3-минутного отрезка времени наказывается снятием баллов по усмотрению распорядителя соревнования.
- Во время соревнования операторы должны оставаться в отведённых им местах за пределами арены.
- Если робот в ходе соревнования выйдет из строя, ваше выступление завершится.
- Чтобы получить баллы за сбор шариков для настольного тенниса, робот должен вернуться на исходный участок до истечения отведённого времени.
- Баллы подсчитываются по истечении 3-минутного отрезка времени или после выхода робота из строя.
- Распорядитель соревнования выступит в роли конечного арбитра по всем вопросам, касающимся учёта баллов и соблюдения правил честного состязания.

### **Правила начисления баллов:**

- за каждый мяч для гольфа, загнанный в накопитель: 10 баллов
- за каждый шарик для настольного тенниса: 15 баллов; баллы присуждаются только при условии, что робот по истечении времени окажется на стартовом участке

### **Правила начисления штрафных баллов:**

- за каждый шарик/мяч за пределами арены: -10 баллов
- за каждый повреждённый шарик/мяч: -20 баллов
- По -10 баллов за загрязняющий предмет (шарик или мячик, доставленный не по назначению: шарик для настольного тенниса, попавший в накопитель, или мячик для гольфа, оказавшийся в ловушке робота)

## **Дополнительные сведения**

### **Советы по проведению соревнований**

- Проектируя, конструируя и испытывая своего робота, следуйте алгоритму проектирования технических конструкций.
- Записывайте в технический журнал все результаты мозгового штурма, варианты проектов, процессов, решений, роли членов команды, ежедневные заметки и прочее.

## Раздел № 2 — Восполнение потребностей промышленности

### Отправные положения

Техника и технологии разрабатываются исключительно для восполнения нужд и потребностей человека. Обычно эти нужды или потребности обусловлены особенностями культуры и жизненного уклада разных стран, групп, компаний и даже отдельных людей. Так что именно общество даёт толчок и выступает движущей силой разработки и применения разных технологий и технических средств.

Человек, к примеру, часто бывает озабочен созданием для себя удобства и уюта. Многим людям начинает казаться, что удобнее заказать продовольствие и прочие товары по интернету и получить заказ там, где захочется, а не идти в магазин самим. Это вызвало бурный рост отраслей, связанных со складированием и доставкой таких товаров потребителям. Спрос на подобную услугу заставил некоторые компании изобрести новую технологию, улучшающую удовлетворение этой потребности. Однако спрос на услугу такого рода характерен для определённых социальных групп, социальных слоёв или стран. Многие другие предпочитают пойти на местный рынок, самостоятельно выбрать что-то из съестного и вещей, и напрямую пообщаться с продавцами этих товаров. Этим людям нужны разные технологии.

Итак, появление и развитие технологии диктуется особенностями общественного и культурного уклада, но спрос на определённую технологию сильно зависит от других факторов. Например: хорошо ли некая компания рекламирует новое техническое решение? Могут ли потребители позволить себе приобрести эту техническую новинку? Считается ли эта техническая новинка модной? Какого рода система поддержки потребителей внедрена? Отвечает ли эта техническая новинка стандартам на эксплуатационный ресурс? Эти и другие факторы влияют на то, какой приём окажут технологии потребители с разным социальным и культурным укладом.

Иногда факторы, определяющие приемлемость некой конкретной технологии, вступают в прямое противоречие друг с другом. Необходимо сопоставить и учесть положительные и отрицательные последствия использования некой технологии. Вернёмся к примеру складирования и доставки товаров для интернет-покупателей. Чтобы сделать свою работу рациональнее и прибыльнее, некоторые складские компании переходят на использование роботизированной техники, которая заполняет складские стеллажи и забирает с них товары, комплектуя сделанные покупателями заказы. У роботов выше скорость, больше грузоподъёмность, рациональнее траектория перемещения по складу, чем у людей, выполняющих ту же работу. Это положительные факторы: они увеличивают число довольных потребителей и умножают прибыль складской компании. С другой стороны, из-за роботов люди лишаются работы. Сокращение рабочих мест усугубляет проблему безработицы. Чем выше безработица, тем ниже качество жизни. Ухудшение качества жизни ослабляет экономику и людям становится труднее делать покупки через интернет. Уменьшение числа покупок через интернет означает уменьшение прибыли складских компаний, использующих роботов.

### Научные исследования и опытно-конструкторские разработки

Чтобы уравновесить положительные и отрицательные стороны разработки и развития технических средств, технологические компании ведут **научные исследования и опытно-конструкторские разработки**, сокращённо НИОКР. НИОКР — это разработка технических систем и подготовка их к выводу на рынок, где потребители, возможно, захотят их купить и использовать. Многие думают, что между НИОКР и проектированием технических конструкций нет никакой разницы. На самом деле проектирование технических конструкций лишь небольшая часть всего процесса НИОКР.



Технические изделия всегда рождаются в виде идеи. Если идея оказывается совершенно новой, она считается **изобретением**. Изобретения труднее пройти через НИОКР — ведь там столько всего неизвестного. Но изобретение может окупиться сторицей, если правильно пройти все этапы НИОКР, особенно если изобретение запатентовать, чтобы другие компании не смогли позаимствовать идею. Впрочем, большинство идей в мире техники и технологий отталкивается от предыдущих изделий или услуг. Такого рода идеи считаются **новшествами**. Все новшества основаны на некоем изобретении. Многие новшества возникают, когда технология, имеющая одно назначение, используется в другой области или по другому назначению.

Чтобы определить, можно ли реализовать техническую идею или извлечь из неё прибыль, компания должна сначала провести исследования. Ведь нельзя решить задачу, прежде не разобравшись в ней. Вообще, исследования бывают двух видов: фундаментальные и прикладные. Фундаментальные исследования не преследуют никакой чётко заданной цели, кроме обретения понимания и получения знаний. Большинство научных исследований считается фундаментальным. Напротив, прикладные исследования направлены на поиск ответа на некий конкретный вопрос, решение некой задачи или достижение некой цели. Ведя исследования, технологические компании обычно руководствуются такими целями, как удовлетворение некой потребности, оказание некой услуги, изготовление какого-то товара или получение прибыли. Из-за своих мотивов они, как правило, занимаются прикладными исследованиями.

Для НИОКР обычно привлекают экспертов нескольких типов. Разумеется, в их число входят инженеры и технологи, разрабатывающие и конструирующие изделие или услугу. Но иногда определить, как та или иная идея товара соотносится с окружающим миром, помогают учёные. Математики разрабатывают математические модели и составляют прогнозы относительно успеха того или иного товара. Маркетологи определяют, есть ли желание приобрести товар или услугу, или потребность в них, и у кого. Экономисты и бухгалтеры высчитывают, окажется ли товар прибыльным для компании. Рекламодатели решают, как распустить молву о товаре или услуге. Вот лишь немногие из тех специалистов, которые могли бы приложить руку к НИОКР.

По мере перехода от исследования идей к опытно-конструкторской разработке они превращаются в конструктивные схемы. Коллектив специалистов, работающих над проектом, использует метод оптимизации, чтобы прийти к наилучшему проектному решению. **Оптимизация** — это постоянная оценка конструктивной схемы изделия с учётом проектных критериев и ограничений с последующим улучшением изделия. Оптимизация в ходе НИОКР никогда не прекращается. Иногда постоянная проверка и оценка конструктивных схем требует доработки, доработки или даже полной переделки конструктивных схем. Это нормально, потому что цель — получить самую лучшую из возможных конструкций.

После оптимизации конструктивной схемы разрабатывается прототип.

**Прототип** — это рабочая модель придуманной конструкции. Прототипы могут быть грубыми моделями уменьшенного размера или отлаженными и безукоризненными вариантами окончательного изделия в натуральную величину. Прототипы делают для того, чтобы испытать физические характеристики изделий, получить данные наблюдений и сравнить их с критериями и ограничениями.

НИОКР невозможно вести без налаженного обмена информацией. Обмен информацией может быть многообразным. Прежде всего, важно отражать в документах порядок действий, наблюдений, результаты испытаний и изменения в конструкции. Правильное ведение записей помогает, когда приходит время рассказать об идеях, конструктивных схемах, прототипах и окончательных технических решениях специалистам. Обмениваться информацией можно в самых разных формах: письменные проектные задания, скетчи и чертежи, выполненные в программном приложении CADD, лабораторные отчёты о научных опытах, внутриколлективные совещания, мультимедийные показы прототипов и устные доклады для руководителей корпораций и акционеров.



По мере продолжения НИОКР прототипы превращаются в законченные изделия и поступают в продажу, где их покупают потребители. Однако НИОКР не прекращается никогда. Потребители зачастую используют изделие не только с нарушением правил эксплуатации, но и не по назначению. Порой изделие используется в невообразимых условиях и сочетаниях с другими изделиями для удовлетворения потребностей и прихотей потребителей. Важно понять, как потребители используют изделие, и восполняет ли оно их специфические потребности. Сбор таких данных и соответствующая доработка изделия помогает улучшить его и привести в соответствие с меняющимися желаниями потребителей.

### Советы по представлению докладов

Как уже было отмечено, обмен информацией — это главное для НИОКР. Один из типичных способов донесения информации до аудитории — выступление с докладом. Выступление с речью на людях может вызвать испуг. Кому-то бывает боязно выступать при большом стечении народа. Может возникнуть страх сделать ошибку, сказать что-то не так, или того хуже, вообще не знать, что сказать. Но репетиции и подготовка способны отчасти смягчить остроту переживаний из-за доклада. Вот памятка из нескольких простых советов по представлению доклада.

- Составьте план доклада. Не забудьте добавить введение, тезисы и заключительное обобщение. Во время доклада используйте пункты плана, как вехи своего выступления. Говорите, не глядя в написанный текст.
- Разумно подбирайте слова. Чтобы донести свою мысль, используйте точные и уместные средства языка.
- Не забывайте о регламенте выступления. Пусть у слушателей лучше останется желание узнать больше, чем ощущение, что вам пора бы закончить.
- Сделайте это по-своему. Используйте свои достоинства и личностные особенности, чтобы завладеть вниманием ваших слушателей. В мире нет другого такого человека, как вы. Воспользуйтесь этим.
- Завладейте вниманием слушателей: смотрите им в глаза и жестикулируйте.
- Чтобы усилить впечатление от доклада, продумайте, как использовать зрительно-звуковые эффекты и наглядность. Изображения и фильмы иногда понятнее, чем любые ваши слова. Говорите как можно меньше, прибегая только к ключевым словам и фразам.
- Внешний вид должен быть приятным и профессиональным. Аудитория должна считать, что вы выглядите представительно и пристойно. Это не значит, что вы всегда должны одеваться, словно напоказ. Например, выступающая с докладом команда могла бы облачиться в футболки единого фасона и так заявить о своём единодушии и сплочённости.
- Подготовьтесь. Перед докладом отрепетируйте всё, что собираетесь сказать. Держите под рукой все наглядные пособия и реквизит.
- Старайтесь свести на нет употребление звуковых заполнителей вроде "эээ" и "ааа". Мы часто прибегаем к ним, когда наша речь опережает мысль. Такие заполнители нужны, чтобы мозг успел подстроиться. Обойтись без них можно, снизив темп речи и говоря с расстановкой.
- Говорите чётко и с подъёмом. Соразмеряйте пыл своей речи с аудиторией. Чтобы не потерять внимание слушателей, избегайте монотонности в голосе и разнообразьте темп речи.
- Ведите себя уверенно, но скромно. Если не знаете ответа на вопрос, ничего не выдумывайте. Вместо этого смиренно скажите: "Замечательный вопрос. Пока ответа на него не знаю, но я разберусь и отвечу вам позже".



**Совет:** Следуйте этим советам, когда будете выступать с докладом, относящимся к соревнованию из раздела № 2.



## Цель соревнования

В этом разделе главным будет проектирование и конструирование роботизированного оборудования для складской отрасли, которая стремится к равновесию между плюсами от возросшей производительности труда и минусами от потери людьми рабочих мест. Складскими роботами, которых вы создаёте, будут управлять люди, то есть компании-заказчики всё же вынуждены будут нанять кого-то, кто будет водить и обслуживать роботов, предназначенных для подъёма тяжестей.

Роботы в этом разделе устроены чуть сложнее и им понадобится немного больше систем, чем роботам из раздела № 1. В этом разделе происходит переход от роботов, которые сгребают и собирают предметы, к роботам, которые захватывают, поднимают, сбрасывают и штабелируют предметы. Помимо захватных и подъёмных устройств, главным в этом разделе станет управляемость и манёвренность вашего робота

Никаких объяснений ожидаемого порядка сборки не будет. Вместо этого, в пособии для вас изложены простейшие методики и сведения, опираясь на которые вы строите собственного робота, реализуя собственные творческие замыслы и находки. Чтобы понять, как создать складского робота, вы выполните три упражнения. Чтобы спроектировать и собрать своего робота в ходе упражнений, вам понадобятся приёмы НИОКР, а также алгоритм проектирования технических конструкций и системный подход. Прежде чем приступить к упражнениям в этом разделе, вам, пожалуй, стоит прочесть всё о соревновании складских роботов, начав со страницы 32.

В соревнование, предусмотренное для этого раздела, входит и выступление с докладом. Вам надо будет подготовить и представить устный доклад — полученные за это баллы станут частью вашей заключительной оценки. Ваш доклад будет оценивать учитель с помощью рубрики "Доклад" на странице 35. Чтобы знать, как вас будут оценивать и за что присуждать баллы, полезно целиком прочесть рубрику "Доклад".



## Упражнение № 4 — Системы навигации

### Задача и цель

**Задача** — Изучая и испытывая разные системы навигации для складского робота, руководствуйтесь принципами ведения НИОКР. Чтобы робот был проворным и лёгким в управлении, в системах навигации следует использовать силовые передачи, зубчатые передачи и настройки контроллера электродвигателя.

**Цель** — Изучить разные стили управления передвижением роботов. Нахождение хорошего способа точного управления роботом поможет вам подготовиться к предстоящему соревнованию складских роботов.

### Обзор

Одна из главных составляющих успеха в робототехнических соревнованиях — умение управлять движениями робота. Есть несколько способов управлять передвижением робота по прямой и его поворотами. Исходя из вида соревнований и задач, которые предстоит решать роботу, какие-то способы будут работать лучше других. В ходе этого упражнения вы исследуете разные способы управления вашим роботом. Затем вы перейдете к экспериментам и соберёте модели роботов, оснащённые теми средствами управления, которые представляются вам наиболее перспективными в предстоящем соревновании складских роботов.

### Требуемые учёта обстоятельств

Перед проектированием и конструированием системы навигации для складского робота примите во внимание такие факторы:

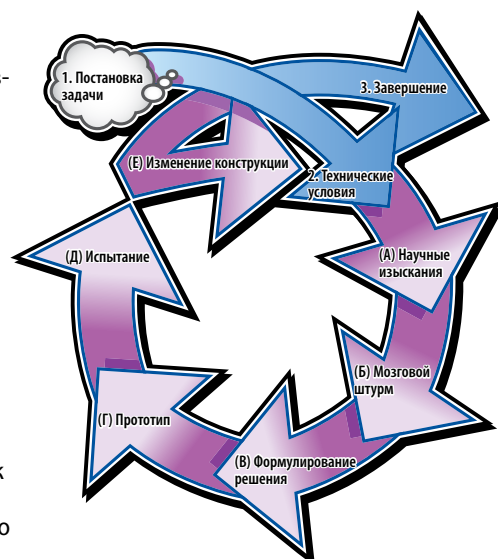
- Обдумайте, как именно будет двигаться ваш робот. В этом разделе самое главное — его проворство и управляемость.
- Важна сила сцепления с опорной поверхностью. Схема распределения массы вашего робота существенно скажется на силе сцепления колёс с опорной поверхностью.
- Разрабатывая конструктивную схему, держите в уме число находящихся в вашем распоряжении сервоприводов и электродвигателей.
- Конструируйте с прицелом на заключительное соревнование. В итоге ваш робот получит захватные и подъёмные устройства. Работая над системой навигации не забудьте оставить место для этих приспособлений.
- Проектируя конструктивную схему и собирая по ней модель примените то, чему научились в ходе предыдущих упражнений и соревнований.
- Ведите подробные записи в своём техническом журнале.

### Методика и порядок действий

Цель упражнения — опробовать разные способы управления складским роботом. Сосредоточиться следует на исследованиях и экспериментах, а не на окончательной конструктивной схеме вашего робота. К концу упражнения вы должны хорошо понимать, какая система навигации подойдёт для вашего складского робота.

1. Просмотрите технические условия для масштабируемой модели складского робота к соревнованию № 2 на странице 33. Вам надо будет спроектировать систему навигации с учётом этих технических условий.
2. Проведите исследования систем навигации роботов. В приложении к этому пособию множество полезных сведений о настройках контроллеров электродвигателей роботов, о приводах и зубчатых передачах, — всё это по-своему поможет вам управлять роботом. В ходе исследования используйте приложение и любые подручные ресурсы. На роботов из конструктора TETRIX MAX, оснащённых некоторыми из этих систем навигации, можно посмотреть на сайте [Pitsco.com/CompetitionToTheMAX](http://Pitsco.com/CompetitionToTheMAX).

3. Выберите несколько разных способов управления маневрированием, которые будете испытывать. Выберите те способы, которые, по вашему мнению, обеспечат машин наилучшую подвижность и управляемость. Запишите эти идеи в свой технический журнал.
4. Для разработки одной из выбранных вами идей управления передвижением используйте алгоритм проектирования технических конструкций.
  - A. Если понадобится, изучите эту идею более пристально.
  - B. Устройте мозговой штурм, чтобы решить, как внедрить свою идею в модель складского робота. Не забывайте использовать уместные принципы мозгового штурма и записывать всю сопутствующую информацию в свой технический журнал.
  - C. Отметьте наиболее, по вашему мнению, многообещающие идеи и начните формулировать решение задачи по оснащению складского робота системой навигации. Обдумайте, как будут взаимодействовать шасси, система навигации и электрооборудование. Отрадите разработанную конструктивную схему своего робота в техническом журнале: нарисуйте эскизы, составьте описания, разработайте методику сборки и перечислите нужные материалы. Пока что не усложняйте придумываемые конструкции. И ещё: проектируя конструктивную схему, не забывайте о технических условиях.
  - D. Приступайте к созданию прототипа по разработанной конструктивной схеме. Соберите придуманную конструкцию из деталей конструктора TETRIX MAX. Начните с шасси, а затем добавьте составные части системы навигации. В конце установите составные части электрооборудования и бортовой электроцепи. Не забывайте оптимизировать свои конструкции во время сборки, по мере необходимости записывая изменения в планах в свой технический журнал. Проследите, чтобы прототип отвечал техническим условиям для масштабированной модели складского робота на странице 33.
  - E. Пока что вы будете испытывать систему навигации своего робота в месте, назначенном учителем. Возможно, это будет место вне соревновательной арены. Спросите у учителя, где лучше испытать вашего робота.
  - F. Повторяя этапы проектирования технических конструкций, доработайте робота, чтобы улучшить его эксплуатационные показатели.
    - i. Проследите, чтобы на испытательном участке не было препятствий и помех.
    - ii. Поставьте робота на пол в отведённом для испытания участке.
    - iii. Включите робота и контроллер.
    - iv. Проведите испытания робота, по ходу испытания собирая результаты измерений и наблюдений. Запишите результаты измерений и наблюдений в свой технический журнал.
    - v. По завершении испытания выключите робота и контроллер.



**Совет:** При необходимости см. электромонтажную схему аппаратуры Д/У на странице 71.

- Г. Иногда, сколько изменений и улучшений ни вноси в первоначально придуманную схему, толку не будет, потому что сама идея негодная и работать как нужно не будет. Если почувствуете, что возникшие у вас идеи по обеспечению управления передвижением робота, не имеют перспективы, прекратите их разработку и перейдите к другой идее. Не тратьте время впустую на то, что работать не будет.
5. Когда почувствуете, что довольны своей системой навигации, прекратите разработку этой идеи. Запишите заключительные наблюдения и выводы в свой технический журнал.
  6. Выберите в пункте № 3 ещё одну идею для системы навигации. Для превращения этой идеи в прототип повторите алгоритм проектирования технических конструкций в пункте 3. Оптимизируйте и дорабатывайте придуманную конструктивную схему в ходе мозгового штурма, разработки проекта, создания и испытания прототипа. Исправно записывайте в свой технический журнал всё, что касается придуманных конструктивных схем, испытаний, наблюдений и результатов. Продолжайте исследование, проектирование и создание прототипа разных систем навигации всё оставшееся для этого время. Когда время, отведённое на это упражнение, истечёт, вы должны иметь довольно сносное представление о системе навигации, которой оснастите своего робота.



**Рулевой привод Аккермана  
(с поворотными кулаками  
и трапецией)**



**Рулевой привод  
с шарнирно-  
сочленённым шасси**

## Упражнение № 5 — Подъёмные системы

### Задача и цель

**Задача** — Придумайте и соберите робота, способного поднять предельно тяжёлый груз как можно выше.

**Цель** — Изучить разные подъёмные механизмы и получить опыт их проектирования и конструирования. Упражнение подготовит вас к соревнованию, завершающему этот раздел.

### Обзор

Во многих робототехнических соревнованиях проверяется способность роботов поднимать грузы и переносить их в другие места. В предстоящем соревновании грузоподъёмные роботы будут подбирать грузы и укладывать их либо в штабеля, либо в ящики. Вы начнёте упражнение с исследования подъёмных механизмов. Окончив исследование, вы должны спроектировать, сконструировать, испытать и доработать подъёмный механизм для прототипа своего складского робота. Главная тема упражнения — подъёмный механизм. Грузозахватными механизмами вы займётесь в упражнении № 6.



### Требующие учёта обстоятельства

- Определите, насколько тяжёлый груз вы будете поднимать. Помните, что следует учесть массу подъёмной системы, в том числе всяких подъёмных приспособлений вроде грузозахватных устройств или ковшей.
- Проследите, чтобы робот занимал устойчивое положение. Как правило, подъёмные системы сильно утяжеляют переднюю часть робота. Эту дополнительную нагрузку можно уравновесить, если сдвинуть батарею и другие более тяжёлые элементы к тыльной части робота.
- Разрабатывая конструктивную схему, держите в уме число находящихся в вашем распоряжении сервоприводов и электродвигателей. Как правило, подъёмные системы делят их с системой навигации.
- Конструируйте с прицелом на заключительное соревнование.
- Проектируя конструктивную схему и собирая по ней модель примените то, чему научились в ходе предыдущих упражнений и соревнований.
- Ведите подробные записи в своём техническом журнале.

### Методика и порядок действий

Цель этого упражнения — спроектировать, сконструировать и оптимизировать робота, обладающего максимально возможной грузоподъёмностью. Пока что не переживайте по поводу способности устройства хватать или сжимать предметы. Грузозахватные устройства и ковши будут рассмотрены в следующем упражнении. А сейчас сосредоточьтесь на рычаге, которым будете поднимать груз.

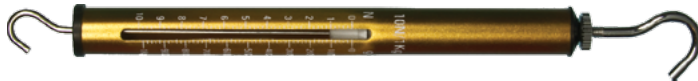
1. Просмотрите технические условия для масштабируемой модели складского робота к соревнованию № 2 на странице 33. Вам надо будет спроектировать роботизированный подъёмник с учётом этих технических условий.
2. Для создания прототипа грузоподъёмного робота вам понадобится алгоритм проектирования технических конструкций и системный подход.
  - А. Проведите исследования робототехнических грузоподъёмных механизмов. Начните с информации из приложения в этом пособии. Другую информацию черпайте из любых доступных вам источников. Некоторые примеры роботов из конструктора TETRIX MAX с разными грузоподъёмными системами можно посмотреть на [Pitsco.com/CompetitionToTheMAX](http://Pitsco.com/CompetitionToTheMAX).
  - В. Рассмотрите системы навигации, которые вы исследовали и разрабатывали в упражнении № 4. Выберите систему навигации, которой оснастите своего робота, предназначенного для соревнования.
  - С. Устройте мозговой штурм, чтобы придумать, как объединить грузоподъёмную систему с системой навигации. Подумайте, как объединить сервоприводы, подъёмные стрелы и шестерни.

Запишите появившиеся в результате мозгового штурма идеи в свой технический журнал.

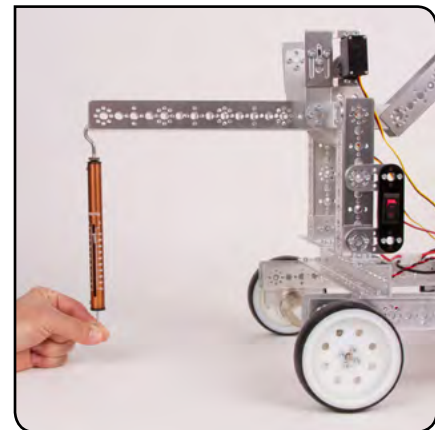
- **Примечание:** Вы, вероятно, уже сконструировали несколько разных грузоподъёмных систем, собирая из конструктора TETRIX MAX мини-робота и робота с поворотным захватом из *Руководства по сборке управляемых моделей*. В ходе мозгового штурма обращайтесь к извлечённому из этого опыту.
- D. На основе этих идей сформулируйте реализуемое техническое решение. Изобразите конструктивную схему своего решения в техническом журнале. Проектируя систему навигации, продумайте, как она будет взаимодействовать с грузоподъёмной системой и её подсистемами. Необходимо добиться слаженности этих систем и подсистем, иначе от робота не будет проку.
- E. Превратите придуманные конструктивные схемы в рабочий прототип. Для постройки спроектированного робота используйте детали из конструктора TETRIX MAX. Помните, что единственные детали не из конструктора TETRIX MAX, которые разрешено использовать, — это резиновые ленты (резиновые кольца). В ходе конструирования, возможно, придётся что-то изменить в конструктивных схемах. Это составная часть процесса оптимизации.
- F. Перед завершением сборки не забудьте придумать, как прикрепить крюк пружинных весов к оконечности вашей грузоподъёмной системы.



**Совет:** Проектируя и собирая конструкцию, не забывайте о технических условиях для соревнования № 2.



3. Пружинные весы понадобятся, чтобы измерить тяговое усилие грузоподъёмной системы. Учитель уже должен был подготовить в классе испытательные станции. Чтобы воспользоваться испытательной станцией, придётся, вероятно, подождать своей очереди. Проведите время ожидания с пользой. Когда будете готовы испытать своего робота, действуйте в следующем порядке.
- A. Перенесите робота на испытательную станцию.
  - B. Прикрепите пружинные весы одним концом к грузоподъёмной системе. Другой конец пружинных весов удерживайте неподвижно.
  - C. Включите робота и пульт управления.
  - D. При помощи пульта управления введите в действие грузоподъёмную систему. Она должна потянуть пружину весов вверх.
  - E. Измерьте поднятый вес по пружинным весам. Если ваш робот выйдет за предел 1000 г, подсоедините вторые пружинные весы и объедините результаты замеров веса на обоих весах. Запишите данные в свой технический журнал.
  - F. Измерьте максимальную высоту, на которую робот поднимает груз. Запишите результаты измерения в свой технический журнал.
  - G. Проведите другие наблюдения за поведением робота во время подъёма грузов. Например, способен ли робот удерживать груз на максимальной высоте или со временем она снижается? Сохраняет ли робот устойчивость во время подъёма? Мешает ли подъём груза работать другим системам робота? Насколько высоко способен робот поднять груз?
4. Внесите в конструкцию робота изменения, чтобы улучшить его грузоподъёмные характеристики. Находите идеи методом мозгового штурма, формулируйте решения, создавайте прототипы конструктивных схем и испытывайте новые модификации, повторяя этапы проектирования технических конструкций. Помните: грузоподъёмная система должна поднимать как можно более тяжёлый груз как можно выше.



## Упражнение № 6 — Захваты и ковши

### Задача и цель

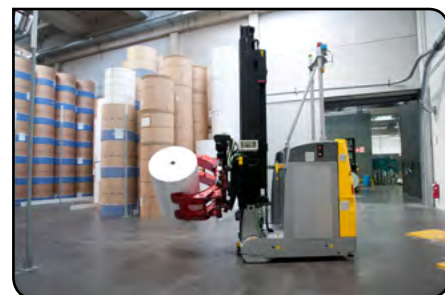
**Задача** — Оснастить робота, собранного в ходе предыдущего упражнения, грузозахватным устройством.

**Цель** — Изучить разные системы захвата грузов и оснастить робота грузозахватным механизмом. Полученный здесь опыт подготовит вас к соревнованию складских роботов, завершающему этот раздел.

### Обзор

В предыдущем упражнении вы спроектировали и собрали грузоподъемный механизм для штабелирования или подъема предметов и складывания их в накопительные ёмкости. Ваш грузоподъемный робот не добьётся особого успеха, если не будет оснащён приспособлением для захвата груза, который вы хотите поднять. Для предстоящего соревнования складских роботов понадобится робот, способный подхватывать предметы вроде катушек и стаканчиков, поднимать их и либо сбрасывать в ёмкости, либо складывать их стопками. Это будет имитацией того, как складские роботы забирают товарные единицы со стеллажей и либо складывают их в отгрузочную тару, либо штабелируют их, рационально заполняя пространство хранилища.

В ходе упражнения вам предстоит изучить разные захватные механизмы, решить, каким именно механизмом оснастить своего робота, затем придумать, собрать, испытать и доработать грузозахватную систему, которая бы предельно увеличила производительность вашего складского робота.



### Требующие учёта обстоятельства

- Определите, с какой стороны лучше брать за поднимаемый груз. Подхватывать его лучше снизу, сбоку или сверху?
- Спроектируйте захватную систему специально для грузов, которые будете подбирать во время соревнования.
- Обеспечьте устойчивость робота при подъёме грузов. Утяжеление передней части можно уравновесить, если сдвинуть батарею и другие более тяжёлые элементы к тыльной части робота. Так центр масс вашего робота окажется посередине.
- Разрабатывая конструктивную схему, держите в уме число находящихся в вашем распоряжении сервоприводов и электродвигателей.
- Отрадите в конструктивной схеме идеи и опыт, появившиеся в ходе предыдущих упражнений и соревнований.
- Ведите подробные записи в своём техническом журнале.

### Методика и порядок действий

В ходе этого упражнения вы навесите на конец грузоподъемной системы захватное приспособление.

1. Просмотрите технические условия для масштабируемой модели складского робота к соревнованию № 2 на странице 33. Вам надо будет спроектировать систему захвата для своего робота с учётом этих технических условий.
2. Для разработки грузозахватной системы и оснащения ею прототипа складского робота вам понадобится алгоритм проектирования технических конструкций и системный подход.
  - A. Проведите исследования робототехнических захватных систем. Начните с информации из приложения в этом пособии. Другую информацию черпайте из любых доступных вам источников. Некоторые примеры роботов из конструктора TETRIS MAX с разными захватными системами можно посмотреть на [Pitsco.com/CompetitionToTheMAX](http://Pitsco.com/CompetitionToTheMAX).
  - B. Устройте мозговой штурм, чтобы придумать, как объединить захватную систему с грузоподъемной. Запишите появившиеся в результате мозгового штурма идеи в свой технический журнал.

- C. На основе этих идей сформулируйте реализуемое техническое решение. Изобразите конструктивную схему своего решения в техническом журнале. Проектируя захватную систему, продумайте, как она будет взаимодействовать с грузоподъёмной системой и системой навигации. Необходимо добиться слаженности этих систем и подсистем, иначе от робота не будет проку.
  - D. Превратите придуманные конструктивные схемы в рабочий прототип. Для изготовления и прикрепления грузозахватной системы к роботу используйте детали конструктора TETRIX MAX. В ходе конструирования, возможно, придётся что-то изменить в конструктивных схемах. Оптимизируйте собираемую конструкцию робота, записывая вносимые в проект изменения в свой технический журнал.
3. Есть два способа испытать взаимодействие грузозахватной системы с грузоподъёмной системой и системой навигации вашего робота. Для мелких функциональных испытаний робота подойдёт рабочая станция или другой испытательный участок, назначенный учителем. Для серьёзных эксплуатационных испытаний придётся вывести робота на соревновательную арену.

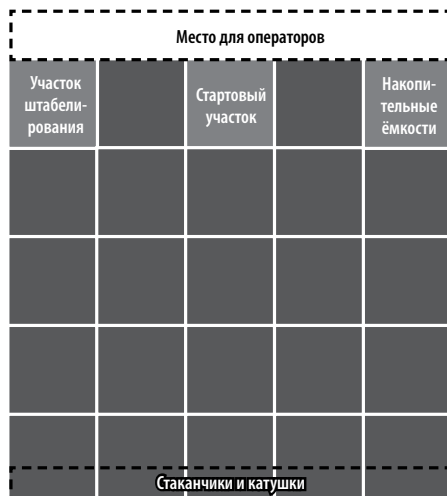
**Совет:** Проектируя и собирая конструкцию, не забывайте о технических условиях для соревнования № 2.

Для испытания за пределами арены:

- A. Проследите, чтобы на испытательном участке не было препятствий и помех.
- B. Поставьте робота на пол в отведённом для испытания участке.
- C. Включите робота и контроллер.
- D. Проведите испытания. Собранные во время испытаний результаты измерений и наблюдений запишите в свой технический журнал.
- E. По завершении испытания выключите робота и контроллер.

Для испытания в пределах арены:

- A. Для испытания робота на арене вам, возможно, придётся подождать своей очереди — это зависит размера вашего класса и числа учеников в нём. Когда подойдёт ваша очередь, у вас будет три минуты на проведение испытаний на соревновательной арене.
- B. Подготовьте арену к испытаниям. Если понадобится, поместите стаканчики, катушки и накопительные ёмкости в исходные места.





- C. Поставьте на арену робота. Включите робота и контроллер.
  - D. Проведите испытания. Собранные во время испытаний результаты измерений и наблюдений запишите в свой технический журнал.
  - E. По истечении ваших трёх минут на арене выключите робота и контроллер. Уберите робота с арены.
  - F. Уберитесь на арене и подготовьте её для следующей группы.
4. Проанализируйте данные собранные в ходе испытаний. Что они говорят о рабочих показателях робота? Сделайте все возможные для вас выводы и запишите ход и результаты анализа в технический журнал.
5. Продолжите проектирование технической конструкции, внося изменения и улучшения в придуманную конструктивную схему своего робота. Сосредоточьтесь на системе навигации, грузоподъёмной системе, грузозахватной системе и на их взаимодействии, без которого робот работать не будет. Повторите все этапы цикла, чтобы определить, спроектировать и сконструировать робота улучшенной конструкции. Улучшая робота, помните о критериях и ограничениях конкурсного задания.

## Соревнование № 2 — Складские роботы

### Обзор

"Впрок" — крупная обслуживающая компания, сдающая в аренду склады и оказывающая услуги по распространению товаров небольшим специализированным и начинающим компаниям. У этих компаний зачастую нет денег, рабочей силы, оборудования или площадей для хранения и распределения выпускаемых товаров. За ежемесячное вознаграждение компания "Впрок" предлагает результативное оказание этих услуг, уменьшающее расходы клиентов, так что обслуживаемые компании могут заняться тем, в чём сильны — выпуском своих товаров.

Для своего развития компании "Впрок" необходимо найти новые действенные способы оказания услуг и уменьшения издержек. "Впрок" хочет внедрить автоматизированную систему складирования, комплектования заказов, упаковки и доставки товаров, в которой всю работу выполняют роботы. Руководители компании "Впрок" уверены, что у применения для этой цели роботов есть несколько плюсов. Снимать товары со стеллажей у роботов получается лучше, так как они вычисляют наилучшую траекторию движения по складу. А ещё у них больше грузоподъёмность и они реже заезжают в проходы между складскими стеллажами. Можно сделать так, чтобы роботы дотягивались до более высоких стеллажей и увеличить высоту складирования товаров. Используя роботов на территории склада люди, создают для себя безопасные условия, так как им уже не нужно поднимать грузы или бояться попасть под вилочные автопогрузчики.

Для самостоятельного создания роботов у специалистов компании "Впрок" нет знаний в области робототехники. Поэтому они хотят привлечь к созданию складских роботов других разработчиков робототехники. Компания "Впрок" распространила запрос на предложения подрядчиков (ЗПП) в ЗПП включены критерии и ограничения проекта.

В этом соревновании вы со своей командой должны откликнуться на ЗПП и побороться против других команд за контракт, предлагаемый компанией "Впрок". Соревнование состоит из двух важных частей:

1. Соревнование роботов. Вы должны спроектировать и сконструировать прототип складского робота по техническим условиям, заданным компанией "Впрок". Вы должны продемонстрировать эффективность вашего прототипа на соревновательной арене с макетом склада. Выступление робота будет оценивать представитель компании "Впрок". На этом этапе соревнования ваш робот должен показать, что умеет подбирать предметы, перемещать их в другие места и либо складывать в накопительные ёмкости, либо штабелировать на соответствующем участке.
2. Доклад. Вам также предстоит выступить с докладом перед представителем компании "Впрок". Вы должны представить доклад, наглядно описывающий ход разработки придуманного вашей командой робота, обосновать принятые решения, доказать, почему компании "Впрок" стоит вложить средства в вашего робота. Баллы за представление доклада будут начисляться в рубрике "Доклад".

Результаты обоих этапов соревнования будут суммированы и дадут окончательный счёт. Команда с наибольшим числом баллов победит в соревновании и получит контракт от компании "Впрок". ЗПП компании "Впрок" можно найти в разделе с дополнительной информацией к этому соревнованию.



**Совет:** Не разбирайте робота по завершении этого соревнования. Он пригодится для начала упражнения № 7 из раздела № 3.

## Соревновательная арена

Складом станет квадратная или прямоугольная игровая площадка с бортиками по периметру, назначение которых — препятствовать выезду робота за её пределы. Точный размер арены будет зависеть от условий в вашем классе. С одной стороны арены необходимо устроить стартовый участок. Один угол арены надо отвести под участок штабелирования. В другом углу арены — поставить две накопительные ёмкости (одна низкая, другая высокая). На стороне, лежащей напротив стартового участка, необходимо расставить 20 стаканчиков с катушками поверх каждого стаканчика.

С наружной стороны, у края, ближайшего к стартовому участку, будет место для операторов. Именно там вы будете стоять, управляя своим роботом во время соревнования. За образец соревновательной арены возьмите эти два схематичных вида сверху.

## Правила и учёт баллов

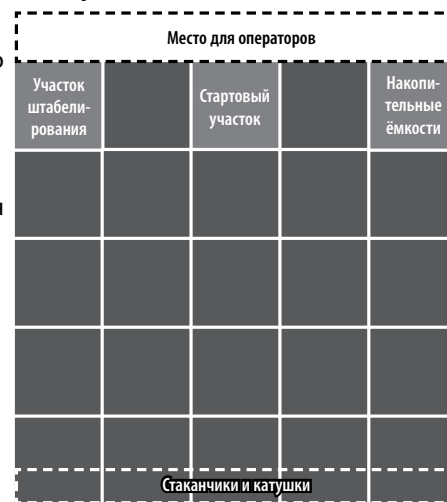
### Правила соревнований

- Робот, предназначенный для соревнования, должен отвечать следующим условиям:
  - Командам разрешается использовать только детали из их конструктора TETRIX MAX.
  - Командам запрещено гнуть или портить детали из конструктора TETRIX.
  - Единственные детали не из конструктора TETRIX, которые разрешено использовать, — это резиновые ленты/кольца.
  - Робот не должен быть больше 40 см в ширину, 55 см в длину и 40 см в высоту.
  - Робот должен управляться дистанционно.
  - Робот должен быть в состоянии захватывать предметы, поднимать их на высоту не менее восьми сантиметров, перемещать с места на место и оставлять в заданных местах.
- Робот должен начать движение с назначенного стартового участка.
- Каждой команде даётся 3 минуты, за которые она должна набрать как можно больше баллов.
- Во время соревнования операторы должны оставаться в отведённых им местах за пределами арены.
- Если робот в ходе соревнования выйдет из строя, команда завершает свою попытку участия в соревновании.
- Баллы подсчитываются по истечении 3-минутного отрезка времени или после выхода робота из строя.
- Распорядитель соревнования выступит в роли конечного арбитра по всем вопросам, касающимся учёта баллов и соблюдения правил честного состязания.

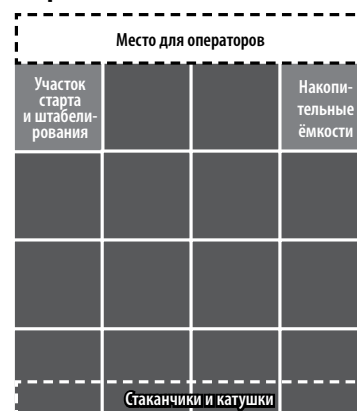
### Правила выступления с докладом

- Доклад вашей команды должен длиться 3–4 минуты.
- Доклад вашей команды должен:
  - Быть изложен применительно к задаче создания складского робота для компании "Впрок". Объяснить, как вы собираетесь превратить свою модель в складского робота натуральной величины.
  - Объяснить, как в вашей команде шло проектирование и конструирование.
  - Вкратце сообщить, с какими решениями и компромиссами пришлось согласиться.
  - Указать достоинства конструкции вашего робота.
  - Объяснить, почему компании "Впрок" следует отдать предпочтение проектному предложению вашей команды.

Соревновательное поле 3 x 3 м



Соревновательное поле 2,44 x 2,44 м



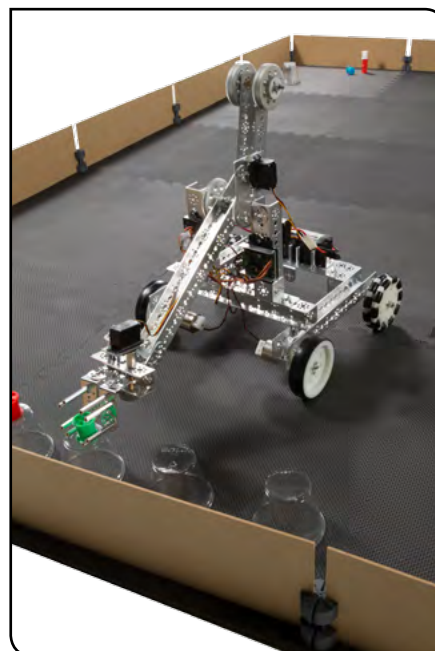
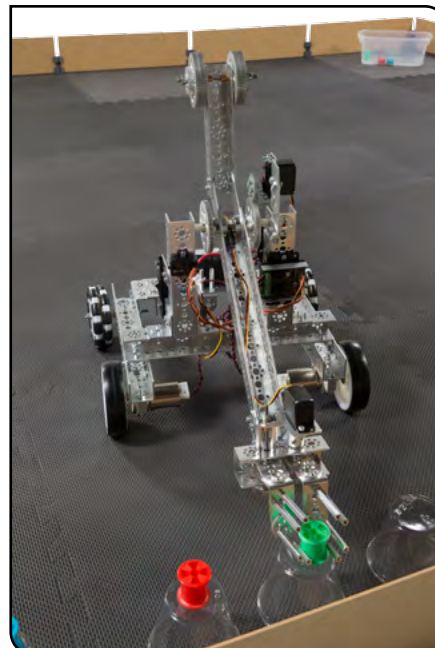
- Ваш доклад должен быть представлен профессионально.
  - Выбор слов должен быть зрелым и точным.
  - Захватите внимание слушателей и следите за выражением их лиц. Когда уместно, пользуйтесь жестикуляцией.
  - Говорите отчётливо и с нужной громкостью.
  - Избегайте использовать для связности речи и заполнения паузы слова-паразиты и нечленораздельные звуки вроде "э-э-э", "mmm", "типа", "короче" и т. п.
  - Оденьтесь подобающим образом.
  - Наглядно-графические средства должны усилить впечатление от доклада, а не отвлекать внимание слушателя.
- Постарайтесь подключить к докладу всех, кто есть в команде.
- После доклада некоторое время будет отведено на вопросы-ответы. На вопросы распорядителя соревнования давайте точные и исчерпывающие ответы, покажите знания и уверенное владение темой.

#### Правила начисления баллов:

- Баллы можно получить за укладку катушек в ящики:
  - за каждую катушку, положенную в низкую ёмкость: 5 баллов
  - за каждую катушку, положенную в высокую ёмкость: 10 баллов
- Баллы можно получить за укладку катушек одной на другую на участке штабелирования.
  - за стопку из двух катушек: 20 баллов
  - за стопку из трёх катушек: 30 баллов
  - за стопку из четырёх катушек: 50 баллов
  - за стопку из пяти катушек (максимум): 100 баллов
- Стаканчики можно ставить один на другой на участке штабелирования после снятия с их верха катушек:
  - за каждый стаканчик в стопке из 2 и более стаканчиков: 10 баллов
- Доклады команд будет оценивать по таблице из рубрики "Доклад" распорядитель соревнования.
  - За представление доклада присуждается максимум 100 баллов.

#### Правила начисления штрафных баллов:

- за каждую упавшую и оставленную на соревновательном настиле катушку снимается 10 баллов; за катушки, остающиеся на стаканчиках, штрафные баллы не начисляются.
- за каждый стаканчик, оставленный на соревновательном настиле за пределами стартового участка или участка штабелирования, будет снято 10 баллов; то есть, за стаканчики, не сдвинутые во время соревнования, штрафные баллы не начисляются.



**Дополнительные сведения****Рубрика "Доклад"**

Категория	4	3	2	1	0	Максимальное число баллов
<b>Время</b>	Уложились в заданное время 3–4 минуты.	Превысили заданное время меньше чем на 1 минуту.	Превысили заданное время на 1–2 минуты.	Превысили заданное время больше чем на 2 минуты.	Доклад не представлен.	20 баллов (Баллы x 5)
<b>Содержание</b>	<p>Доклад отвечает всем критериям:</p> <ul style="list-style-type: none"> <li>Представлен в соответствии с условиями ЗПП компании "Впрок".</li> <li>Ход проектирования описан достаточно подробно.</li> <li>Есть описание решений и компромиссов.</li> <li>Освещены достоинства конструкции робота.</li> </ul>	<p>Доклад отвечает трём из указанных критериев:</p> <ul style="list-style-type: none"> <li>Представлен в соответствии с условиями ЗПП компании "Впрок".</li> <li>Ход проектирования описан достаточно подробно.</li> <li>Есть описание решений и компромиссов.</li> <li>Освещены достоинства конструкции робота.</li> </ul>	<p>Доклад отвечает двум из указанных критериев:</p> <ul style="list-style-type: none"> <li>Представлен в соответствии с условиями ЗПП компании "Впрок".</li> <li>Ход проектирования описан достаточно подробно.</li> <li>Есть описание решений и компромиссов.</li> <li>Освещены достоинства конструкции робота.</li> </ul>	<p>Доклад отвечает одному из указанных критериев:</p> <ul style="list-style-type: none"> <li>Представлен в соответствии с условиями ЗПП компании "Впрок".</li> <li>Ход проектирования описан достаточно подробно.</li> <li>Есть описание решений и компромиссов.</li> <li>Освещены достоинства конструкции робота.</li> </ul>	<p>Доклад не отвечает ни одному из указанных критериев:</p> <ul style="list-style-type: none"> <li>Представлен в соответствии с условиями ЗПП компании "Впрок".</li> <li>Ход проектирования описан достаточно подробно.</li> <li>Есть описание решений и компромиссов.</li> <li>Освещены достоинства конструкции робота.</li> </ul>	32 балла (Баллы x 8)
<b>Профессионализм</b>	<p>Доклад отвечает всем критериям:</p> <ul style="list-style-type: none"> <li>Докладчики завладели вниманием аудитории (наблюдали за выражением лиц слушателей, использовали жестикюляцию и т. п.).</li> <li>Выбор слов был зрелым и точным.</li> <li>Речь была ясной (темп, громкость, чёткость артикуляции).</li> <li>Докладчики были вежливыми, опрятными и раскованными.</li> <li>Одежда была надлежащей.</li> </ul>	<p>Доклад отвечает четырём из указанных критериев:</p> <ul style="list-style-type: none"> <li>Докладчики завладели вниманием аудитории (наблюдали за выражением лиц слушателей, использовали жестикюляцию и т. п.).</li> <li>Выбор слов был зрелым и точным.</li> <li>Речь была ясной (темп, громкость, чёткость артикуляции).</li> <li>Докладчики были вежливыми, опрятными и раскованными.</li> <li>Одежда была надлежащей.</li> </ul>	<p>Доклад отвечает двум трём из указанных критериев:</p> <ul style="list-style-type: none"> <li>Докладчики завладели вниманием аудитории (наблюдали за выражением лиц слушателей, использовали жестикюляцию и т. п.).</li> <li>Выбор слов был зрелым и точным.</li> <li>Речь была ясной (темп, громкость, чёткость артикуляции).</li> <li>Докладчики были вежливыми, опрятными и раскованными.</li> <li>Одежда была надлежащей.</li> </ul>	<p>Доклад отвечает одному из указанных критериев:</p> <ul style="list-style-type: none"> <li>Докладчики завладели вниманием аудитории (наблюдали за выражением лиц слушателей, использовали жестикюляцию и т. п.).</li> <li>Выбор слов был зрелым и точным.</li> <li>Речь была ясной (темп, громкость, чёткость артикуляции).</li> <li>Докладчики были вежливыми, опрятными и раскованными.</li> <li>Одежда была надлежащей.</li> </ul>	<p>Доклад не отвечает ни одному из указанных критериев:</p> <ul style="list-style-type: none"> <li>Докладчики завладели вниманием аудитории (наблюдали за выражением лиц слушателей, использовали жестикюляцию и т. п.).</li> <li>Выбор слов был зрелым и точным.</li> <li>Речь была ясной (темп, громкость, чёткость артикуляции).</li> <li>Докладчики были вежливыми, опрятными и раскованными.</li> <li>Одежда была надлежащей.</li> </ul>	20 баллов (Баллы x 5)
<b>Команда</b>	В представлении доклада участвовала вся команда.	В представлении доклада участвовало большинство команды.	В представлении доклада участвовала некоторая часть команды.	Доклад был представлен одним членом команды.	Доклад не представлен.	20 баллов (Баллы x 5)
<b>Вопросы</b>	Ответы были развернутыми и точными, свидетельствовали об уверенности и хорошем знании предмета.	Ответы были по большей части развернутыми и точными, свидетельствовали о некоторой неуверенности или нехватке знаний по предмету.	Ответы свидетельствовали о спутанности мыслей и нехватке знаний по предмету.	Ответы были невнятными и не раскрыли тему заданных вопросов.	Ответов не было.	8 баллов (Баллы x 2)

**Советы по проведению соревнований**

- Полностью прочтите ЗПП компании "Впрок". Убедитесь, что полностью понимаете, как провести соревнование, и что для этого требуется.
- Есть несколько способов набрать баллы в этом соревновании. Сосредоточьтесь на сильных сторонах вашего робота.
- Проектируя, конструируя и испытывая своего робота, следуйте алгоритму проектирования технических конструкций.
- Записывайте в технический журнал все результаты мозгового штурма, варианты проектов, процессов, решений, роли членов команды, ежедневные заметки и прочее.
- Отведите некоторое время на планирование своего доклада. Просмотрите рубрику "Доклад", чтобы знать, как будет оцениваться ваше выступление с докладом. Отрепетируйте своё выступление с докладом.



## Запрос на предложения подрядчиков

### Отправные положения

Компания "Впрок" ищет творчески мыслящих разработчиков с новаторским подходом к роботостроению и намерена заключить с ними партнёрский договор на разработку инновационных складских роботов, управляемых дистанционно.

Компания "Впрок" оказывает услуги по складированию и доставке товаров. Своё призвание мы видим в том, чтобы точно и рационально складировать, сортировать, учитывать и отгружать для других компаний запасы их товаров, делая это безопасно и в срок. Для быстрого снабжения заказчиков у нас есть склады в нескольких городах и во многих странах.

### Объём и содержание работ

В настоящее время мы ищем возможность оптимизировать свои услуги и работы. Вот одна из идей по уменьшению издержек: использовать на складских площадях дистанционно управляемых роботов, способных взять находящийся на складе товар, а затем безопасно и эффективно доставить его в зону упаковки в тару. У этих роботов будут и другие обязанности, в частности сортировка товаров и заполнение ими стеллажей.

Ввиду необходимости выбрать поставщика из определённого числа роботостроительных компаний, мы предлагаем заинтересованным коллективам разработчиков спроектировать и сконструировать модели роботов в уменьшенном масштабе по нижеприведённым техническим условиям. Коллективы разработчиков должны будут продемонстрировать свои модели в ходе условного соревнования складских роботов.

Однако компанию "Впрок" интересуют не просто конструктивные схемы и модели роботов. Мы ценим партнёрские отношения с людьми. Поэтому нам важно знать о вашем коллективе больше. Перед робототехническим соревнованием вы получите возможность представить свои идеи и проектные решения в виде доклада представителю компании "Впрок". Для вас это шанс проявить свои творческие способности, новаторство, трудовую этику, умение решать задачи и прочие важные, ценимые нами свойства.

### Мы преследуем следующие цели:

- Провести честное и объективное соревнование робототехнических моделей.
- Выявить команды, явно обладающие необходимыми качествами: творческим подходом к делу, новаторским мышлением и умением решать задачи.
- Предоставить одной отобранной нами команде контракт на разработку складских роботов в натуральную величину.

### Модель складского робота в уменьшенном масштабе должна отвечать следующим техническим условиям:

- Командам разрешается использовать только детали из их конструктора TETRIX MAX.
- Командам запрещено гнуть или портить детали из конструктора TETRIX.
- Единственные детали не из конструктора TETRIX, которые разрешено использовать, — это резиновые ленты/кольца.
- Робот не должен быть больше 40 см в ширину, 55 см в длину и 40 см в высоту.
- Робот должен управляться дистанционно.
- Робот должен быть в состоянии захватывать предметы, поднимать их на высоту не менее восьми сантиметров, перемещать с места на место и оставлять в заданных местах.

### Условия допуска к партнёрству

Если вы считаете, что именно ваша команда должна стать нашим партнёром, просим как можно скорее откликнуться на этот запрос. С вопросами и заявками на дополнительную информацию обращайтесь к местному представителю компании "Впрок".

## Раздел № 3 — Автономные спасательные роботы

### Отправные положения

В 2011 году в прибрежных водах Японии случилось землетрясение силой 8,9 балла. Землетрясение и последовавшее за ним цунами причинили страшное опустошение этому островному государству. Когда землетрясение стихло и вода, принесённая цунами, схлынула, множество домов и прочих сооружений оказалось в настолько шатком состоянии, что заходить в них людям было опасно. Для оценки ущерба и поиска выживших внутрь отправили спасательных роботов. С 2011 года, после землетрясения в Японии, роботов используют по всему миру для оценки причиняемого природными катастрофами ущерба, для поиска и спасения уцелевших.

В этом разделе главным будет конструирование придуманного вами спасательного робота. В отличие от других разделов, требовавших постройки дистанционно управляемых роботов, в этом разделе робот будет автономным. Автономными называют роботов, действующих без вмешательства человека. При помощи датчиков они собирают информацию об окружающей обстановке и благодаря управляющей программе сами решают, как реагировать на изменение обстановки. Итак, для создания автономного спасательного робота вашей конструкции мало просто спроектировать и собрать крепкого робота, надо ещё и запрограммировать его так, чтобы он считывал окружающую обстановку и реагировал на складывающиеся вокруг условия.

### Разомкнутые и замкнутые системы

Применительно к прежним соревнованиям в этом пособии говорилось о том, что технические системы состоят из подсистем, а системы взаимодействуют с подсистемами при решении задач и удовлетворении нужд и потребностей человека. Все системы и подсистемы обрабатывают по определённому алгоритму входные элементы, преобразуя их в выходные элементы. По способу регулирования системы и подсистемы можно разделить на две главные группы: разомкнутые системы (системы без обратной связи, системы с внешним регулированием) и замкнутые системы (системы с обратной связью, саморегулирующиеся системы).

Элементы, поступающие в **разомкнутые системы** извне, после обработки преобразуются в желаемые выходные элементы. Затем выходные элементы покидают систему. Соответствие выходных элементов требованиям должен оценить человек. Если требования нарушены, человеку приходится вмешиваться, менять алгоритмы обработки и отправлять входные элементы на повторную обработку, чтобы получить на выходе желаемые результаты. Как это происходит, прекрасно видно на примере управляемых роботов, которые вы конструировали при прохождении предыдущих разделов. Робот выполняет задание по указке человека. Управлять роботом приходится джойстиком на пульте управления. Робот выполняет задания только после получения информации от вас. Если выходной элемент не совпадает с намеченной целью, приходится менять входные элементы или алгоритмы их обработки, чтобы добиться искомого результата.



### Разомкнутые системы управления







Как видно из этого примера, чем сложнее устроена система, тем больше в ней уровней управления и контуров обратной связи, подсказывающих всей системе, как правильно делать своё дело. Ещё раз посмотрите на схему контура обратной связи спасательного робота. Подумайте, как вёл бы себя робот, не поступай нужное напряжение по электрической цепи в его системы. Сбой в работе всего одного контура обратной связи способен вывести из равновесия всю систему.

### Устранение неисправностей

В автономных роботах контуры управления и обратной связи возникают не сами собой. Их необходимо ввести в управляющую программу робота. Программирование автономного робота требует совершенного иного набора умений, помимо умения проектировать и конструировать робота. Программирование или написание управляющего кода требует способности мыслить логически и рассуждать при решении задач. Программисты должны быть невероятно внимательны к мелочам. Из-за пропуска одного знака в компьютерной программе вся работа может оказаться напрасной. Ещё программистам надо быть организованными. Аккуратность и рациональность программы способствует оптимизации робота и выполнению им заданий предусмотренным образом.

**Устранение неисправностей** — ещё одна важная составляющая в числе обязательных для хороших программистов умений. Как видно из предыдущего примера со спасательным роботом, системы автономных роботов могут быть довольно сложно устроены, поэтому программисты должны уметь вычленять в системах подсистемы и находить программные средства управления и контуры обратной связи, относящиеся к каждой из подсистем. Если та или иная подсистема работает не так, программисты должны найти раздел программного кода, который управляет этой подсистемой, выявить ошибки в коде и написать новый код, устраняющий проблему. При устранении неисправностей нередко приходится экспериментировать, чтобы найти верные параметры и настройки, позволяющие оптимизировать выполнение роботом заданий.

Вот памятка из нескольких простейших советов по отладке программного кода:

- Используйте комментарии — они определяют нужные секции и строки кода, и узнать назначение этих секций и строк.
- Проверьте синтаксис программы. Возможно, не хватает круглой или квадратной скобки или точки с запятой.
- Тестируйте небольшие секции кода поочерёдно, одну за другой. Это поможет отыскать нерабочую секцию.
- Исправляя код, начните с того, который, как вы уже знаете, работает. Вносите в него необходимые изменения поочерёдно, одно за другим.
- После каждого небольшого изменения, запускайте исполнение кода. Так вы узнаете, позволяет ли изменение добиться искомого результата, и как вносить дальнейшие изменения.
- Диагностировать проблемы вам помогут средства контроля и сообщения об ошибках.
- Если ошибку в коде найти не удаётся, немного отдохните и вернитесь к делу позже. Иногда после непродолжительного занятия чем-то ещё в мозгу возникают новые идеи и открываются новые перспективы.
- Не мешаает и чей-то свежий взгляд со стороны. Иногда, подолгу всматриваясь в один и тот же код, не замечаешь того, что очевидно окружающим. Попросите кого-то ещё помочь и высказать своё мнение.

### Цель соревнования

Для участия в третьем соревновании вы займётесь проектированием, конструированием и программированием автономного робота. Вам предстоит выполнить четыре упражнения, которые в конце подведут вас прямо к соревнованию автономных роботов. Прежде чем заняться упражнениями в этом разделе, вам, пожалуй, стоит прочесть всё о соревновании по очистке мусорной свалки, начав со страницы 66.

Для этого соревнования в вашего робота надо будет встроить датчики и контроллер PRIZM. С их помощью робот сможет решать, чем заниматься. Как и в случае с другими соревнованиями, никаких подсказок о рекомендуемом порядке сборки не будет. Вместо этого вам надо будет видоизменить робота из раздела № 2.

После выполнения упражнений из *Руководства по программированию контроллера PRIZM* у вас уже должен быть опыт программирования. Выполнять упражнения в этом разделе придётся, обращаясь к знаниям и опыту, почерпнутым при программировании прежних роботов. Если вы ещё не проштудировали *Руководство по программированию контроллера PRIZM*, обязательно сделайте это, прежде чем работать с этим разделом.

Перед завершением этого раздела вам также надо будет обязательно скачать и установить *ПО Arduino (IDE)*. Следуйте инструкциям в *Руководстве по программированию контроллера PRIZM*.



## Упражнение № 7 — Внесение изменений

### Задача и цель

**Задача** — Установить на робота из раздела № 2 контроллер PRIZM, датчик линии и ультразвуковой датчик.

**Цель** — Превратить радиоуправляемого робота из раздела № 2 в автономного робота, использующего контроллер PRIZM.

### Обзор

Автономные роботы действуют самостоятельно на основе данных об обстановке, собираемых ими для принятия решений о том, какие из полученных заданий выполнять, и когда. Но что делать, автономные роботы не знают, если их не запрограммировать. Это упражнение вы начнёте с роботом из раздела № 2. Чтобы сделать этого робота автономным, надо снять с него приёмник сигналов дистанционного управления и контроллер электродвигателей, заменив их контроллером PRIZM. Кроме того, к роботу надо будет прикрепить ультразвуковой датчик и датчик линии. Программированием робота вы займётесь в последующих упражнениях.

Если робот из раздела № 2 уже разобран, его надо заново собрать по записям в техническом журнале и только потом приступить к упражнению.

### Требующие учёта обстоятельства

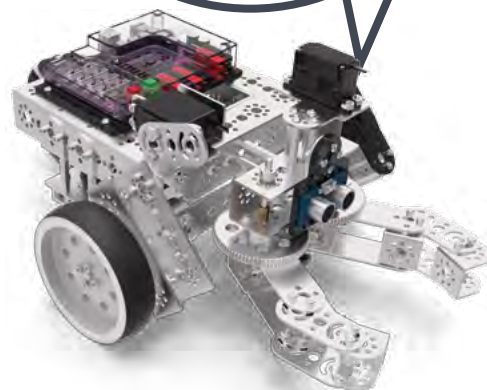
- Контроллер PRIZM крупнее радиоуправляемого контроллера электродвигателей. После замены радиоуправляемого контроллера электродвигателей контроллером PRIZM, возможно, понадобится установить дополнительные конструктивные элементы для его закрепления.
- Место установки и положение датчика линии, а также степень его приближенности к земле может сказаться на его способности обнаруживать линию. Если заметите, что робот не очень хорошо движется по линии, попробуйте опустить или поднять датчик.
- Общий свет в помещении также может повлиять на датчик линии, отбрасывая на землю тени от робота, людей или других объектов. Придайте датчику линии на роботе такое положение, чтобы изменения в общем свете минимально воздействовали на датчик.
- Высота установки ультразвукового датчика повлияет на его способность обнаруживать более низкие предметы. Прикрепляя датчик к роботу, учитывайте высоту и размер предметов, которые предстоит обнаружить.
- Ведите подробные записи в своём техническом журнале.

### Методика и порядок действий

Начните выполнять упражнение с роботом из раздела № 2.

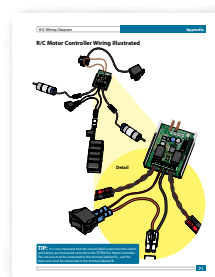
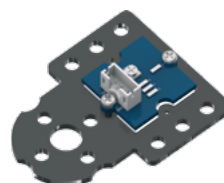
1. Отсоедините от робота все электрические узлы и детали.
  - A. Отсоедините батарею. Пока что снимите и отложите её в сторону.
  - B. Отсоедините от электродвигателей силовые кабели.
  - C. Отсоедините кабели радиоуправляемого контроллера электродвигателей и все кабели радиоуправляемого приёмника сигналов дистанционного управления.
  - D. Отсоедините от выключателя провода.
2. Снимите с робота беспроводной приёмник сигналов дистанционного управления.
3. Снимите с робота радиоуправляемый контроллер электродвигателей.
4. Установите на робота контроллер PRIZM. Если места не хватает, перекомпонуйте составные части робота или добавьте к его шасси конструктивные элементы. Устанавливая контроллер PRIZM, учитывайте следующее:
  - Установите контроллер PRIZM в том месте, где к нему будет удобно подсоединять все кабели электродвигателей постоянного тока, сервоприводов и датчиков.
  - Позаботьтесь об удобном доступе к кнопке пуска и кнопке сброса параметров.

Вложите  
в робота  
МОЗГ.

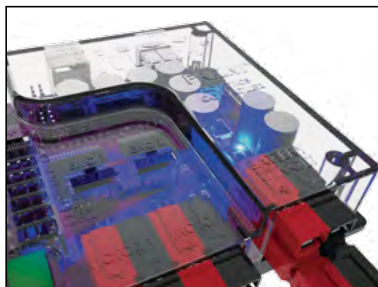
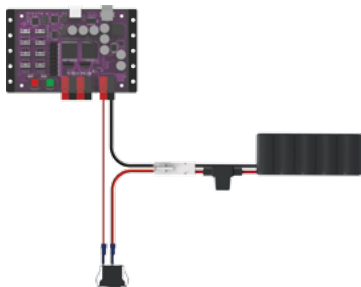


**Совет:** Уберите радиоуправляемый приёмник сигналов дистанционного управления и радиоуправляемый контроллер электродвигателей в сохранное место. Они вам больше не понадобятся.

- Разверните контроллер PRIZM так, чтобы кабель USB было легко подсоединить и отсоединить.
5. Прикрепите к роботу датчик линии. Если понадобится, оснастите шасси какими-нибудь конструктивными элементами для фиксации датчика. Прикрепляя датчик, учитывайте следующее:
    - Позаботьтесь, чтобы кабель датчика дотягивался до предусмотренных для него портов на контроллере PRIZM.
    - Датчик должен быть обращён вперёд и находиться на центральной оси робота. Проследите, чтобы датчик стоял перед механизмом, служащим для поворачивания робота.
    - На работу датчика влияет расстояние его подвески над грунтом. Постарайтесь установить датчик приблизительно в 2,5 см над полом.
  6. Прикрепите к роботу ультразвуковой датчик. Если понадобится, добавьте какие-нибудь конструктивные элементы для фиксации датчика. Прикрепляя датчик, учитывайте следующее:
    - Позаботьтесь, чтобы кабель датчика дотягивался до предусмотренных для него портов на контроллере PRIZM.
    - Установите в передней части робота датчик, развернутый чувствительным элементом вперёд.
    - Проследите, чтобы рабочие органы (ковши, грузозахватные и грузоподъёмные приспособления), особенно когда они с грузом, не мешали датчику.
  7. Подсоедините к контроллеру PRIZM электродвигатели постоянного тока при помощи разъёмов Powerpole.
  8. Вставьте в порты для сервоприводов на контроллере PRIZM все силовые кабели сервоприводов.
  9. В один из портов для цифровых датчиков на контроллере PRIZM вставьте кабель датчика линии. Порты для цифровых датчиков обозначены D2...D5.
  10. Вставьте кабель ультразвукового датчика в незанятый порт для цифровых датчиков (с обозначением D2...D5) на контроллере PRIZM.
  11. Установите на прежнее место батарею.
  12. Завершите монтаж электропроводки контроллера PRIZM, подсоединив его силовые кабели к выключателю и батарее. За справкой по электропроводке, предусмотренной для контроллера PRIZM, обращайтесь к приложению в этом пособии.
  13. Включите питание и убедитесь, что синий светодиод подачи питания на контроллере PRIZM горит. Если всё именно так, можно переходить к следующему этапу упражнения. Если синий светодиод подачи электропитания не горит, устраните неисправность в цепи источника электропитания:
    - Проверьте проводные соединения. Если понадобится помощь, см. электромонтажную схему в приложении или в разделе о подготовке контроллера в *Руководстве по программированию контроллера PRIZM*.
    - Проследите, чтобы батарея была полностью заряжена.



Страница 71



## Упражнение № 8 — Сочетание датчиков (повторение)

### Задача и цель

**Задача** — Запрограммировать робота на движение по линии и объезд препятствий.

**Цель** — Критически проанализировать некоторые команды и функции, управляющие автономным роботом. Многие из таких же команд и функций будут использованы для управления вашим роботом в соревновании спасательных роботов в конце этого раздела.

### Обзор

Поисково-спасательные роботы должны быть сообразительными. Они должны собирать информацию о внешних условиях, обрабатывать эту информацию и решать, что делать. Запрограммировать спасательного робота с учётом множества разных условий и обстоятельств, типичных для чрезвычайной ситуации, может быть непросто. Но не волнуйтесь — вы начнёте с чего-то уже знакомого.

Это упражнение станет повторением заключительного упражнения из *Руководства по программированию контроллера PRIZM*, которое вы уже делали. До выполнения этого упражнения на ваш компьютер необходимо установить *ПО Arduino (IDE)*, а также библиотеку Arduino для контроллера PRIZM. Как скачать, установить и настроить *ПО Arduino (IDE)* см. в разделе по подготовке ПО в *Руководстве по программированию контроллера PRIZM*.

Чтобы самостоятельно принимать правильные решения, робот должен знать, что происходит вокруг. В ходе этого упражнения вы запрограммируете робота на движение по линии, пока не обнаружится препятствие. Обнаружив препятствие, робот должен будет остановиться и ждать, чтобы препятствие убрали. Вы начнёте упражнение с запуска скетча, представленного в качестве примера в упражнении № 15 из *Руководства по программированию контроллера PRIZM*. Затем вам предстоит изменить скетч с учётом особенностей своего робота.

### Требующие учёта обстоятельства

- Выполняя упражнение, делайте заметки и ведите подробные записи в техническом журнале.
- Обратите внимание на то, как упражнение связано с соревнованием, завершающим этот раздел.
- При записи управляющего кода легко допустить мелкие ошибки, которые могут сделать всю программу бесполезной. Пусть товарищи по команде проверяют вашу работу ещё раз.
- Занимаясь отладкой управляющего кода, сверяйтесь с памяткой, упомянутой в обзорной части этого раздела на странице 39.
- При программировании в среде *ПО Arduino (IDE)* к наиболее распространённым относятся синтаксические ошибки вроде пропущенной точки с запятой в конце команды или не поставленной закрывающей круглой или квадратной скобки. Помните это, исправляя в своём коде синтаксические ошибки.

### Методика и порядок действий

#### Методика составления программного кода

1. Запустите на своём компьютере *ПО Arduino (IDE)*.
2. В меню файлов (File) выберите **Examples > TETRIS\_PRIZM > TaskBot\_Act15\_Combining\_Sensors**. В новом окне откроется пример скетча.
3. Для начала сохраните скетч под другим именем. В меню файлов (File) выберите **Save As** и дайте файлу новое имя.
4. Вы наверняка уже видели этот скетч, когда проходили *Руководство по программированию контроллера PRIZM*. Далее, изучите скетч и некоторые его особенности.



- A. В начале стоят два оператора, позволяющие использовать контроллер PRIZM. Оператор включения позволяет использовать в скетче библиотечные функции контроллера PRIZM. Оператор "prizm" **создаёт экземпляр** объекта PRIZM. Попросту говоря, это означает, что оператор готовит контроллер PRIZM к использованию.

```
#include <PRIZM.h>
PRIZM prizm;
```

- B. Далее идёт секция **void setup()**. Секция выполняется однократно в начале скетча. Оператор **prizm.PrizmBegin()**; запускает контроллер PRIZM. Следующий оператор запускает вращение электродвигателей постоянного тока в обратном направлении. Благодаря этому электродвигатели, повернутые в противоположные стороны (один влево, другой вправо), вращаются в одном направлении. Последняя строка задаёт скорость вращения сервоприводов.

```
void setup() {
  prizm.PrizmBegin();
  prizm.setMotorInvert(1,1);

  prizm.setServoSpeed(1,50);
}
```

- C. В основной цикл **loop()** входит условный оператор для считывания сигналов датчика линии. Попросту говоря, программа говорит: "Если датчик линии обнаружит линию, я поверну направо, остановив для этого правое колесо и вращая левое колесо вперёд с мощностью 30 %". Часть **else** в операторе означает, что линия не обнаружена, поэтому скетч повернёт робота влево, остановив левое колесо и вращая правое колесо вперёд с мощностью 30 процентов. Кроме того, скетч включает-выключает красный светодиод в зависимости от того, обнаружена линия или нет.

```
if(prizm.readLineSensor(3) == 1){
  prizm.setMotorPowers(30,125);
  prizm.setRedLED(HIGH);
}
else
{
  prizm.setMotorPowers(125,30);
  prizm.setRedLED(LOW);
}
```

За оператором **if/else** в скетче стоит цикл **while()**. Эта секция кода будет повторяться снова и снова до соблюдения некоего заданного условия. В данном случае условие таково: обнаружение ультразвуковым датчиком объекта на расстоянии менее 25 сантиметров. Если условие соблюдено, тогда скетч включает зелёный светодиод, сразу останавливает оба электродвигателя, задавая им обороты 125, и меняет положение вала сервопривода. Эти команды будут повторяться, пока объект не уберут или пока не истощится заряд батареи. Но если объект не обнаружен или если объект обнаружен, но находится на расстоянии больше 25 см, тогда эти команды исполняться не будут.



**Совет:** Хотите увидеть, как это работает? Тогда посмотрите серию наших видеоочерков про рабочее место роботостроителя, дополняющую Руководство по программированию контроллера PRIZM. Вся серия представлена по адресу [video.tetrixrobotics.com](http://video.tetrixrobotics.com), а также на канале Pitsco в YouTube.



**Совет:** В окне скетча библиотечные функции контроллера PRIZM меняют цвет, если напечатаны или введены без ошибок. Соответственно, если в их написании есть ошибка, то цвет не поменяется. Программное обеспечение в языке Arduino распознаёт функции контроллера PRIZM как ключевые слова и окрашивает их в оранжевый цвет, если не нарушены синтаксические правила.

```
while(prizm.readSonicSensorCM(4) < 25){
  prizm.setGreenLED(HIGH);
  prizm.setMotorPowers(125,125);
  prizm.setServoPosition(1,0);
}
```

За циклом **while()** следуют ещё две команды. Первая гасит зелёный светодиод, если он загорелся во время цикла **while()**. Вторая команда возвращает вал сервопривода в нейтральное положение, если это положение изменилось во время цикла **while()**.

```
prizm.setGreenLED(LOW);
prizm.setServoPosition(1,90);
```

Все эти операторы в составе основного цикла **loop()** будут повторяться всё время, пока исполняется скетч.

5. Помните, что этот скетч написан для вашего робота. Попробуйте что-то изменить в скетче так, чтобы это отразилось на действиях робота.
- A. В секции **void setup()** скетча есть команда, устанавливающая обороты сервопривода № 1 на 50 процентов.

```
prizm.setServoSpeed(1,50);
```

Прежде всего, следует учесть, что у робота может быть не один сервопривод. Если это так, тогда для второго сервопривода надо добавить ещё одну команду. Для этого следует скопировать уже имеющуюся команду и вставить копию прямо под первой командой. Проследите, чтобы команда была вставлена перед закрывающей скобкой секции **void setup()**.

```
void setup() {
  prizm.PrizmBegin();           // initialize PRIZM
  prizm.setMotorInvert(1,1);    // invert the direction of DC Motor 1
                                // to harmonize the direction of
                                // opposite facing drive motors
  prizm.setServoSpeed(1,50);    // set servo 1 speed to 50;
  prizm.setServoSpeed(2,50);    // set Servo 2 speed to 50
}
```

Проследите, чтобы порты, указанные в вашем скетче (в данном случае — для сервоприводов № 1 и № 2) соответствовали портам на контроллере PRIZM, к которым на самом деле подсоединены сервоприводы.

- B. Следующая секция кода, подлежащая изменению, находится внутри цикла **while()**. Внутри цикла вы задаёте валу 1-го сервопривода положение на 0 градусов. Создайте ещё одну команду, такую же, которая тоже задаёт валу 2-го сервопривода положение на 0 градусов. Полезно выработать у себя привычку следить за точностью комментариев к управляющему коду. В комментарии к положению вала первого сервопривода написано: "Поднять флажок обнаружения!" У вашего робота скорее всего нет флажка обнаружения. Измените этой комментарий так, чтобы было ясно, что должно произойти с роботом, когда положение вала сервопривода будет задано значение 0 градусов. То же самое сделайте для второго сервопривода.

```
while(prizm.readSonicSensorCM(4) < 25){ // object is in path; loop here until cleared
  prizm.setGreenLED(HIGH);           // turn on green LED
  prizm.setMotorPowers(125,125);     // stop; obstacle detected
  prizm.setServoPosition(1,0);       // close the gripper
  prizm.setServoPosition(2,0);       // lower the gripper
}
```

- С. В конце скетча стоит команда, которая возвращает 1-й сервопривод в положение 90. Добавьте под первой командой ещё одну, чтобы вернуть и 2-й сервопривод в положение 90. Добавляйте или меняйте комментарии к этим командам по мере необходимости.

```
prizm.setServoPosition(1,90);  
prizm.setServoPosition(2,90);
```

- D. Возможно, придётся изменить порт датчика линии в операторе **if()** и порт ультразвукового датчика в цикле **while()** — это зависит от того, в какие порты вы вставили датчики. Определите, в какие порты контроллера PRIZM вставлены датчик линии и ультразвуковой датчик. Измените команды в скетче так, чтобы они соответствовали фактическим портам.
- E. Если заметите, что в скетче надо изменить для вашего робота что-то ещё, внесите изменения сразу.
6. Проверьте синтаксис скетча инструментом контроля (Verify) в *ПО Arduino (IDE)*. Этот инструмент проверяет, насколько код точен и удобочитаем для контроллера PRIZM. Если появится сообщение об ошибке, проведите поиск и устранение неисправности, руководствуясь приведёнными ранее советами.
  7. Убедившись, что в коде нет помарок и ошибок, сохраните внесённые изменения.

### Методика испытаний

Пора проверить на роботе, как работает ваш управляющий код. Учитель уже должен был подготовить в классе несколько испытательных станций.

Придерживайтесь следующей методики:

1. Вставьте кабель USB в контроллер PRIZM. Проследите, чтобы другой конец кабеля был вставлен в порт USB на вашем компьютере.
2. Включите робота.
3. В меню инструментов (Tools) *ПО Arduino (IDE)* выберите **Port**, и проследите, чтобы был выбран порт COM, связанный с вашим контроллером PRIZM. Если он не выбран, отметьте соответствующий порт COM.
4. В *ПО Arduino (IDE)* щёлкните по кнопке Upload, чтобы загрузить скетч в контроллер PRIZM. Вы наверняка обратите внимание на жёлтый светодиод у разъёма USB на контроллере PRIZM — он мигает, пока скетч загружается. Когда светодиод перестанет мигать, это означает, что код загружен и готов к запуску.
5. Отсоедините от робота кабель USB.
6. Отнесите робота на одну из испытательных станций, подготовленных учителем. Поставьте робота на пол так, чтобы датчик линии оказался прямо над чёрной линией.
7. Чтобы запустить исполнение программы, нажмите на зелёную пусковую кнопку. Робот должен двигаться по линии, пока ультразвуковой датчик не обнаружит на пути какое-то препятствие. Когда это случится, робот должен остановиться и ждать, пока препятствие уберут. Уберите препятствие с пути и наблюдайте за тем, как робот продолжит движение по линии.
8. Попробуйте поменять параметры в скетче. Вот что можно попробовать:
  - A. Поменяйте дальность обнаружения ультразвуковому датчику, который останавливает робота, если на пути окажется препятствие.
  - B. Измените скорость движения робота по линии. Насколько быстро может двигаться робот, при этом оставаясь на линии?



**Совет:** Если контроллер PRIZM не исполняет скетч после загрузки, попробуйте выключить-включить выключатель питания робота.



- С. Для условия, когда робот обнаруживает препятствие, измените обороты и положения сервоприводов.
9. Поэкспериментируйте, чтобы определить поле периферийного зрения ультразвукового датчика. Иными словами, насколько далеко необходимо передвинуть препятствие, чтобы ультразвуковой датчик его не обнаружил.
  10. Закончив экспериментировать, сохраните внесённые в код изменения. Обязательно записывайте любую важную информацию, данные, результаты и выводы в свой технический журнал.

## Упражнение № 9 — В какую сторону?

### Задача и цель

**Задача** — Запрограммировать робота так, чтобы он обнаруживал стенку и определял наилучший маршрут объезда препятствий.

**Цель** — Узнать, как запрограммировать робота на обнаружение и объезд препятствий при помощи ультразвукового датчика. Многие из команд и функций, которые вы используете в этом упражнении, понадобятся во время соревнования спасательных роботов в конце этого раздела.

### Обзор

Поисково-спасательные роботы, проникающие в обрушившиеся здания, натываются на множество разных препятствий, мешающих их продвижению. Если мелкие препятствия ещё можно было бы отодвинуть с пути, то более крупные, в виде завалившихся стенок, обломков и кусков мебели, сдвинуть нелегко. Важно, чтобы робот умел опознавать эти препятствия и находить путь объезда.

В ходе этого упражнения вам предстоит запрограммировать своего спасательного робота так, чтобы он двигался вперёд, пока не обнаружит препятствие, а потом посмотрел влево-вправо, чтобы решить, как это препятствие лучше объехать. После этого робот должен двинуться в избранном направлении, пока не обнаружит новое препятствие.

### Переменные

В упражнении появляется новое понятие из области программирования, которое не затрагивалось в *Руководстве по программированию контроллера PRIZM* — переменные. Записывая управляющий код, программисты используют **переменные**, в которых выражаются настройки или параметры в составе кода. Переменные, кроме того, облегчают сохранение информации и последующее её использование в программе.

Допустим, у вас есть поисково-спасательный робот, который должен передвигаться с определённой скоростью. Пока что сообщите, что хотите ограничить мощность, затрачиваемую на передвижение, 75 % от максимальной. Всякий раз, когда захотите привести робота в движение, можно использовать такую команду:

```
prizm.setMotorPowers(75,75);
```

Но что, если 75 процентов — слишком быстро, и надо перейти на 50 процентов от максимальной мощности? Тогда необходимо найти в коде все места, где задана величина мощности электродвигателя, и вместо 75 указать 50 процентов. Если программа сложная, это может потребовать большой работы.

Лучше в таком случае было бы сохранить величину мощности электродвигателя в виде переменной. В начале скетча можно объявить переменную и сохранить в ней некую величину. Например, можно ввести команду:

```
int mPower = 50;
```

Здесь "int" указывает на то, что это целочисленная переменная. Есть переменные и других видов, но в этом пособии по соревнованиям рассматриваются только целочисленные. В имени переменной **mPower** представляет собой сокращённое обозначение мощности электродвигателя. Имя для переменной можно выбрать почти любое, если оно не совпадает с именем команды или функции. В этой кодовой строке вы задаёте переменной **mPower** значение 50. Теперь, если захотите задать мощность электродвигателям, можно выразить это так:

```
prizm.setMotorPowers(mPower, mPower);
```

В результате мощность электродвигателей будет установлена на 50 процентов. Это хорошо тем, что, если вы захотите ещё раз изменить мощность, понизив её до 25 процентов, тогда останется заменить только одну строку кода (**int mPower = 25;**), а не искать по всему скетчу каждое место, в котором задана мощность электродвигателей.



Переменные можно использовать не только для сохранения вызываемых потом настроек или параметров. Переменные пригодны и для сохранения собранных данных, сравнения информации, подсчёта повторений и математических расчётов. Выполняя это упражнение, вы подробнее познакомитесь с некоторыми видами использования переменных.

### Вызываемые функции

К средствам программирования, которые вам предстоит использовать в этом упражнении, относится и вызываемая функция. Вызываемые функции должны быть вам знакомы после работы с *Руководством по программированию контроллера PRIZM*. Помните, что вызываемые функции — это своеобразные мини-программы, встречающиеся за пределами основных функций настройки и цикла. Всякий раз, когда в программе есть повторяющееся задание, которое надо выполнить, вне основного цикла **loop()**, который включает в себя все повторяющиеся команды, создаётся вызываемая функция. Затем, внутри основного цикла **loop()**, как только возникает необходимость исполнить эти команды, вызывается и исполняется эта функция. После завершения функции исполнение программы продолжается с того места, где функция была вызвана.

Вот скетч из упражнения № 10 в *Руководстве по программированию контроллера PRIZM*. В скетче был использован цикл **for()**, повторяющийся четырежды. Объявленная целочисленная переменная под именем **x** отслеживает число исполнений цикла. Внутри цикла **for()** две вызываемые функции. Одна с именем **forward()**, другая с именем **rightTurn()**. Обратите внимание, что определения функций находятся вне основного цикла **loop()** в конце скетча. Когда начинается исполнение цикла **for()**, робот движется вперёд, а потом поворачивает направо. Эти команды повторяются четыре раза, в результате чего робот движется по квадратной траектории.

```
#include <PRIZM.h>      // include PRIZM library
PRIZM prizm;           // instantiate a PRIZM object "prizm" so we can use its functions
void setup() {
  prizm.PrizmBegin();   // initialize PRIZM
  prizm.setMotorInvert(1,1); // invert the direction of DC Motor 1 to harmonize direction
}
void loop() {
  for(int x=0; x<=3; x++){ // do this four times, increment x by + 1
    forward();
    rightTurn();
  }
  prizm.PrizmEnd();
}
void forward(){        // function to go forward
  prizm.setMotorPowers(50,50); // go forward at 50% power
  delay(3000);         // wait here for 3 seconds while motors spin
  prizm.setMotorPowers(125,125); // stop both motors with in brake mode
  delay(1000);        // wait here for 1 second
}
void rightTurn(){     // function for a right turn
  prizm.setMotorPowers(50,-50); // make a right turn
  delay(600);         // wait here for .6 seconds while motors spin
  prizm.setMotorPowers(125,125); // stop both motors with in brake mode
  delay(1000);        // wait here for 1 second
}
```

### Счисление пути

В только что приведённом примере скетча для управления перемещением робота используется процесс **счисления пути**. При любом применении автономного робота важно знать, где находится и что делает робот в некое заданное время. Но без нужных датчиков роботу трудно точно знать, где он находится и как ему реагировать на обстановку. Счисление пути — это процесс определения положения и направленности робота. Для счисления пути в представленном примере скетча используются отсрочки. В функции движения передним ходом отсрочка в 3 тысячи миллисекунд означает, что роботу нужно двигаться вперёд 3 секунды.

```
void forward(){ // function to go forward
  prizm.setMotorPowers(50,50); // go forward at 50% power
  delay(3000); // wait here for 3 seconds while motors spin
  prizm.setMotorPowers(125,125); // stop both motors with in brake mode
  delay(1000); // wait here for 1 second
}
```

Но какое расстояние покрывает робот, двигаясь передним ходом? Он не знает. Дальность продвижения робота вперёд зависит от частоты вращения электродвигателей, размера колёс, силы трения между колёсами и полом, и заряда батареи. Захоти вы передвинуть робота на определённое расстояние (скажем, на один метр), пришлось бы перепробовать разные величины отсрочки. В конце концов, методом проб и ошибок, вы могли бы найти точную величину отсрочки, при которой робот проходит передним ходом один метр.

Так что, точное управление передвижением робота с помощью счисления пути, в котором время служит переменной, может оказаться сложноватым делом. Чтобы определить положение робота, можно вместо времени использовать датчик, например, датчик углового положения вала электродвигателя, или энкодер. Энкодеры снабжают контроллер робота информацией о положении вала электродвигателя и скорости его оборотов. Исходя из этого, контроллер может с помощью математики вычислить положение робота или рассчитать, сколько оборотов должен совершить электродвигатель, чтобы робот прошёл определённое расстояние. Датчики углового положения вала электродвигателя делают управление передвижением робота гораздо более точным.

У электродвигателей в вашем роботе могут быть энкодеры, но их может и не быть. В этом пособии управление роботом при помощи энкодеров электродвигателей подробно не рассматривается. Если у вас есть электродвигатели с энкодерами и применить их вам было бы интересно, обсудите это с учителем. Если учитель даст добро, вам надо будет самостоятельно исследовать способы их использования.

### Требующие учёта обстоятельства

- Обратите внимание на то, как упражнение связано с соревнованием, завершающим этот раздел.
- Давая переменным имена, используйте описательные названия, легко узнаваемые и непохожие друг на друга.
- Имена функций также лучше давать такие, чтобы они описывали решаемые функциями задачи.
- При записи управляющего кода легко допустить мелкие ошибки, которые могут сделать всю программу бесполезной. Пусть товарищи по команде проверят вашу работу ещё раз.
- Занимаясь отладкой управляющего кода, сверяйтесь с памяткой, упомянутой в обзорной части этого раздела на странице 39.
- При программировании в среде *ПО Arduino (IDE)* к наиболее распространённым относятся синтаксические ошибки вроде пропущенной точки с запятой в конце команды или не поставленной закрывающей круглой или квадратной скобки. Помните это, исправляя в своём коде синтаксические ошибки.
- Своевременно пополняйте записи в своём техническом журнале. Вносите туда заметки, данные, рабочие характеристики робота, полезные блоки (или кадры) управляющего кода, и всё, что могло бы пригодиться, когда наступит время соревнования.



**Совет:** В окне скетча библиотечные функции контроллера PRIZM меняют цвет, если напечатаны или введены без ошибок. Соответственно, если в их написании есть ошибка, то цвет не поменяется. Программное обеспечение в языке Arduino распознаёт функции контроллера PRIZM как ключевые слова и окрашивает их в оранжевый цвет, если не нарушены синтаксические правила.

## Методика и порядок действий

### Методика составления программного кода

1. Запустите на своём компьютере *ПО Arduino (IDE)*.
2. Чтобы создать новый скетч, зайдите в меню файлов (File) и выберите **New**.
3. Сохраните скетч. В меню файлов (File) выберите **Save As** и дайте файлу какое-нибудь имя.
4. Перед секцией **void setup()** надо добавить команду включения для библиотеки PRIZM. Напечатайте вверху скетча **#include <PRIZM.h>**. Для пояснения строки добавьте комментарий.
5. Далее понадобится команда для создания экземпляра объекта PRIZM. Напечатайте **PRIZM prizm;** под оператором включения. Для пояснения этой строки добавьте ещё один комментарий. Пока управляющий код должен выглядеть так:

```
#include <PRIZM.h>           // include PRIZM library
PRIZM prizm;                // instantiate a PRIZM object "prizm" so we can use its functions

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

6. Далее надо объявить несколько переменных. Как правило, переменные объявляются перед секцией настройки скетча. Из-за этого переменные обретают глобальную значимость: их можно использовать в любом месте скетча. Пока что начните с пяти переменных.
  - A. Начните с переменной с именем **mPower** — она будет регулировать мощность электродвигателя. Объявите её целочисленной переменной и присвойте значение 50. Чтобы сделать это, впечатайте **int mPower = 50;** перед секцией **void setup()**.
  - B. У следующей переменной имя **turnTime**, и она будет отвечать за длительность поворота робота вокруг центральной оси на 90 градусов. Напечатайте **int turnTime = 1000;** — этот оператор присваивает **turnTime** значение 1000, означающее 1000 миллисекунд или 1 секунду.
  - C. Понадобится измерить расстояние в двух местах: слева и справа. Объявите две переменные: одну с именем **leftDist**, а другую с именем **rightDist**. Объявите эти переменные целочисленными и равными 400 — это почти максимальная дальность действия ультразвукового датчика в сантиметрах.
  - D. Задать дальность обнаружения препятствий можно тоже с помощью переменной. Объявите целочисленную переменную с именем **leftCount**, равную 25.

- Е. В конце объявления каждой переменной добавьте комментарии, которые будут напоминать, что обозначает каждая переменная. Управляющий код должен выглядеть так:

```
#include <PRIZM.h> // include PRIZM library
PRIZM prizm; // instantiate a PRIZM object "prizm" so we can use its functions

int mPower = 50; // variable for motor power
int turnTime = 1000; // variable for the time it takes to make a 90-degree turn
int leftDist = 400; // variable for left distance
int rightDist = 400; // variable for right distance
int obstacleDist = 25; // variable for the stopping distance from an obstacle

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

7. Для секции **void setup()** остаётся ввести только две команды. Оператор **prizm.PrizmBegin();** задаёт начальные условия работы контроллера PRIZM, а команда **prizm.setMotorInvert(1,1);** задаёт 1-му электродвигателю постоянного тока обратное направление вращения. Секция **void setup()** должна выглядеть так:

```
void setup() {
  prizm.PrizmBegin(); // initialize PRIZM
  prizm.setMotorInvert(1,1); // invert the direction of DC Motor 1 to harmonize direction
}
```

8. Для основного цикла **loop()** выберите начало с оператором **if()**. Это позволит определить, находится ли препятствие в заданных пределах действия датчика робота. Запомните, что объявленной переменной **obstacleDist** вы установили дальность 25. Для обнаружения препятствий вы будете использовать ультразвуковой датчик. Создайте оператор **if()**, аналогичный показанному ниже.

```
void loop() {
  if(prizm.readSonicSensorCM(2) > obstacleDist) // determine if obstacle is within range
  { // conditions for no obstacle detected
  }
}
```

Проследите, чтобы оператор **prizm.readSonicSensorCM()** указывал на правильный порт на контроллере PRIZM. В представленном выше примере ультразвуковой датчик подсоединён ко 2-му порту для цифровых датчиков на контроллере PRIZM. Если у вас датчик вставлен в другой порт, в этой команде надо поменять номер, чтобы он соответствовал порту на вашем контроллере PRIZM.

9. Чтобы завершить формирование условий оператора **if()**, надо указать роботу, что делать, когда ультразвуковой датчик обнаруживает расстояние, превышающее дальность, заданную переменной **obstacleDist**, которая равна 25 сантиметрам. То есть, такое происходит, когда препятствий в пределах действия датчика нет. Поэтому роботу не надо делать ничего, кроме как продолжать движение вперёд.



**Совет:** Если понадобится быстро узнать, какие функции имеются в вашем распоряжении, см. в *Руководстве по программированию контроллера PRIZM* памятку с библиотечными функциями на языке Arduino для контроллера PRIZM компании TETRIX.

- A. Внутри скобок, относящихся к оператору **if()**, введите команду, которая задаёт мощность обоих электродвигателей переменной **mPower**.
- B. Чтобы показать, что всё хорошо, выключите на контроллере PRIZM красный светодиод и включите зелёный. Внутри скобок оператора **if()** напечатайте команду для каждого из них.
- C. Добавьте комментарии, которые напомнят вам или кому-то ещё, какие действия вызывает каждая команда. Теперь ваш условный оператор должен выглядеть примерно так:

```
if(prizm.readSonicSensorCM(2) > obstacleDist) // determine if obstacle is within range
{
    prizm.setMotorPowers(mPower,mPower);      // conditions for no obstacle detected
    prizm.setRedLED(LOW);                      // move foward at the determined motor power level
    prizm.setGreenLED(HIGH);                  // turn off the red LED
}                                             // turn on the green LED
```

10. Надо также проверить второе условие: если препятствие обнаружено. То есть, когда расстояние, считываемое ультразвуковым датчиком, меньше или равно заданному расстоянию. Поскольку есть всего два условия (одно для случая, когда препятствие не обнаружено, второе для случая, когда препятствие обнаружено), можно превратить условный оператор **if()** в оператор **if/else**. Добавьте к условному оператору часть **else** так:

```
if(prizm.readSonicSensorCM(2) > obstacleDist) // determine if obstacle is within range
{
    prizm.setMotorPowers(mPower,mPower);      // conditions for no obstacle detected
    prizm.setRedLED(LOW);                      // move foward at the determined motor power level
    prizm.setGreenLED(HIGH);                  // turn off the red LED
}                                             // turn on the green LED
else
{
    // conditions for obstacle detected
}
```

11. Внутри скобок, относящихся к части **else** условного оператора, будет несколько команд. Запомните, что это относится к случаю, когда робот обнаруживает препятствие в пределах действия датчика. После ввода каждой команды добавляйте комментарии — они помогут вам и вашим товарищам по команде вспомнить, какие действия вызывает каждая команда.

- A. Для начала включите красный светодиод и выключите зелёный, чтобы указать на наличие препятствия в пределах действия датчика. Добавьте команду на выполнение каждого действия.
- B. Необходимо остановить робота, чтобы он не врезался в препятствие. Поскольку эти команды должны исполняться неоднократно, создайте функцию с именем **stopMotors()**. Для начала добавьте в часть **else** вызов функции. Чтобы сменить светодиоды на контроллере PRIZM, напечатайте "**stopMotors();**" в виде следующей строки под командами. Теперь надо дать функции определение. Это необходимо сделать вне основного цикла **loop()**. Перейдите в самый конец скетча и напечатайте функцию так, как показано ниже:

```
void stopMotors(){
    prizm.setMotorPowers(125,125);           // function to come to a dead stop
    delay (250);                             // stop all motors
}
```

Теперь эту функцию можно использовать всякий раз, как требуется остановить электродвигатели.

- C. Вернитесь наверх к части **else** скетча. Дальше, остановив электродвигатели, необходимо чуть отодвинуть робота назад. Так у него будет чуть больше пространства для манёвра и он не наедет на препятствие. Дайте роботу команду медленно отъехать назад, включив электродвигатели на половину указанной в начальных настройках мощности. Запомните, что для включения заднего хода переменная **mPower** должна быть отрицательной. Команда должна выглядеть так:

```
prizm.setMotorPowers(-mPower/2,-mPower/2); // back up at half the motor power level
```

- D. Добавьте небольшую отсрочку в 1000 миллисекунд, чтобы у робота было время отъехать назад.
- E. После включения заднего хода робота надо опять остановить. Ещё раз вызовите функцию **stopMotors()**. Чтобы остановить обратное вращение электродвигателей, напечатайте **"stopMotors();"**.
- F. Теперь надо заставить робота повернуться и посмотреть влево, чтобы проверить, пригодно ли это направление для объезда. Надо повернуть робота влево. Создайте ещё одну функцию, которая будет отвечать за поворот влево. Вызовите функцию **leftTurn()**. Перейдите в конец скетча под функцию **stopMotors()** и создайте для поворота влево эту функцию **leftTurn()**:

```
void leftTurn(){ // called function for making left turn
  prizm.setMotorPowers(-mPower,mPower); // rotate motors to turn left
  delay(turnTime); // turn for the designated time
  prizm.setMotorPowers(125,125); // stop all motors
}
```

В этой функции одному колесу переменная **mPower** задаёт вращение назад, а другому колесу — вращение вперёд с той же скоростью. В результате робот вращается на месте. Чтобы у робота было время развернуться на месте, добавляется отсрочка. Переменная **turnTime** задаёт отсрочку, поскольку пока неизвестно, сколько времени займёт у робота поворот на 90 градусов. Вы узнаете это методом проб и ошибок позже. Наконец, электродвигатели останавливаются, завершая поворот.

- G. Вернитесь к части **else** скетча. Теперь, развернувшись влево, робот должен определить расстояние до возможного препятствия слева. Читанное в этом направлении расстояние он сохранит в виде переменной **leftDist**. Чтобы это произошло, напечатайте следующую команду:

```
leftDist = prizm.readSonicSensorCM(2); // read distance to the left and store it in variable leftDist
```

- H. Измерив расстояние до препятствия слева, робот должен повернуться вправо и измерить расстояние до возможного препятствия справа. Создайте ещё одну функцию, которая будет отвечать за поворот на 180 градусов. Вызовите функцию **aboutTurn()**. Чтобы дать этой функции определение, перейдите в конец скетча и создайте следующую функцию:

```
void aboutTurn(){ // called function for making about turn
  prizm.setMotorPowers(mPower,-mPower); // rotate motors to turn right
  delay(turnTime*2); // turn for twice the designated time
  prizm.setMotorPowers(125,125); // stop all motors
}
```

В этой функции величины мощности электродвигателей заданы таким образом, чтобы робот развернулся в сторону, противоположную той, которая задала функция **leftTurn()**. Поскольку совершается поворот на 180, а не на 90 градусов, ожидание для вращения робота должно быть вдвое больше величины **turnTime**. В заключение, когда робот повернётся, электродвигатели остановятся. Теперь робот должен быть развёрнут вправо.



- I. Вернитесь в часть **else** скетча: робот должен считать расстояние до возможного препятствия справа и сохранить показания в виде переменной **rightDist**. Задайте переменной **rightDist** значение, равное измеренному ультразвуковым датчиком расстоянию, так же, как делали это для переменной **leftDist**.

```
rightDist = prizm.readSonicSensorCM(2); // read distance to the right and store it in variable rightDist
```

- J. Теперь робот должен решить, в какую сторону двинуться. Для этого он сравнит расстояние справа с расстоянием слева. После сравнения робот двинется туда, где расстояние больше. Если расстояние справа больше расстояния слева, тогда роботу не надо предпринимать никакие дополнительные действия. Он уже смотрит в нужную сторону, потому что последний его поворот был совершен вправо. А если расстояние слева больше расстояния справа, тогда роботу придётся ещё раз развернуться на 180 градусов. Вам надо будет записать условный оператор для проверки этого условия:

```
if(leftDist > rightDist){aboutTurn();} // compare leftDist and rightDist and make about turn if necessary
```

В этом условном операторе повторно используется ранее записанная функция **aboutTurn()**.

- K. На этом часть **else** основного условного оператора должна завершиться. Это одновременно и конец основного цикла **loop()** — здесь робот знает, куда двигаться и развернут в нужном направлении. Остаётся только повторить команды. Итак, если всё сложить вместе, ваш основной цикл **loop()** должен выглядеть так:

```
void loop() {
  if(prizm.readSonicSensorCM(2) > obstacleDist)
  {
    prizm.setMotorPowers(mPower,mPower); // conditions for no obstacle detected
    prizm.setRedLED(LOW); // move foward at the determined motor power level
    prizm.setGreenLED(HIGH); // turn off the red LED
    prizm.setGreenLED(HIGH); // turn on the green LED
  }
  else
  {
    // conditions for obstacle detected
    prizm.setRedLED(HIGH); // turn on the red LED
    prizm.setGreenLED(LOW); // turn off the green LED
    stopMotors(); // call the function to stop the motors
    prizm.setMotorPowers(-mPower/2,-mPower/2); // back up at half the motor power level
    delay (1000); // back up for 1 second
    stopMotors(); // call the function to stop the motors

    leftTurn(); // execute the leftTurn function
    leftDist = prizm.readSonicSensorCM(2); // read distance to the left and store it in variable leftDist

    aboutTurn(); // execute the aboutTurn function to face to the right
    rightDist = prizm.readSonicSensorCM(2); // read distance to the right and store it in variable rightDist

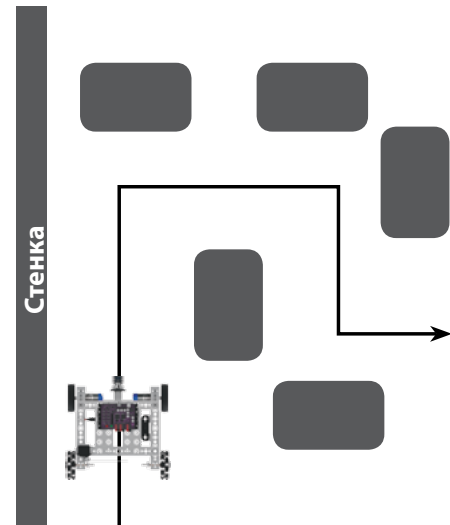
    if(leftDist > rightDist){aboutTurn();} // compare leftDist and rightDist and make about turn if necessary
  }
} // end the main loop
```

12. Проверьте синтаксис скетча инструментом контроля (Verify) в ПО *Arduino (IDE)*. Этот инструмент проверяет, насколько код точен и удобочитаем для контроллера PRIZM. Если появится сообщения об ошибке, устранили её причину. Используйте памятку по отладке управляющего кода, помещённую на странице 39. Обращайте также внимание на сообщения об ошибках и коды неисправностей, появляющиеся в нижней части окна с ПО *Arduino (IDE)*.
13. Убедившись, что в коде нет помарок и ошибок, сохраните внесённые изменения.

**Методика испытаний**

Пора проверить, как робот исполняет вашу программу для объезда препятствий. Учитель уже должен был подготовить в классе несколько испытательных участков. Придерживайтесь следующей методики:

1. Вставьте кабель USB в контроллер PRIZM. Проследите, чтобы другой конец кабеля был вставлен в порт USB на вашем компьютере.
2. Включите робота.
3. В меню инструментов (Tools) *ПО Arduino (IDE)* выберите **Port**, и проследите, чтобы был выбран порт COM, связанный с вашим контроллером PRIZM. Если он не выбран, отметьте соответствующий порт COM.
4. В *ПО Arduino (IDE)* щёлкните по кнопке Upload, чтобы загрузить скетч в контроллер PRIZM. Вы наверняка обратите внимание на жёлтый светодиод у разъёма USB на контроллере PRIZM — он мигает, пока скетч загружается. Когда светодиод перестает мигать, это означает, что код загружен и готов к запуску.
5. Отсоедините от робота кабель USB.
6. Отнесите робота на одну из испытательных станций, подготовленных учителем. Поставьте робота на грунт, не ближе чем в 25 сантиметрах от препятствия.
7. Чтобы запустить исполнение программы, нажмите на зелёную пусковую кнопку. Робот должен двигаться вперёд, пока не обнаружит препятствие. Здесь он должен остановиться, подать назад, повернуть налево, повернуть направо, решить куда двинуться и направиться по наиболее свободному пути. Понаблюдайте за поведением робота, затем остановите программу, нажав красную кнопку сброса параметров на контроллере PRIZM.
8. Даже если управляющий код исполняется правильно, есть большая вероятность, что робот ведёт себя не так, как было задумано. Регулируя параметры робота, можно точно отладить код и улучшить поведение робота. Поскольку многие из этих параметров выражены переменными, их точная отладка не составит труда. Попробуйте отрегулировать эти настройки в своём скетче.
  - A. Измените заданное переменной **mPower** значение с 50 на 25. В результате робот должен замедлиться. Загрузите скетч в робота и проверьте его на испытательном участке.
  - B. Время, за которое выполняется поворот на 90 градусов, сильно зависит от заряда батареи. Каждый робот будет вести себя по-своему. Если робот поворачивает больше чем на 90 градусов, уменьшите значение переменной **turnTime**. Если робот поворачивает меньше чем на 90 градусов, увеличьте значение переменной **turnTime**. Загрузите скетч в робота и проверьте его на испытательном участке.
  - C. Задав значение переменной **turnTime**, вы можете обнаружить, что поворот на 180 градусов занимает не ровно вдвое больше заданного времени. На самом деле, он должен занять чуть меньше времени, чем удвоенное значение переменной **turnTime**. Поразмышляйте о возможной причине этого. Затем скорректируйте функцию **aboutTurn**, умножив **turnTime** на 1,75, а не на 2. Загрузите скетч в робота и проверьте его на испытательном участке.
  - D. Если выяснится, что робот наезжает на препятствия, особенно приближаясь к ним под углом, попробуйте увеличить значение переменной **obstacleDist** с 25 до 50. Загрузите скетч в робота и проверьте его на испытательном участке.



**Совет:** Если контроллер PRIZM не исполняет уже загруженный скетч, попробуйте выключить-включить выключатель питания робота.

- E. Продолжайте отладку управляющего кода, стремясь улучшить поведение робота.
- 9. Когда робот начнёт всё делать правильно и будет объезжать препятствия, сохраните окончательную версию кода. Обязательно записывайте любую важную информацию, данные, результаты и выводы в свой технический журнал.

## Упражнение № 10 — Всего-то гончий пёс

### Задача и цель

**Задача** — Запрограммировать робота так, чтобы он мог двигаться по линии, совершая крутые повороты.

**Цель** — Узнать, как запрограммировать робота на движение по линии при совершении крутого поворота. Многие из команд и функций, которые вы используете в этом упражнении, понадобятся во время соревнования спасательных роботов в конце этого раздела.

### Обзор

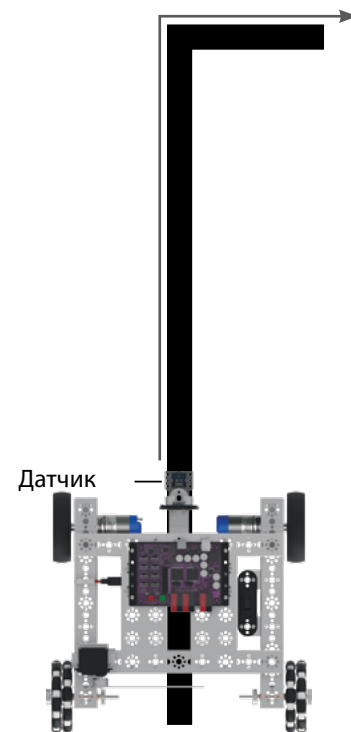
Существуют всевозможные виды поисково-спасательных роботов. Некоторые оснащены датчиками, чуткими к движению, звуку и запаху. Роботы с такими датчиками хороши для обнаружения людей, которые могли оказаться под завалами в обрушившемся здании. Это технические гончие, нюх которых помогает найти живых. В вашем наборе конструктора TETRIX нет таких датчиков, которые позволили бы запрограммировать своего робота в виде гончего пса, но зато в нём есть датчик линии, способный взять след линии и идти по нему. Основной принцип использования показаний датчика для принятия решения не меняется.

У вас уже есть опыт работы с роботами-следопытами. Но до сих пор все линии, по которым шли роботы, были прямыми или слегка изогнутыми, так что робот менял траекторию движения постепенно. А что случится, если линия круто повернёт под прямым углом? Сумеете ли вы запрограммировать робота так, чтобы он распознавал крутые повороты и разворачивался на месте, чтобы продолжить движение по таким линиям? Конечно, сумеете. Это пригодится, когда дело дойдёт до соревнования в этом разделе.

Вы начнёте упражнение со знакомой программы, управляющей движением по линии. Вы проверите эту программу и посмотрите, как робот круто поворачивает под прямым углом. Затем вы доработаете программу так, чтобы робот, двигаясь по линии, уверенно совершал крутые повороты. Внося изменения в управляющий код, чтобы сделать его эффективным и удобным для отладки действий робота, вы будете по-прежнему использовать переменные.

### Требующие учёта обстоятельства

- Разработанный вами в ходе упражнения управляющий код пригодится для заключительного соревнования. Заносите в свой технический журнал заметки, а также важные сведения и результаты наблюдений.
- При записи управляющего кода легко допустить мелкие ошибки, которые могут сделать всю программу бесполезной. Пусть товарищи по команде проверят вашу работу ещё раз.
- Занимаясь отладкой управляющего кода, сверяйтесь с памяткой, упомянутой в обзорной части этого раздела на странице 39.
- При программировании в среде *ПО Arduino (IDE)* к наиболее распространённым относятся синтаксические ошибки вроде пропущенной точки с запятой в конце команды или не поставленной закрывающей круглой или квадратной скобки. Помните это, исправляя в своём коде синтаксические ошибки. Для проверки синтаксиса программы используйте инструмент контроля (*Verify*), встроенный в *ПО Arduino (IDE)*.



**Совет:** В окне скетча библиотечные функции контроллера PRIZM меняют цвет, если напечатаны или введены без ошибок. Соответственно, если в их написании есть ошибка, то цвет не поменяется. Программное обеспечение на языке *Arduino* распознаёт функции контроллера PRIZM как ключевые слова и окрашивает их в оранжевый цвет, если не нарушены синтаксические правила.

## Методика и порядок действий

### Методика предварительных испытаний

1. Вы начнёте со скетча, уже знакомого вам по *Руководству по программированию контроллера PRIZM*.
  - A. Запустите ПО *Arduino (IDE)*.
  - B. В меню файлов (File) выберите **Examples > TETRIX\_PRIZM > TaskBot\_Act12\_Follow\_A\_Line**.
  - C. Сохраните этот скетч под новым именем, например "Line\_Following\_Sharp\_Turns."
2. Проследите, чтобы оба оператора **prizm.readLineSensor** указывали на правильный порт на вашем контроллере PRIZM. В представленном примере скетча датчик линии подсоединён ко 3-му порту для цифровых датчиков на контроллере PRIZM. Если у вас датчик вставлен в другой порт, в этих командах надо поменять номер, чтобы он соответствовал порту на вашем контроллере PRIZM. Сохраните все изменения, внесённые в скетч.
3. Вставьте кабель USB в контроллер PRIZM. Проследите, чтобы другой конец кабеля был вставлен в порт USB на вашем компьютере.
4. Включите робота.
5. В меню инструментов (Tools) *ПО Arduino (IDE)* выберите **Port**, и проследите, чтобы был выбран порт COM, связанный с вашим контроллером PRIZM. Если он не выбран, отметьте соответствующий порт COM.
6. В *ПО Arduino (IDE)* щёлкните по кнопке **Upload**, чтобы загрузить скетч в контроллер PRIZM. Вы наверняка обратите внимание на жёлтый светодиод у разъёма USB на контроллере PRIZM — он мигает, пока скетч загружается. Когда светодиод перестает мигать, это означает, что код загружен и готов к запуску.
7. Отсоедините от робота кабель USB.
8. Отнесите робота на испытательный участок, подготовленный учителем. Поставьте робота на землю прямо над линией, чтобы он следовал по ней.
9. Чтобы запустить исполнение программы, нажмите на зелёную пусковую кнопку. Робот должен начать движение по линии. Заметьте, что происходит, когда линия круто поворачивает. Остановите робота и запишите свои наблюдения в технический журнал. Предположите, почему это произошло.



**Совет:** Если контроллер PRIZM не исполняет уже загруженный скетч, попробуйте выключить-включить выключатель питания робота.

### Методика составления программного кода

В предыдущем испытании вы могли заметить, что робот с трудом двигался по линии, которая поворачивала под крутым углом. Далее вы переделаете управляющий код так, чтобы убрать эти затруднения.

1. Вернитесь в *ПО Arduino (IDE)* к скетчу, который сохранили в разделе "Методика предварительных испытаний" (Preliminary Testing Procedures). Начните с объявления переменных, которые будут использованы в видеоизменённом скетче. Запомните, что объявления переменных в скетче предшествуют части **void setup()**.
  - A. Создайте для нормальной мощности электродвигателя целочисленную переменную с именем **normPower**. Задайте переменной значение, равное 30.
  - B. Создайте для пониженной мощности электродвигателя ещё одну целочисленную переменную с именем **lowPower**. Задайте переменной значение, равное 10.

- C. Для подсчёта числа непрерывных повторений одной и той же задачи в программе вам понадобятся две переменные. Эти переменные помогут нам определить, сошёл робот с линии или остаётся прямо над ней. Подробнее эту переменную рассмотрим позднее. Пока что создайте равную нулю целочисленную переменную с именем **leftCount**. Создайте ещё одну целочисленную переменную с именем **rightCount** и также присвойте ей нулевое значение.
- D. Вам понадобится ещё одна переменная, чтобы задать программе максимальное число повторений одной и той же группы команд, прежде чем она определит, что что-то не так. Подробные объяснения этого также будут даны позднее. Создайте целочисленную переменную с именем **maxIt** (сокращение для "максимального числа повторений") и присвойте ей значение 3.
- E. Дополните объявления переменных комментариями, чтобы не забыть, что означает каждая переменная. Объявления ваших переменных должны выглядеть так:

```
int normPower = 30;    // normal motor power
int lowPower = 10;    // low motor power
int leftCount = 0;    // counts the number of times the robot turns left
int rightCount = 0;   // counts the number of times the robot turns right
int maxIt = 3;        // maximum number of iterations allowed
```

2. В части **void setup()** скетча предположительно всё в порядке. Здесь ничего менять не нужно.
3. Перейдите к секции основного цикла **loop()** скетча. Здесь кое-что в коде надо поменять. Сейчас есть два однострочных оператора **if()**. К этим условным операторам вы добавите ещё несколько команд. Разбейте эту пару операторов **if()** на многострочные операторы.
- A. В обоих условных операторах добавьте возврат перед командой **prizm.setMotorPowers()**.
- B. В обоих условных операторах добавьте ещё один возврат перед командой **prizm.setRedLED()**.
- C. В заключение добавьте ещё один возврат перед скобкой, закрывающей каждый условный оператор.
- D. Если понадобится, замените или добавьте комментарии, чтобы знать, что делает каждая строка кода. В законченном виде код должен выглядеть примерно так:

```
void loop() {
    if(prizm.readLineSensor(3) == 0){           // beam reflected, no line detected
        prizm.setMotorPowers(125,30);         // turn the robot left
        prizm.setRedLED(LOW);                 // turn off the red LED
    }

    if(prizm.readLineSensor(3) == 1){         // no reflected beam, line detected
        prizm.setMotorPowers(30,125);        // turn the robot right
        prizm.setRedLED(HIGH);               // turn on the red LED
    }
}
```

4. Вспомните, что для нормальной мощности электродвигателя вы создали переменную с именем **normPower**. В обоих операторах **prizm.setMotorPowers()** вместо значения 30 поставьте **normPower**. Впоследствии благодаря этому вы легко сможете изменить скорость робота, тестируя код.
5. Теперь вам надо подсчитать, сколько раз подряд робот не обнаружил линию и совершил поворот влево. То есть надо сосчитать, сколько раз было истинным условие первого оператора **if()**. Вот почему это важно: если робот не обнаружит линию заданное число раз подряд, тогда ему придётся найти её иначе. Для подсчёта этих повторений вы уже создали две переменные: **leftCount** и **rightCount**. Поскольку первый оператор **if()** отвечает за поворот влево, вы используете **leftCount**, чтобы отследить, сколько раз подряд соблюдается условие этого оператора. Значение переменной **leftCount** должно увеличиваться на единицу при каждом соблюдении этого условия.
  - A. Перед закрывающей скобкой первого оператора **if()** добавьте команду:

```
leftCount = leftCount + 1;
```

- B. Добавьте к этой строке комментарий, описывающий, что именно подсчитывает эта команда.
    - C. То же самое надо сделать для второго оператора **if()**, на котором робот поворачивает вправо. Но в этом случае строка обнаруживается много раз подряд. Эта команда помогает понять, когда линия круто сворачивает в сторону и робот пересекает линию. Перед закрывающей скобкой второго оператора **if()** добавьте команду:
- ```
rightCount = rightCount + 1;
```
- D. Добавьте к этой строке комментарий с описанием того, что именно подсчитывает эта команда.
  6. Если робот выполняет одно и то же действие слишком много раз подряд, либо поворачивая налево, потому что не обнаружил линию, либо направо, потому что несколько раз подряд обнаружил линию, тогда вам ясно, что он больше не отслеживает край линии. Он либо потерял линию из виду, если значение переменной **leftCount** слишком высоко, либо оказался посередине линии, если слишком высоко значение переменной **rightCount**. В любом из этих случаев линия повернула под более крутым углом, чем тот, на который способен робот, поэтому роботу надо выполнить другое действие, чтобы снова найти край линии.
    - A. В первый оператор **if()** надо добавить ещё один условный оператор. Это называется вложенным условным оператором, потому что один оператор **if()** находится внутри другого оператора **if()**. Перед закрывающей скобкой первого оператора **if()** добавьте вложенный оператор **if()** вот так:

```
if(prizm.readLineSensor(3) == 0){
    prizm.setMotorPowers(125,normPower); // beam reflected, no line detected
    prizm.setRedLED(Low); // turn the robot left
    leftCount = leftCount + 1; // turn off the red LED
    if (leftCount > maxIt) { // count the number of times in a row no line was detected
        // determine if the robot has lost the line
    }
}
```

Здесь вы сравниваете переменную **leftCount** с переменной **maxIt**, объявленной в начале скетча. То есть, если робот повернул влево, превысил максимально допустимое число, заданное в начале скетча, тогда он выполнит некоторые другие команды, чтобы снова найти линию.

- В. Нечто подобное можно сделать во втором операторе **if()** для поворота вправо, но на сей раз с переменной **maxIt** будет сравниваться переменная **rightCount**. Создайте вложенный оператор **if()** в конце условия для обнаружения линии (второй оператор **if()**) вот так:

```
if(prizm.readLineSensor(3) == 1){           // no reflected beam, line detected
  prizm.setMotorPowers(normPower,125);     // turn the robot right
  prizm.setRedLED(HIGH);                   // turn on the red LED
  rightCount = rightCount + 1;             // count the number of times in a row the line was detected
  if (rightCount > maxIt) {                 // determine if the robot is in the middle of the line
  }
}
```

- С. Дополните оба вложенных оператора **if()** комментариями.

7. Внутри вложенного оператора **if()** должно произойти некое решительное действие, чтобы робот снова нашёл край линии. Необходимо круто повернуть либо влево, либо вправо. Роботу надо продолжить эти крутые повороты до тех пор, пока он снова не найдёт край линии. Чтобы крутые повороты продолжались, пока не будет выполнено условие, надо использовать циклы **while()**.

- А. В первом вложенном операторе **if()** надо дать роботу команду круто поворачивать влево, пока он снова не найдёт линию. Чтобы робот круто повернул, одно его колесо должно вращаться назад согласно значению переменной **lowPower**, а другое — вперёд согласно значению переменной **lowPower**. Из-за этого робот перестаёт двигаться вперёд и начинает кружиться на месте. Внутри первого вложенного оператора **if()** создайте цикл **while()**, который выглядит так:

```
if (leftCount > maxIt) {                     // determine if the robot has lost the line
  while (prizm.readLineSensor(3) == 0) {    // repeat while no line is detected
    prizm.setMotorPowers(-lowPower,lowPower); // pivot the robot sharply to the left
  }
}
```

- В. Аналогичный цикл **while()** можно создать во втором вложенном операторе **if()**, чтобы робот повернулся на месте вправо. Учтите, что для этого цикла **while()** команда **prizm.readLineSensor()** должна равняться единице, а не нулю, потому что нужно, чтобы команда внутри цикла исполнилась, когда будет обнаружена линия.

```
if (rightCount > maxIt) {                    // determine if the robot is in the middle of the line
  while (prizm.readLineSensor(3) == 1) {    // repeat while the line is detected
    prizm.setMotorPowers(lowPower,-lowPower); // pivot the robot sharply to the right
  }
}
```



8. Пора внести ещё одно изменение в порядок использования светодиодов, помогающих определить, совершает робот обычный поворот или входит в цикл **while()**, чтобы совершить крутой поворот. Впоследствии, если робот будет сбиваться с линии, это поможет устранить неполадки.

- A. В первом операторе **if()** вы уже выключаете красный светодиод. Добавьте прямо под ним ещё одну команду, чтобы выключить и зелёный светодиод.

```
prizm.setRedLED(LOW);           // turn off the red LED
prizm.setGreenLED(LOW);        // turn off the green LED
```

- B. То же самое надо сделать и во втором операторе **if()**. В команде для красного светодиода замените переменную HIGH (высокий уровень) переменной LOW (низкий уровень). Добавьте прямо под ней ещё одну команду, чтобы выключить зелёный светодиод.
- C. Для обозначения крутого поворота влево используйте красный светодиод. В цикле **while()** для левого поворота добавьте перед закрывающей скобкой команду включения красного светодиода.

```
while (prizm.readLineSensor(3) == 0) { // repeat while no line is detected
  prizm.setMotorPowers(-lowPower,lowPower); // pivot the robot sharply to the left
  prizm.setRedLED(HIGH); // turn on the red LED
}
```

- D. Для обозначения крутого поворота вправо используйте зелёный светодиод. В цикле **while()** для правого поворота добавьте перед закрывающей скобкой команду включения зелёного светодиода.

```
while (prizm.readLineSensor(3) == 1) { // repeat while the line is detected
  prizm.setMotorPowers(lowPower,-lowPower); // pivot the robot sharply to the right
  prizm.setGreenLED(HIGH); // turn on the green LED
}
```

- E. По мере необходимости добавляйте или меняйте комментарии к новым строкам.
9. Теперь, когда робот снова в строю и движется по линии, осталось последнее — обнулить счётчики. Иначе счётчики продолжат отсчёт, и робот будет просто кружиться на месте.
- A. Внутри первого вложенного оператора **if()** после закрывающей скобки для цикла **while()** добавьте команду на обнуление переменной **leftCount**. Добавьте ещё одну команду, чтобы обнулить переменную **rightCount**.
- B. Внутри второго вложенного оператора **if()** после закрывающей скобки для цикла **while()** добавьте ещё две команды на обнуление переменной **leftCount** и переменной **rightCount**.

- С. Добавьте для этих строк комментарии. Теперь ваш основной цикл `loop()` должен выглядеть примерно так:

```
void loop() {
  if(prizm.readLineSensor(3) == 0){           // beam reflected, no line detected
    prizm.setMotorPowers(125,normPower);      // turn the robot left
    prizm.setRedLED(LOW);                     // turn off the red LED
    prizm.setGreenLED(LOW);                  // turn off the green LED
    leftCount = leftCount + 1;                // count the number of times in a row no line was detected
    if (leftCount > maxIt) {                  // determine if the robot has lost the line
      while (prizm.readLineSensor(3) == 0) { // repeat while no line is detected
        prizm.setMotorPowers(-lowPower,lowPower); // pivot the robot sharply to the left
        prizm.setRedLED(HIGH);               // turn on the red LED
      }
      leftCount = 0;                          // reset the leftCount variable
      rightCount = 0;                          // reset the rightCount variable
    }
  }
  if(prizm.readLineSensor(3) == 1){           // no reflected beam, line detected
    prizm.setMotorPowers(normPower,125);      // turn the robot right
    prizm.setRedLED(LOW);                     // turn off the red LED
    prizm.setGreenLED(LOW);                  // turn off the green LED
    rightCount = rightCount + 1;              // count the number of times in a row the line was detected
    if (rightCount > maxIt) {                 // determine if the robot is in the middle of the line
      while (prizm.readLineSensor(3) == 1) { // repeat while the line is detected
        prizm.setMotorPowers(lowPower,-lowPower); // pivot the robot sharply to the right
        prizm.setGreenLED(HIGH);            // turn on the green LED
      }
      rightCount = 0;                          // reset the rightCount variable
      leftCount = 0;                          // reset the leftCount variable
    }
  }
}
```

- Проверьте синтаксис скетча инструментом контроля (Verify) в *ПО Arduino (IDE)*. Этот инструмент проверяет, насколько код точен и удобочитаем для контроллера PRIZM. Если появятся сообщения об ошибке, устраните её причину. Используйте памятку по отладке управляющего кода, помещённую на странице 39. Обращайте также внимание на сообщения об ошибках и коды неисправностей, появляющиеся в нижней части окна с *ПО Arduino (IDE)*.
- Убедившись, что в коде нет помарок и ошибок, сохраните внесённые в него изменения.

#### Методика заключительного испытания

- Вставьте кабель USB в контроллер PRIZM. Проследите, чтобы другой конец кабеля был вставлен в порт USB на вашем компьютере.
- Включите робота.
- В меню инструментов (Tools) *ПО Arduino (IDE)* выберите **Port**, и проследите, чтобы был выбран порт COM, связанный с вашим контроллером PRIZM. Если он не выбран, отметьте соответствующий порт COM.
- В *ПО Arduino (IDE)* щёлкните по кнопке **Upload**, чтобы загрузить скетч в контроллер PRIZM. Вы наверняка обратите внимание на жёлтый светодиод у разъёма USB на контроллере PRIZM — он мигает, пока скетч загружается. Когда светодиод перестанет мигать, это означает, что код загружен и готов к запуску.
- Отсоедините от робота кабель USB.
- Отнесите робота на испытательный участок, подготовленный учителем. Поставьте робота на пол так, чтобы датчик линии оказался прямо над чёрной линией.

7. Чтобы запустить исполнение программы, нажмите на зелёную пусковую кнопку. Робот должен начать двигаться по линии. Если линия круто поворачивает, робот должен прекратить движение вперёд и начать вращаться на месте, пока не найдёт линию снова. После обнаружения края линии робот должен продолжить движение по ней.
8. Поведение робота может зависеть от нескольких факторов, в том числе от уровня освещённости в помещении, расстояния датчика линии от пола, положения датчика перед осью поворота робота, размера робота и скорости движения робота. Попробуйте отладить и оптимизировать работу робота, меняя переменные в настройках. Если понадобится, поместите датчик в другое место.
  - A. В ходе испытаний следите за светодиодами: они подскажут, когда робот делает обычные повороты, а когда — повороты вокруг своей вертикальной оси. Это облегчит устранение неисправностей и подстройку переменных с целью оптимизации робота.
  - B. Если робот сбивается с линии, попробуйте замедлить его, подстроив переменные **normPower** и **lowPower**.
  - C. Если у робота большой радиус поворота, попробуйте увеличить переменную **maxIt**, чтобы сделать поправку на увеличенный радиус поворота. В случае малого радиуса поворота попробуйте уменьшить переменную **maxIt**.
  - D. Если у вас датчик линии вынесен далеко перед осью поворота робота, то ширина заноса при повороте робота увеличится. Уменьшение скорости робота и увеличение переменной **maxIt** должно помочь роботу удержаться на линии.
  - E. Может оказаться, что, следуя линии, ваш робот лучше поворачивает в одну сторону, чем в другую. Попробуйте поменять значения мощности электродвигателя в цикле **while()** так, чтобы робот поворачивал круче. Возможно, придётся задать две разные переменные: одну для **lowPowerLeft**, а другую для **lowPowerRight**, чтобы поворачивать с разной скоростью.
  - F. Если покажется, что робот реагирует на тени, попробуйте отрегулировать освещение в комнате так, чтобы убрать тени. Чтобы датчик слабее реагировал на перепады освещённости из-за бликов на опорной поверхности, можно также попробовать изменить его положение, опустив ближе к опорной поверхности или задвинув дальше под днище робота.
9. После оптимизации робота для движения по линии с крутыми поворотами сохраните окончательную версию своего скетча. Обязательно записывайте любую важную информацию, данные, результаты и выводы в свой технический журнал.



**Совет:** Если контроллер PRIZM не исполняет уже загруженный скетч, попробуйте выключить-включить выключатель питания робота.

## Соревнование № 3 — Спасательные роботы

### Обзор

Министерство по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий (МЧС) мобилизует и отправляет обученных профессионалов и добровольцев в пострадавшие от бедствия районы для оказания помощи и устранения тяжелых последствий. Будь то ураганы, смерчи, землетрясения, наводнения, пожары или даже рукотворные бедствия, МЧС прибывает на место, чтобы оценить обстановку и поддержать первых спасателей.

Одна из главных задач МЧС — поиск и спасение потерпевших. Организация помогает пожарным и полицейским отыскать людей, оказавшихся в ловушке в повреждённых зданиях и обвалившихся строениях. Из-за вероятности обрушения повреждённых сооружений отправлять людей искать и выводить оттуда тех, кто остался в живых, опасно. Принимая во внимание совершенствование технологии, МЧС хотело бы разработать малоразмерных поисково-спасательных роботов, которые спокойно проникают в такие здания, оценивают состояние строения, находят выживших, оставляют им еду и питьё на время, которое пройдёт до прибытия спасателей.

Для сбора и оценки предлагаемых конструкций спасательных роботов МЧС устраивает соревнование, во время которого команды должны спроектировать, сконструировать и запрограммировать автономного поисково-спасательного робота. Повреждённым зданием будет соревновательная арена в виде лабиринта. Внутри здания будут находиться двое выживших, которых надо отыскать. Баллы присуждаются, исходя из того, какое расстояние в лабиринте прошёл робот, сколько времени он затратил на прохождение всего лабиринта, а также нашёл ли он искомым выживших и оставил ли им ёмкости с водой. Катуски в этом соревновании будут изображать ёмкости с водой. За наезды на стенки лабиринта, чреватые дальнейшим разрушением сооружения, будут сниматься баллы.

В соревнование спасательных роботов входит также оценка программного кода: за чистоту и эффективность программы начисляются дополнительные баллы по таблице в рубрике "Программирование". Рубрика "Программирование" с помощью которой будут распределяться баллы, находится на странице 69.



Министерство по чрезвычайным ситуациям

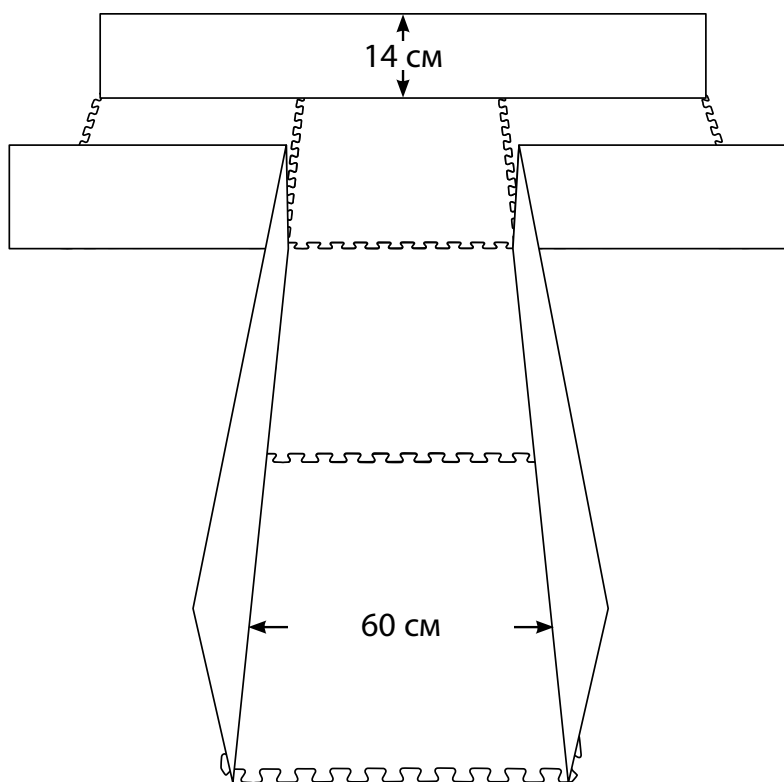
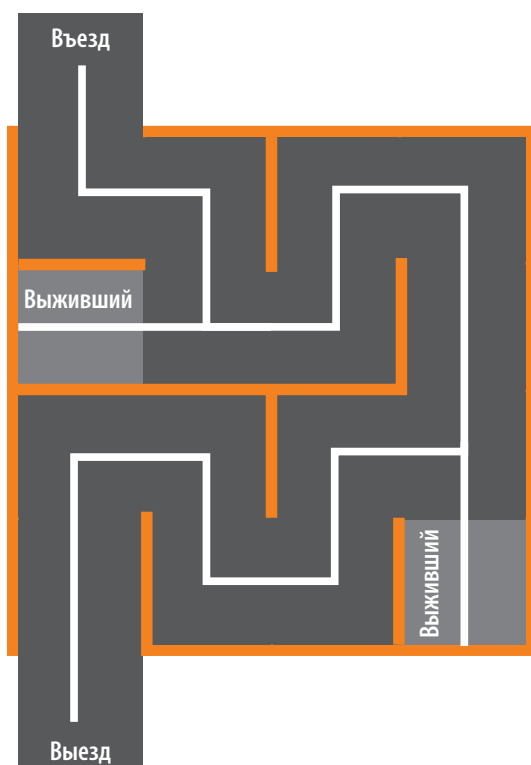


### Соревновательная арена

Соревновательная арена должна быть квадратом со стороной 2,44 метра. Внутри арены из должен быть лабиринт из бортиков, поставленных под прямым углом. Бортики лабиринта должны быть 14 см в высоту, а проезды лабиринта — 60 см в ширину. На полу, посередине проездов лабиринта, должна быть наклеена белая лента. Эта лента будет обозначать траекторию движения от въезда в лабиринт до выезда из него. Внутри лабиринта в двух местах должны находиться выжившие, которых предстоит найти. Только в местах нахождения выживших линия на полу должна касаться стенки лабиринта. Кроме того, пол в местах нахождения выживших, должен отличаться цветом от остальных частей лабиринта.

Окончательная схема лабиринта должна стать известна соревнующимся только в день соревнования. Показанная ниже схема — лишь пример того, как мог бы выглядеть лабиринт.

### Пример соревновательной арены



## Правила и учёт баллов

### Правила соревнований

- Роботы, предназначенные для соревнования, должны отвечать следующим условиям:
  - Командам разрешается использовать только детали из их конструктора TETRIX. Использовать что-то дополнительное можно только с разрешения распорядителя соревнования.
  - Командам запрещено гнуть или портить детали из конструктора TETRIX.
  - Робот не должен быть больше 45 см в ширину, 50 см в длину и 40 см в высоту.
  - Робот должен быть в состоянии перевезти две катушки и сбросить их выжившим перед выездом из лабиринта.
- Каждой команде предстоит спроектировать и сконструировать робота, который самостоятельно пройдёт по лабиринту. После пуска секундомера какое-либо вмешательство со стороны человека, кроме нажатия на кнопки "Пуск" и "Сброс параметров" в начале и конце лабиринта, запрещено.
  - Если какой-то робот начнёт безудержно ломать трассу, его остановит распорядитель соревнования. Будут подсчитаны баллы, и выступление команды на этом завершится.
- Накануне соревнования вы должны создать окончательный вариант скетча, загрузить его в робота и распечатать копию для учителя.
- На прохождение лабиринта каждому роботу отводится не больше 5 минут (300 секунд).
- Робот должен начать движение с назначенного стартового участка.
- Распорядитель соревнования вслух даст обратный отсчёт с цифры 3 и на слове "Марш" включит секундомер, а вам надо тут же нажать на пусковую кнопку на роботе.
- Время будет остановлено, когда робот полностью выедет из лабиринта в назначенном месте. Баллы будут присуждаться, исходя из времени, затраченного на достижение цели соревнования, и (или) пройденное для этого расстояние. Если робот станет неуправляемым или примется крушить трассу, его остановят, и время его выступления прекратится. Будут присуждены баллы за расстояние, пройденное до цели соревнования, но за время, потраченное на это, баллы не присуждаются.
- Ваш учитель выступает в роли распорядителя соревнования, за ним остаётся последнее слово при решении вопросов по справедливости.

### Правила начисления баллов:

- За каждую пройденную плитку (квадрат 61 x 61 см) лабиринта, включая обнаружение двух искомых выживших, присуждается 10 баллов. Наибольшее число баллов за пройденное расстояние может равняться 160 баллам (16 плиток × 10 баллов = 160 баллов) за прохождение всего лабиринта.
- Если пройден весь лабиринт, баллы присуждаются с учётом времени выполнения заезда. Число баллов и длительность заезда обратно пропорциональны, то есть чем быстрее робот пройдёт лабиринт, тем больше баллов заработает.
  - Баллы за длительность заезда начисляются по такой формуле:

$$\text{Баллов} = \frac{25\,000}{(\text{Время в секундах})}$$

- По 50 баллов присуждается за сброс катушки рядом с каждым искомым выжившим. Чтобы получить эти баллы, робот должен найти искомого выжившего и оставить катушку на том участке настла, который отличается цветом от остальных плиток настла в лабиринте.
- А за структуру, чистоту, эффективность и правильность управляющей программы можно также получить в общей сложности ещё 60 баллов. Баллы будут начисляться по таблице в рубрике "Программирование".

#### Правила начисления штрафных баллов:

- За каждый наезд робота на стенку или касание стенки: -5 баллов
  - Если робот долгое время касается стенки, тогда каждые 5-секунд такого касания будут считаться новым наездом, за который вычитаются дополнительные 5 баллов.
- За повреждение трассы могут вычитаться дополнительные баллы по усмотрению распорядителя соревнования.

#### Дополнительные сведения

##### Рубрика "Программирование"

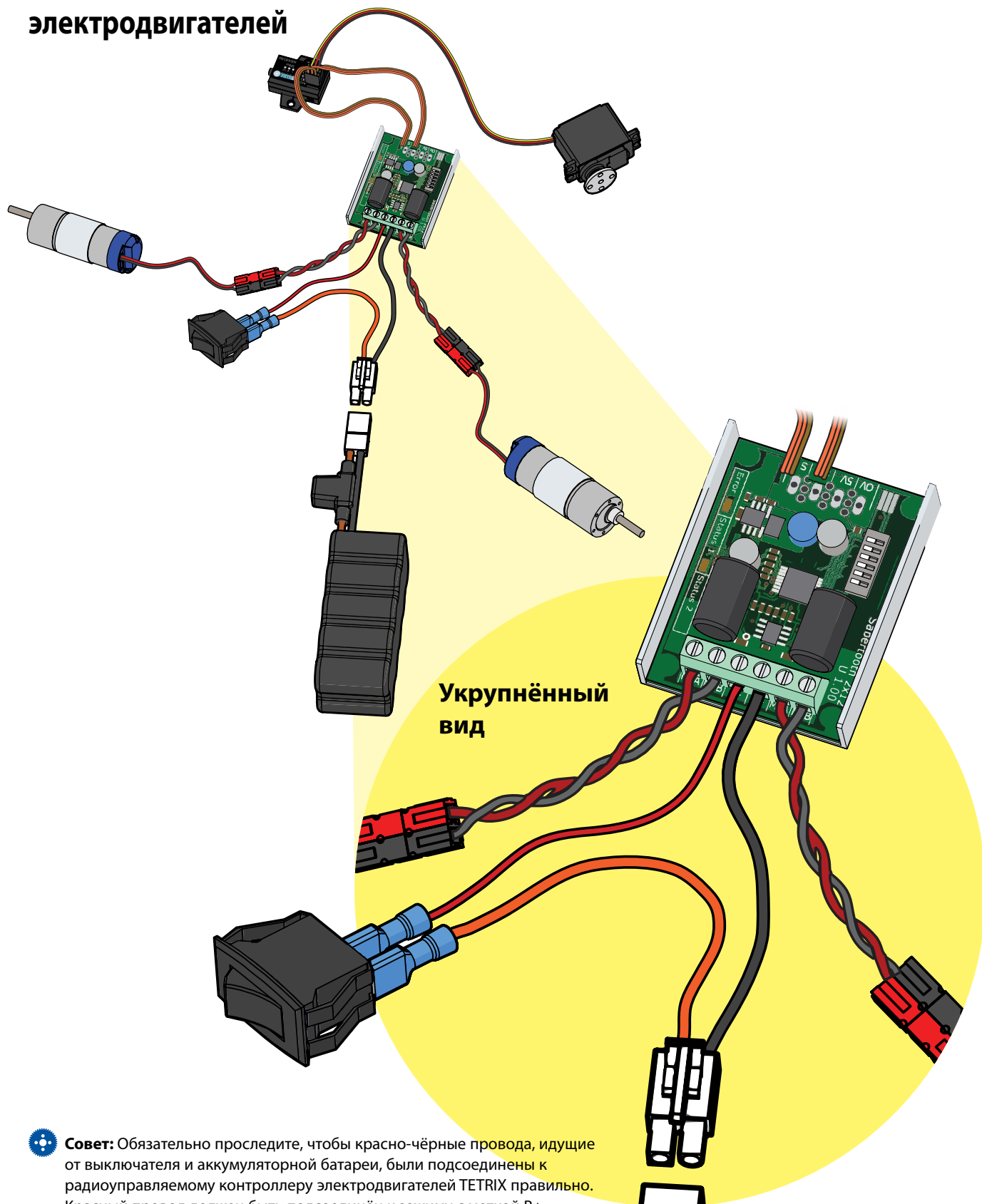
| Категория                                           | 4                                                                                                                                                                                                                                                                                     | 3                                                                                                                                                                                                                                                                                     | 2                                                                                                                                                                                                                                                                                     | 1                                                                                                                                                                                                                                                                                      | 0                                                                                                                                                                                                                                                                                              | Максимальное число баллов |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Псевдокод и комментарии                             | Псевдокод и комментарии используются во всех случаях, а базу управляющего кода нетрудно отследить.                                                                                                                                                                                    | Псевдокод и комментарии используются в большинстве случаев, а базу управляющего кода в общем нетрудно отследить.                                                                                                                                                                      | Псевдокод и комментарии используются в некоторых случаях, а базу управляющего кода в некотором усилии.                                                                                                                                                                                | Псевдокод и комментарии почти не используются, а базу управляющего кода трудно отследить.                                                                                                                                                                                              | Псевдокод и комментарии не используются.                                                                                                                                                                                                                                                       | 8 баллов<br>(Баллы x 2)   |
| Логическая схема и структура базы управляющего кода | База управляющего кода удовлетворяет всем требованиям: <ul style="list-style-type: none"> <li>• Команды выстроены в логическом порядке.</li> <li>• Синтаксис правильный.</li> <li>• Структура одинакова во всех случаях.</li> <li>• Функции команд используются правильно.</li> </ul> | База управляющего кода удовлетворяет трём требованиям: <ul style="list-style-type: none"> <li>• Команды выстроены в логическом порядке.</li> <li>• Синтаксис правильный.</li> <li>• Структура одинакова во всех случаях.</li> <li>• Функции команд используются правильно.</li> </ul> | База управляющего кода удовлетворяет двум требованиям: <ul style="list-style-type: none"> <li>• Команды выстроены в логическом порядке.</li> <li>• Синтаксис правильный.</li> <li>• Структура одинакова во всех случаях.</li> <li>• Функции команд используются правильно.</li> </ul> | База управляющего кода удовлетворяет одному требованию: <ul style="list-style-type: none"> <li>• Команды выстроены в логическом порядке.</li> <li>• Синтаксис правильный.</li> <li>• Структура одинакова во всех случаях.</li> <li>• Функции команд используются правильно.</li> </ul> | База управляющего кода не удовлетворяет ни одному требованию: <ul style="list-style-type: none"> <li>• Команды выстроены в логическом порядке.</li> <li>• Синтаксис неправильный.</li> <li>• Структура одинакова во всех случаях.</li> <li>• Функции команд используются правильно.</li> </ul> | 20 баллов<br>(Баллы x 5)  |
| Полнота и эффективность управляющего кода           | Управляющий код полный, чистый и эффективный. Отсутствуют ненужные строки и команды.                                                                                                                                                                                                  | Управляющий код полный, почти везде чистый и эффективный. Мало ненужных строк и команд.                                                                                                                                                                                               | Управляющий код полный и до некоторой степени чистый и эффективный. Есть несколько ненужных строк и команд.                                                                                                                                                                           | Управляющий код почти везде полный, но не чистый и не эффективный. Много ненужных строк и команд.                                                                                                                                                                                      | Неполный управляющий код.                                                                                                                                                                                                                                                                      | 20 баллов<br>(Баллы x 5)  |
| Использование вызываемых функций                    | Вызываемые функции всегда используются для повторяющихся задач.                                                                                                                                                                                                                       | Вызываемые функции почти всегда используются для повторяющихся задач.                                                                                                                                                                                                                 | Вызываемые функции иногда используются для повторяющихся задач.                                                                                                                                                                                                                       | Вызываемые функции редко используются для повторяющихся задач.                                                                                                                                                                                                                         | Вызываемые функции не используются для повторяющихся задач.                                                                                                                                                                                                                                    | 12 баллов<br>(Баллы x 3)  |

**Советы по проведению соревнований**

- Убедитесь, что полностью понимаете, как провести соревнование, и что для этого требуется.
- Прочтите целиком рубрику "Программирование", чтобы знать, к чему стремиться, программируя робота.
- Есть несколько способов набрать баллы в этом соревновании. Сосредоточьтесь на сильных сторонах вашего робота.
- Используйте описания и схему библиотечных функций Arduino для контроллера TETRIS PRIZM (см. приложение к *Руководству по программированию контроллера PRIZM*).
- Распорядитель соревнования подготовит к заключительному соревнованию небольшой испытательный лабиринт с похожими условиями. Испытайте в нём своего робота.
- Проектируя, конструируя и испытывая своего робота, следуйте алгоритму проектирования технических конструкций.
- Записывайте в технический журнал все результаты мозгового штурма, варианты проектов, процессов, решений, роли членов команды, ежедневные заметки и прочее.
- Проследите, чтобы на распечатке скетча, который сдаётся учителю, было название вашей команды и список тех, кто в неё входит. Свои имена и фамилии можно указать в комментариях внутри скетча или просто надписать на распечатке.



## Наглядная электромонтажная схема радиоуправляемого контроллера электродвигателей

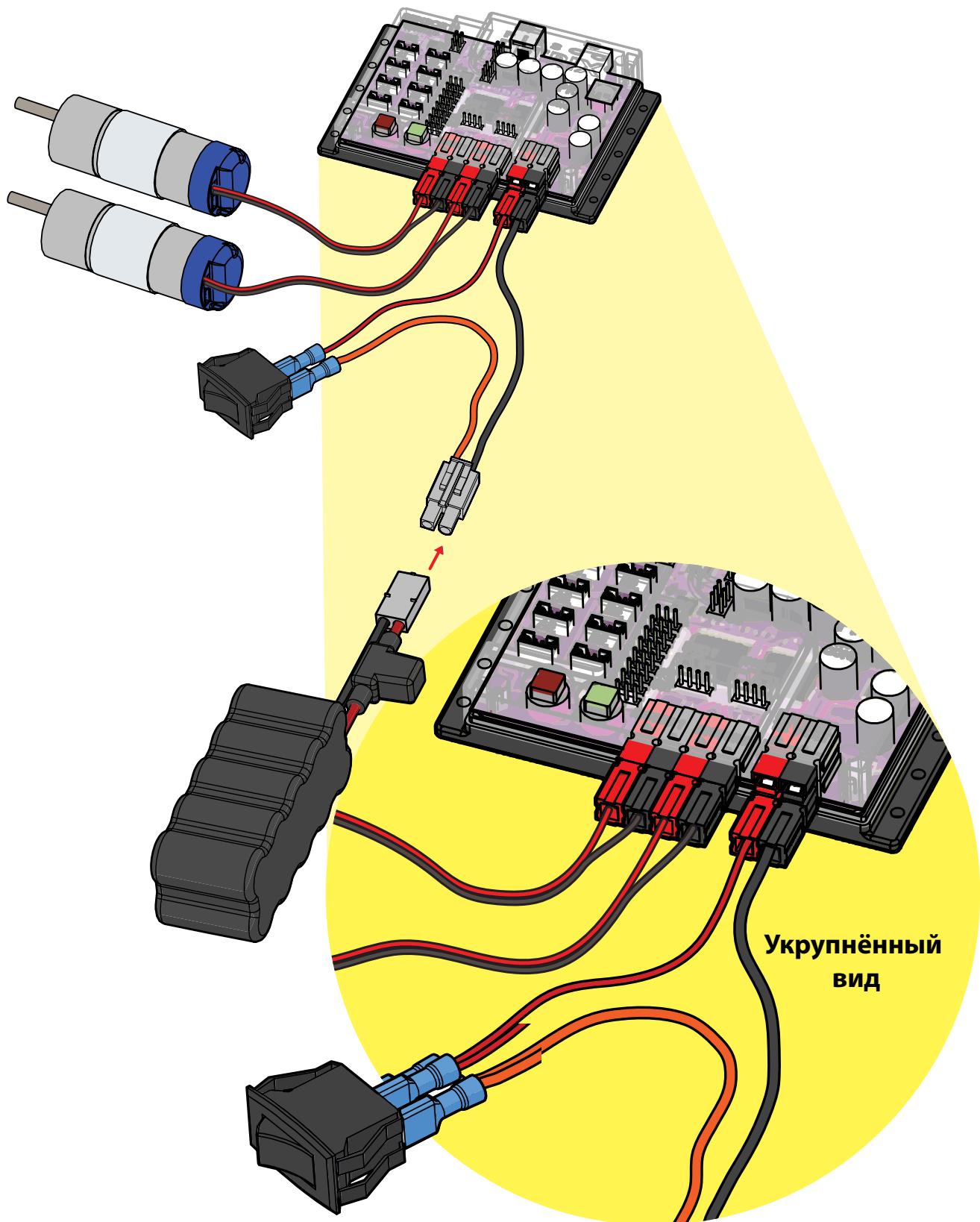


Укрупнённый вид



**Совет:** Обязательно проследите, чтобы красно-чёрные провода, идущие от выключателя и аккумуляторной батареи, были подсоединены к радиоуправляемому контроллеру электродвигателей TETRIX правильно. Красный провод должен быть подсоединён к зажиму с меткой V+, а чёрный провод — к зажиму с меткой V-.

# Наглядная электромонтажная схема контроллера PRIZM

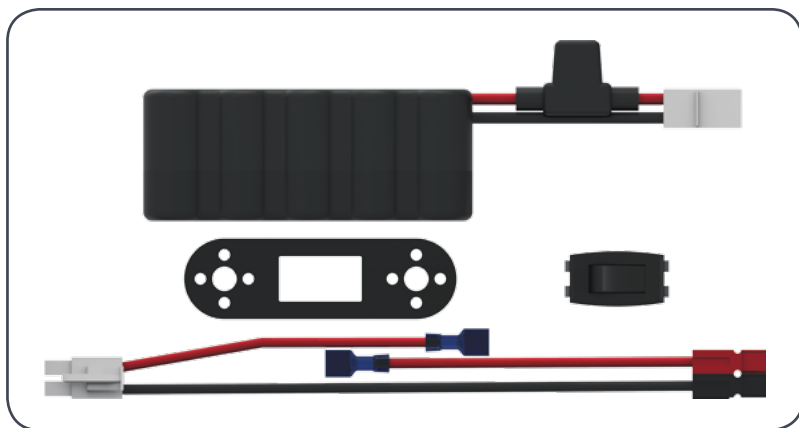


**Включение контроллера PRIZM:**

Для питания контроллера PRIZM используется перезаряжаемая никель-металлгидридная аккумуляторная батарея TETRIX напряжением 12 В.

К контроллеру PRIZM прилагается выключатель электропитания в сборе — через него аккумуляторная батарея подсоединяется к предназначенному для неё порту Powerpole, выполненному в стиле контроллера PRIZM.

Указания по монтажу батареи и комплекта выключателя электропитания приведены в упражнении с роботом-исполнителем из *Руководства по программированию контроллера PRIZM*.

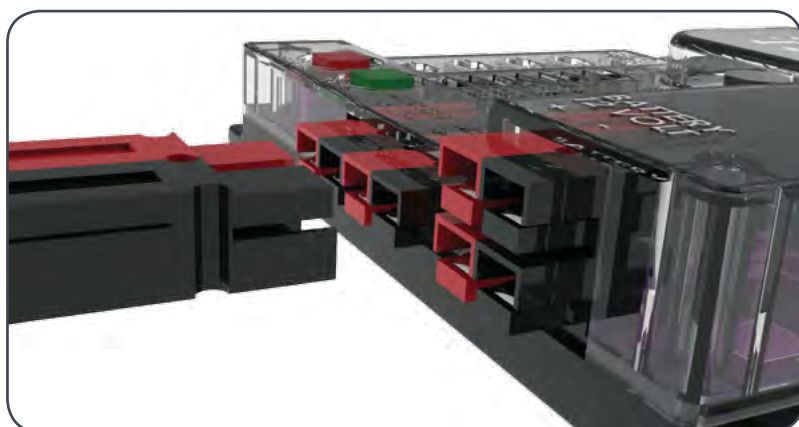


**Предупреждение!** Использовать с контроллером PRIZM аккумуляторные батареи сторонних производителей запрещено. Аккумуляторные батареи TETRIX защищены плавким предохранителем и являются единственными батареями аккумуляторов, пригодными для данной системы. При повреждении продукта вследствие нарушения данного требования гарантия аннулируется.

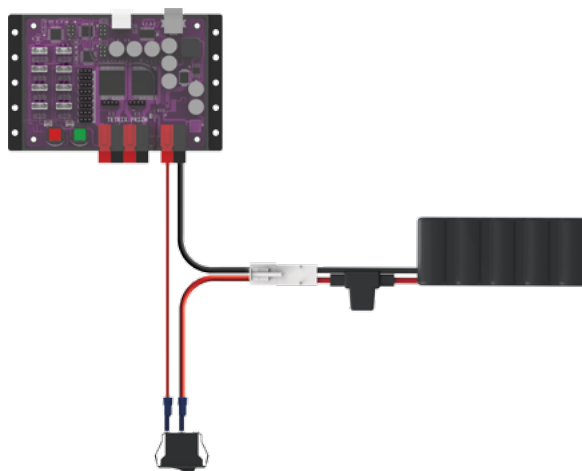
**Подключение аккумуляторной батареи к контроллеру PRIZM:**

Для подключения аккумуляторной батареи к контроллеру PRIZM вставьте её цветные соединители в один из двух предназначенных для этого портов на корпусе контроллера. Чтобы всё работало, соединители следует вставлять в порты того же цвета: КРАСНЫЕ в КРАСНЫЕ, а ЧЁРНЫЕ в ЧЁРНЫЕ. Избегайте выполнять подсоединение с обратной полярностью.

Аккумуляторную батарею можно подсоединить к верхнему или нижнему ряду портов.



**Предупреждение!** Подсоединять к контроллеру PRIZM сразу две аккумуляторные батареи запрещено. Такая попытка неминуемо приведёт к повреждению и отказу. Для электроснабжения цепи контроллера PRIZM и последовательной раздачи электроэнергии любым дополнительно подключаемым контроллерам используйте только одну аккумуляторную батарею.



## Предусмотренные для робота варианты управления и навигации

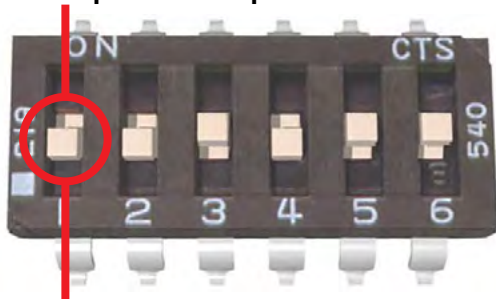
### Функции контроллера электродвигателей

Чтобы роботом было проще управлять, в радиоуправляемом контроллере электродвигателей предусмотрены разные режимы. Найдите на контроллере электродвигателей восемь переключателей. Переводить переключатели вверх-вниз можно небольшой отвёрткой.

#### Режимы бортового и танкового поворота

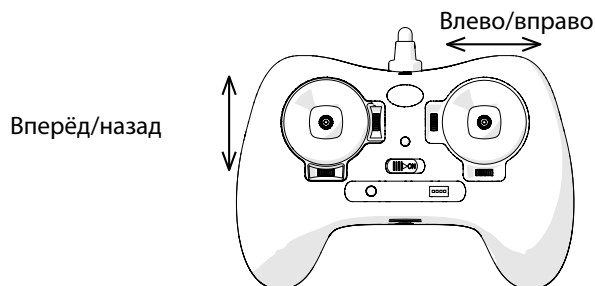
В верхнем положении переключатель № 1 на контроллере электродвигателей включает режим бортового поворота, в нижнем — режим танкового поворота.

Вверх: режим бортового поворота

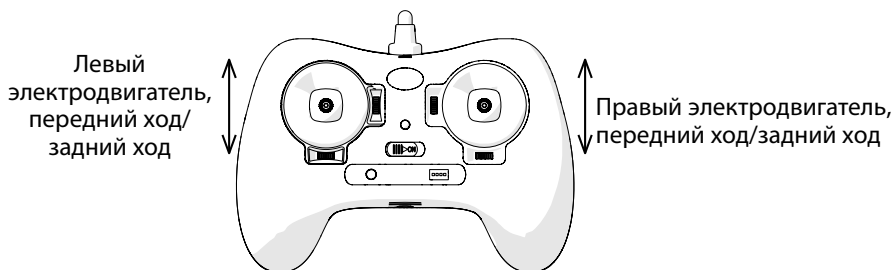


Вниз: режим танкового поворота

**Режим бортового поворота** — один из наиболее распространённых видов дистанционного управления. При таком варианте один джойстик на пульте управления управляет передним и задним ходом робота. Переднее и заднее вращение обоих электродвигателей регулируется по одному каналу на пульте управления; обычно это 2-й канал. Отдельный канал, обычно 1-й или 4-й, отвечает за повороты робота из стороны в сторону.



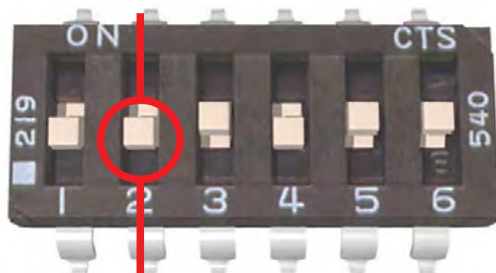
В **режиме танкового поворота** каждый электродвигатель управляется по отдельности двумя разными джойстиками на пульте управления. Обычно 2-й канал управляет передним и задним ходом левого электродвигателя, а 3-й канал управляет передним и задним ходом правого электродвигателя. Для движения передним ходом оба джойстика следует с равной скоростью отклонить вперёд. Для поворота следует отклонить только один джойстик.



### Экспоненциальный и линейный режимы

Переключатель № 2 на контроллере электродвигателей переводит контроллер в экспоненциальный режим, когда поднят, и в линейный, когда опущен.

#### Экспоненциальный режим



#### Линейный режим

В **экспоненциальном режиме** контроллер менее чутко откликается на близкие к вертикальному положения джойстиков. Обороты электродвигателей увеличиваются экспоненциально отклонению джойстика. Так проще управлять мелкими движениями робота и легче добиться управляемости от быстродействующих роботов.

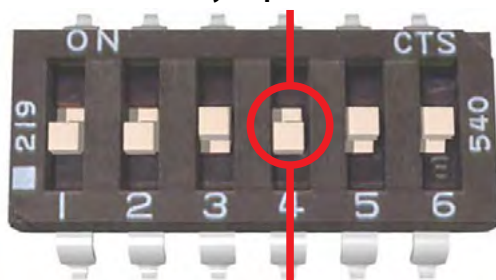
В **линейном режиме** чуткость джойстика меняется линейно. Обороты электродвигателя увеличиваются с постоянной скоростью по мере отклонения джойстика от центрального положения. В результате робот увеличивает скорость передвижения.



### Линейное ускорение

Переключатель № 4 на контроллере электродвигателя включает и выключает линейное ускорение.

#### Линейное ускорение выкл.



#### Линейное ускорение вкл.

Когда включено **линейное ускорение**, контроллер электродвигателей будет плавно переключать скорость приблизительно за 0,25 секунды. Линейное изменение параметров позволяет управлять роботом плавнее. Если хотите включить линейное ускорение, переведите переключатель № 4 в нижнее положение, а если хотите выключить — в верхнее.

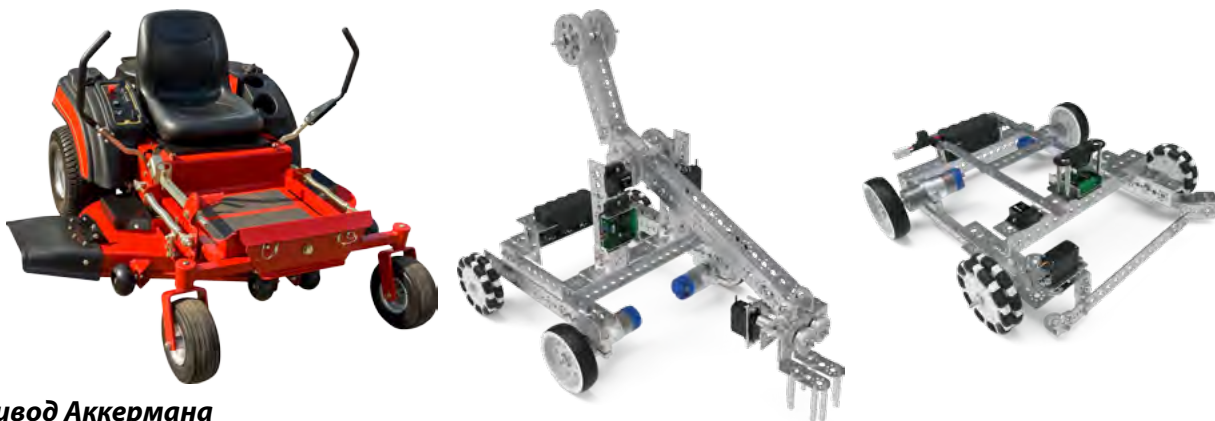
**Важное примечание!** Переключатели № 3, 5 и 6 на роботах TETRIX MAX должны всегда быть в верхнем положении. Переключатель № 3 имеет отношение к типу батареи, используемой на роботе. Переключатели № 5 и 6 имеют отношение к калибровочным параметрам. Откалибровать каналы управления на вашем роботе можно при помощи подстроечных колёсиков на пульте управления.

## Приводные механизмы и способы управления поворотами

**Приводной механизм** робота — это система, заставляющая робота передвигаться и поворачивать. На соревнованиях приводной механизм робота требует самого пристального внимания. Приводной механизм обеспечивает управление скоростью, мощностью, силой тяги, манёвренностью и радиусом поворота робота. Без хорошего приводного механизма управлять роботом, а значит добиться успеха в соревновании, будет трудно. Проектируя приводной механизм, рассмотрите несколько его типов. Одни приводные механизмы работают лучше других в определённых соревнованиях. Другие приводные механизмы безусловно устроены сложнее, потому что в их состав входит больше электродвигателей и сервоприводов. Выбор приводного механизма — одно из наиболее важных решений при проектировании робота.

### Прямой привод

В роботах для соревнований наиболее часто используют схему рулевого управления с **прямым приводом**. У робота с прямым приводом два ведущих колеса вращаются независимо двумя разными электродвигателями. Продвижение и повороты зависят от того, с какой скоростью вращаются электродвигатели по отдельности. Одно из достоинств прямого привода заключается в том, что он помогает совершать все виды поворотов, от самых постепенных с большой амплитудой, до разворотов на месте, когда робот поворачивает вокруг своей вертикальной оси. Недостаток прямого привода заключается в том, что на высоких скоростях роботом бывает трудно управлять. В газонокосилках с нулевым радиусом поворота используется прямой привод, благодаря которому газонокосилка может огибать деревья и другие препятствия под крутым углом. Каждое колесо управляется независимо отдельным рычагом.



### Привод Аккермана

**Рулевой привод Аккермана (с поворотными кулаками и трапецией)** широко используется в легковых и грузовых автомобилях. Одну пару колёс одновременно поворачивают рулевые тяги, используемые для управления поворотами. Рулевой привод Аккермана значительно улучшает управляемость поворотами при движении на высоких скоростях. А недостаток заключается в том, что поворачивать так же круто, как в случае рулевого управления с прямым приводом, даже близко не получится. У роботов с рулевым приводом Аккермана, положением передних управляемых колёс управляет стандартный сервопривод, а ведущие задние колёса вращает по меньшей мере один электродвигатель постоянного тока. Как правило, направляющие колёса не проводят автомобиль в движение.



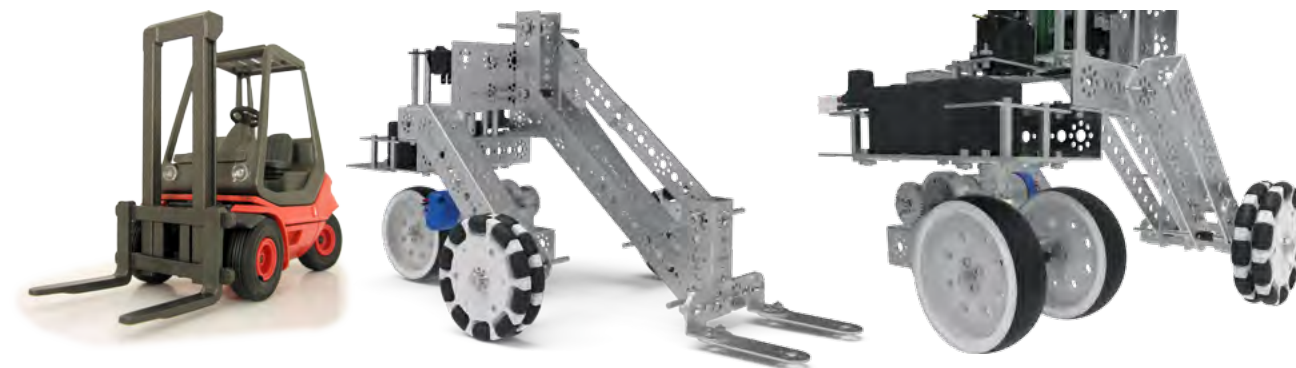
### **Шарнирное сочленение**

Роботы, оснащённые рулевым приводом с **шарнирно-сочленённым шасси**, могут изгибаться посередине — там у них ось поворота. Управление осью вращения с помощью сервопривода высвобождает мощность электродвигателей для других применений. В некоторых мощных тракторах сельскохозяйственного назначения и строительных машинах используется рулевой привод с шарнирно-сочленённым шасси. Но у шарнирно-сочленённых роботов часто увеличенные радиусы поворота. Кроме того, на высоких скоростях робот может потерять устойчивость из-за шарнирного сочленения.



### **Привод с ведущими полноповоротными колёсами**

**Рулевой привод с ведущими полноповоротными колёсами** снабжён полноповоротными колёсами, способными менять угол атаки. Робот, оснащённый рулевым приводом с ведущими полноповоротными колёсами, способен двигаться в заданном направлении, при этом вращаясь вокруг собственной вертикальной оси. Построить привод с ведущими полноповоротными колёсами можно разными способами, иногда довольно сложными, особенно, если у робота все колёса — ведущие полноповоротные. Но можно и упростить привод с ведущими полноповоротными колёсами, оставив для управления поворотами только одно такое колесо или один комплект таких колёс. Для приведения ведущего полноповоротного колеса во вращение можно было бы использовать один электродвигатель постоянного тока, а для поворота такого колеса в ту или иную сторону — один стандартный сервопривод. У некоторых малогрузных вилочных погрузчиков сзади установлен рулевой привод с единственным ведущим полноповоротным колесом, который и приводит машину в движение, и поворачивает её.



### **Привод с всенаправленными роликовыми колёсами**

Есть несколько разных видов рулевых приводов с всенаправленными роликовыми колёсами. Такие приводные механизмы обеспечивают роботу высокую манёвренность и проворство. Однако нередко приходится жертвовать мощностью толкания. Кроме того, проектировать и конструировать их бывает сложнее, а для приведения в действие может потребоваться больше двух электродвигателей постоянного тока. К общеупотребительным типам приводов с всенаправленными роликовыми колёсами относятся, например, рулевые приводы с двумя комплектами поперечно направленных колёс, приводы поперечного скольжения и голономные приводы.



## Зубчатые передачи

**Зубчатая передача** — это ещё одна система, облегчающая управление роботом. Зубчатые передачи могут прибавить роботу скорости, увеличить его крутящий момент, изменить направление вращения или передать вращательное движение в другое место сообразно компоновке и положению зубчатых колёс. В зубчатой передаче **ведущая шестерня** (иногда называемая первичной шестернёй) прикреплена к валу электродвигателя. Затем ещё одно зубчатое колесо, называемое ведомой или вторичной шестернёй, входит в зацепление с ведущей шестернёй.

**Ведомая шестерня** обычно соединена с колесом или другим вращающимся механизмом.

Отношение числа зубьев ведомой шестерни к числу зубьев ведущей шестерни называется **передаточным отношением**. Передаточное отношение часто записывают в виде Ш1:Ш2, где Ш1 — число зубьев ведомой шестерни, а Ш2 — число зубьев ведущей шестерни. Допустим, у вас есть зубчатая передача, составленная из шестерён, взятых из конструктора TETRIX MAX, в которой у ведомой шестерни 80 зубьев, а у ведущей — 40. Тогда передаточное отношение можно записать как 80:40. Но обычно отношение сводится к простейшему выражению, так что для этой зубчатой передачи отношение будет 2:1. Если записать передаточное отношение в виде простой или десятичной дроби, тогда передаточное отношение будет равно выигрышу в силе, который обеспечивает эта система.

$$\text{Передаточное отношение} = \frac{\text{Зубья на ведомой шестерне}}{\text{Зубья на ведущей шестерне}} = \text{выигрыш в силе}$$

### **Редуктор (уменьшает обороты/увеличивает крутящий момент)**

Если число зубьев на ведомой шестерне больше, чем на ведущей, тогда у системы выигрыш в силе больше единицы. Вращение ведущей шестерни замедляет ведомую, в результате снижается частота вращения колеса вращательного механизма, прикреплённого к ведомой шестерне. Но от понижения оборотов есть польза: возрастает крутящий момент, то есть вращательная сила. Применение зубчатой передачи для увеличения крутящего момента ценой уменьшения оборотов — типичный инженерный компромисс. Обдумайте использование зубчатой передачи в виде редуктора в роботе, которому нужна большая мощность для толкания или буксировки.

**Передаточное отношение = 2:1**



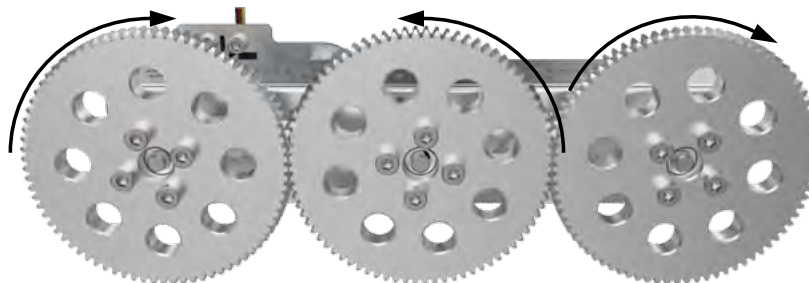
**Ведущая шестерня**

**Ведомая шестерня**

### **Механизм изменения направления вращения**

Если у ведущей и ведомой шестерён одинаковое число зубьев, тогда обе шестерни вращаются с одинаковой скоростью. Передаточное отношение 1:1, а выигрыш в силе у системы равен 1. Частота вращения ведущей шестерни равна частоте вращения ведомой шестерни. Теоретически крутящий момент на ведущей шестерне равен крутящему моменту на ведомой шестерне, если не принимать во внимание трение в системе. Единственная польза от такого типа зубчатой передачи — её способность менять направление вращения или передавать вращение в другое место.

**Передаточное отношение = 1:1**



**Ведущая шестерня**

**Ведомая шестерня**



### Мультипликатор (увеличивает обороты/уменьшает крутящий момент)

Если число зубьев на ведомой шестерне меньше, чем на ведущей, тогда у системы выигрыш в силе меньше единицы. Это означает, что ведомая шестерня должна вращаться быстрее ведущей. Поскольку этот механизм увеличивает обороты, его иногда называют мультипликатором. Но ведущей шестерне труднее вращать ведомую, так что крутящий момент уменьшается. Эту зубчатую передачу можно также назвать мультипликатором. Обдумайте использование зубчатой передачи в виде мультипликатора в роботе, который должен двигаться очень быстро.

Передаточное отношение = 1:2

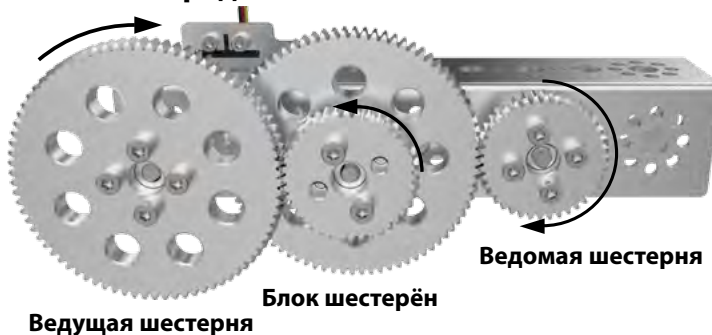


### Сложные зубчатые передачи с блоками шестерён

Сложные зубчатые передачи с блоками шестерён можно использовать для дополнительного увеличения оборотов или момента в зависимости от компоновки зубчатой передачи. В **блоке шестерён** не менее двух шестерён, соединённых на общем валу. Для смены передаточного отношения у этих зубчатых колёс обычно разное число зубьев.

В примере показана ведущая шестерня (80 зубьев), соединённая с ведомой шестернёй (40 зубьев) на блоке шестерён. Передаточное отношение двух этих звеньев 1:2, то есть мы имеем дело с мультипликатором — блок шестерён будет вращаться вдвое быстрее ведущей шестерни. Блок шестерён состоит из насаженных на один вал двух шестерён: с 40 и 80 зубьями. Эти две шестерни вращаются с одинаковой скоростью. Шестерня с 80 зубьями в составе блока шестерён становится ведущей и входит в зацепление с другой ведомой шестернёй, имеющей 40 зубьев. В результате во второй половине зубчатой передачи тоже складывается передаточное отношение 1:2. У этого сложного зубчатого механизма с блоком шестерён два передаточных отношения 1:2 и 1:2, произведение которых даёт общее передаточное отношение 1:4. У последней ведущей шестерни четверть крутящего момента ведущей шестерни, но при этом обороты вчетверо больше оборотов ведущей шестерни.

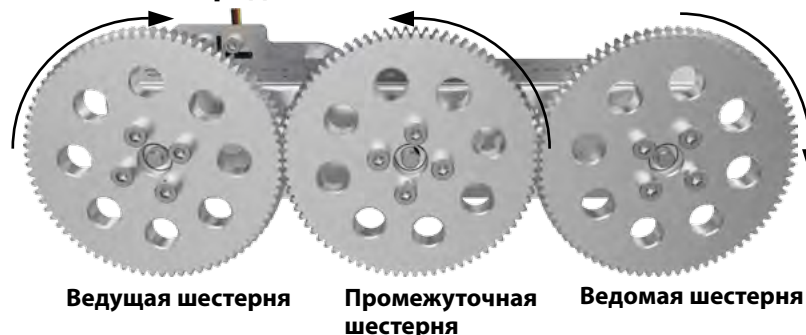
Передаточное отношение = 1:4



### Промежуточные шестерни

В некоторых приводных механизмах между ведущей и ведомой шестернями есть **промежуточные шестерни**. Эти зубчатые колёса меняют направление вращения и соединяют далеко разнесённые ведущую и ведомую шестерни. Промежуточные шестерни не влияют на передаточное отношение системы.

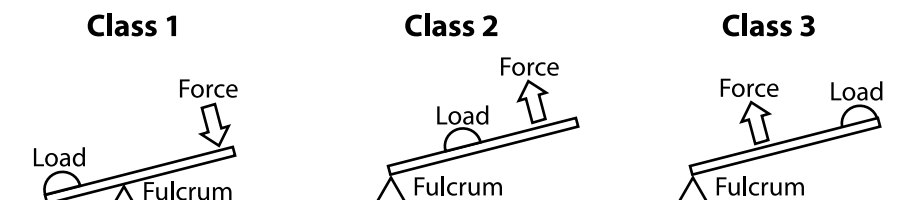
Передаточное отношение = 1:1



## Робототехнические подъёмные механизмы

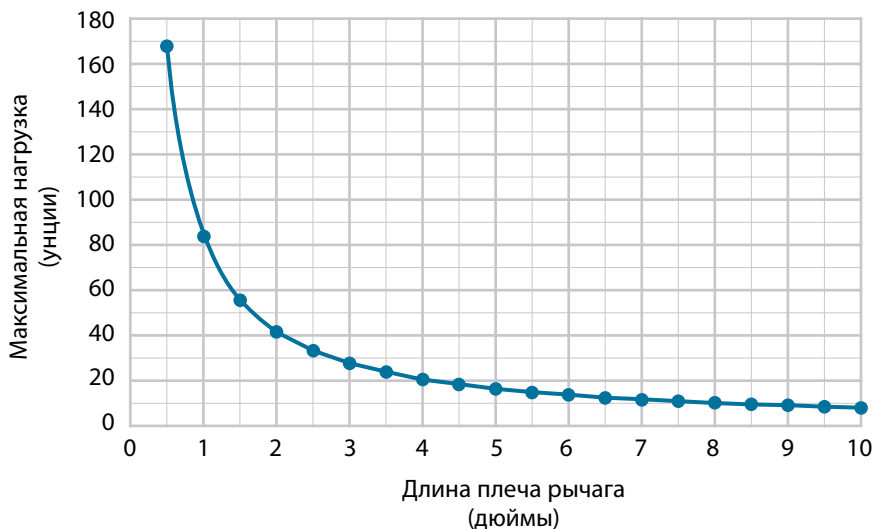
### Рычаги

Во многих роботах, создаваемых для соревнований, в качестве подъёмной системы используются рычаги. Рычаги относятся к числу шести простых механизмов и состоят из двух главных частей: плеча рычага и точки опоры (оси вращения). Подъём происходит именно благодаря плечу рычага, а **точка опоры** — это такая точка, вокруг которой рычаг вращается или поворачивается. В зависимости от положения точки опоры, нагрузки (поднимаемого веса) и места приложения силы различают три рода рычагов. Большинство роботизированных подъёмников представляют собой рычаги либо первого, либо третьего рода.



К плечу рычага необходимо приложить силу, чтобы поднять им груз. Робот обычно получает эту силу от электродвигателя или сервопривода. Вес, который может поднять рычаг зависит от мощности сервопривода и длины плеча рычага. Длина плеча нагрузки и величина груза обратно пропорциональны, то есть чем длиннее плечо рычага со стороны груза, тем меньший груз способен поднять рычаг. У стандартного сервопривода из конструктора TETRIX MAX номинальный крутящий момент равен 0,59 Н·м. То есть с плечом рычага длиной 2,54 см сервопривод сможет поднять вес 2,38 кг. Если удлинить плечо рычага на 5 см, то сервопривод сможет поднять вес наполовину меньше: 1,19 кг. Вот график сравнения длины плеча рычага и максимального груза, посильного для стандартного сервопривода из конструктора TETRIX MAX.

**Нагрузка стандартного сервопривода с учётом соотношения длины и грузоподъёмности плеча рычага**

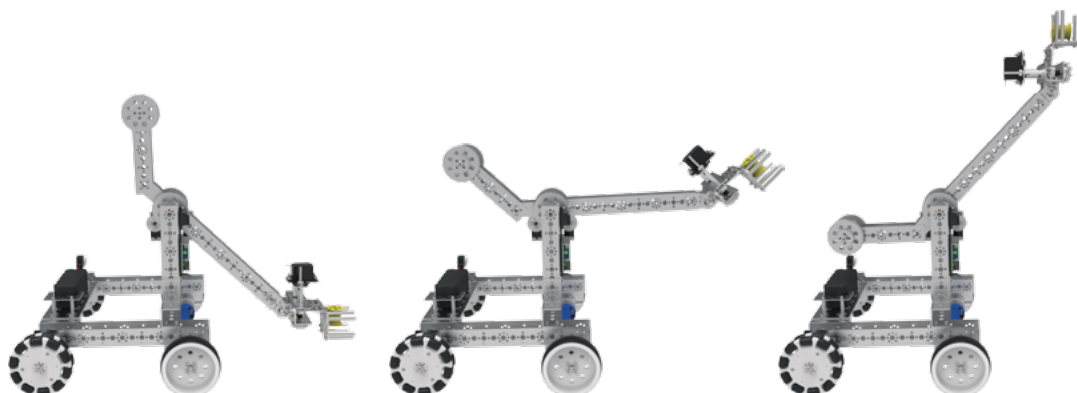


Определяя, насколько большой вес может поднять сервопривод с помощью плеча рычага, необходимо учесть вес самого рычага.

При использовании рычагов в качестве подъёмных стрел следует учесть несколько вариантов.

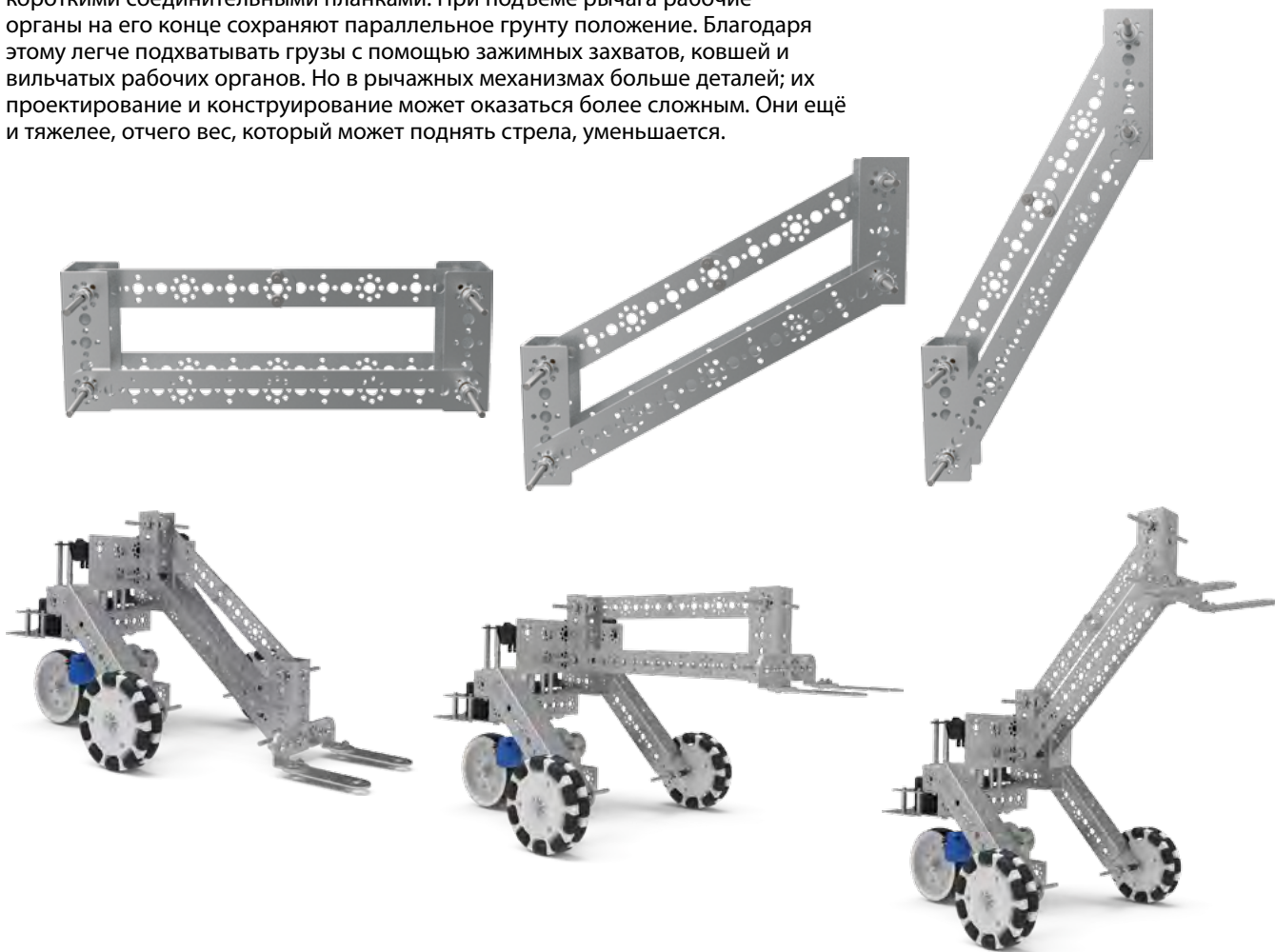
### ***Стрелы с вертикальным вращением***

С обычно состоят из одиночной стрелы, которая вращается вокруг опорной точки. Грузоподъёмные органы, прикреплённые к концу стрелы, также вращаются вокруг этой опорной точки. У стрелы с вертикальным вращением есть достоинства и недостатки. К достоинствам относятся лёгкость проектирования и конструирования, использование меньшего числа деталей, то есть, как правило, облегчённость конструкции, а также то, что при подъёме стрелы с вертикальным вращением грузоподъёмные органы отклоняются назад и фиксируют грузы, опирая их на заднюю часть рабочего органа. Впрочем, вращательное движение стрелы может быть и недостатком. Когда стрела с вертикальным вращением опускается, грузоподъёмный орган наклоняется вниз, что затрудняет подхват некоторых грузов. Могут возникнуть затруднения и при штабелировании грузов, поскольку основание поднимаемого груза непараллельно верху штабеля.



### Параллельно соединённые рычажные механизмы

**Параллельно соединённые рычажные механизмы** — ещё один типичный для робототехнических соревнований вид подъёмного рычага. Эти подъёмные рычаги состоят из параллельных балок, скреплённых с обоих концов короткими соединительными планками. При подъёме рычага рабочие органы на его конце сохраняют параллельное грунту положение. Благодаря этому легче подхватывать грузы с помощью зажимных захватов, ковшей и вилчатых рабочих органов. Но в рычажных механизмах больше деталей; их проектирование и конструирование может оказаться более сложным. Они ещё и тяжелее, отчего вес, который может поднять стрела, уменьшается.



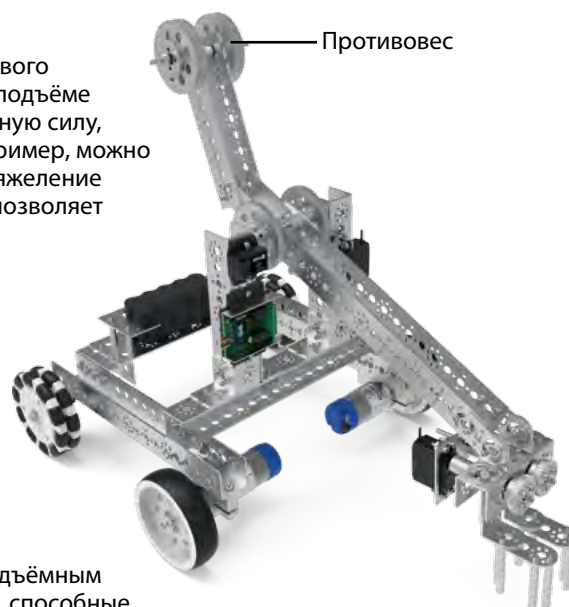
### Лифты

Лифты — ещё одна категория подъёмных систем, часто используемая в робототехнических соревнованиях. Лифтовые системы обычно компактнее и, если сконструированы правильно, поднимают грузы гораздо выше подъёмных рычагов. Но для лифтов часто требуются нестандартные детали TETRIX MAX, которые не входят в набор для создания автономных и управляемых моделей. Для обеспечения подъёма часто используют цепи и звёздочки, механизмы реечной передачи или комплекты гусеничных лент. В наборах, используемых на ваших уроках, может и не оказаться этих спецдеталей. Если захотите собрать какой-нибудь мачтовый подъёмник, придётся использовать то, что найдётся в наборе.



## Противовес и пассивное усиление

Исходя из конструктивной схемы подъёмного рычага или мачтового подъёмника, может понадобиться противовес. **Противовес** при подъёме помогает сервоприводу, силой тяжести усиливая его вращательную силу, называемую также крутящим моментом. Подъёмный рычаг, например, можно удлинить за точку опоры и превратить его в рычаг 2-го рода. Утяжеление ненагруженного конца рычага помогает уравновесить рычаг и позволяет стреле поднять более тяжёлый груз.



Увеличить грузоподъёмность сервопривода, соединённого с подъёмным рычагом, можно и с помощью **пассивного усиления**. Предметы, способные упруго растягиваться или сжиматься, вроде пружин и резиновых колец, можно использовать для получения дополнительной силы, помогающей сервоприводу поднимать груз. Например, вместо противовеса на подъёмном рычаге можно было бы закрепить резиновые ленты, растягивающиеся по мере опускания рычага к грузу. Резиновые ленты сообщают подъёмному рычагу при движении вверх дополнительную силу.



## Робототехнические грузозахватные механизмы

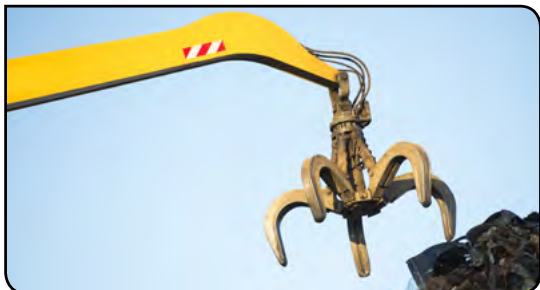
### Зажимные захваты

Пальцеобразные захваты относятся к наиболее часто используемым рабочим органам роботов во время соревнований. Пальцеобразный захват состоит из двух и более разнесённых механических пальцев на шарнирах, которые с разных сторон смыкаются вокруг груза. Пальцы сжимаются на грузе и удерживают его, пока робот перемещает груз в новое место. Как правило, при сближении робота с предметом, который предстоит поднять, зажимные захваты берут этот предмет горизонтально, сбоку. Хороший пример зажимного захвата есть у робота с поворотным захватом TETRIS MAX, которого вы уже наверняка собирали, когда проходили *Руководство по сборке управляемых моделей*.



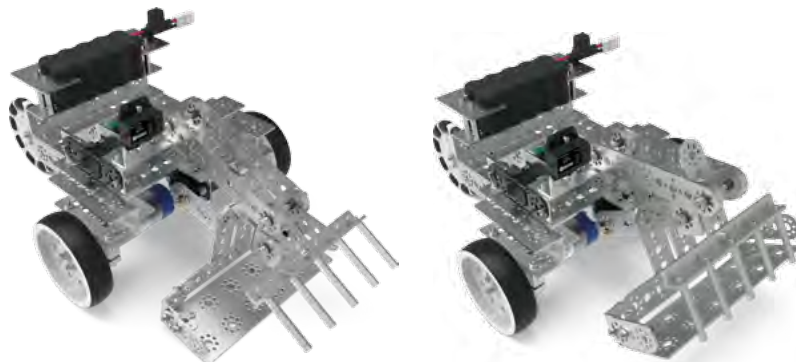
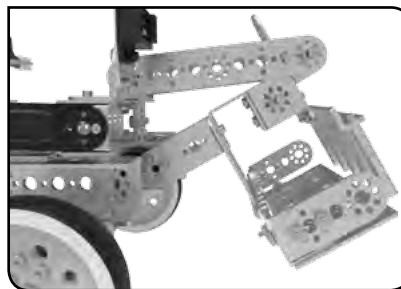
## Когтевые и грейферные захваты

Когтевые и грейферные захваты похожи на зажимные захваты, только обычно их подводят к грузу сверху, а не сбоку. Вспомните когтевые грузозахватные приспособления, которые вы наверняка видели в игровых автоматах, где предлагается ухватить приз и перенести его к месту сброса. Робототехнический когтевой захват действует почти так же. Робот подводит когтевой захват сверху к грузу, при помощи подъёмной системы опускает, смыкает вокруг груза и поднимает груз, чтобы перенести в другое место. Когтевые механизмы могут быть простыми, как двухпальцевый зажим, повёрнутый вверх, или сложными, как захватное устройство для металлолома, используемое на станции переработки металлических отходов.



## Ковши

Ковши — ещё одно замечательное приспособление для подхвата и подъёма грузов. Ковши подводят под груз снизу, как бывает при работе на экскаваторе. Преимущество от использования ковшей заключается в том, что можно спроектировать ковши с увеличенной грузоподъёмностью. Зажимной или когтевой захват обычно удерживает только один предмет, а ковш — несколько. Впрочем, ковши ещё и весят больше, а значит для их подъёма нужны более мощные сервоприводы. Ещё один недостаток ковшей заключается в том, что их бывает непросто подвести под грузы, предназначенные для подъёма. Чтобы подцепить эти грузы, их иногда приходится придвигать к стене или другой преграде. Есть один способ обойти это: добавить захватную челюсть, которая, опускаясь сверху, подтягивает груз внутрь ковша.



## Толковый словарь

**линейное ускорение:** вариант настройки контроллера электродвигателей, при котором контроллер электродвигателей будет переключать обороты плавно.

**ловушка:** робототехническая система для сбора и накопления предметов

**рулевой привод Аккермана (с поворотными кулаками и трапецией):** Система рулевого управления, в котором поворотом спаренных колёс на шкворнях управляют рулевые тяги в виде трапеции;

**рулевой привод с шарнирно-сочленённым шасси:** система рулевого управления с вертикальным шарниром в середине шасси, благодаря которому шасси робота может изгибаться

**автономный:** действующий самостоятельно; не подчиняющийся внешнему управлению

**автономный робот:** робот, совершающий действия под управлением микропроцессора, без вмешательства человека

**замкнутая система:** система, в которой для её управления и получения желательных результатов без вмешательства человека используются механизмы обратной связи

**блок шестерён:** не менее двух шестерён, соединённых на одном валу; используется для переключения передач

**ограничение:** некий предел или рамки, часто обусловленные наличными ресурсами

**противовес:** утяжеление, помогающее при подъёме увеличить прилагаемую силу или момент силой тяжести

**критерии:** желательные результаты проекта, технического решения или конкурсного задания

**счисление пути:** процесс, позволяющий определить будущее положение на основе направления движения и ранее пройденного расстояния

**прямой привод:** система рулевого управления, в которой два ведущих колеса вращаются независимо двумя разными электродвигателями; передвижение и повороты зависят от частоты вращения электродвигателей по отдельности

**ведущая шестерня:** силовая шестерня в зубчатой передаче, вращающая ведомую шестерню; иногда называется первичной шестернёй

**ведомая шестерня:** несиловая шестерня в зубчатой передаче, вращаемая ведущей шестернёй; иногда называется вторичной шестернёй

**приводной механизм:** система, позволяющая роботу передвигаться и поворачивать; регулирует скорость, мощность, силу тяги, манёвренность и радиус поворота

**экспоненциальный режим:** вариант настройки контроллера электродвигателей, при котором у джойстика в средних положениях менее чуткий отклик, позволяющий более тонко управлять движениями; обороты электродвигателей возрастают экспоненциально расстоянию смещения джойстика

**механизм обратной связи:** процесс, снабжающий систему информацией о том, как изменить входные элементы и алгоритмы их обработки, чтобы получить искомые выходные элементы

**точка опоры:** точка вокруг которой вращается или поворачивается рычаг

**передаточное отношение:** отношение числа зубьев ведомой шестерни к числу зубьев ведущей шестерни; определяет поведение зубчатой передачи

**зубчатая передача:** система зубчатых колёс, которая регулирует частоту вращения, крутящий момент, направление вращения или определяет место вращательного движения

**идеация:** процесс порождения идей

**промежуточная**

**шестерня:** зубчатое колесо, которое меняет направление вращения и соединяет далеко разнесённые ведущую и ведомую шестерни

**новшество:** некая техническая идея, в основе которой лежат существовавшие ранее изделия или услуги.

**создать экземпляр:** термин программирования, означающий "распознать предопределённый объект"

**изобретение:** совершенно новая идея или изделие

**линейный режим:** вариант настройки контроллера электродвигателей, при котором чуткость джойстика меняется линейно; при отклонении джойстика от центрального положения обороты электродвигателей возрастают на одну и ту же величину

**разомкнутая система:** система, регулируемая человеком, который контролирует поступление входных элементов, оценивает выходные элементы и видоизменяет алгоритмы их обработки с целью получения искомых результатов

**оптимизация:** постоянная оценка конструктивной схемы изделия или алгоритма обработки с учётом проектных критериев и ограничений с последующим улучшением изделия

**параллельно соединённый рычажный механизм:** подъёмный рычаг, состоящий из параллельных балок, скреплённых короткими соединительными планками с обоих концов; при подъёме рабочие органы на конце рычага остаются параллельными земле

**пассивное усиление:** несилловые звенья системы, служащие источником дополнительной силы или крутящего момента при подъёме или передвижении.

**прототип:** рабочая модель придуманной конструктивной схемы.

**псевдокод:** описание шагов (с помощью простого языка), которые должны выполняться в компьютерной программе

**научные исследования и опытно-конструкторские разработки:** разработка технических систем и подготовка их к выводу на рынок, где потребители, возможно, захотят их купить и использовать

**режим бортового поворота:** способ управления робототехнической моделью, при котором один джойстик на пульте управления отвечает за передвижение робота передним и задним ходом, одновременно включая оба электродвигателя; при этом за поворот робота из стороны в сторону отвечает отдельный джойстик

**технические условия:** подробное описание точных требований, предъявляемых к разработке

**рулевой привод с ведущими полноповоротными колёсами:** рулевая система, оснащённая полноповоротными колёсами, способными менять угол атаки

**система:** набор взаимосвязанных составных частей, совместно выполняющих задание или функцию; в технических системах процессы используются для превращения входных элементов в выходные элементы.

**системный подход:** способ оценки взаимного влияния составных частей некоего целого; рассмотрение взаимоотношений частей системы, а не на самих этих частях

**режим танкового поворота:** способ управления передвижением робототехнической модели, при котором каждый электродвигатель управляется по отдельности одним из двух джойстиков на пульте управления

**компромисс:** ограничение того или иного качества, свойства или количества с целью получения другого качества, свойства или количества

**устранение неисправностей:** выявление причины неисправности, нахождение и применение решения, направленного на её устранение

**переменная:** в программировании — некая заданная величина, которая хранится в памяти и может принимать разные значения





# Учебное пособие к набору TETRIX® "Соревнование в коробке"



Бесплатный звонок  
800-835-0686

Загляните на наш сайт  
[Pitsco.com](http://Pitsco.com)