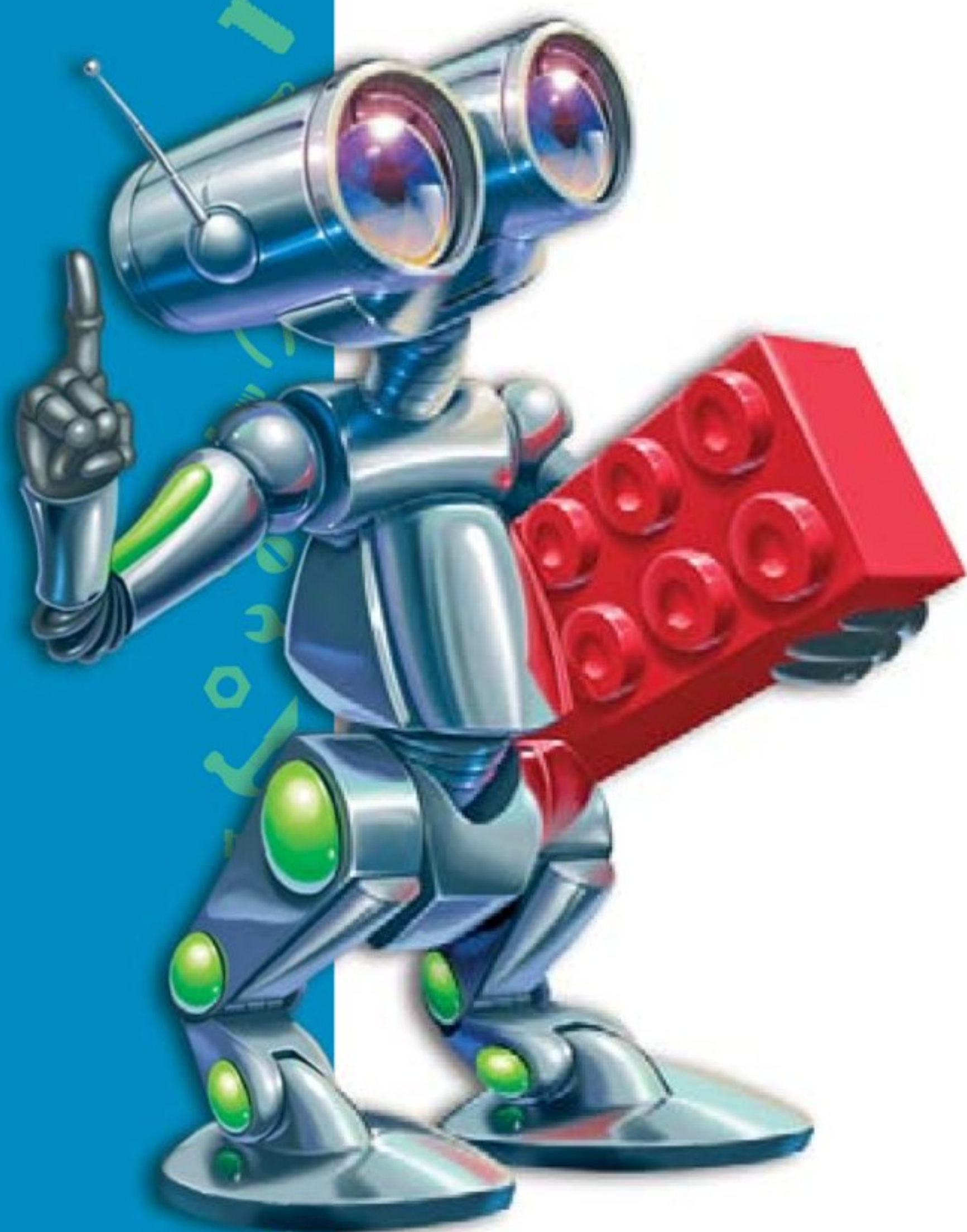


**С. А. Филиппов**

# УРОКИ РОБОТОТЕХНИКИ

Конструкция  
Движение  
Управление



ЛАБОРАТОРИЯ

**ПИЛОТ**



**С. А. Филиппов**

# **УРОКИ РОБОТОТЕХНИКИ**

Конструкция  
Движение  
Управление

Электронное  
издание



Москва  
Лаборатория знаний  
2017

УДК 621.86/.87(072)  
ББК 74.262:32.816  
Ф53

**Филиппов С. А.**

Ф53 Уроки робототехники. Конструкция. Движение. Управление [Электронный ресурс] / С. А. Филиппов ; сост. А. Я. Щелкунова. — Эл. изд. — Электрон. текстовые дан. (1 файл pdf : 179 с.). — М. : Лаборатория знаний, 2017. — Систем. требования: Adobe Reader XI ; экран 10".

ISBN 978-5-00101-553-6

Учебное пособие знакомит с основами моделирования автоматических устройств на основе робототехнических конструкторов LEGO и TRIK и создания алгоритмов управления роботами в среде TRIK Studio. Рассмотрены физические основы робототехники. Приведены интересные факты, касающиеся истории робототехники и ее современных достижений.

Предназначено для школьников 5-6 классов и старше, а также всех, интересующихся робототехникой. Может быть использовано для самостоятельного обучения, а также на уроках технологии, занятиях робототехнических кружков, при выполнении проектов и подготовке к участию в соревнованиях и олимпиадах.

УДК 621.86/.87(072)  
ББК 74.262:32.816

**Деривативное электронное издание на основе печатного аналога:** Уроки робототехники. Конструкция. Движение. Управление / С. А. Филиппов ; сост. А. Я. Щелкунова. — М. : Лаборатория знаний, 2017. — 176 с. : ил. — ISBN 978-5-00101-074-6.

**В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации**

ISBN 978-5-00101-553-6

© Лаборатория знаний, 2017

## *Дорогие ребята!*

Книга, которую вы держите в руках, — необычная. Собственно, это даже не книга, а дверь, за ней — невероятно увлекательный мир робототехники. Сегодня создание роботов — самое перспективное направление развития мировой научной и конструкторской мысли.

Роботы-манипуляторы давно уже несут вахту на производствах, в службах контроля за качеством продукции и в складских комплексах — там, где можно заменить человека при выполнении тяжелого и монотонного труда.

Нечастые пока медицинские роботы осуществляют адресную доставку лекарств в организме пациента, собирают анализы, исследуют состояние больного и передают эту информацию врачам. Их более продвинутые «коллеги» проводят сложные обследования и хирургические операции — разумеется, под присмотром людей в белых халатах, но это пока...

В условиях долговременных космических полетов роботы станут надежными помощниками космонавтов, обеспечивая им комфортное пребывание на борту корабля. А в открытом космосе при устранении возможных неполадок оборудования им не понадобятся ни скафандры, ни кислород для дыхания. Так что вполне возможно, на красные пески Марса одновременно ступят ребристый ботинок космонавта и металлическая «нога» робота.

Но и на нашей планете работы у роботов хоть отбавляй. Они наведут идеальный порядок в жилище, быстро доставят почту, выстроят наилучший маршрут и довезут грузы и пассажиров в нужное место без аварий и опозданий... Да мало ли что могут сделать для человека такие устройства, не требующие пищи и воды, не знающие, что такое усталость и сон!

И всё это не преувеличение. Когда вы вырастаете, многие, даже самые невероятные сегодня направления использования роботов, станут чем-то привычным и само собой разумеющимся.

Хотите стать творцами будущего? В этом вам поможет наша книга. Вы научитесь собирать из деталей конструкторов модели робототехнических устройств и программировать их для выполнения заданных действий. Они помогут лучше понять, по каким законам и правилам существует мир реальных машин и механизмов.

Книга состоит из глав, разделенных на параграфы. Самые важные определения и термины выделены *жирным курсивом*, *светлым курсивом* даны те слова, на которые требуется обратить особое внимание. В разделе «Это интересно!» собраны любопытные факты из мира робототехники. Такие материалы расширят ваш кругозор.

На полях страниц вы увидите специальные символы, которые облегчат навигацию по книге:



«Ключевые термины и определения» — указаны в тексте параграфа. Их необходимо запомнить.



«Проверьте себя» — эти материалы размещены в конце каждого параграфа. Вопросы повышенной сложности отмечены звездочкой \*. Работа над вопросами и заданиями поможет понять и освоить пройденный материал. Некоторые задания связаны с поиском новой информации. Её можно найти в книгах и журналах из личной или школьной библиотеки, а также в сети Интернет.



«Запомните» — так называется заключительный раздел в конце параграфа. Здесь повторяются новые понятия, которые вы встретили в тексте. Будет совсем нелишним проверить, хорошо ли вы их запомнили и можете ли правильно объяснить.

Не нужно робеть! Все рассматриваемые в этой книге объекты и сюжеты — не для каких-то необыкновенных гениев. Это совершенно привычная, будничная деятельность ваших сверстников из города Санкт-Петербурга — воспитанников замечательного педагога и энтузиаста робототехники Сергея Александровича Филиппова.

Итак, вперед, друзья! Смелее открывайте дверь в чудесный мир робототехники!

# Предисловие автора

С помощью этой книги вы познакомитесь с удивительным миром роботов и разберетесь в основах новой прикладной науки — робототехники.

Робототехника постепенно становится частью нашей жизни. Она включает в себя уже знакомые школьнику предметы: математику, физику, информатику. Используя знания, полученные на обычных уроках, вы научитесь конструировать и программировать автоматические устройства, очень похожие на настоящих роботов. Для этого вам понадобятся специальный робототехнический конструктор и среда программирования.

Самые известные робототехнические конструкторы — это LEGO® Mindstorms® и ТРИК. Специально для школьников в России разработана среда программирования роботов ТРИК Студия (TRIK Studio). С её помощью вы научитесь создавать простые и сложные алгоритмы управления, которые сможете проверить как на реальных роботах, так и в виртуальном мире.

Даже если у вас нет конструктора, в ТРИК Студию встроена двумерная модель мира, в которой маленький двухколесный робот исполняет множество различных алгоритмов: ездит по линиям, объезжает предметы, находит выход из лабиринта, рисует узоры.

Робот может быть надежным помощником и послушным исполнителем воли человека. Для этого необходимо научиться программировать его с помощью специальных алгоритмов управления. В пособии вам встретятся новые термины: регулятор, отклонение, управляющее воздействие. Используя простые математические формулы, вы сможете сделать поведение робота аккуратным и предсказуемым.

Полученные знания пригодятся на уроках технологии, занятиях робототехнического кружка, при выполнении творческих проектов, подготовке к участию в олимпиадах и соревнованиях.

*С. А. Филиппов*

# 1

## Простейшие механизмы

Люди постоянно совершенствуют среду своего обитания, дополняя её новыми элементами. Уже в начале XXI века человека повсюду сопровождают автоматизированные устройства, которые помогают ему быстро и эффективно решать многие задачи. Самые сложные и умные из этих устройств называются роботами. Изначально роботы задумывались для замены человека при физической работе в сложных условиях. Но теперь диапазон их применения расширяется с каждым днём. Учёные создают роботов, которые смогут существовать рядом с людьми, быть похожими на них (рис. 1.1).

Вам предстоит познакомиться с увлекательной наукой — робототехникой, в которой применяются знания большинства школьных дисциплин. Создавая роботов, вы не только узнаете, как они



Рис. 1.1. Роботы: а — Atlas; б — iCub

устроены, но и научитесь использовать элементы робототехники в разных областях жизни.

Робот — это технически сложное устройство, которое включает в себя несколько обязательных составляющих: механическую, электронную, программную.

Давайте начнём по порядку.

## § 1.1. Механизм, автомат, робот

**Механизм** — это устройство, с помощью которого человек выполняет физическую работу, или подвижная часть некоторого более сложного устройства. Механизмы служат для передачи движения и преобразования энергии.

**Автомат** — это устройство, которое может работать по заложенной в него программе без участия человека.

**Робот** — это автомат, который способен ориентироваться в окружающей среде и обладает элементами искусственного интеллекта, то есть он может принимать «самостоятельные решения», например управлять автомобилем вместо водителя.

### Механизм

Человек ограничен в своих физических возможностях, поэтому всегда стремился расширить их с помощью технических приспособлений. Одним из первых таких приспособлений, скорее всего, был рычаг.

**Рычаг** — это простейший механизм, представляющий собой балку, способную поворачиваться вокруг точки опоры (рис. 1.2 и 1.3).

Примером рычага может служить палка, с помощью которой первобытные люди выкапывали из земли съедобные корни или выворачивали камни. Нетрудно встретить рычаг в повседневной жизни: лом, лопата, тачка



Рис. 1.2. Рычаг







**Рис. 1.3.** Примеры механизмов: *а* — использование рычага при подъёме грузов; *б* — использование рычага в часовом механизме



**Рис. 1.4.** Автоматы: *а* — внутренний механизм музыкальной шкатулки; *б* — часы с кукушкой

**Рис. 1.5.** Пишущая кукла: *а* — внешний вид; *б* — внутренний механизм

и другие механизмы. Человеческое тело тоже состоит из множества рычагов: пальцы, руки, ноги. Со временем человек изобрёл приспособление, аналогов которому нет в организмах большинства живых существ, — *колесо*. С помощью колеса стало значительно легче перемещать грузы. Благодаря своим изобретениям, человек начал выполнять работу более эффективно, что, несомненно, улучшило его жизнь.

Таким образом, механизмом можно назвать устройство, с помощью которого человек выполняет физическую работу. С появлением более сложных устройств, которые при работе обходятся без участия человека, механизмы стали их составными частями.

## Автомат

Со временем человек научился создавать механические устройства, которые работают самостоятельно, выполняя сложные цепочки действий (рис. 1.4 и 1.5). Например, музыкальная шкатулка или часы с кукушкой. Такие устройства работают по заложенной человеком программе и являются автоматами.

Механические автоматы с давних пор занимали умы изобретателей. И, хотя первые автоматы появились ещё до нашей эры, их расцвет приходится на XVIII–XIX века, когда были созданы механические животные, птицы и даже люди, которые действовали почти как настоящие. Механические куклы играли на музыкальных инструментах, писали длинные предложения, а животные были так похожи на настоящих, что церковь даже осуждала их как орудия дьявола.

И всё же, как бы ни были чудесны механические устройства прошлых веков, назвать роботами их нельзя. Расцвет робототехники начался только после того, как человечество освоило электричество, а, точнее, в XX веке.

## Робот

В 1920 году чешский писатель *Карел Чапек* (рис. 1.6) впервые употребил термин «робот» в пьесе под названием «Р.У.Р.» (Россумские универсальные роботы). Основой послужило чешское слово «*robot*», что означает «тяжёлый труд». В этой пьесе роботы выходят из-под контроля людей.

Фантасты XX века в своих сочинениях приписывали роботам невероятные возможности искусственного интеллекта, которые до сих пор не достигнуты. Например, «три закона робототехники» (обязательные правила поведения роботов) *Айзека Азимова* (рис. 1.7), которые впервые были опубликованы в рассказе «Хоровод» в 1942 году, до



Рис. 1.6. Карел Чапек



**Рис. 1.7.** Айзек Азимов

сих пор не могут быть соблюдены ни одним из современных роботов. Технологии ещё не достигли того уровня, при котором робот сможет самостоятельно выполнять эти законы, трактуя их как человек. Для этого робот должен обладать полноценным искусственным интеллектом, то есть запрограммированной способностью обучаться, воспринимать окружающий мир и действовать как человек.

В первой половине XX века начали появляться *заводы-автоматы*. Автоматические линии оказались очень выгодными при производстве крупных

партий изделий. Впоследствии именно для заводов были изготовлены первые подобия роботов — программируемые манипуляторы (рис. 1.8). Таким образом, можно сказать, что робототехника берёт своё начало в промышленности.

Современных роботов разрабатывают учёные, используя достижения высшей математики, физики, информатики и многих других наук.

Основное отличие робота от автомата — присутствие элементов *искусственного интеллекта*. Эта грань строго не определена, поэтому в жизни некоторые автоматы тоже называются роботами. Их объединяет наличие специаль-



**Рис. 1.8.** Конвейер по сборке автомобилей промышленными манипуляторами


ных устройств — *датчиков*, с помощью которых автомат получает информацию из окружающего мира.


## Проверьте себя

1. Какие механизмы вы используете в своей жизни дома, на даче, в школе?
2. Приведите примеры устройств, частью которых является механизм.
3. Перечислите автоматы, которые есть почти в каждом доме, на улице, в торговом центре.
4. Как задаётся мелодия в музыкальной шкатулке?
5. Какие автоматы работают без использования электричества? Как можно задать программу механическим способом?
6. Подготовьте небольшое сообщение о любом из известных вам роботов. Обратите внимание на то, чем этот робот отличается от автомата.
7. Принесите на занятие механизм, который может работать только при участии человека, и расскажите о нём.
8. Найдите в Интернете информацию о российских изобретателях механизмов и механических автоматов: А. К. Нартове, И. П. Кулибине, П. Л. Чебышёве или других.

**Запомните** ♦ Механизм ♦ Автомат ♦ Робот ♦ Рычаг


### Это интересно!

 Название «робот» предложил Карелу Чапеку его брат. До того предполагалось название «лабор». Как бы тогда называлась наука по созданию «лаборов»?

 Впервые три закона робототехники были сформулированы писателем-фантастом Айзеком Азимовым в рассказе «Хоровод» в 1942 году. Впоследствии им было создано множество произведений, в которых показано, как сложно даже совершенному роботу выполнять эти законы в различных жизненных ситуациях. Но как бы ни вёл себя робот, ответственным за его поступки всегда является человек.


### *Законы робототехники*

1. Робот не может причинить вред человеку или своим бездействием допустить, чтобы человеку был причинён вред.

- 
2. Робот должен повиноваться всем приказам, которые даёт человек, кроме тех случаев, когда эти приказы противоречат Первому Закону.
  3. Робот должен заботиться о своей безопасности в той мере, в которой это не противоречит Первому или Второму Законам.

## § 1.2. Знакомство с конструктором

Для изучения основ робототехники вам потребуется специальный конструктор — набор из деталей, а также электромеханических и электронных устройств. С детства вы встречались с подобными конструкторами, но они, как правило, предназначались для игр. Робототехнический конструктор предназначен для обучения и моделирования настоящих устройств.

 **Конструктор** — это набор стандартных деталей, из которых можно собрать много разных моделей. Благодаря ему вы изучите множество механизмов.

Материалы, из которых сделаны детали конструкторов, могут различаться. Обычно это пластик или металл. Пластиковые детали легче соединять между собой, но конструкции из них менее прочные. Для соединения металлических деталей может потребоваться отвёртка или гаечный ключ.

В магазинах и Интернете можно найти много разнообразных робототехнических наборов. Однако не все они подходят для учебных задач. В состав школьного конструктора обычно входят следующие составные части:

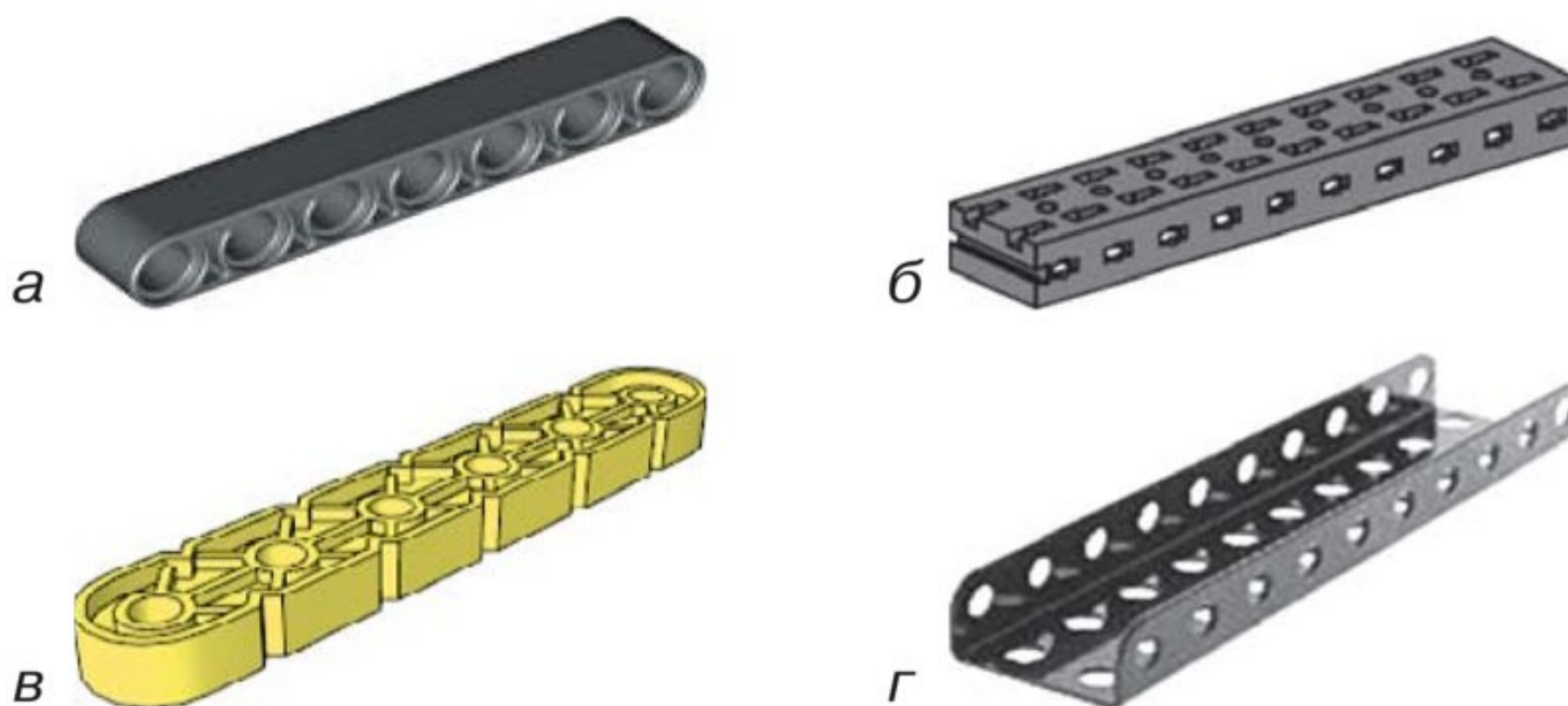
- 1) комплект деталей для построения простейших механизмов;
- 2) моторы и источник электропитания (батарейный блок или аккумулятор);
- 3) контроллер («мозг» будущего робота);
- 4) датчики (устройства, реагирующие на свет, звук, температуру и передающие данные о них «мозгу»).

С моторами, контроллерами и датчиками вы познакомитесь чуть позже. Первым делом следует рассмотреть основные детали конструктора, которые потребуются при сборке базовых механизмов. В качестве примера нам подойдут конструкторы «ТРИК», «ЛЕГО», «Фишертехник», «ВЕКС», «Роботрек», «Науробо», «Роботис».

## Несущие детали

**Балка** — несущая часть большинства конструкций с крепёжными отверстиями или выступами (рис. 1.9). *Несущей частью* называют опорный элемент конструкции. Как правило, балка имеет четыре грани и прямоугольное сечение. Характерная особенность балок в современных конструкторах — наличие отверстий по всей длине на определённом расстоянии друг от друга. Расстояние между центрами двух отверстий называют *модулем*. Таким образом, длину любой детали можно измерять числом модулей. В металлических конструкторах для придания жёсткости в качестве балок используют *швеллеры* — детали, имеющие П-образное сечение.

Аналогичную деталь в других конструкторах называют планкой. **Планка** — это плоская упругая полоска с отверстиями (рис. 1.10). Особенность металлической планки в том, что после сгиба она сохраняет изогнутую форму.



**Рис. 1.9.** Балки из робототехнических наборов: а — «ЛЕГО»; б — «Фишертехник»; в — «ВЕКС»; г — швеллер из «ТРИК»



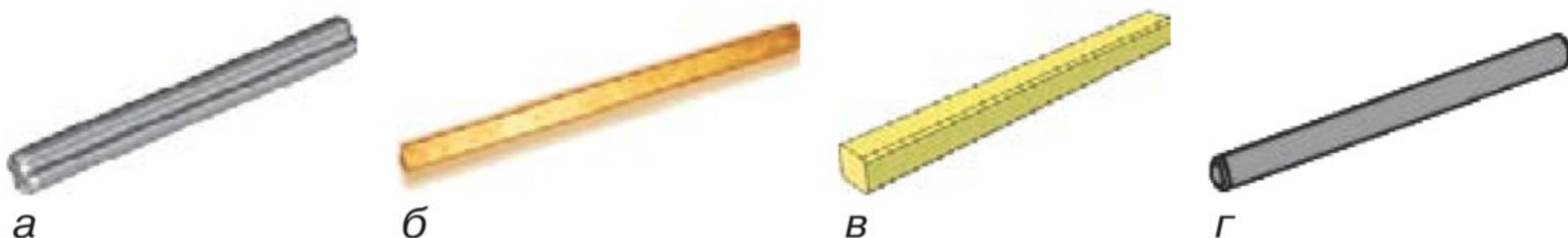
**Рис. 1.10.** Планки из робототехнических наборов: а — «ЛЕГО»; б — «ТРИК»



**Рис. 1.11.** Пластины из робототехнических наборов: а — «ЛЕГО»; б — «ВЕКС»; в — «Роботис»



**Рис. 1.12.** Изогнутые балки и уголки из робототехнических наборов: а — «ЛЕГО»; б — «ТРИК»; в — «ВЕКС»




**Рис. 1.13.** Оси из робототехнических наборов: а — «ЛЕГО»; б — «ТРИК»; в — «ВЕКС»; г — «Фишертехник»

Как правило, на планке бывает не более двух отверстий в ряд. Более широкие полосы из упругого материала с крепёжными отверстиями или выступами называются *пластинами* (рис. 1.11).

Для крепления деталей под углом используются изогнутые балки или уголки (рис. 1.12). Если балка имеет изгиб, то при измерении учитывается длина каждого плеча в модулях.





**Ось** — это стержень, свободно вращающийся в отверстиях опор и обеспечивающий вращение колёс. В идеале ось имеет круглое сечение, но для фиксации колёс и других деталей оси могут иметь плоские грани и даже крестообразное сечение (рис. 1.13).

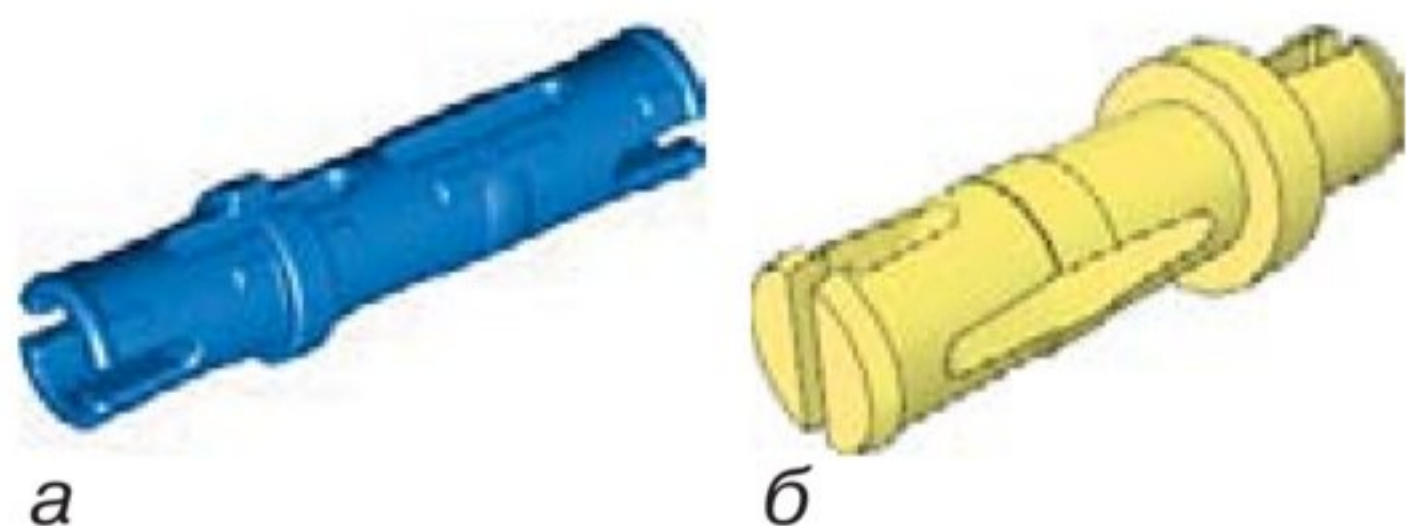
## Крепёжные элементы

Для надёжного соединения несущих деталей друг с другом используются *крепёжные элементы*. Как правило, они небольшого размера и присутствуют в наборе в большом количестве.

**Штифт** в робототехническом конструкторе — крепёжный элемент для соединения деталей через отверстия (рис. 1.14). Штифт может соединять две и более деталей. Как правило, штифт плотно входит в детали и держит их за счёт трения или специальных рёбер. Различаются гладкие штифты, допускающие некоторое вращение деталей друг относительно друга, и штифты с выступами, полностью блокирующие вращение.

Аналог штифта в металлическом конструкторе — винт с гайкой. **Винт** — это крепёжный элемент со спиральной нарезкой и головкой со шлицем. **Шлиц** — это прорезь под отвёртку в головке винта. В металлических конструкторах чаще всего используют плоские, крестовые или шестигранные отвёртки.

**Гайка** — кольцо со спиральной нарезкой изнутри для навинчивания на винт (рис. 1.15). Снаружи гайка чаще всего имеет форму шестигранника и для её удержания



**Рис. 1.14.** Штифты из робототехнических наборов: а — «ЛЕГО»; б — «ВЕКС»



**Рис. 1.15.** Винт (а) и гайка (б) из робототехнического набора «ТРИК»







**Рис. 1.16.** Втулки из робототехнических наборов: *а* — «ЛЕГО»; *б* — «ВЕКС»; *в* — «Амперка»; *г* — «ТРИК»



**Рис. 1.17.** *а*, *б* — фиксаторы из робототехнического набора «ЛЕГО»

**Рис. 1.18.** Резиновая шина из робототехнического набора «ЛЕГО»

и закручивания применяется гаечный ключ соответствующего размера.

Для более надёжной фиксации используют гайки, снабжённые специальным элементом, блокирующим самопроизвольное развинчивание (самоконтрящиеся гайки).



**Втулка** — это кольцо или короткая трубка для фиксации оси (рис. 1.16). Втулки, как правило, держатся за счёт трения. В металлических конструкторах применяют втулки с потайным винтом (без головки), с помощью которого втулка фиксируется на оси.

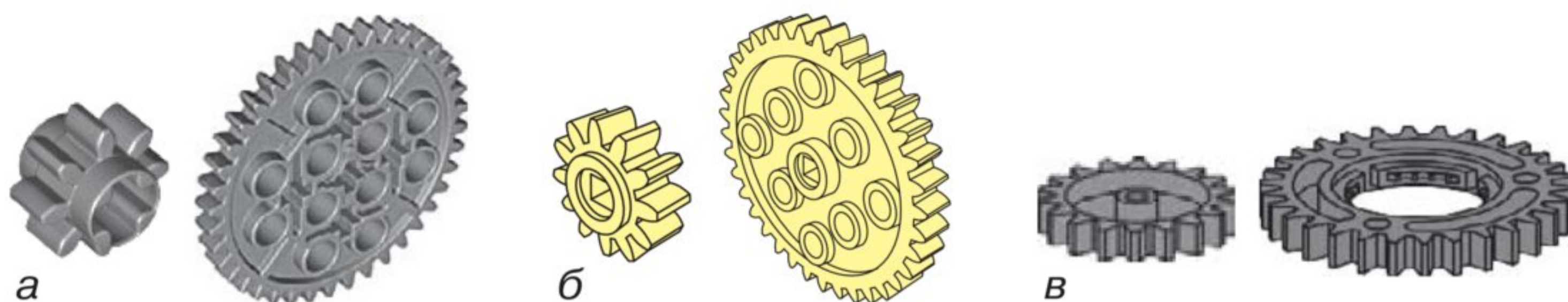


**Фиксатор** — это элемент, закрепляющий ось в нужном положении. Фиксаторы имеют различную форму, углы поворота и количество отверстий для крепления (рис. 1.17).

## Колёса



**Колесо** — это круг, вращающийся на оси. Колесо состоит из *ступицы* (колёсного диска) с отверстием для оси и *резиновой шины* (рис. 1.18). Размер колеса характеризуют его диаметром и шириной.



**Рис. 1.19.** Шестерёнки из робототехнических наборов: а — «ЛЕГО»; б — «ВЕКС»; в — «Фишертехник»

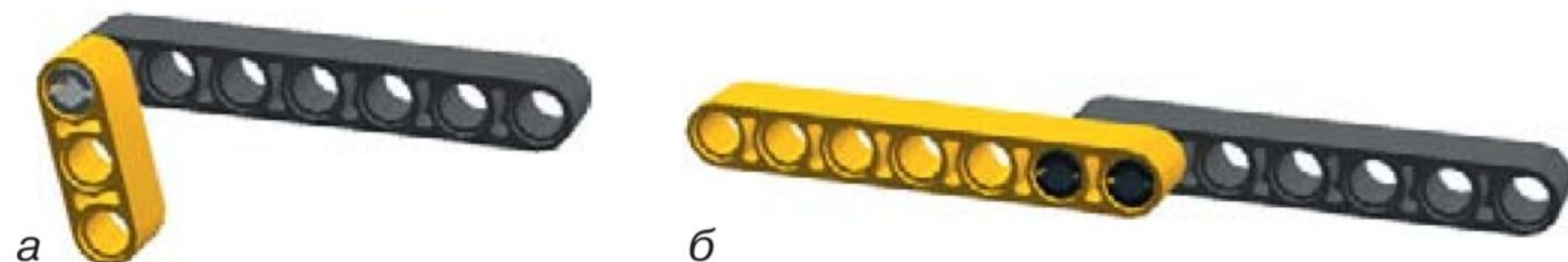
**Шестерёнка** (зубчатое колесо) — колесо с зубчиками и отверстием для оси (рис. 1.19). Чаще всего шестерёнка устанавливается на ось и контактирует зубчиками с другой шестерёнкой. Размер шестерёнки определяется числом зубчиков. Размер зубчика у шестерёнок одного типа всегда одинаков.

## Принципы крепления

Рассмотрим два основных принципа крепления деталей: шарнир и жёсткий узел.

**Шарнир** — это подвижное соединение двух частей механизма, обеспечивающее их вращение (рис. 1.20, а). Самый простой пример шарнира можно получить, соединив две балки (планки) одним штифтом, винтом или осью. Даже если затянуть винт или зажать ось втулками, под большой нагрузкой шарнир никогда не будет жёстким.

**Жёсткий узел** — конструктивное соединение, в котором концы деталей соединены между собой жёстко без возможности вращения (рис. 1.20, б). Такое соединение достигается, как правило, с помощью двух штифтов или винтов.



**Рис. 1.20.** Соединения: а — шарнирное; б — жёсткое





**Рис. 1.21.** а—в — придание жёсткости конструкции

При шарнирном соединении детали всегда будут под нагрузкой поворачиваться друг относительно друга, если такая возможность присутствует. Так, четырёхугольник на рисунке 1.21, а быстро потеряет форму. Однако треугольник (рис. 1.21, б) — жёсткая конструкция. Используя принцип треугольника, можно сделать более жёсткой и квадратную конструкцию (см. рис. 1.21, в). Этот приём вы могли видеть в калитке у забора деревенского дома.

Познакомившись более подробно со своим конструктором, вы узнаете, как называются все его детали, и это поможет в будущем совместном с одноклассниками и учителем творчестве.



### Проверьте себя

1. Перечислите основные типы несущих деталей вашего учебного конструктора.
2. Перечислите основные крепёжные элементы вашего учебного конструктора.
3. Какие из крепёжных деталей позволяют создавать наиболее прочные конструкции?
4. По виду детали определите её полное название. Например, пластина  $4 \times 10$  модулей или колёсный диск диаметром 24 мм.
5. Используя все детали одного конструктора, постройте башню максимальной высоты, которая будет стоять на своём основании без дополнительных опор. Для связок используйте жёсткие узлы.
6. В современной технике распространены ножничные механизмы, построенные на основе многозвенных парных шарниров (рис. 1.22), похожих на несколько скреплённых между собой ножниц. Используя этот принцип, постройте захватывающий



**Рис. 1.22.** а—в — ножничные механизмы

ножничный механизм наибольшей длины и научитесь перемещать с его помощью грузы.

7. Игра «Фантастическое животное». Разделитесь на пары. Придумайте фантастическое животное (рис. 1.23). Постройте его, используя не более 25 деталей конструктора. Используйте не более половины деталей каждого типа. Дайте название получившемуся животному. Никому не показывайте его.

Договоритесь, кто в вашей паре конструктор, а кто изобретатель. Изобретатель прячет животное и остаётся на месте спиной к набору. Конструктор отправляется за стол к другому изобретателю. В новых парах конструктор и изобретатель садятся друг к другу спинами.

Изобретатель не смотрит на работу конструктора и не показывает ему ничего. Конструктор не смотрит на фантастическое животное. Изобретатель говорит конструктору, какую деталь взять и куда прикрепить. Конструктор на слух строит близнеца фантастического животного.

Какие знания нужны, чтобы копия животного получилась похожей на оригинал?



**Рис. 1.23.** Пример фантастического животного



**Запомните** ♦ Конструктор ♦ Балка ♦ Планка ♦ Пластина ♦ Ось  
♦ Штифт ♦ Винт ♦ Шлиц ♦ Гайка ♦ Втулка ♦ Фиксатор ♦ Колесо  
♦ Шестерёнка ♦ Шарнир ♦ Жёсткий узел

### Это интересно!



Самая высокая башня в мире — это Бурдж-Халифа, построенная в 2010 году в г. Дубаи (ОАЭ). Её высота — 828 метров. Основание башни представляет собой трилистник, поскольку конструкция, имеющая три точки опоры, наиболее устойчива. Масса здания — 500 тыс. тонн. Самая тяжёлая часть башни — основание, самая лёгкая — шпиль.



## § 1.3. Механическая передача



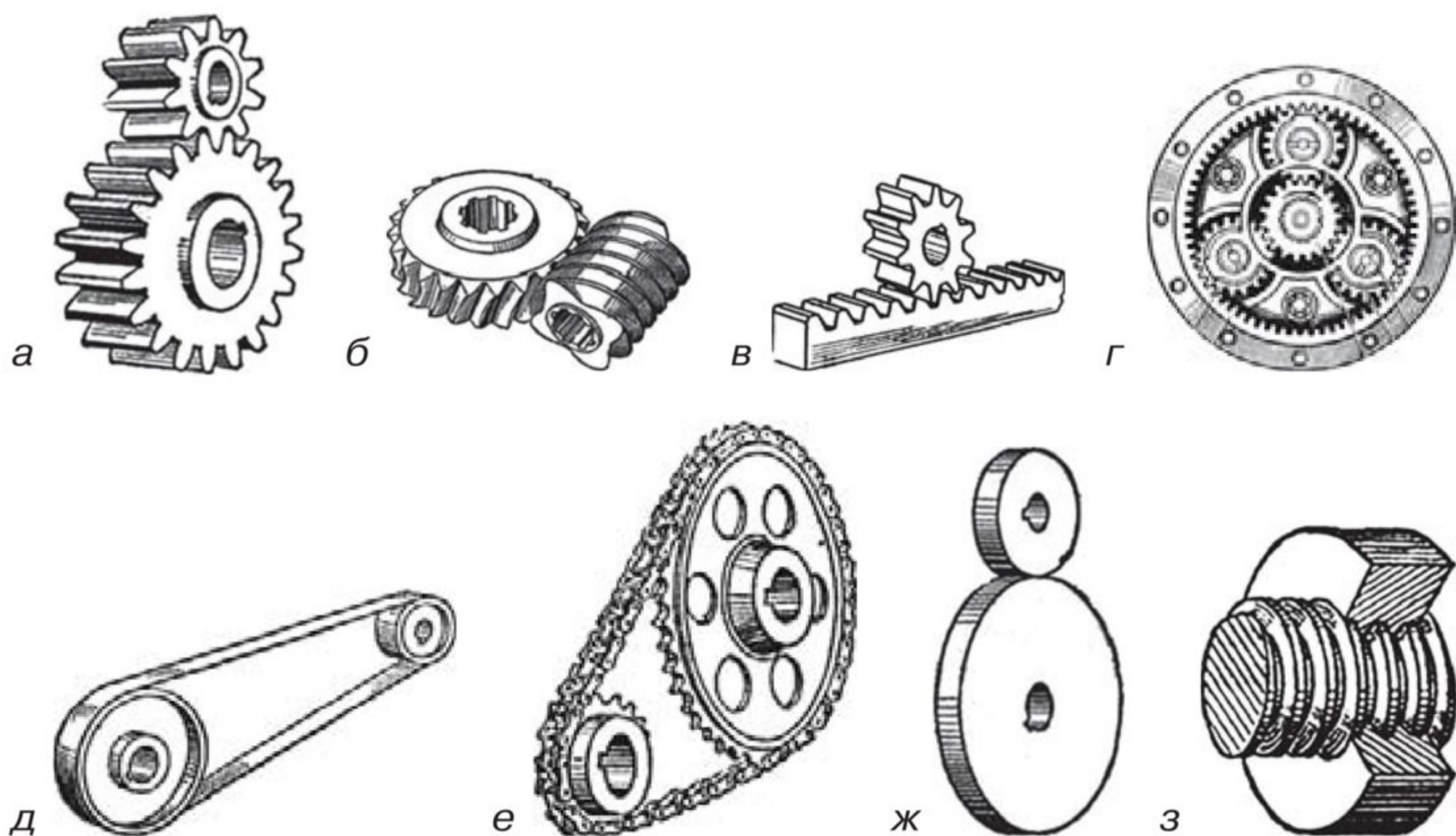
Важной частью многих устройств является механическая передача. **Механическая передача** — это механизм, обеспечивающий передачу и преобразование механического движения. Чаще всего используют передачу вращательного движения от двигателя к исполнительному механизму, например колесу. Исполнительный механизм производит нужные действия: перемещение, подъём тяжестей, обработку деталей и другие. Потребность в передаче возникает, когда необходимо изменить скорость и силу вращения,

передать вращение на расстояние или изменить его направление.

## Виды механической передачи

Существует несколько видов механических передач. Самый распространённый вид передачи — *зубчатая*. Её разновидностями являются также планетарная, червячная, реечная, цепная и другие (рис. 1.24, *а—г* и *е*). Из названия понятно, что в этих передачах основным элементом является *зуб*, который обеспечивает зацепление вращающихся деталей. Также распространены ремённая и фрикционная передачи, в которых зацепление основано на трении элементов.

Зубчатое колесо ещё называют шестернёй. Это связано с тем, что для надёжного и плавного зацепления на нём должно быть не меньше шести зубьев.



**Рис. 1.24.** Виды передачи: *а* — зубчатая прямая; *б* — червячная; *в* — реечная; *г* — планетарная; *д* — ремённая; *е* — цепная; *ж* — фрикционная; *з* — винт-гайка



**Рис. 1.25.** Пример реечной передачи, использовавшейся на железной дороге Стефенсона

Реечная передача является разновидностью зубчатой. Вместо одного из колёс используется зубчатая рейка, благодаря чему вращательное движение переходит в поступательное или наоборот (рис. 1.24, в). Такая передача применяется, например, на зубчатой железной дороге, придуманной английским инженером-механиком *Джорджем Стефенсоном* (рис. 1.25), или в телескопе для наводки объектива на резкость.



С помощью передачи вращение передаётся с ведущего вала на ведомый. То есть **ведущим** называют вал (или ось), к которому обычно подключается двигатель, а **ведомым** — вал, который получает движение через передачу и может выполнять полезную работу, например вращать колёса автомобиля или патрон электродрели (рис. 1.26).

Если в качестве двигателя выступает человек, то ведущим считается вал, который он крутит (рис. 1.27). Далее при обозначении вала будет употребляться термин «ось».



**Рис. 1.26.** Ведомый и ведущий вал механической передачи



**Рис. 1.27.** Рукоятка для руки человека расположена на ведущей оси

## Передаточное отношение

При построении передаточных механизмов необходимо знать, во сколько раз будут изменены скорость вращения и усилие на ведомом валу. Для этого используются понятия *передаточное отношение* и *передаточное число*. Передаточное отношение  $i$  показывает, во сколько раз понижается скорость вращения и одновременно повышается усилие. Оно равно отношению скорости вращения ведущего вала ( $w_1$ ) к скорости вращения ведомого ( $w_2$ ):

$$i = \frac{w_1}{w_2}.$$

Эту величину можно рассчитать через отношение диаметра ведомого колеса ( $d_2$ ) к диаметру ведущего ( $d_1$ ), что применимо, например, в ремённой передаче:

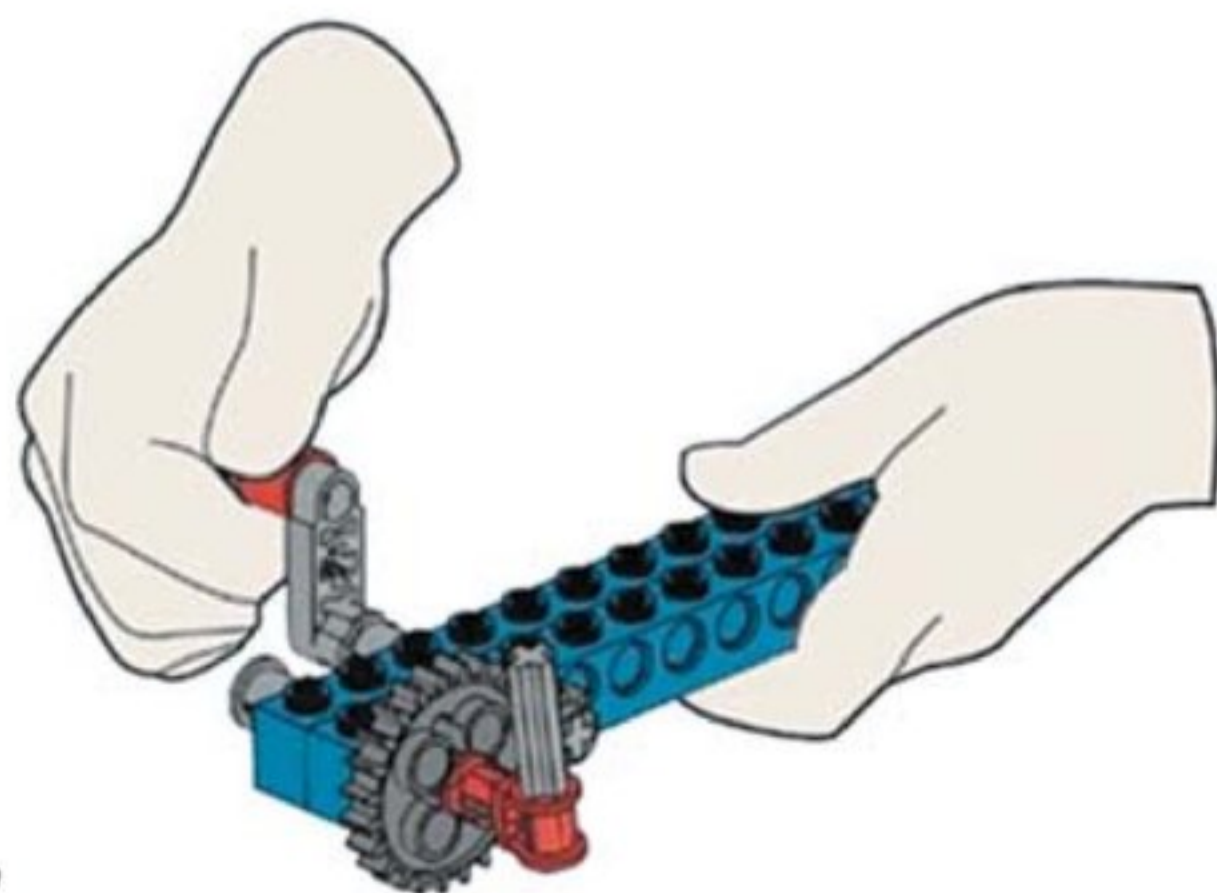
$$i = \frac{d_2}{d_1}.$$

В зубчатой передаче отношение может быть рассчитано через количество зубцов на шестернях ( $z$ ), которое пропорционально их диаметру:

$$i = \frac{z_2}{z_1}.$$

На рисунке 1.28 ведущая шестерня имеет 8 зубцов, ведомая — 24. Тогда:

$$i = \frac{24}{8} = \frac{3}{1}.$$



а



б

Рис. 1.28. Понижающая передача: а — рисунок; б — схема





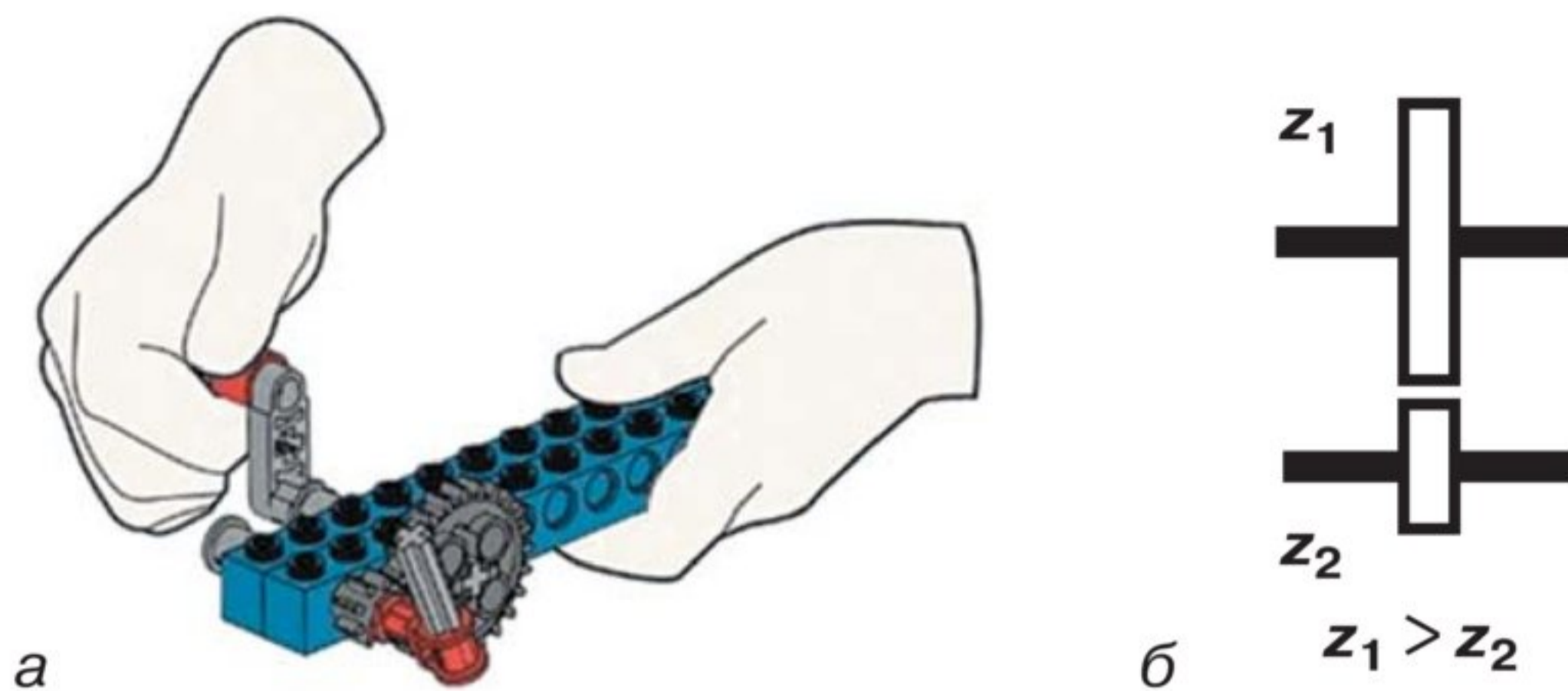


Рис. 1.29. Повышающая передача: а — рисунок; б — схема

Такая передача втрое замедляет вращение и увеличивает тяговую силу.

Если поменять шестерни местами (рис. 1.29) и сделать ведущей 24-зубую, а ведомой — 8-зубую, то получим обратный результат:

$$i = \frac{8}{24} = \frac{1}{3}.$$

Значит, вращение будет в три раза быстрее, но и в силе мы потеряем втрое.

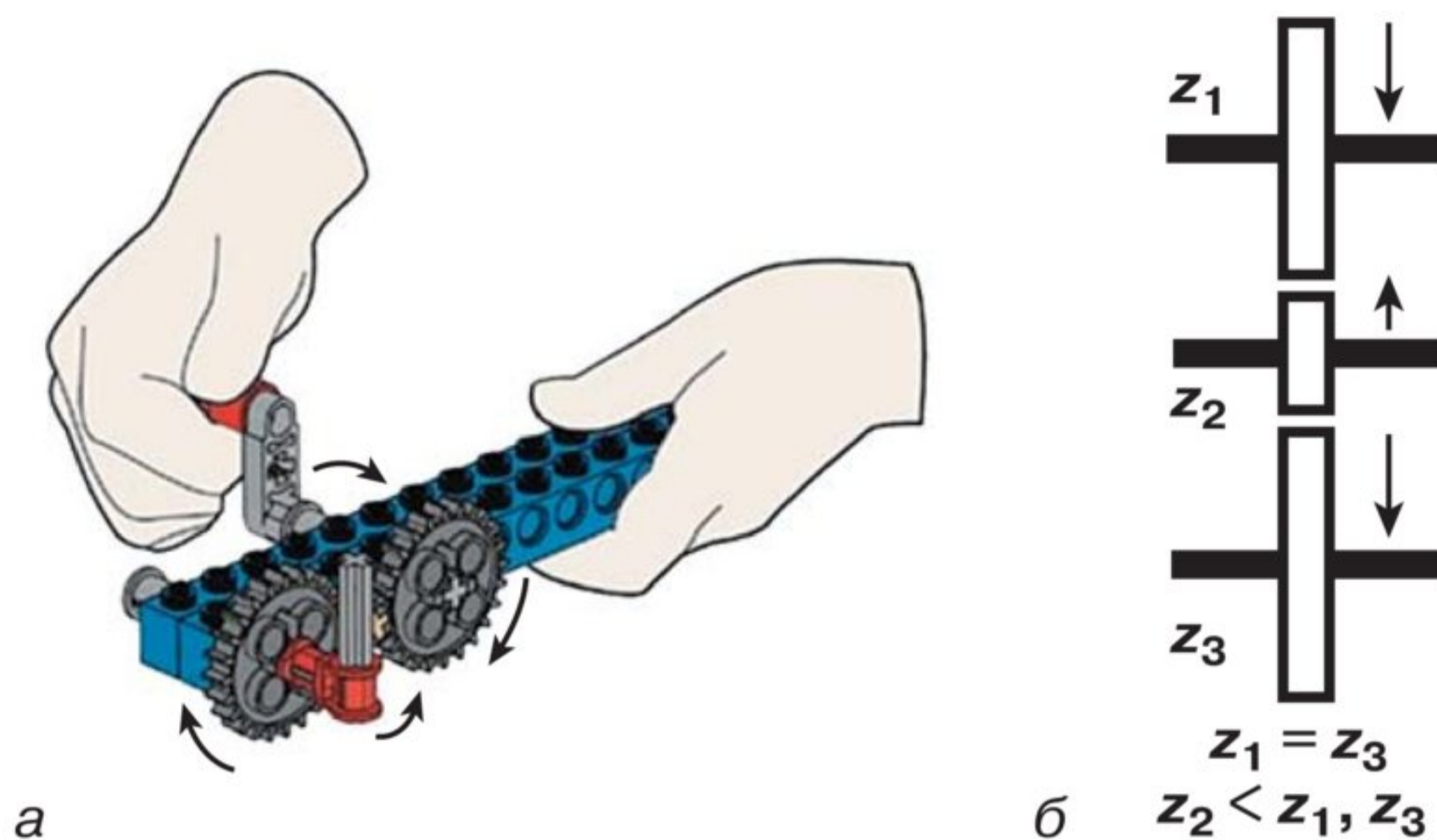
Передачу, замедляющую вращение, называют **понижающей**. В этом случае передаточное отношение  $i > 1$ . Механизм на основе понижающей передачи называется **редуктором** (от англ. «редьюс» — понижать).

Передачу, ускоряющую вращение, называют **повышающей**, и её передаточное отношение  $i < 1$ . Механизм на основе повышающей передачи называют **мультипликатором** (от англ. «мультиплай» — умножать).

Подберите самую малую и самую большую шестерню из вашего конструктора. Какое самое большое передаточное отношение можно построить с их помощью?

## Паразитные шестерни

Постройте передачу из трёх зубчатых колёс, соединённых последовательно. Как ни странно, скорость вращения первого и третьего равны. Тогда какую роль играет второе?



**Рис. 1.30.** Паразитная шестерня: а — рисунок; б — схема

Оно только меняет направление вращения. Такая шестерня называется *паразитной* и не влияет на передаточное отношение (рис. 1.30). Это можно доказать математически:

$$i = \frac{z_2}{z_1} \cdot \frac{z_3}{z_2} = \frac{z_3}{z_1}.$$

Из расчёта видно, что количество зубцов  $z_2$  паразитной шестерни не играет роли в передаточном отношении. Любая шестерня, соединённая с двумя другими, для них является паразитной. Как правило, от неё возникают потери на трение, а механизм выигрывает только в размерах.

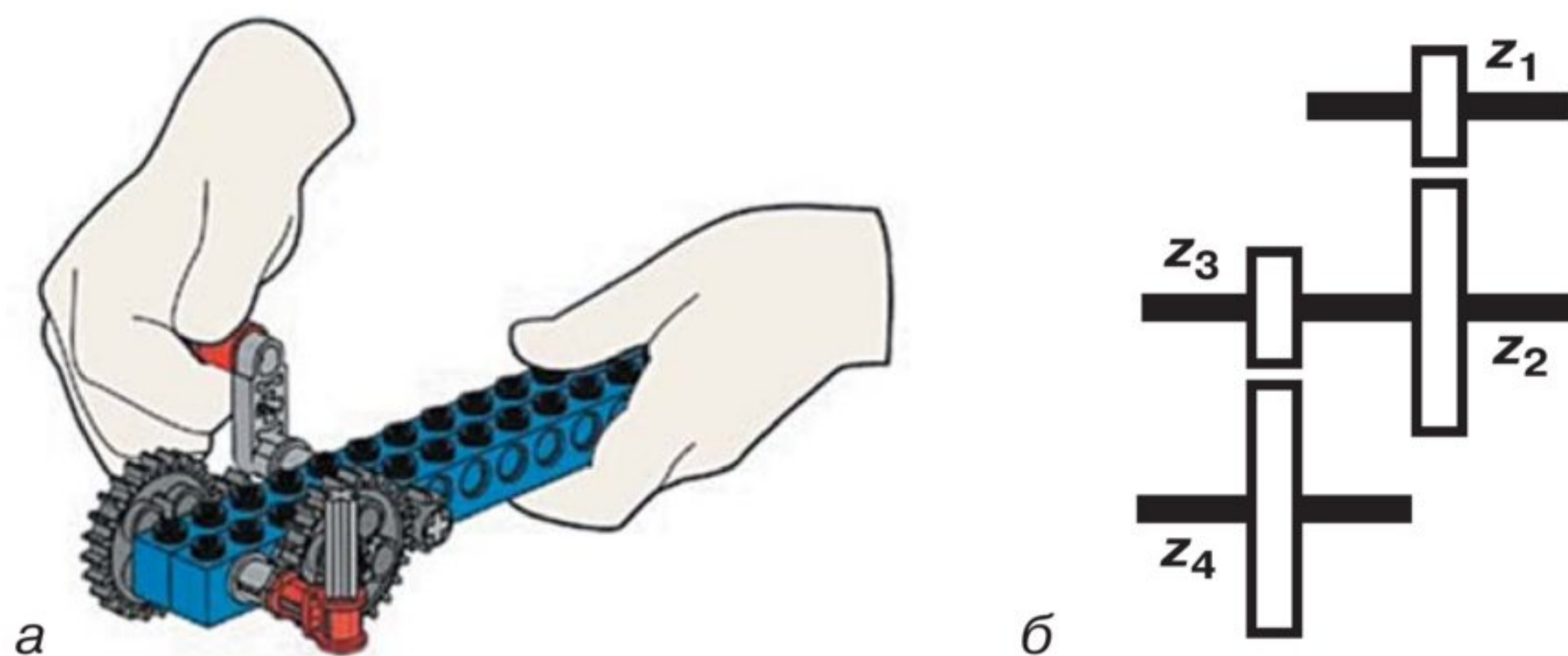
## Многоступенчатая передача

Совсем другой результат можно получить, если установить две шестерни разного размера на одну ось (рис. 1.31). Они будут вращаться с одинаковой скоростью, но в соприкосновении ещё с двумя шестернями дадут существенное увеличение отношения:

$$i = \frac{z_2}{z_1} \cdot \frac{z_4}{z_3}.$$

Такая передача называется *многоступенчатой*.





**Рис. 1.31.** Двухступенчатая передача: а — рисунок; б — схема

Сколько раз теперь нужно повернуть ведущую ось для одного поворота ведомой? Проведём расчёт для двух пар шестерёнок на 8 и 24 зубца.

$$i = \frac{24}{8} \cdot \frac{24}{8} = \frac{3}{1} \cdot \frac{3}{1} = \frac{9}{1}.$$

Передаточное отношение многоступенчатой передачи равно произведению отношений всех ступеней.

Используя это правило, можно построить передачу с очень большим передаточным отношением, достигающим нескольких миллиардов и более.


## Червячная передача



**Червячная передача** построена на основе одной обычной шестерни и *червяка* — винта со специальной резьбой (рис. 1.32). В этой паре червяк всегда является ведущим. Один заход червяка поворачивает один зубец ведомой



**Рис. 1.32.** а—в — примеры использования червяка в разных деталях и механизмах



шестерни. Поэтому передаточное отношение можно рассчитать по формуле

$$i = \frac{z_2}{z_1}.$$

Таким образом, с помощью червячной передачи можно получить наибольшее передаточное отношение. В обратном направлении вращение чаще всего блокируется, что даёт возможность использовать червячную передачу при больших нагрузках. Вы могли встречать червячную передачу в разводном гаечном ключе, где вместо ведомой шестерни используется зубчатая рейка.

## Передаточное число

Понятие *передаточное число* служит для обобщённой характеристики передаточного механизма. Передаточное число  $u$  всегда больше или равно 1. Оно определяется через передаточное отношение  $i$ .

$$u = i, \text{ если } i \geq 1 \quad \text{и} \quad u = \frac{1}{i}, \text{ если } i < 1.$$

Иными словами, для редуктора передаточное число равно передаточному отношению, а для мультипликатора — обратно ему.

Для одноступенчатой передачи передаточное число можно определить как отношение числа зубьев большей шестерни к числу зубьев меньшей, в каком бы порядке они ни стояли.

В дальнейшем для однозначного понимания мы будем использовать термин «*передаточное отношение*».

## Проверьте себя

1. Какое количество зубчиков в шестерёнках вашего конструктора?
2. Приведите примеры механической передачи, которые вы встречали в жизни.
3. Приведите расчёт следующих передаточных отношений, используя шестерни с 8, 24 и 40 зубчиками:  
а) 27:1,    б) 45:1,    в) 135:1,    г) 3:5,    д) 5:9.
4. Приведите расчёт следующих передаточных отношений, используя шестерни с 12, 20 и 36 зубчиками:  
а) 9:1,    б) 9:5,    в) 25:9.





5. Из всех шестерёнок одного конструктора постройте механизм на устойчивом основании с максимальным передаточным отношением. Приведите его расчёт.

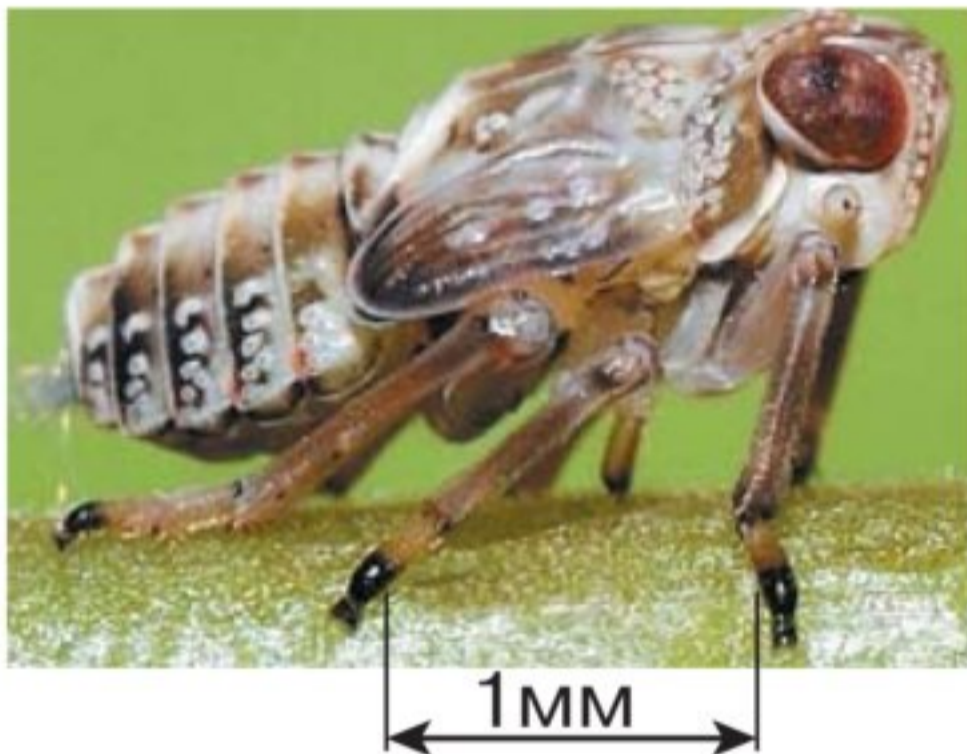


**Запомните** ♦ Механическая передача ♦ Ведущий вал  
♦ Ведомый вал ♦ Передаточное отношение ♦ Передаточное число  
♦ Понижающая передача ♦ Повышающая передача  
♦ Редуктор ♦ Мультипликатор ♦ Паразитная шестерня  
♦ Многоступенчатая передача ♦ Червячная передача

### Это интересно!



Недавно учёные обнаружили на задних лапках личинок прыгающих насекомых *Issus coleoptratus* орган, в точности напоминающий шестерёнку. Благодаря ему лапки насекомого движутся необыкновенно быстро и скоординировано. С помощью только нервной системы этого достичь не удалось бы.



«Шестерёнки»  
на задних лапках насекомого

## § 1.4. Ремённая и фрикционная передачи

### Ремённая передача



**Ремённую передачу** используют в механизмах, в которых не требуется постоянное передаточное отношение, а расстояние между осями должно быть достаточно большим. Например, в автомобильном генераторе электрического тока или в приводе станка (рис. 1.33). Ремённая передача позволяет передавать вращение на достаточно большое расстояние.



**Рис. 1.33.** а—в — примеры ремённых передач

Ведущее и ведомое колесо называются *шкивами*, а соединяет их упругий ремень — *натяжитель*. Как правило, ремень помещается в специальный жёлоб, чтобы избежать соскакивания. Передача вращения происходит за счёт трения между ремнём и шкивами.

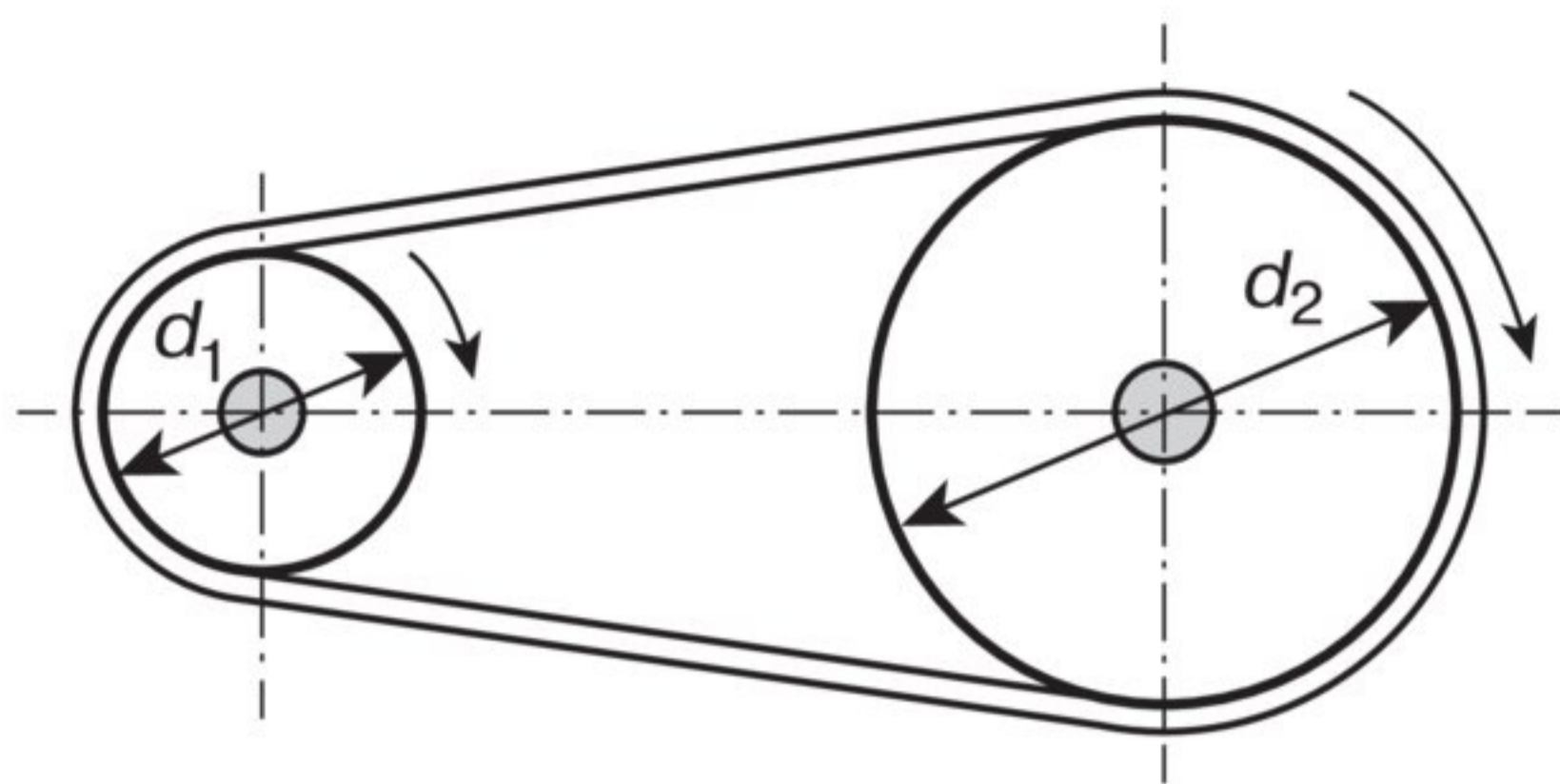
Ремённая передача — самая распространённая из всех механических передач после зубчатой.

При создании механизма необходимо знать его передаточное отношение. Но как его посчитать для колёс, у которых нет зубчиков? Передаточное отношение ремённой передачи рассчитывается через отношение диаметров ведущего ( $d_2$ ) и ведомого ( $d_1$ ) шкивов (рис. 1.34):

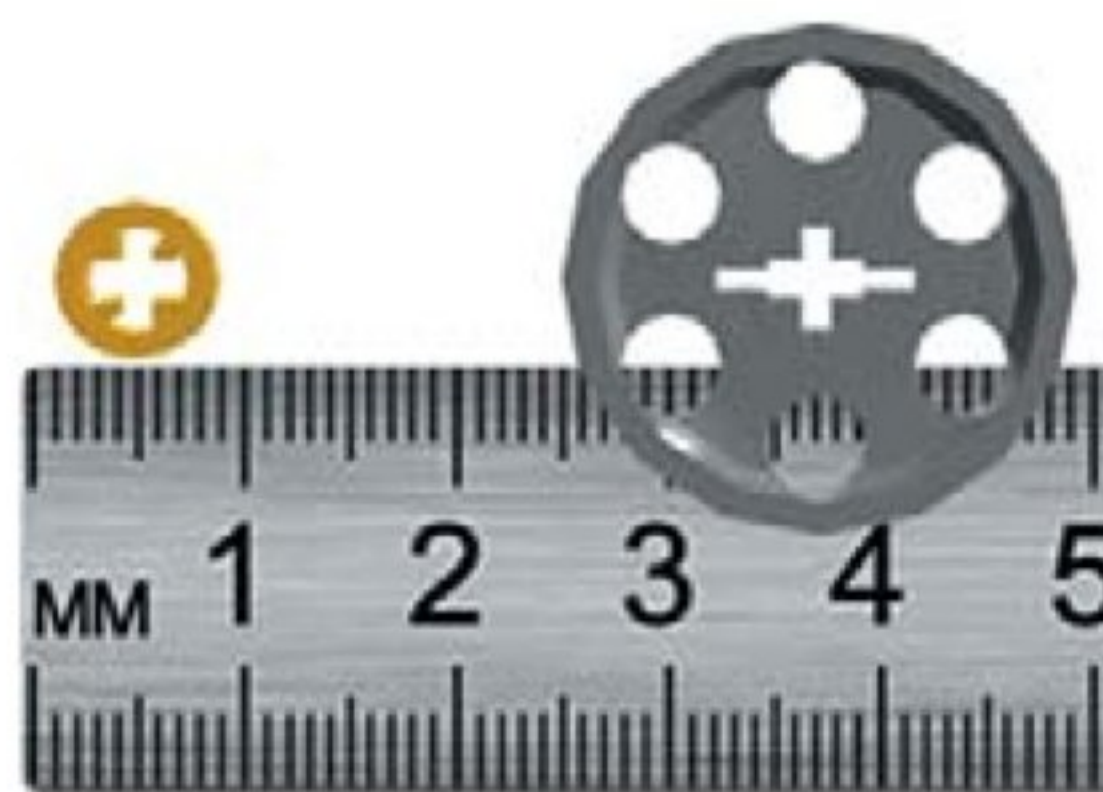
$$i = \frac{d_2}{d_1}.$$

Постройте ремённую передачу из деталей вашего конструктора. В качестве ведущего шкива используйте малую втулку, в качестве ведомого — ступицу узкого колеса. С помощью линейки измерьте диаметр каждого из шкивов и запишите в тетрадь (рис. 1.35).





**Рис. 1.34.** Диаметры шкивов и направление их вращения

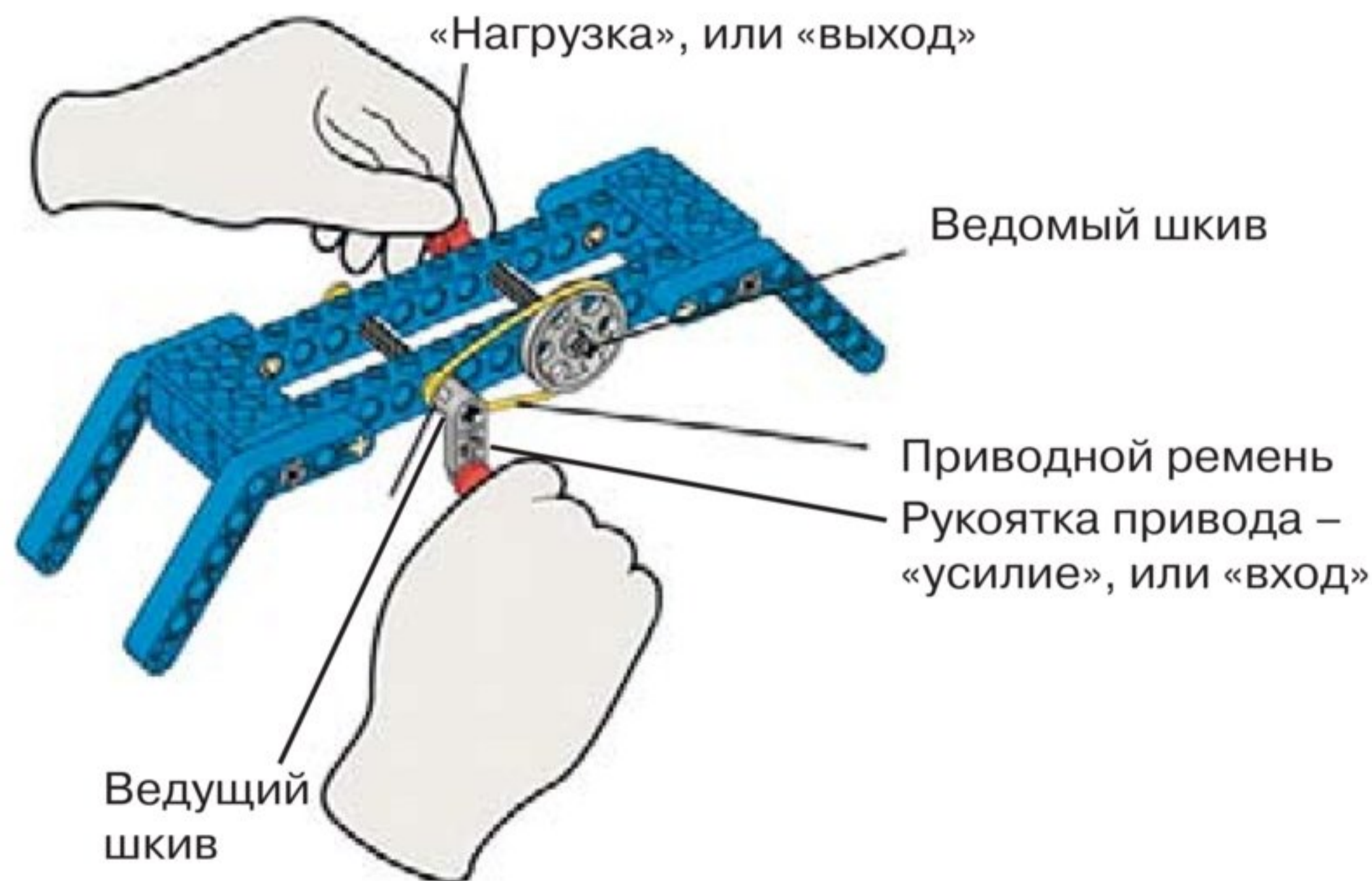


**Рис. 1.35.** Измерение диаметров шкивов с помощью линейки

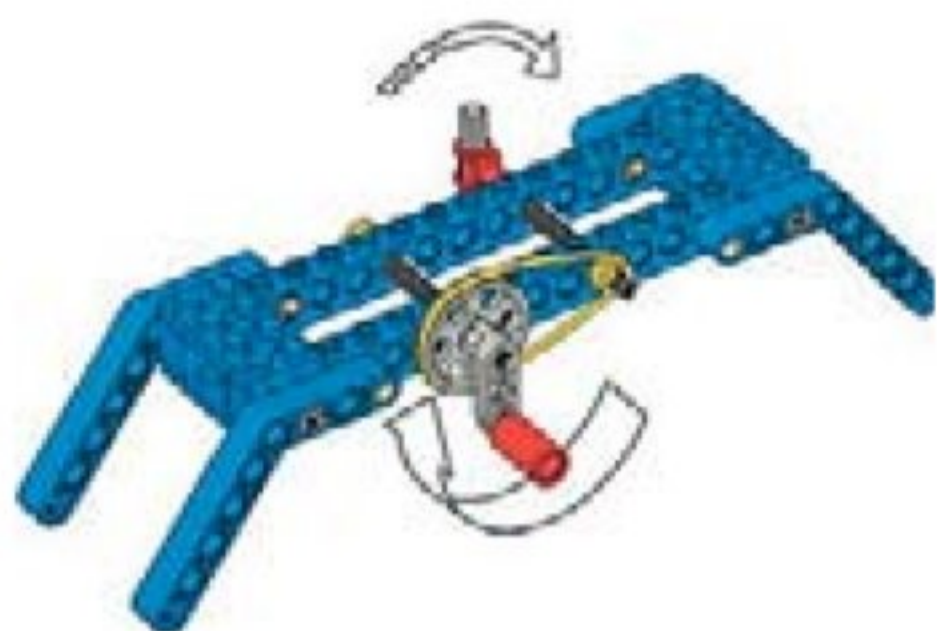
Постройте конструкцию с ремённой передачей, воспользовавшись станком, на котором ранее строилась зубчатая передача. В качестве ремня можно использовать резинку для плетения браслетов или обычную банковскую резинку (рис. 1.36).

Проверьте на практике простые правила:

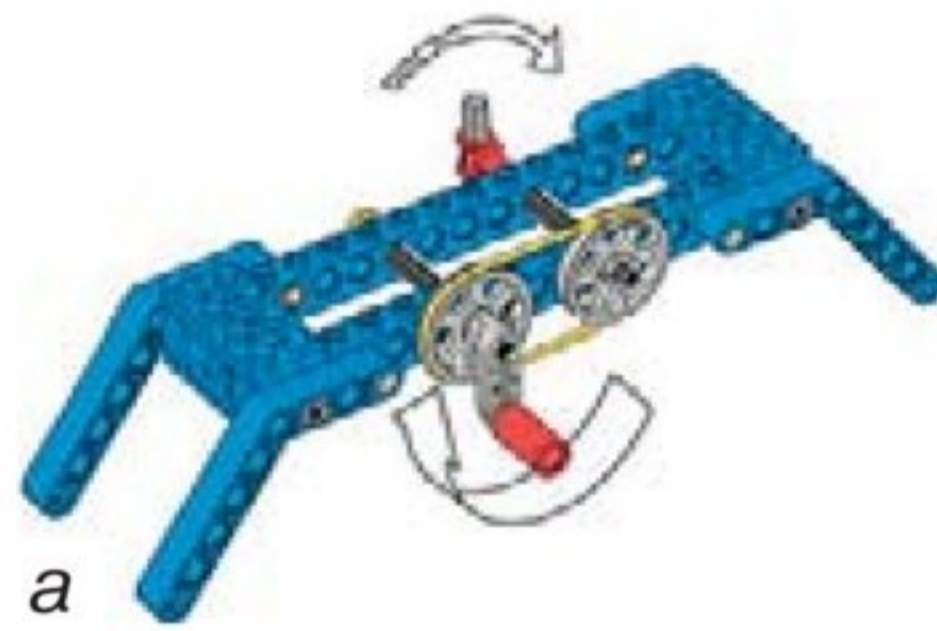
- 1) число оборотов ведомого шкива во столько раз меньше числа оборотов ведущего, во сколько раз его диаметр больше диаметра ведущего шкива;



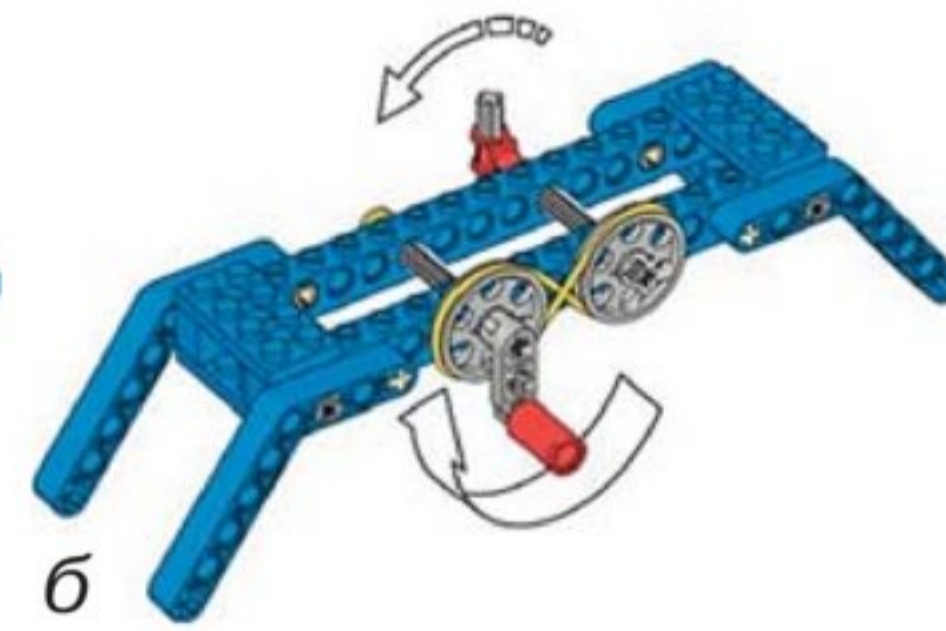
**Рис. 1.36.** Построение ремённой передачи



**Рис. 1.37.** Увеличение скорости в ременной передаче



а



б

**Рис. 1.38.** Передачи: а — открытая; б — перекрёстная

2) число оборотов ведомого шкива во столько раз больше числа оборотов ведущего, во сколько раз его диаметр меньше диаметра ведущего шкива.

Чтобы проверить второе правило, поменяйте шкивы местами (рис. 1.37).

Вспомните, каковы диаметры шкивов. Приблизительно 6,9 мм и 23 мм (если учитывать внутренний диаметр желобка). Значит, передаточное отношение получается:

$$i = \frac{23}{6,9} \approx \frac{10}{3} \approx 3 \text{ и } \frac{1}{3}.$$

Иными словами, за 10 поворотов малого шкива большой должен повернуться 3 раза. Это и произошло бы в идеальных условиях. Однако в ременной передаче с гладким ремнём всегда существует проскальзывание, поэтому реальные значения будут отличаться от расчётных.

В отличие от зубчатой передачи, в ременной шкивы одного звена вращаются в одном направлении. Это называется **открытая передача**. Чтобы обеспечить разнонаправленное движение, достаточно перевернуть ремень на одном из шкивов так, как показано на рисунке 1.38. Получится **перекрёстная передача**.

На рисунке 1.38 представлены шкивы одного размера, значит их скорости вращения должны совпадать. Однако, если используется гладкий ремень, возможно проскальзывание, и скорости их вращения будут различаться.

Натянутый на шкивы ремень охватывает их не полностью. Облегающая часть носит название **угла обхвата**.







**Рис. 1.39.** Регулировка натяжения ремня

При малом угле обхвата сцепление ремня со шкивом будет недостаточным, и появится проскальзывание. На рисунке 1.39 сцепление нижнего шкива с ремнём может оказаться недостаточным, и при вращении он будет отставать от расчётного значения скорости. Зато благодаря ему можно регулировать натяжение ремня, что позволяет улучшить сцепление на двух верхних шкивах.

Недостатки ремённых передач:

- ◆ большие размеры;
- ◆ малая долговечность ремня;
- ◆ большая нагрузка на валы из-за натяжения ремня.

Достоинства ремённых передач:

- ◆ простота конструкции;
- ◆ возможность передачи вращения на большие расстояния;
- ◆ плавность и бесшумность;
- ◆ предохранение от перегрузок из-за проскальзывания.

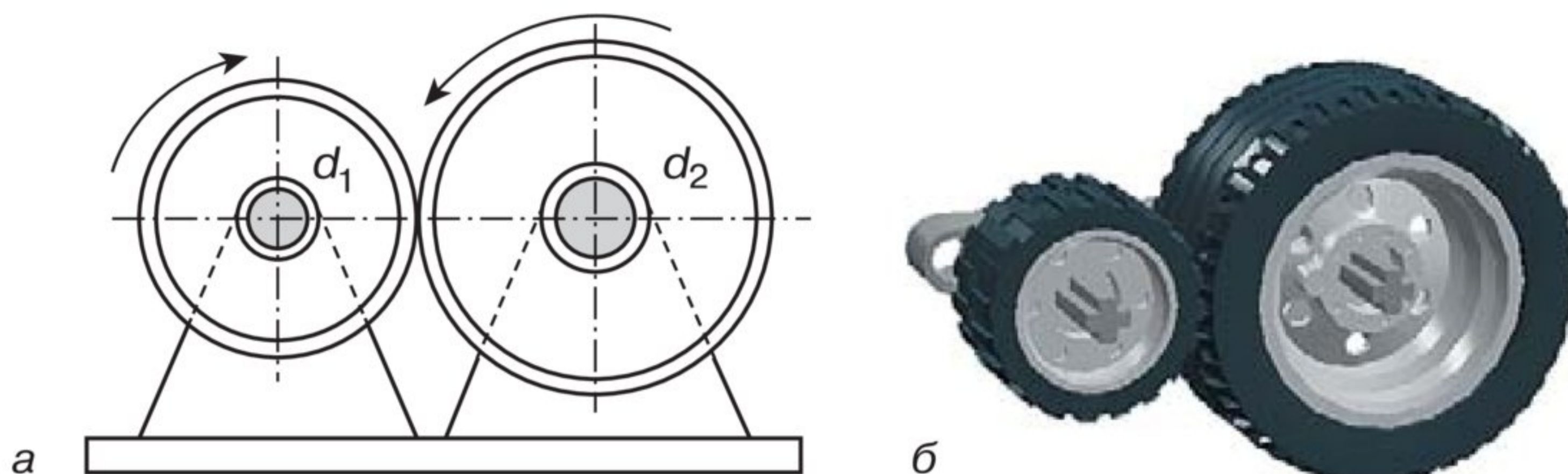
## Фрикционная передача

**Фрикционная передача** — это ещё один вид передачи, в котором вращение передаётся за счёт трения. В ней используются колёса без зубчиков и пазов для ремней, а движение передаётся за счёт трения между соприкасающимися колёсами (рис. 1.40).

Расчёт передаточного отношения для фрикционной передачи производится по той же формуле, что и для ремённой. Однако направление вращения соседних колёс различается, как и у зубчатой.

## Проверьте себя

1. Что общего между ремённой и фрикционной передачами?
2. По какой причине теряется скорость при использовании ремённой передачи?




**Рис. 1.40.** Фрикционная передача: а — схема; б — фотография

3. Какие дополнительные требования появляются при построении ремённой передачи с высоким натяжением ремня?
4. Что произойдёт, если на две оси поставить одновременно ремённую и фрикционную передачи?
5. Почему проскальзывание в ремённой передаче безопаснее, чем в зубчатой?
6. Используя ремённую и фрикционную передачи, постройте понижающий передаточный механизм из нескольких ступеней. Измерьте диаметры колёс и рассчитайте передаточное отношение. Насколько отличается реальное понижение скорости вращения от расчётного?
7. Постройте передаточный механизм из трёх шкивов и одного ремня. Перемещайте один из шкивов и с помощью транспортира определите минимальный угол обхвата, при котором проскальзывание не превышает заданной величины (например, 5%).
8. Постройте самую длинную многоступенчатую ремённую передачу, и поднимите с её помощью какой-нибудь груз.

**Запомните** ♦ Ремённая передача ♦ Шкив ♦ Пассик  
♦ Открытая передача ♦ Перекрёстная передача ♦ Угол обхвата  
♦ Фрикционная передача



### Это интересно!

 Скорость движения ремня в высокоскоростной ремённой передаче может превышать 100 м/с. При нормальной работе ремённой передачи на скольжении теряется 1–2% скорости вращения.

## § 1.5. Соосный редуктор

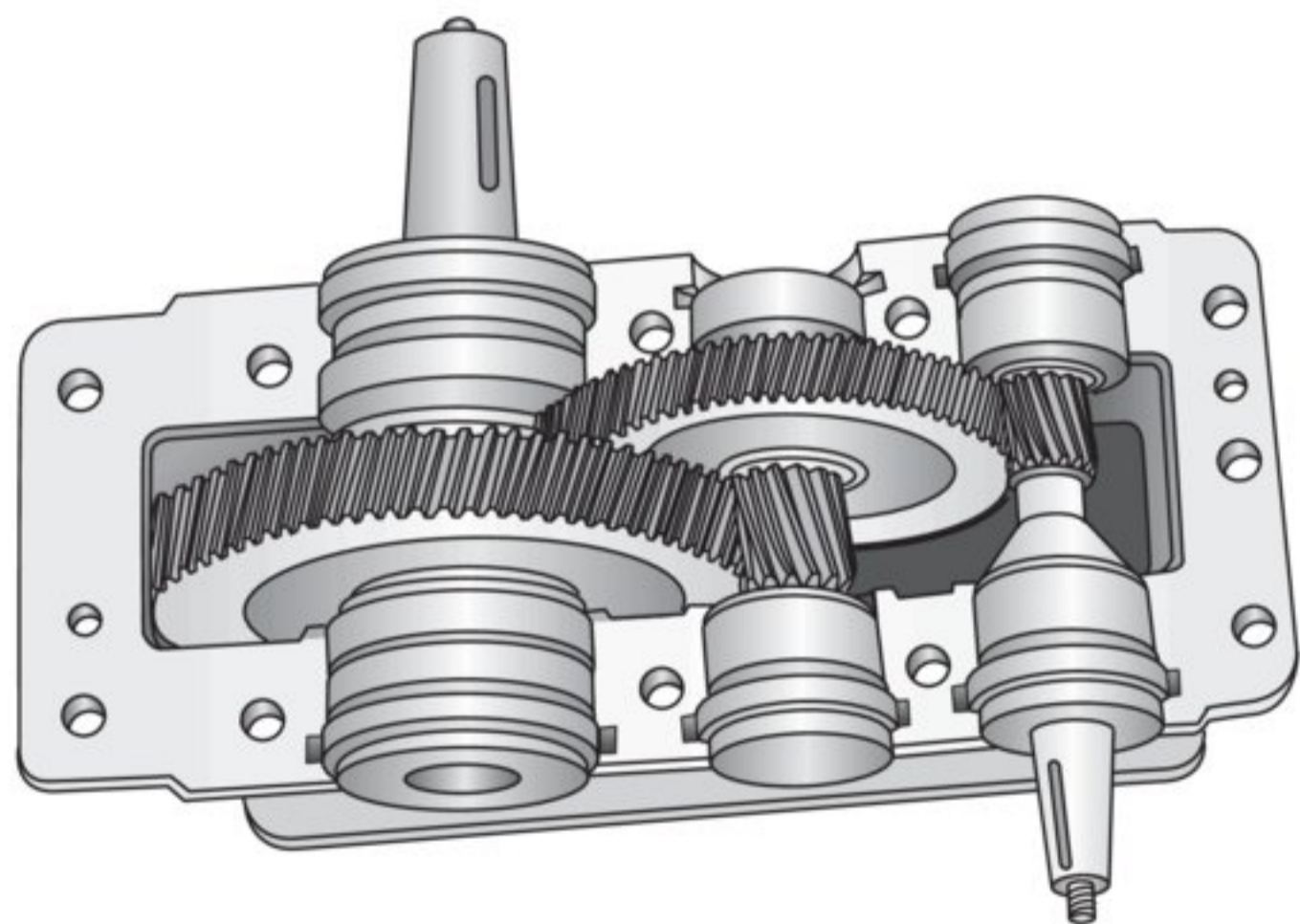
### Развёрнутая и соосная схемы

Большинство электродвигателей имеют малую тяговую силу и высокую скорость вращения. Редуктор понижает скорость вращения и увеличивает силу. Благодаря этому даже небольшим двигателем можно выполнять тяжёлую работу.

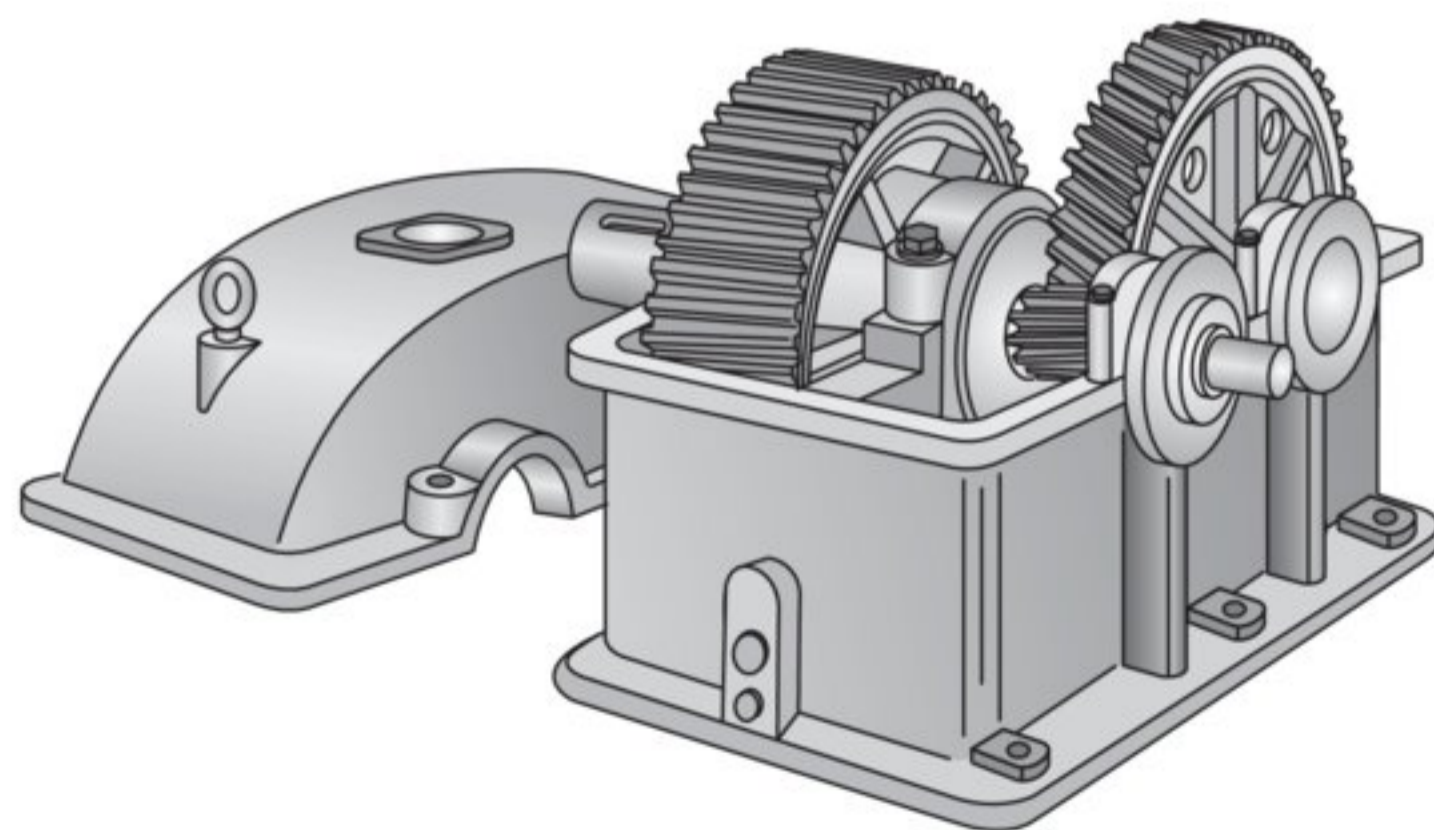
Редукторы, выполненные по *развёрнутой схеме* имеют достаточно большой размер (рис. 1.41). Ведомая и ведущая оси оказываются удалены друг от друга. При создании или совершенствовании передаточного механизма может возникнуть потребность изменить передаточное отношение, сохраняя вращающиеся оси на одной прямой. Для этих целей используется редуктор, выполненный по *соосной схеме* (рис. 1.42). В *соосном редукторе* ведущая и ведомая оси расположены на одной прямой.

Такое расположение осей делает редуктор более компактным. На рисунке 1.43 приведён соосный редуктор с передаточным отношением 9:1.

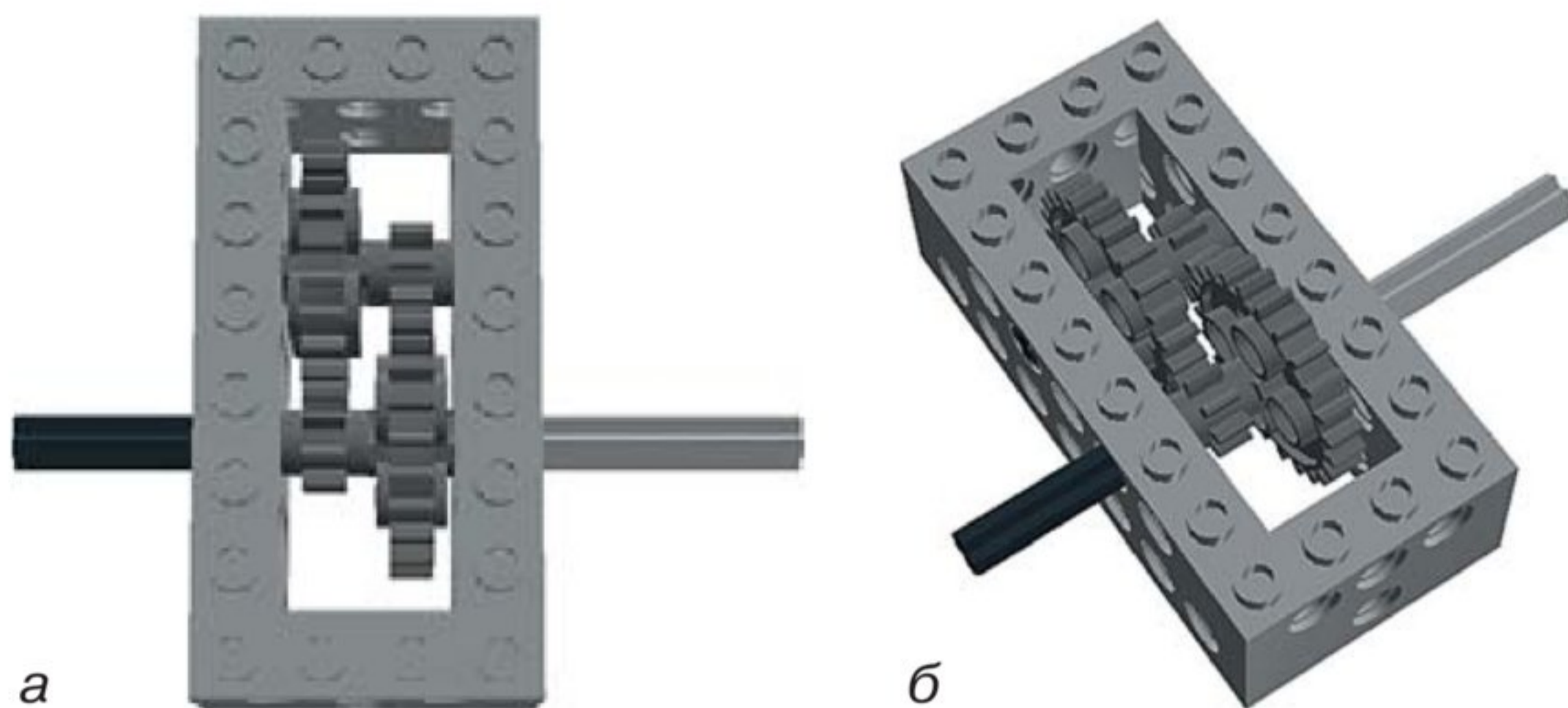
$$i = \frac{24}{8} \cdot \frac{24}{8}.$$



**Рис. 1.41.** Редуктор, выполненный по развёрнутой схеме



**Рис. 1.42.** Редуктор, выполненный по соосной схеме



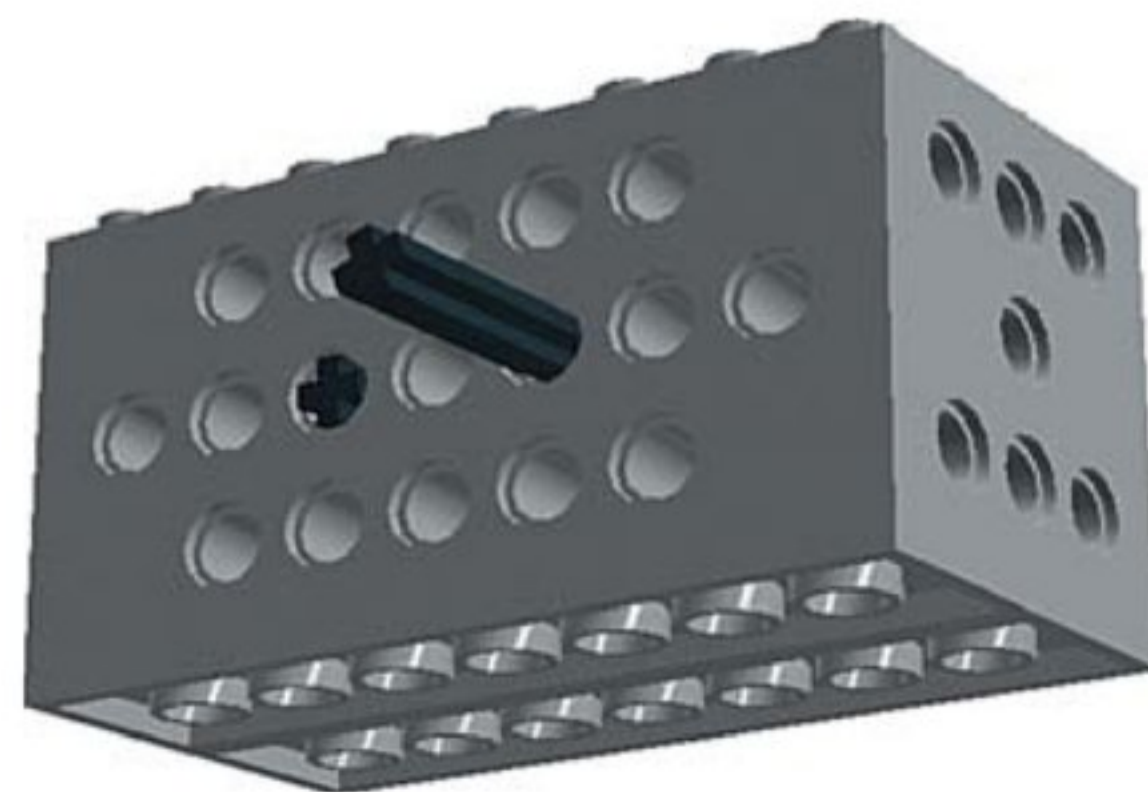
**Рис. 1.43.** а—б — соосный редуктор 9:1

Чёрная ось на рисунке 1.43 — *ведущая*, серая — *ведомая*. Обе оси находятся на одной прямой. Соберите такую модель из конструктора.

### Картер редуктора

Обычно внутри механизма редуктор наполнен смазочным материалом. Это может быть густая или жидкая смазка на основе технического масла. Смазка необходима для того, чтобы шестерни, соприкасаясь друг с другом, не изнашивались. Для защиты от вытекания смазки, а также для защиты от попадания в механизм мусора и посторонних предметов используется картер. **Картер редуктора** — это закрытый корпус с креплениями для осей (рис. 1.44).

Понятно, что из деталей конструктора можно построить только модель редуктора, поскольку в них присутствуют технологические отверстия, которых не будет в реальном устройстве. Будем считать, что эти отверстия не мешают выполнить задачу.



**Рис. 1.44.** Модель редуктора в картере



## Соосный редуктор с картером

Построить *соосный редуктор внутри картера* — непростая задача. Но с помощью конструктора это можно сделать. Необходимо подобрать шестерни соответствующего размера и комплект балок и пластин для построения картера. Первый редуктор имеет передаточное отношение  $9:1$ . Для построения редуктора использована двухступенчатая передача из двух пар шестерёнок: 8- и 24-зубых.

Передаточное отношение второй модели редуктора будет  $3:1$  (рис. 1.45). Значит, потребуется использовать только одну пару шестерёнок с соответствующим числом зубчиков. Но как обеспечить две ступени? Есть минимум два варианта решения: с использованием двух или трёх шестерёнок одного размера, одна из которых будет *паразитной*.

Редуктор на рисунке 1.45, *а* имеет одну паразитную шестерню, и благодаря этому ведомая ось вращается в противоположную сторону по сравнению с ведущей. Редуктор на рисунке 1.45, *б* имеет две 16-зубых шестерни одного размера, благодаря чему ведомая и ведущая оси вращаются в одну сторону.

Редукторы окружают нас повсюду. Они используются в коробках передач автомобилей, в часах, в DVD-приводах, в ручном электроинструменте, в подъёмных кранах и лебёдках. Хотя потери на трение в них велики, редук-

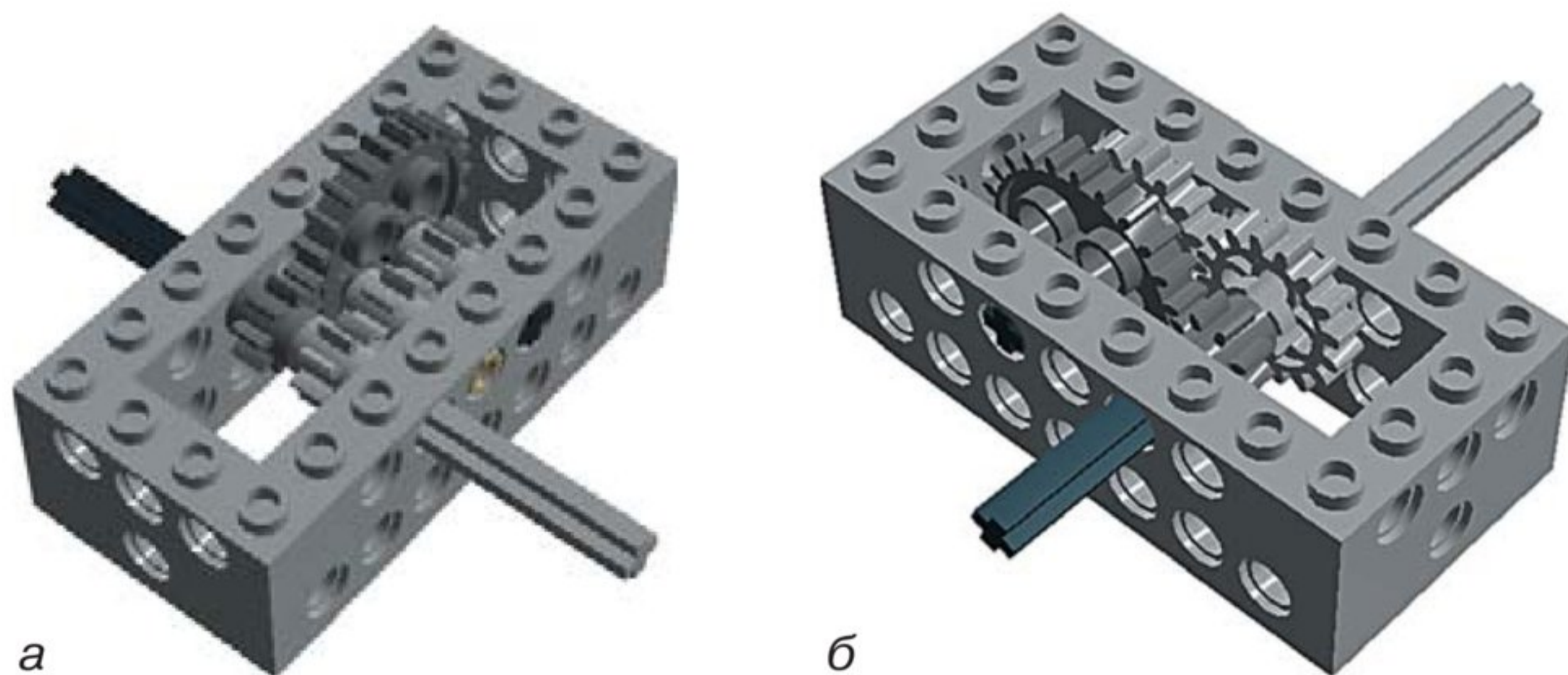



Рис. 1.45. *а—б* — соосные редукторы с отношением  $3:1$




торы незаменимы и позволяют повысить эффективность использования различных двигателей.

## Проверьте себя

1. Чем различаются соосная и развёрнутая схемы построения редуктора?
2. Почему новое машинное масло имеет золотисто-коричневатый оттенок, а отработанное — тёмно-серый?
3. Каким образом соосный редуктор может изменить направление вращения?
4. Используя детали одного конструктора, постройте соосный редуктор с максимальным передаточным отношением.
5. Постройте развёрнутый редуктор с максимальным передаточным отношением, полностью помещённый внутрь картера.

**Запомните** ♦ Соосный редуктор ♦ Развёрнутая схема  
♦ Соосная схема ♦ Картер редуктора

### Это интересно!

 Из всех шестерёнок и колёс конструктора «Технология и физика» можно собрать редуктор с передаточным отношением более 2 триллионов!

## § 1.6. Построение трёхмерной модели

Прежде чем построить настоящий механизм, современные инженеры создают на компьютере его *трёхмерную модель*. Для этого применяются профессиональные системы трёхмерного проектирования. Используя виртуальную модель, можно не только запустить процесс изготовления детали, но и предварительно проверить её на соответствие необходимым параметрам (форма, упругость, прочность и другие). Начать знакомство с виртуальным трёхмерным миром можно с бесплатного программного продукта *Lego Digital Designer* (сокращённо LDD). Каждый может установить его на свой домашний компьютер

и самостоятельно смоделировать небольшой механизм из деталей конструктора.

## Первая модель

При старте предлагаются три режима:

- ◆ Digital Designer;
- ◆ Mindstorms;
- ◆ Digital Designer Extended.

Для построения простейшей механической передачи при запуске можно выбрать любой из режимов, например, Mindstorms (рис. 1.46). Для начала работы следует нажать на пиктограмму «Free build».

В левой части окна редактора расположены палитры деталей. Каждую из них можно открыть, кликнув мыш-

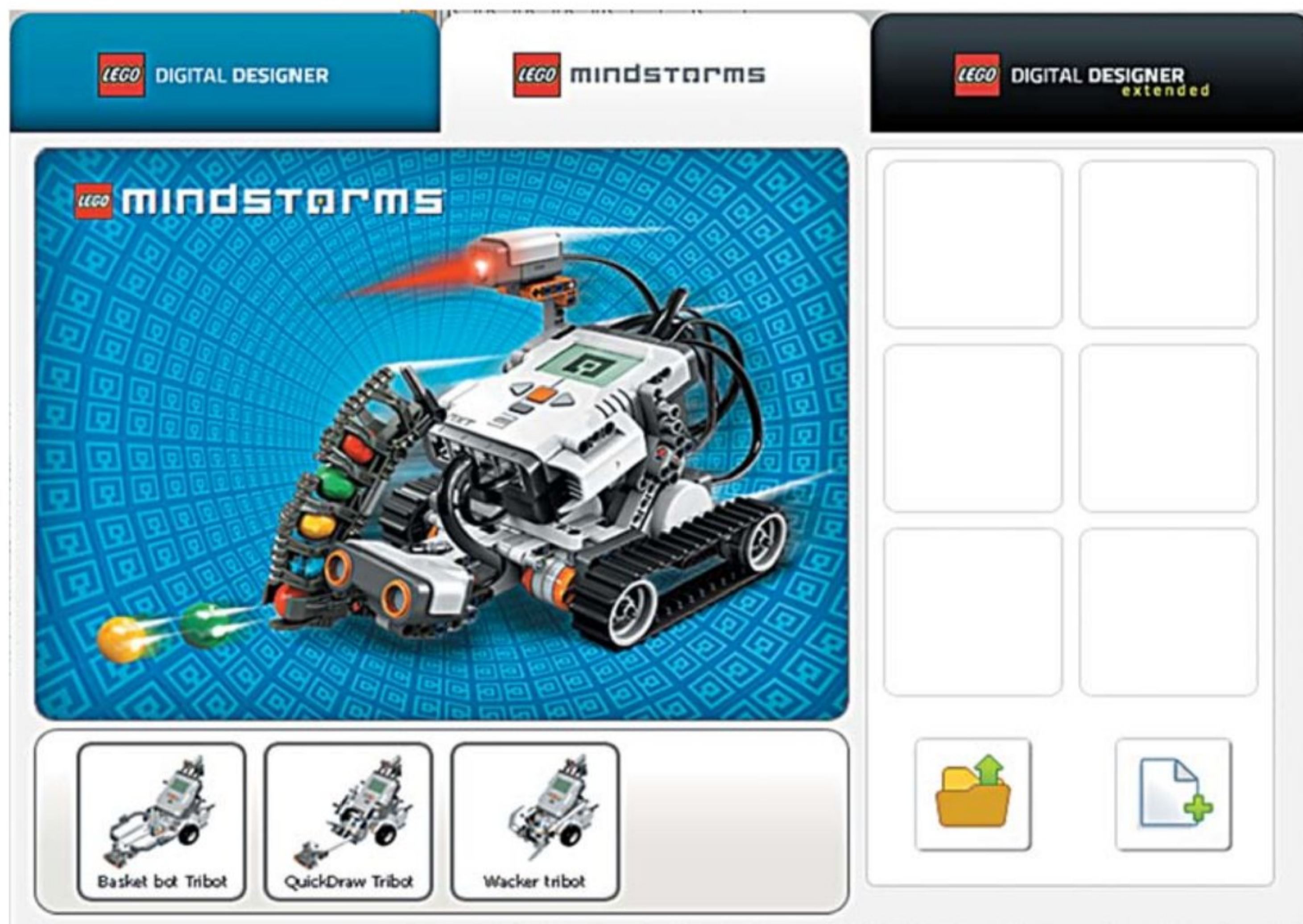
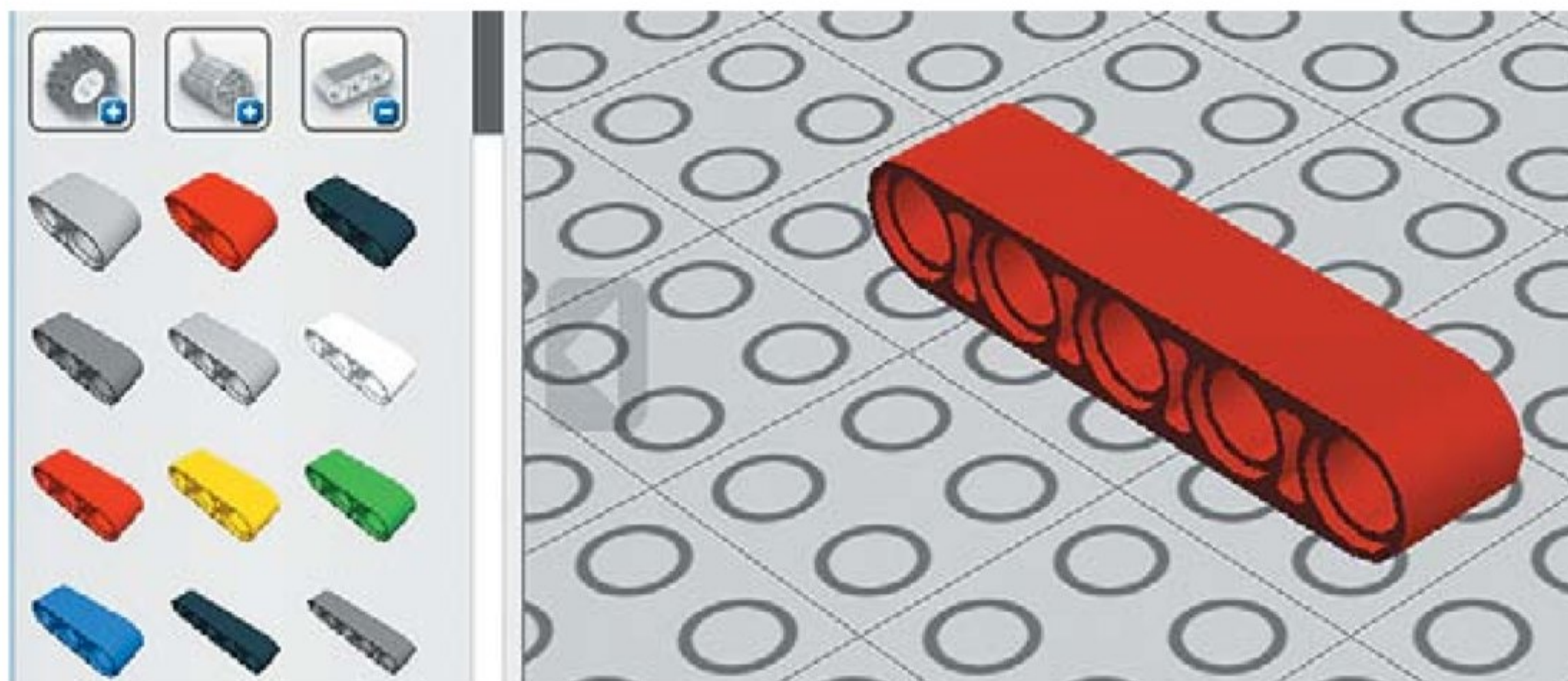
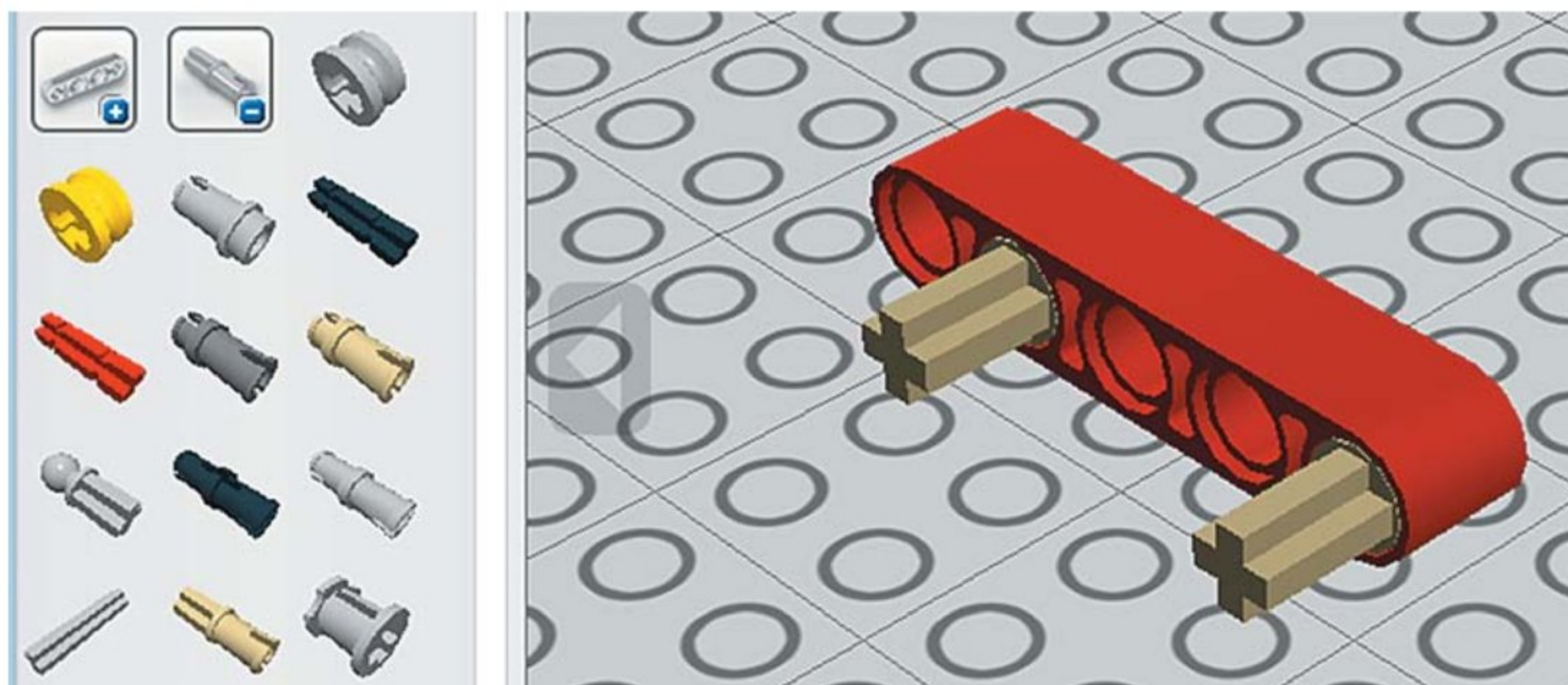


Рис. 1.46. Выбор режима Mindstorms



**Рис. 1.47.** Установка пятимодульной балки

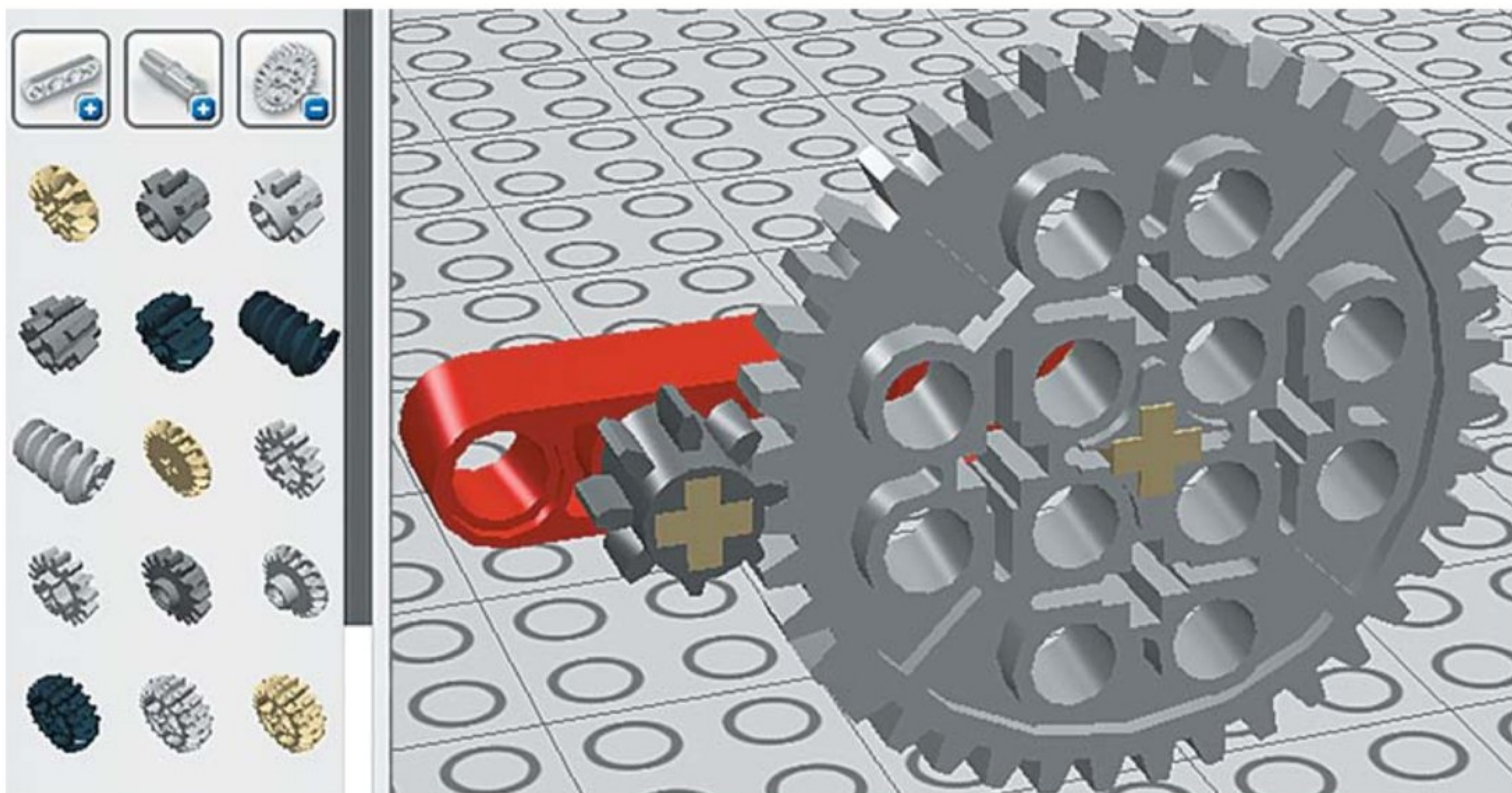


**Рис. 1.48.** Установка штифтов-полуосей

кой по соответствующей пиктограмме. Первая деталь — пятимодульная балка. Установив её в центре поля, можно свернуть палитру балок и раскрыть палитру штифтов (рис. 1.47).

Для создания простой механической передачи потребуются два штифта-полуоси, которые нужно установить в первое и четвёртое отверстия балки (рис. 1.48). Повер-





**Рис. 1.49.** Установка шестерёнок

ните штифты в нужном направлении обычными клавишами со стрелками на клавиатуре компьютера.

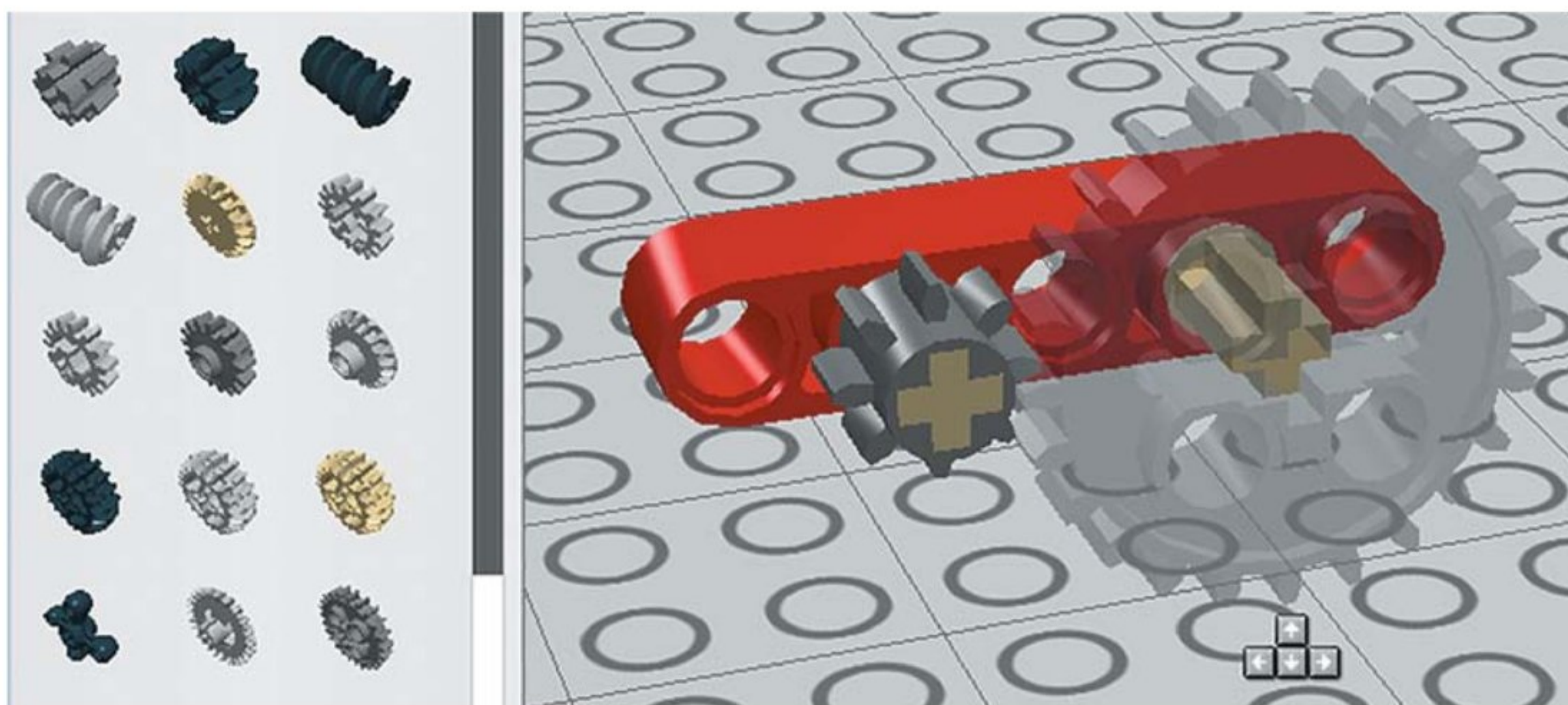
Теперь можно свернуть палитру штифтов и раскрыть палитру зубчатых колёс. На первый штифт установите 8-зубую шестерню, на второй — 40-зубую. Получился механизм с передаточным отношением  $i = 5 : 1$ .

Чтобы повернуть его, можно воспользоваться правой кнопкой мыши. Для приближения и удаления вращайте колёсико мыши (рис. 1.49).

Теперь следует сохранить полученную модель, назвав её, например, «Передача 5 к 1.lxf». Обратите внимание, что в названии нельзя использовать знаки «:» и «/». А расширение файла «.lxf» писать не нужно, его LDD добавит сам.

### **Поворот детали**

Теперь попробуем построить передачу с отношением 3:1. Для этого придётся заменить одну шестерёнку. Выберите



**Рис. 1.50.** Невозможно установить шестерёнку

инструмент «Удаление» в виде красного крестика и кликните мышкой на 40-зубую шестерню. Не беспокойтесь, удалённая шестерёнка не пропадёт, в компьютере их почти бесконечное количество.

Следующим действием переставьте освободившийся штифт-полуось в третье отверстие. Для этого потребуется снова выбрать инструмент в виде белой стрелки. Очевидно, что следующим действием нужно установить 24-зубую шестерню. Что же из этого получается? Нужная шестерёнка остаётся прозрачной, это означает, что установить её нельзя (рис. 1.50).

Видите причину? Дело в том, что совпадают два зубчика разных шестерёнок. Значит, нужно одну из них повернуть, чтобы освободить место для второй. Сделайте это с 8-зубой шестерёнкой. Выберите инструмент «Поворот» и выделите её. Сразу появляются три новых инструмента: зелёная стрелка, синий рычаг на круге и числовое поле. С помощью любого из них можно выполнить поворот шестерни примерно на  $\frac{1}{16}$  полного оборота, или  $22,5^\circ$  (рис. 1.51).

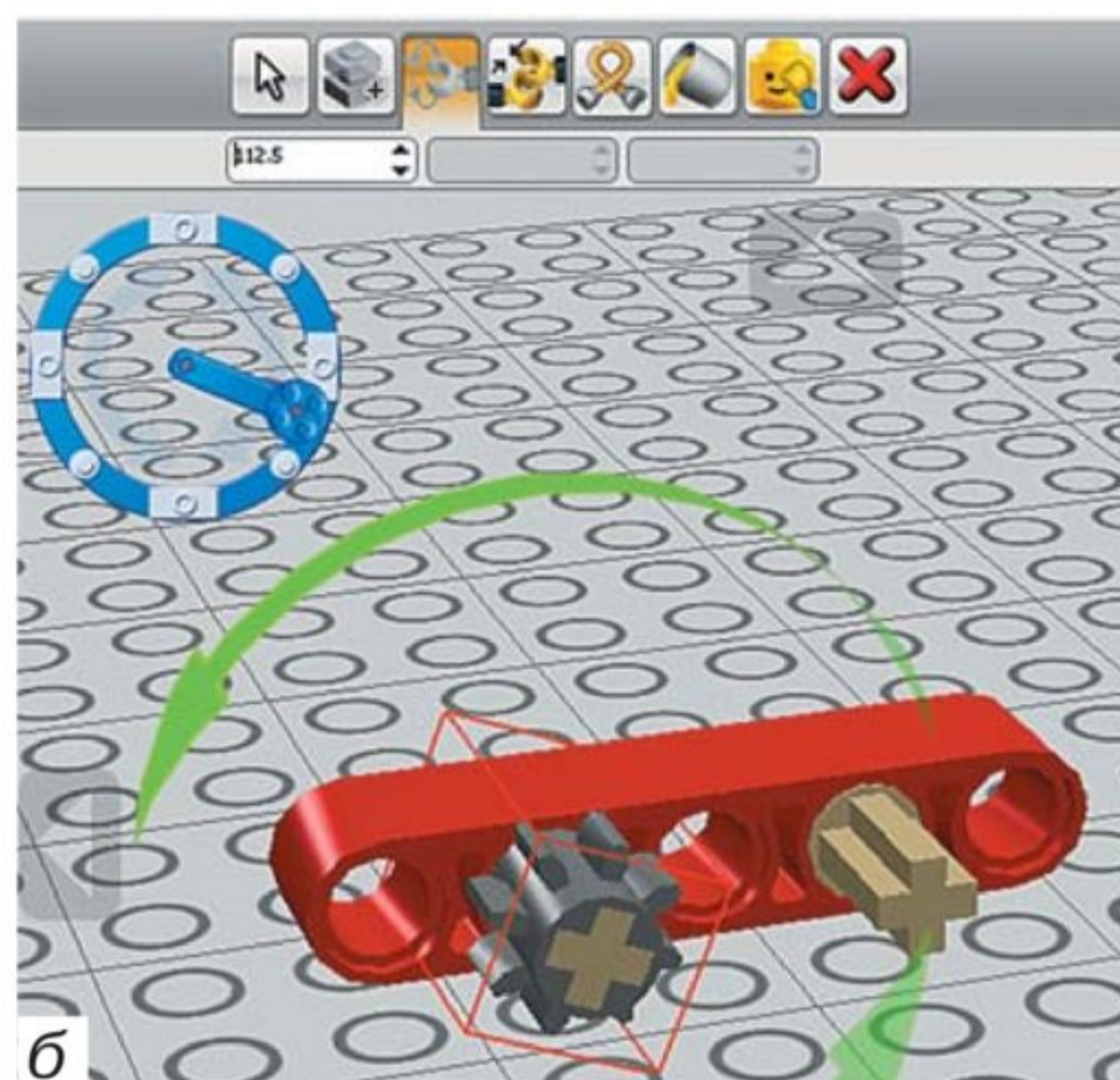
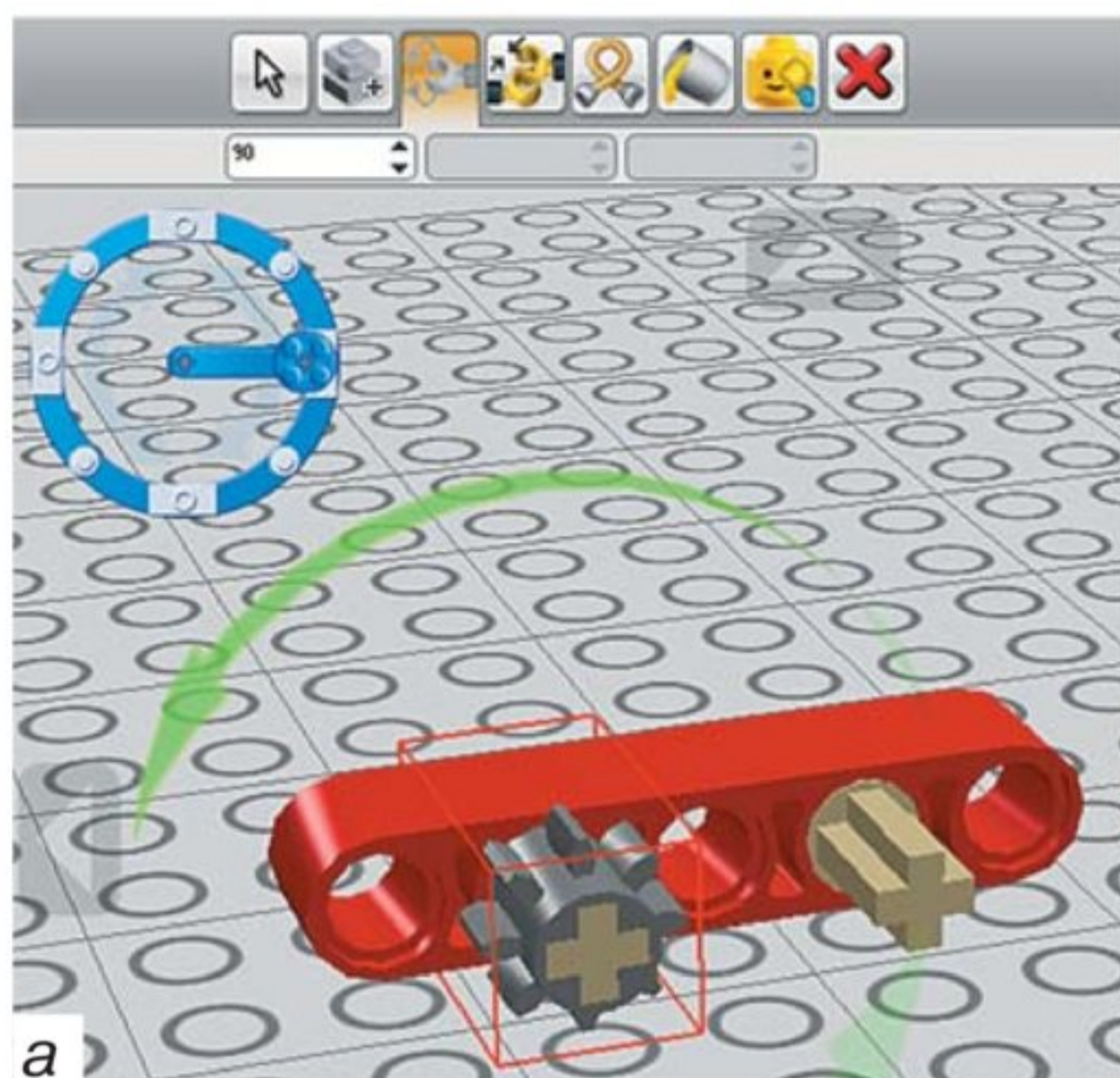


Рис. 1.51. а—б — поворот детали

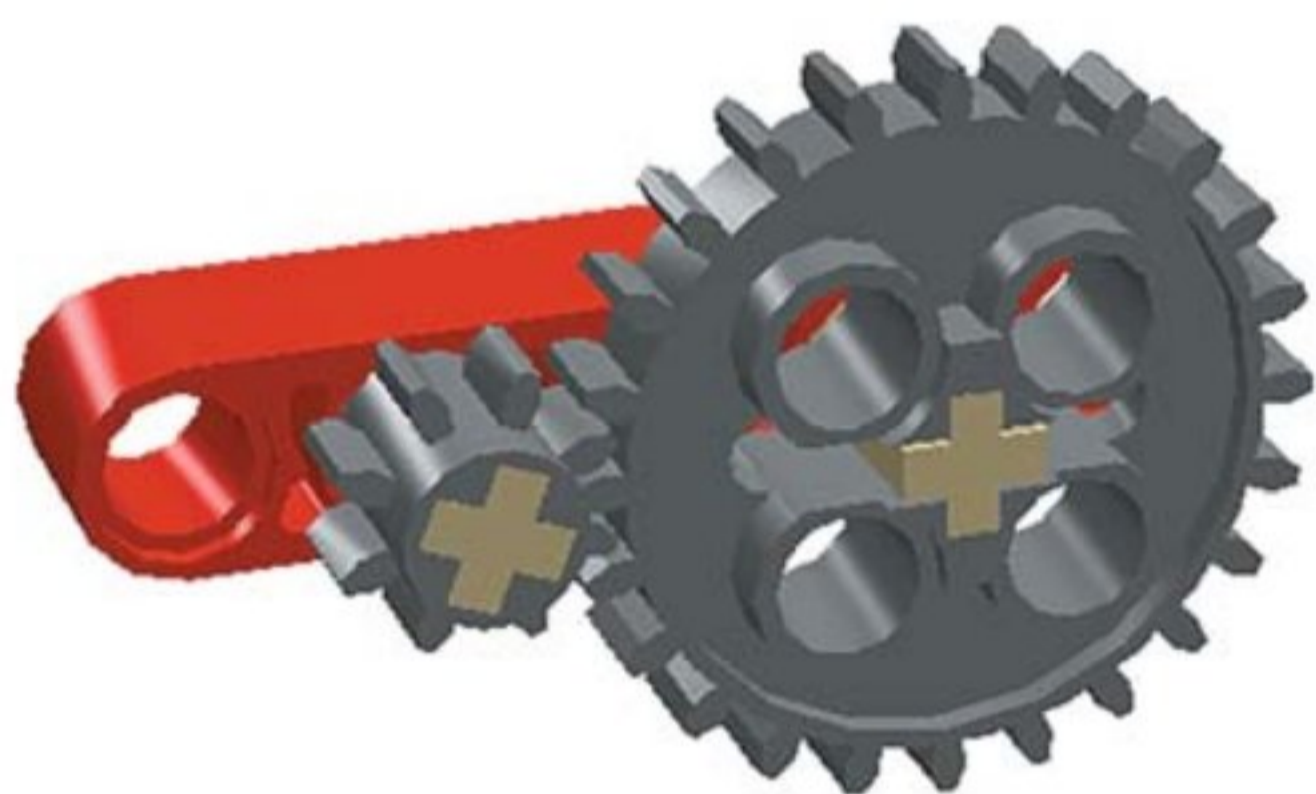


Рис. 1.52. Модель готова

Подобный поворот может быть выполнен по отношению к детали, которая установлена на любую ось вращения.

Теперь остаётся установить 24-зубую шестерню в нужное место, и задача выполнена (рис. 1.52). Не забудьте сохранить модель в файле с новым именем.

Напоследок сделайте «фотоснимок» модели через пункт *Take a screenshot* в меню *Tool Box*.



## Проверьте себя

1. Для чего используются среды трёхмерного проектирования?
2. По какой причине шестерни на соседних осях могут не стыковаться в 3D-модели?
3. Перечислите инструменты трёхмерного редактора и расскажите, какие функции они выполняют.
4. Используя шестерни на 8, 24 и 40 зубчиков, постройте трёхмерные модели мультипликаторов с передаточными отношениями:  
а) 1:9,    б) 1:15,    в) 1:25,    г) 1:27,    д) 1:45.




5. Проследите, чтобы оси были закреплены и конструкция была устойчивой.
6. Постройте трёхмерную модель соосного редуктора в картере с передаточным отношением 1:9.
7. Используя шестерни на 12, 20 и 36 зубчиков, постройте трёхмерные модели мультипликаторов с передаточными отношениями:  
а) 3:5,      б) 1:5,      в) 1:9,      г) 9:25,      д) 1:9.  
Проследите, чтобы оси были закреплены и конструкция была устойчивой.
8. Используя инструмент интеллектуальной стыковки Hing Align Tool, постройте конструкцию калитки с закрепляющей укосиной (см. рис. 1.21, в).

**Запомните** ♦ Трёхмерная модель



### Это интересно!

 В России создан профессиональный пакет трёхмерного проектирования, который называется «Компас-3D». Каждый школьник может получить для себя бесплатную версию. Аналогичная возможность имеется в отношении пакета *Autodesk Inventor*.



# 2

## Моторные механизмы

Всё в природе движется благодаря энергии, которая содержится во всей материи Вселенной. По сути, энергия — это возможность тела выполнить некоторую работу. Человек получает энергию от пищи, воды, воздуха, тепла, света. Но это лишь малая часть существующих возможностей. Природа щедро наделила нашу планету различными видами энергии, от тепловой до ядерной, и человек научился использовать некоторые из них для своих целей. Чтобы заменить тяжёлый ручной труд работой машины, был создан двигатель.

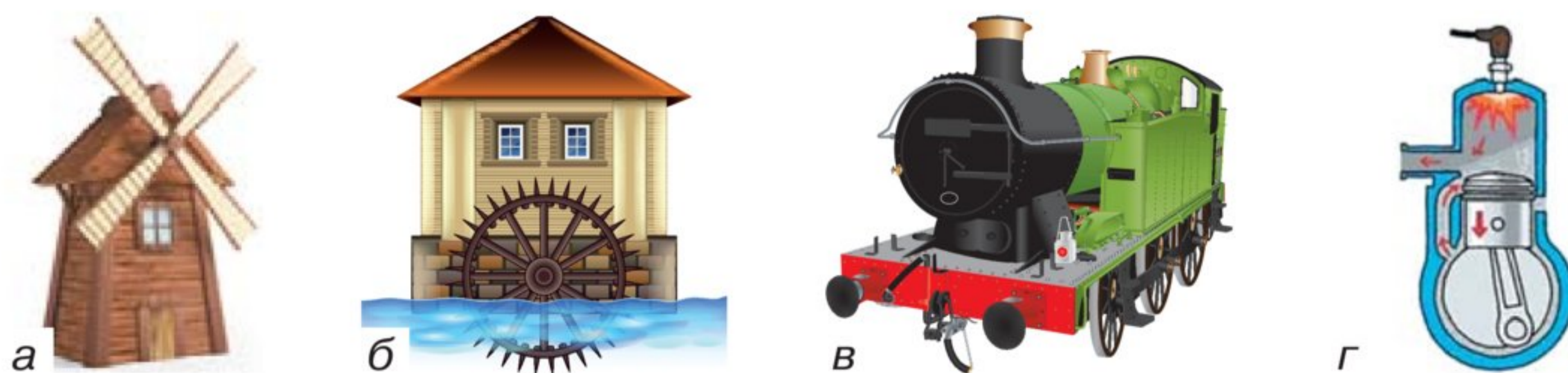
## § 2.1. Источники питания

*Двигатель* — это устройство, преобразующее какой-либо вид энергии в механическую. Синонимом термину «двигатель» является термин «мотор», заимствованный из немецкого языка. Двигатель, как правило, получает энергию от природных источников: парус и ветряная мельница — от ветра, гидротурбина и водяная мельница — от напора воды, паровая машина — от давления горячего пара, двигатель внутреннего сгорания — от сжигания топлива и так далее (рис. 2.1). Электрическая энергия, от которой работает электродвигатель, может быть вторичным источником, то есть вырабатываться с помощью специальных генераторов и других двигателей, работающих от природных источников. Кроме того, электрическую энергию можно получить из энергии Солнца или с помощью специальных химических реакций в элементах питания (в просторечии «батарейках») и аккумуляторах.

С принципом работы электрического двигателя вы познакомитесь на уроках физики. Наша задача — научиться правильно его использовать.

Итак, если двигатель работает от электричества, значит требуется источник питания: электросеть, батарейки, аккумуляторы.

Мы будем использовать двигатели только в учебных целях. Вы знаете, что электроэнергия характеризуется напряжением, например, наши дома оснащают розетками с напряжением 220 вольт (обозначается В), а в некоторых странах — 110 В. Не пытайтесь подключить к розетке с на-



**Рис. 2.1.** Различные типы двигателей используют энергию ветра (а), воды (б), пара (в), огня (г)



пряжением 220 В двигатель из конструктора: можно нанести серьёзный вред своему здоровью и устроить пожар. Подключение возможно с помощью специального блока питания, который преобразует высокое напряжение (110 или 220 В) в низкое (например, 5 или 12 В).

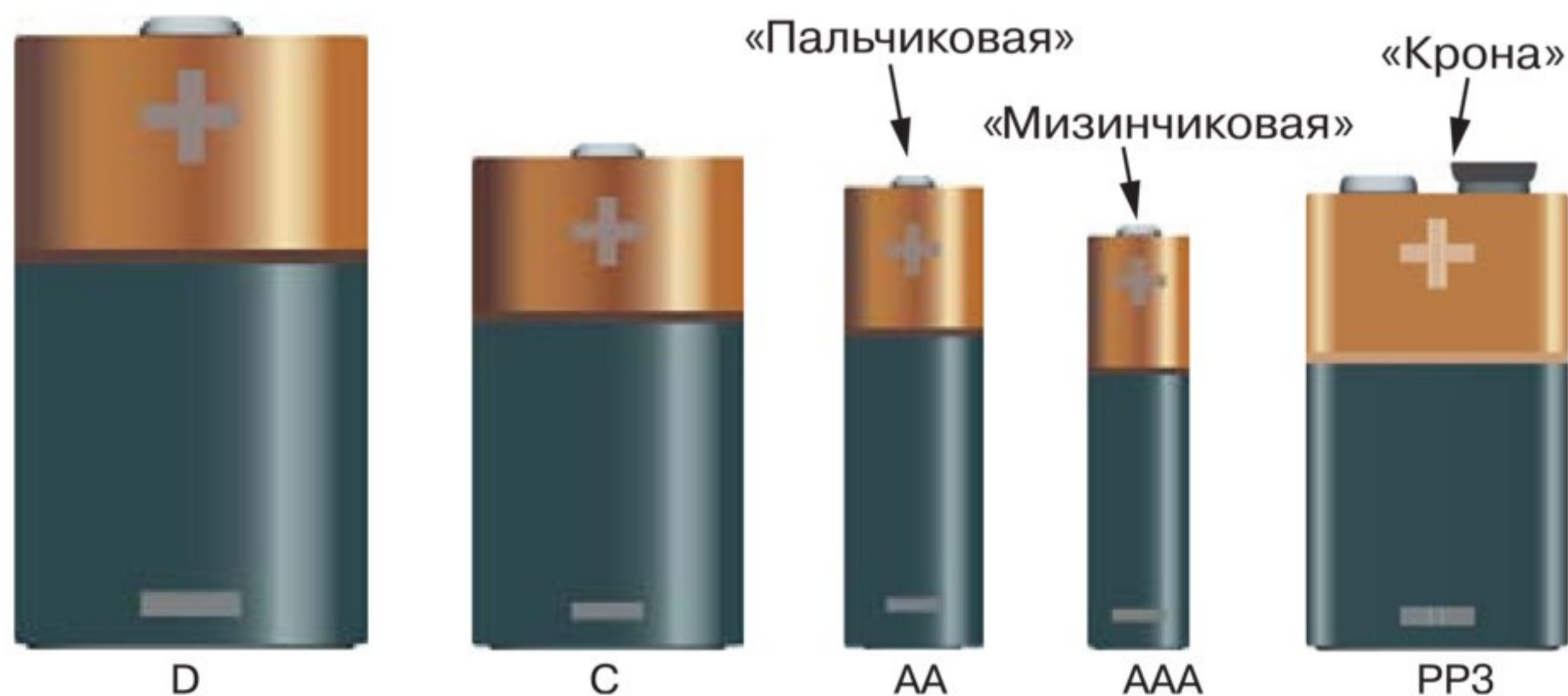
Для питания двигателей из конструктора чаще используют химические источники тока — батарейки или аккумуляторы.



**Элемент питания (батарейка)** — это химический источник электрического тока. Внутри неё протекает сложная химическая реакция, в ходе которой вырабатывается электроэнергия определённого напряжения. Обычно, отдельные элементы обеспечивают напряжение не более 1,5 В. Для того, чтобы получить более высокое напряжение, их соединяют в батареи (батарейные блоки). Например, такой батареей является батарея типа «Крона» (рис. 2.2).

**Химическая реакция** — это взаимодействие различных веществ, в результате которого образуются новые вещества и выделяется или поглощается энергия. В батарейках электрическая энергия выделяется.

Когда реакция заканчивается, напряжение падает, батарейка «садится». Севшие батарейки нельзя разбирать и выбрасывать с обычным мусором, поскольку внутри них содержатся вещества, очень вредные для человека и окружающей среды. Использованные батарейки следу-



**Рис. 2.2.** Типы бытовых батареек



Рис. 2.3. а—в — аккумуляторы

ет сдавать в специальные пункты утилизации, откуда они поступают на мусороперерабатывающие заводы.

В зависимости от типа химической реакции, которая протекает в батарейке, и её физического размера, она может обеспечить различную длительность работы и напряжение.

**Электрический аккумулятор** — это источник тока многократного действия. Когда заряд аккумулятора достигает нижней допустимой границы, его необходимо снова подзарядить с помощью специального зарядного устройства. Современные аккумуляторы позволяют производить тысячи циклов «заряд–разряд».

В повседневной жизни вы встречаетесь с аккумуляторами в мобильных телефонах, видеокамерах, автомобилях и множестве других устройств (рис. 2.3).

Есть ещё одна характеристика, по которой могут сильно различаться батарейки одного типа — **ёмкость**. Ёмкость характеризует длительность разряда батарейки. То есть, сколько энергии может отдать батарейка, пока не разрядится. Ёмкость характеризуется единицей Ампер-час,







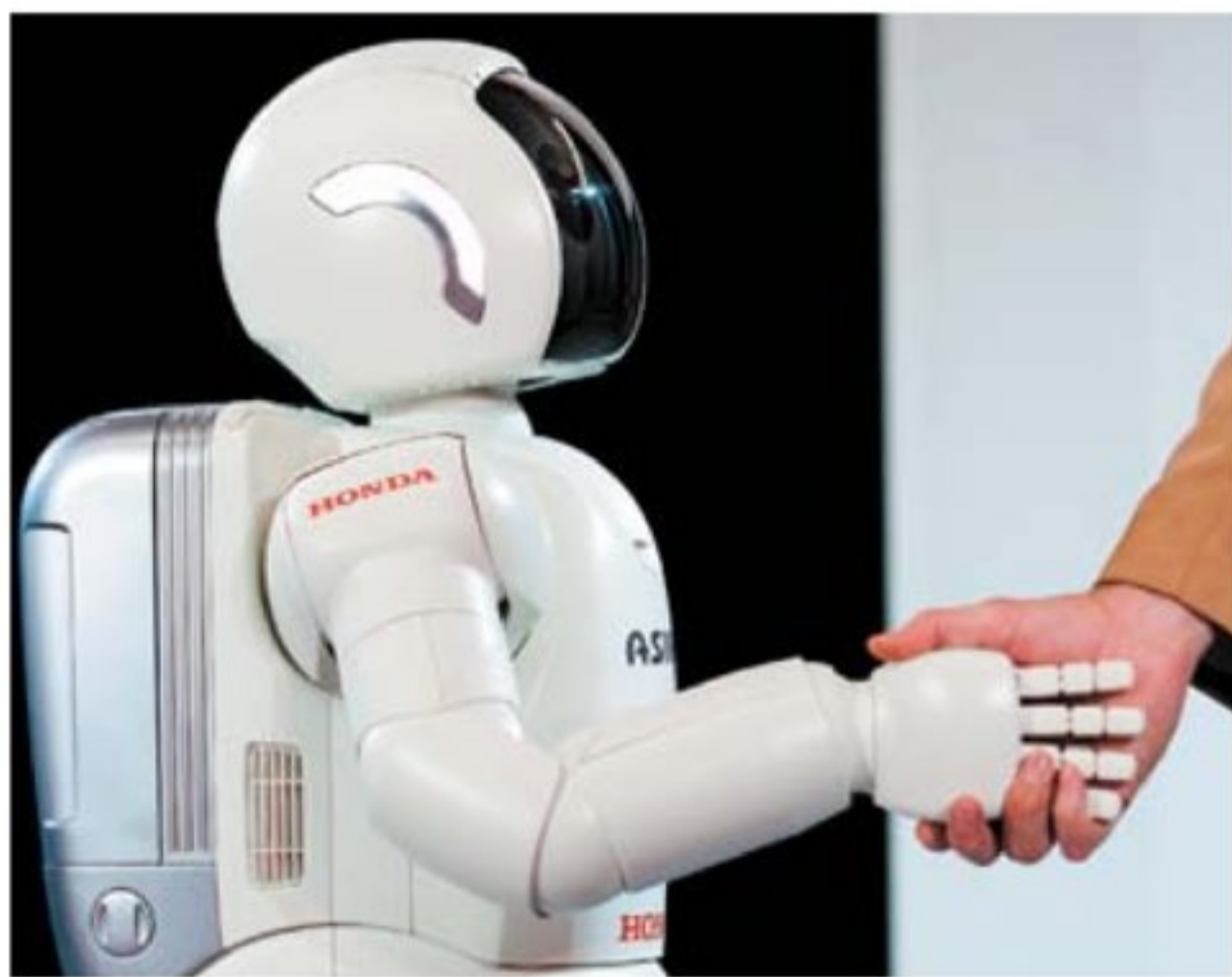
сокращённо Ач (англ. Ah), а в бытовых аккумуляторах используется единица миллиАмпер-час, сокращённо мАч (англ. mAh). Ёмкость 1000 мАч соответствует 1 Ач. Бытовые пальчиковые аккумуляторы имеют ёмкость от 1 до 3 Ач. Со временем ёмкость аккумулятора падает. Кроме того, ёмкость зависит от температуры: у некоторых аккумуляторов на морозе она снижается почти до нуля.

При работе в сложных устройствах с высоким потреблением энергии, например, в аудиоплеере или фотоаппарате, и обычные батарейки, и аккумуляторы разряжаются за несколько дней или даже часов. При низком энергопотреблении, например, в настенных часах или пульте от телевизора батарейка может проработать несколько лет.

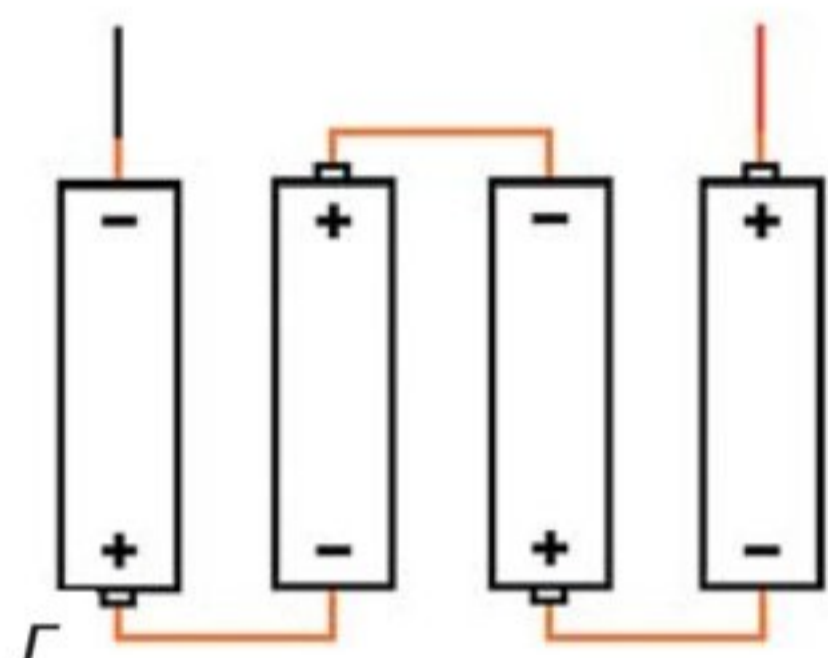
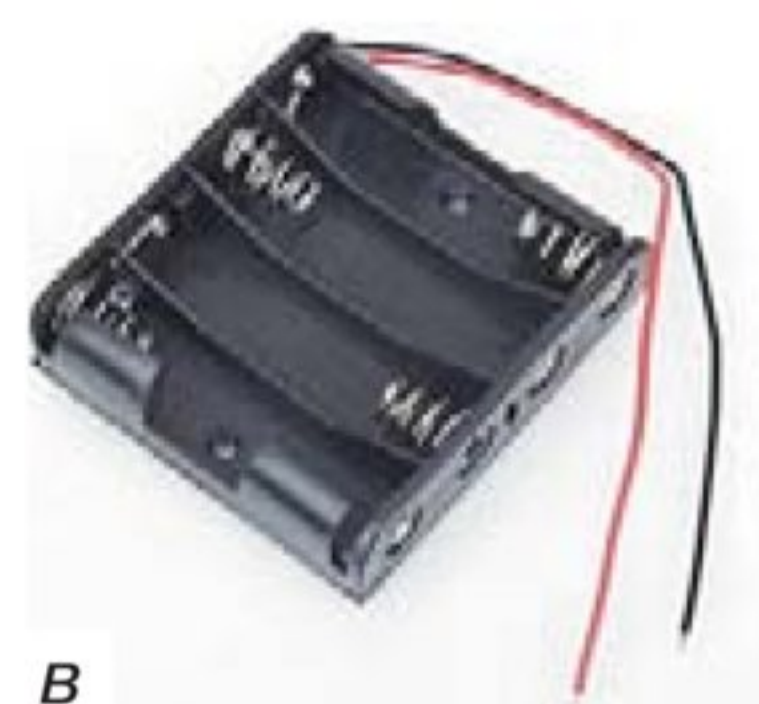
Ёмкость аккумулятора, как правило, зависит от его массы. Современная робототехника имеет серьёзные ограничения в этом вопросе: чтобы долго обеспечивать работа энергией, аккумулятор должен быть либо очень тяжёлым, либо вредным (например, радиоактивным), либо постоянно подзаряжаться от внешних источников питания. Например, человекоподобный робот со встроенным аккумулятором, как правило, может работать без подзарядки чуть более полу-

часа. Истории о ядерных батарейках для терминатора, которые работают годами с высоким напряжением, пока остаются фантастикой и требуют новых открытий и технологических прорывов (рис. 2.4).

Некоторым устройствам для работы недостаточно напряжения 1,2 или 1,5 В. Тогда используется цепочка из батареек, в которой «плюс» одной батареи соединяется с «минусом» другой (будьте внимательны, не перепутайте!). Например, моторы «ЛЕГО» работают от 6 батареек типа АА, что



**Рис. 2.4.** Робот Asimo работает 40 минут без подзарядки на аккумуляторе массой около 8 кг, расположенном в центральной части корпуса



**Рис. 2.5.** Батарейные блоки (а—в) и последовательный способ соединения батарей (г)

даёт от 7,2 до 9 В. В школьном наборе «ТРИК» используется аккумулятор напряжением 11,1 В (рис. 2.5).

Чтобы электроника контроллера (управляющего устройства) не давала сбоев при снижении напряжения батареи, используют специальные стабилизаторы питания. Но моторы, как правило, получают напряжение непосредственно от источника питания, поэтому на севших аккумуляторах мотор вращается медленнее, чем на заряженных недавно. Если ваши моторы стали работать явно медленнее, нужно заменить батарейки или зарядить аккумулятор.

## Проверьте себя

1. Посчитайте, какое напряжение будет у батарейного блока из двух, трёх, четырёх, шести и восьми пальчиковых аккумуляторов.
2. Если в предыдущем задании аккумуляторы заменить на батарейки, какое получится напряжение?

**Запомните** ♦ Двигатель ♦ Элемент питания (батарейка)  
♦ Электрический аккумулятор ♦ Ёмкость



### Это интересно!

💡 Одна пальчиковая батарейка, выброшенная в мусорное ведро, загрязняет тяжёлыми металлами около 20 квадратных метров земли.

💡 Новые литиевые пальчиковые батарейки в первые несколько минут после начала использования имеют напряжение около 1,6 В.

## § 2.2. Электродвигатель

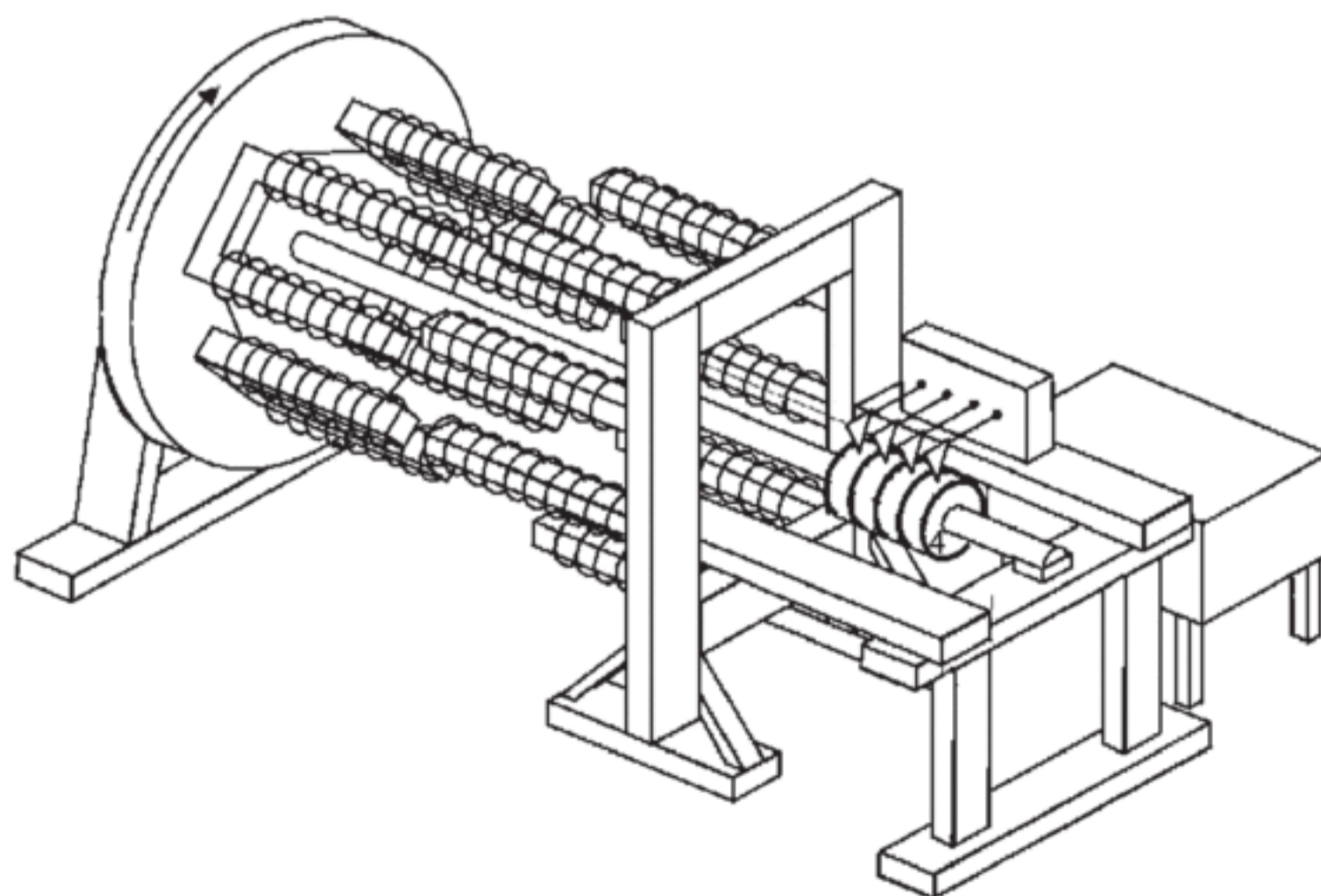
### Из истории электродвигателя

Первый применимый на практике электродвигатель постоянного тока создал в 1834 году русский учёный немецкого происхождения *Б. С. Якоби* (рис. 2.6). Впоследствии он построил лодку с электродвигателем, работающим от батарей, которая двигалась с 14 пассажирами по Неве против течения.

Современные электродвигатели получили широкое применение: от вибрации мобильного телефона до движения железнодорожного электровоза (рис. 2.7). Активно применяются электромобили (они же электрокары — от англ. «электрокар»), электропогрузчики, работающие полностью от аккумуляторов.



а



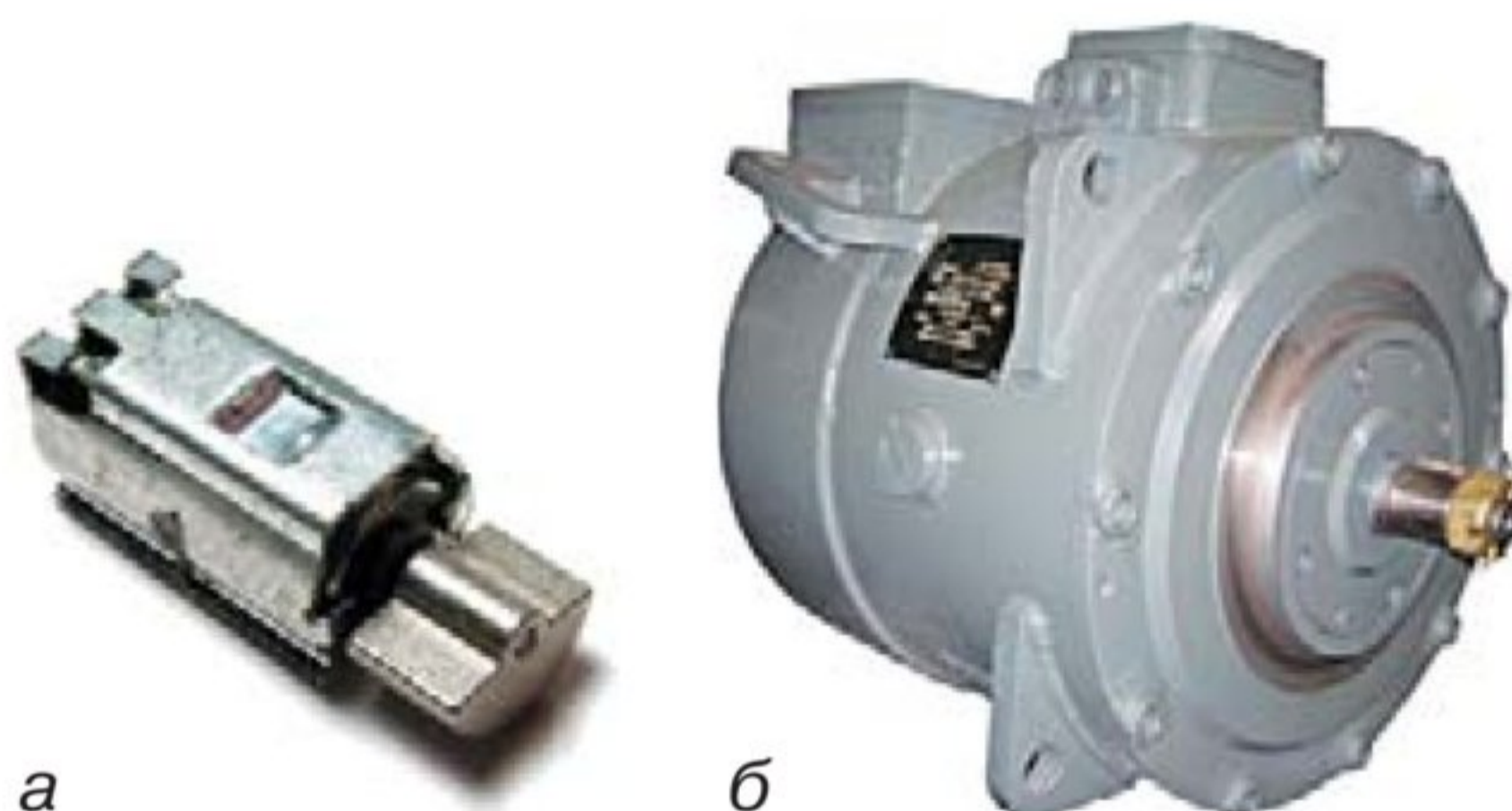
б

Рис. 2.6. Б. С. Якоби (а) и его электродвигатель постоянного тока (б)

## Электромобили

В 1899 году инженер *И. В. Романов* создал первый российский электромобиль под названием «кеб» (рис. 2.8). Он был существенно легче зарубежных моделей, а при торможении обеспечивал автоматическую подзарядку аккумуляторной батареи, что увеличивало длительность работы. Всего кеб мог проехать 60 км без подзарядки, развивая скорость до 35 км/ч.

С тех пор многие технологии изменились, а с началом XXI века интерес человечества к теме электромобилей значительно вырос (рис. 2.9).



**Рис. 2.7.** Вибромотор мобильного телефона (а) и двигатель электровоза (б) различаются размерами в сотни раз



**Рис. 2.8.** Изобретатель И. В. Романов и электромобиль «кеб»



**Рис. 2.9.** Российский электромобиль El Lada (2014 год) со станцией зарядки (а) и электропогрузчик ЗиК (б)



**Рис. 2.10.** Одноместный электрокар Твизи

Самым популярным электромобилем в мире стал одноместный Рено Твизи (рис. 2.10). Он способен проехать без подзарядки до 100 км.

Интересный проект по внедрению электромобилей запустила компания «Тесла моторс». С 2006 года она производит электромобили «Тесла», названные так в честь знаменитого изобретателя *Николы Теслы*. В ряде стран были созданы целые сети станций «Суперзарядка» для быстрой подзарядки или роботизированной замены аккумуляторов

(рис. 2.11). Эти станции в основном работают на солнечных батареях. Заряженный автомобиль может проехать до 350 км. Машины «Тесла», как и большинство современных электромобилей, оснащены функцией автопилота, то есть по сути являются роботами.

*Автопилот* — режим управления транспортным средством, при котором ориентация на дороге и принятие решений производится бортовым компьютером без участия либо с частичным участием человека.

Несмотря на то что электромобили появились более 100 лет назад, на целый век они были вытеснены более дешёвыми автомобилями на бензиновом топливе. Но бла-



**Рис. 2.11.** Электродвигатель «Тесла» (а) и зарядная станция «Суперзарядка» (б)



годаря появлению новых технологий, ситуация меняется. Поскольку электрический двигатель существенно экологичнее и безопаснее бензинового, в будущем можно ожидать появления на дорогах большого количества электромобилей.

Вы тоже сможете построить свой маленький электромобиль. Первая модель будет очень простой — одно моторная тележка, двигающаяся только по прямой. Воспользуйтесь имеющимся конструктором и найдите в нём электродвигатель.

### Характеристики электродвигателя

У электродвигателя из любого конструктора имеется минимум два контакта и, как правило, электроэнергия (питание) на него подаётся двумя проводами: «+» и «-». Причём в зависимости от подключения проводов двигатель будет вращаться в одну или другую сторону. Производители учебных конструкторов обычно создают такие разъёмы (коннекторы), которые не позволяют ошибиться с подключением (рис. 2.12). Однако бывают исключения.

Электродвигатель, как правило, оснащён редуктором для обеспечения достаточной силы вращения, чтобы сдви-



**Рис. 2.12.** а—г — примеры электродвигателей для учебных роботов



Рис. 2.13. а—в — редукторы для электродвигателей

нуть тележку с места или поднять необходимый груз (рис. 2.13).

При выборе электродвигателя в первую очередь оценивают два параметра: скорость вращения и силу.



**Скорость вращения** измеряют в оборотах в минуту (об./мин). Для силы вращения в технике существуют специальные термины: **крутящий момент**, или **момент силы**. Крутящий момент можно определить как усилие, развиваемое двигателем. Представьте себе, что вы едете на велосипеде и, решив ускориться, начали изо всех сил крутить педали. Тогда ваши ноги — двигатель велосипеда — своим усилием увеличили крутящий момент. Эту величину можно измерить в килограммах, умноженных на сантиметр ( $\text{кг} \cdot \text{см}$ ) и определить так: сколько килограммов мотор может поднять на рычаге с плечом длиной 1 см. Плечом рычага будем называть расстояние от оси вращения до точки приложения усилия. Чем длиннее плечо, тем труднее поднимать тяжести (как трудно поднять стул



Рис. 2.14. Изменение плеча рычага на моторе

на вытянутой руке). Например, большой двигатель EV3 имеет скорость вращения 170 об./мин и крутящий момент 2 кг·см. Теперь представьте себе, что плечо удлиннили в 10 раз (рис. 2.14). Какой груз сможет поднять мотор на рычаге 10 см?

$$x \cdot 10 = 2$$

$$x = \frac{2}{10} = 0,2.$$

Ответ: 0,2 кг, или 200 г. Это значит, что при увеличении плеча рычага сила уменьшается. В таблице 2.1 пред-

**Таблица 2.1.** Характеристика учебных электродвигателей

Двигатель	Напряжение	Скорость вращения	Крутящий момент
 Средний мотор PF	9 В	400 об./мин	0,5 кг·см
 Мотор NXT	9 В	170 об./мин	2 кг·см
 Большой мотор EV3	9 В	175 об./мин	2 кг·см
 Средний мотор EV3	9 В	250 об./мин	0,8 кг·см
 Средний мотор из набора «ТРИК»	12 В	95 об./мин	5,5 кг·см
 Микро-мотор из набора «Амперка»	6 В	320 об./мин	2,2 кг·см





ставлена характеристика двигателей из основных робототехнических наборов.

С понятиями «рычаг» и «крутящий момент» вы более подробно познакомитесь на уроках физики. А сейчас полученный результат проверьте на одномоторной тележке, изменяя радиус колеса, по сути — рычаг.

## Одномоторная тележка

Подобно вагонетке, одномоторная тележка может двигаться только вперёд или назад. Постройте простейшую модель с помощью имеющегося в классе конструктора (рис. 2.15). Для этого потребуются мотор, блок питания и две одинаковые 16-зубые шестерни.

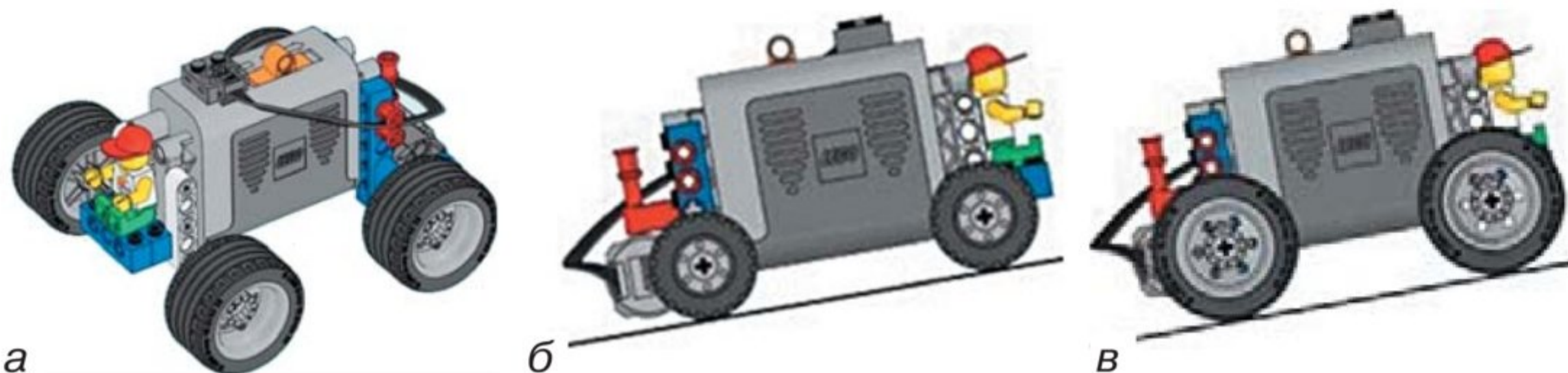
Убедитесь, что тележка может ехать по столу или по полу. Проверьте, может ли она заехать на горку. Если не получается, запустите тележку на горку противоположной стороной. Можно заметить, что при достаточно крутом склоне колёса начинают проскальзывать. Это объясняется тем, что у колёс недостаточное сцепление с дорогой.

Аналогичный опыт проделайте, поставив перед тележкой коробку с грузом. Увеличивайте груз до тех пор, пока тележка может его сдвинуть.

Теперь замените колёса на более крупные и снова проверьте все характеристики: изменение скорости движения,



**Рис. 2.15.** Одномоторная тележка из конструктора «Технология и физика» (а), батарейный блок (б) и мотор (в) для неё



**Рис. 2.16.** а—в — тележки с разными диаметрами колёс

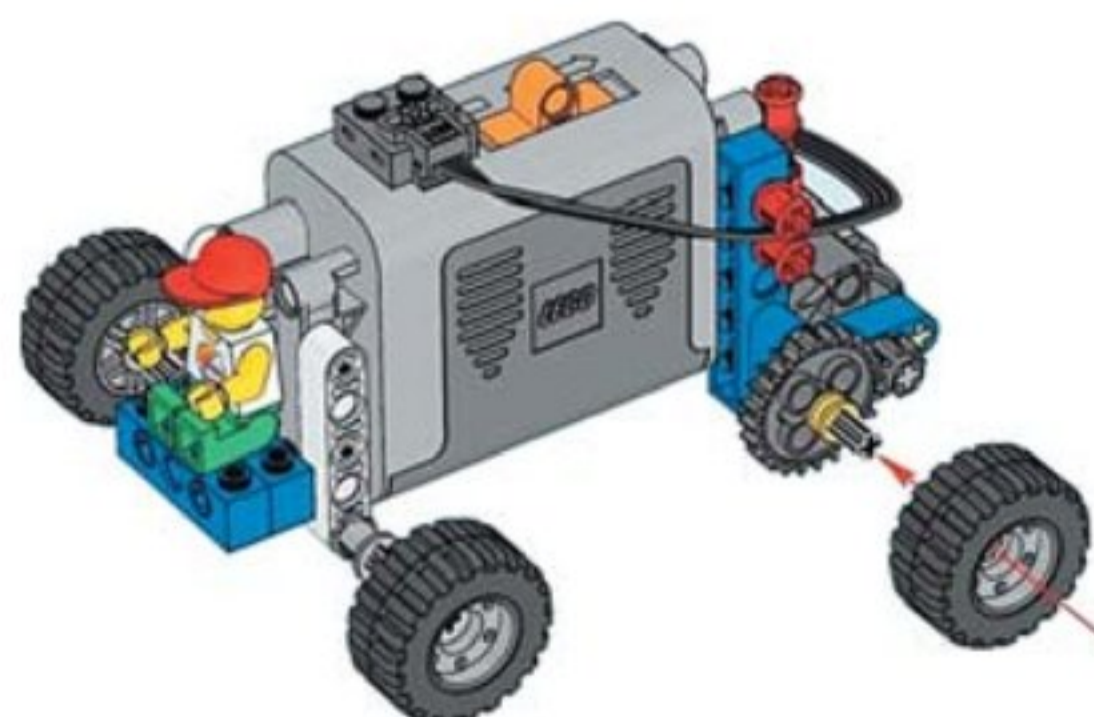
заезд на горку, проскальзывание колёс, толкание груза (рис. 2.16). Можно заметить, что тележка стала ехать быстрее, но на достаточно крутом склоне большие колёса не проскальзывают, а просто останавливаются. Это значит, что двигателю не хватает силы (*крутящего момента*), чтобы сдвинуть тележку. Изучим теперь возможности механической передачи.

Замените пару одинаковых 16-зубых шестерёнок на 24-зубую и 8-зубую так, как показано на рисунке 2.17: на ось мотора — бóльшую шестерню, на ось колеса — меньшую. Проведите те же опыты. Что изменилось? Мы получили подобие гоночного автомобиля. Скорость вращения колёс увеличилась втрое. Только тележка стала очень «слабой» и больше не может заехать на горку или столкнуть груз.

Итак, скоростная тележка не предназначена для выполнения силовых задач. Перестановкой шестерёнок пре-



**Рис. 2.17.** Увеличение скорости тележки



**Рис. 2.18.** Тележка с редуктором с передаточным отношением 1:3



**Рис. 2.19.** Шестерни в коробке передач автомобиля

образуем её в силовую тележку, установив на ось мотора меньшую шестерню, а на ось колеса — большую. Проведите опыты и убедитесь, что все ранее описанные силовые задачи решены (рис. 2.18).

В коробке передач настоящего автомобиля при заездке на горку также устанавливается понижающая передача с минимальным передаточным отношением (рис. 2.19).



### Проверьте себя

1. Почему электромобили распространены меньше, чем автомобили, работающие на бензиновом топливе?
2. Что должен уметь автопилот электромобиля, передвигающегося по улицам города?
3. Почему быстрый автомобиль не значит сильный?
4. Почему слишком быстрый двигатель не может сдвинуть транспортное средство с места?
5. Мальчик массой 40 кг едет на велосипеде с длиной рычага каждой педали 20 см. Какой крутящий момент он создаёт, если, привставая с сиденья, всей своей массой нажимает на педаль?
6. Установите на мотор тележки мультипликатор, повышающий скорость в 5 раз и более. Устройте состязания электромобилей на скорость.
7. Постройте установку для визуального определения скорости вращения мотора под нагрузкой. Используйте многоступенчатый редуктор с большим передаточным отношением (например,  $i = 75 : 1$ ). Засеките время  $t$  в секундах, за которое ведомая ось сделает один полный оборот. Определите скорость по формуле:

$$\omega = i \cdot \frac{60}{t} \quad (\omega \text{ — омега}).$$

8. Повторите тот же опыт для другого передаточного отношения и сравните результаты.



**Запомните** ♦ Скорость вращения ♦ Крутящий момент  
♦ Одномоторная тележка

## Это интересно!

💡 При запуске, чтобы преодолеть инерцию, двигатель развивает большее усилие, чем при нормальном вращении. Характеристика двигателя на старте вращения называется *пусковым моментом*. У двигателя EV3 пусковой момент вдвое выше, чем крутящий момент, то есть равен  $4 \text{ кг} \cdot \text{см}$ .

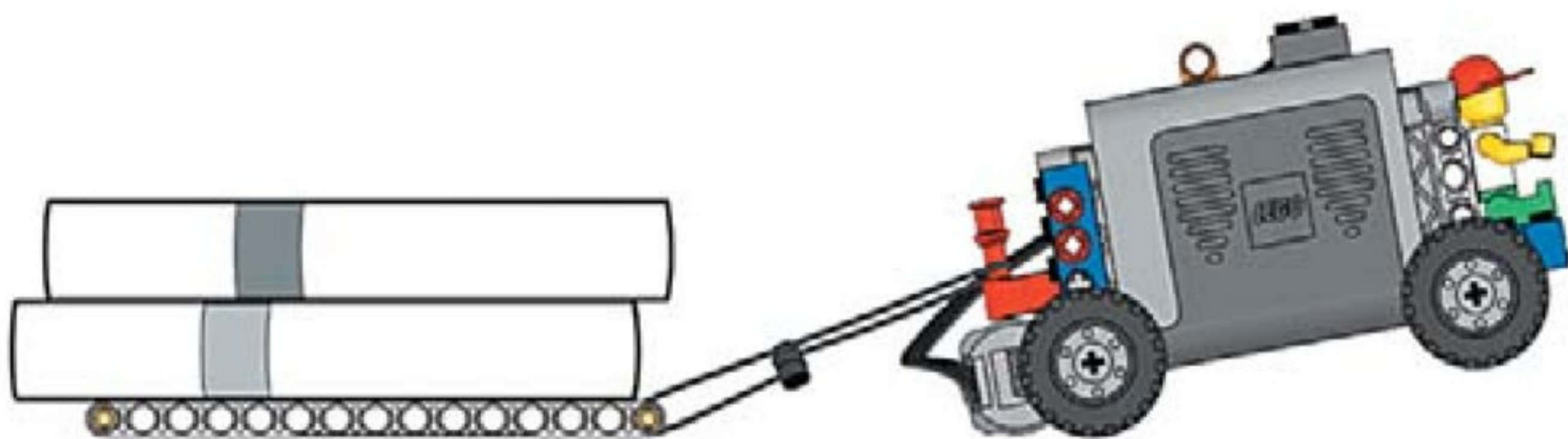
## § 2.3. Тягловые машины

### Тягач из тележки

Транспортные средства нужны не только для перемещения людей и грузов, но и для выполнения различной тяжёлой работы. Такие машины, как бульдозер, трактор, тягач — обладают огромной силой и относительно невысокой скоростью (рис. 2.20). Это достигается не только за счёт мощ-



**Рис. 2.20.** Тягловые машины: а — бульдозер; б — трактор; в — тягач



**Рис. 2.21.** Тележка работает как тягач

ного двигателя, рассчитанного на большие нагрузки, но и за счёт передаточного отношения.

На предыдущем занятии вы собрали силовую тележку с редуктором с передаточным отношением 3:1. Сделайте из неё тягач. Для этого в задней части тележки присутствует специальный штифт, на который можно накинуть крюк или петлю каната. Прицеп постройте из дополнительных деталей, и позвольте ему «волочиться» по земле, чтобы дать тягачу достаточную нагрузку. Сверху можно положить книги (рис. 2.21).

Нагружайте прицеп до тех пор, пока тележка не начнёт подниматься «на дыбы». Подумайте, как можно усовершенствовать конструкцию, чтобы этого не происходило? Поможет ли замена колёс или потребуются дополнительные детали?

## **Перетягивание каната**

В игре «перетягивание каната» выяснится, какой из тягачей самый сильный и устойчивый.

Площадка должна представлять собой линию с тремя перекрёстками — центральным и двумя боковыми. Оба боковых перекрёстка должны находиться на расстоянии 20 см от центрального. В качестве каната можно использовать толстую капроновую нить. Флажок, прикреплённый ровно посередине каната, при старте должен оказаться над центральным перекрёстком. По команде судьи

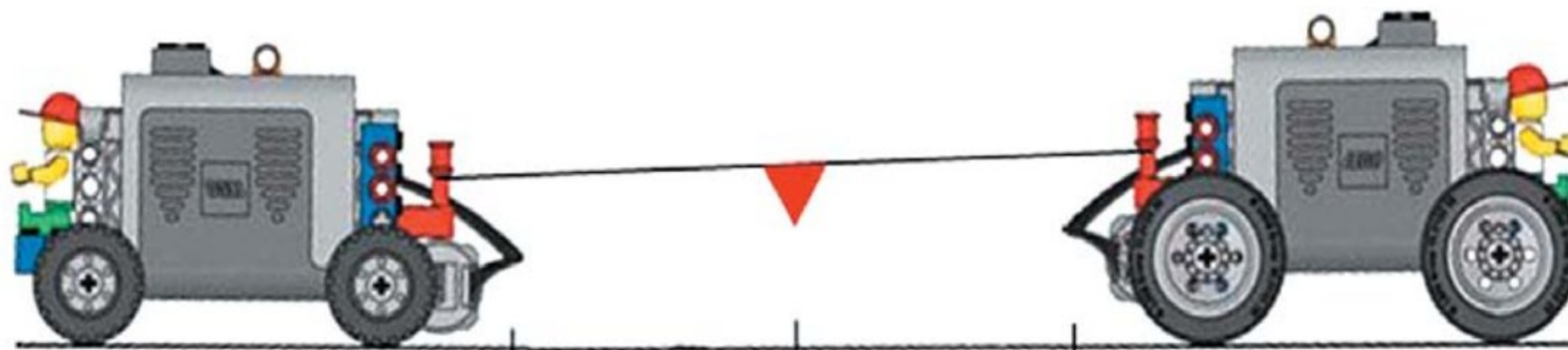
**Таблица 2.2.** Причины недостатков тягачей и их устранение

Недостаток	Причина	Что делать
Тележка переворачивается	Центр масс смещён назад	Установите дополнительный груз в передней части тележки
Колёса блокируются	Не хватает силы (крутящего момента)	Установите редуктор с достаточно большим передаточным числом
Колёса вращаются, но проскальзывают	Не хватает сцепления колёс с поверхностью	Убедитесь, что тележка опирается на ведущие колёса, протрите шины влажной салфеткой, замените колёса, увеличьте массу тележки, сместите центр масс вперёд

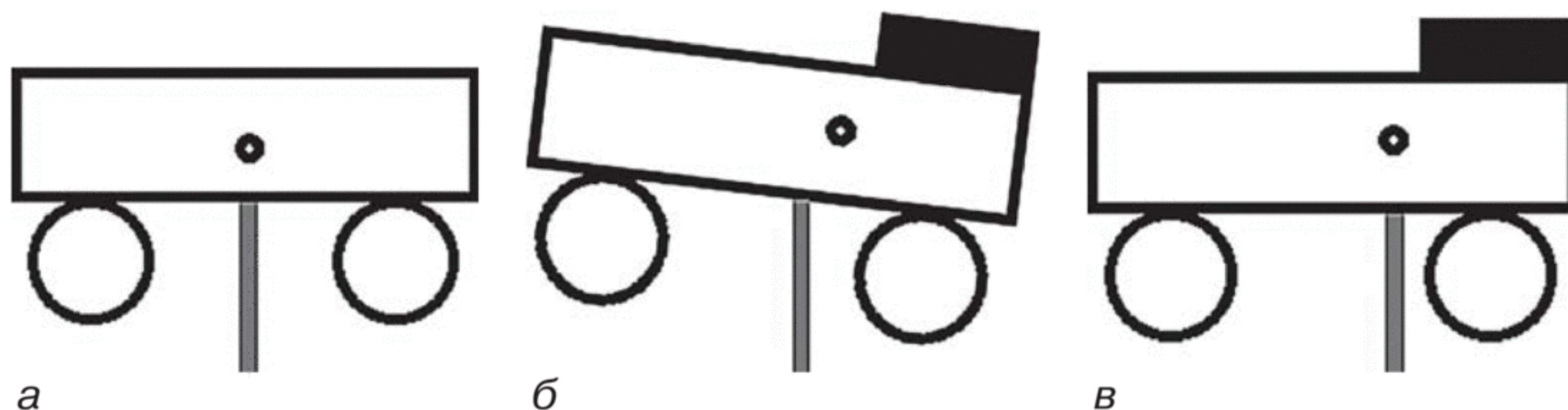
участники включают тележки. Если флажок пересекает крайний перекрёсток, поединок завершается (рис. 2.22).

После первых поединков сразу становятся заметны недостатки тележек: они либо переворачиваются, либо останавливаются колёса, либо колёса проскальзывают. Объяснение, что делать в таких ситуациях, представлено в таблице 2.2. Даже если ваша тележка выступила удачно, всё равно воспользуйтесь советами.

В таблице 2.2 вы встретили новый термин, которому надо дать объяснение. **Центр масс** — это точка, кото-



**Рис. 2.22.** Перетягивание каната



**Рис. 2.23.** *а—в* — поиск центра масс симметричной тележки с утяжелением с помощью линейки

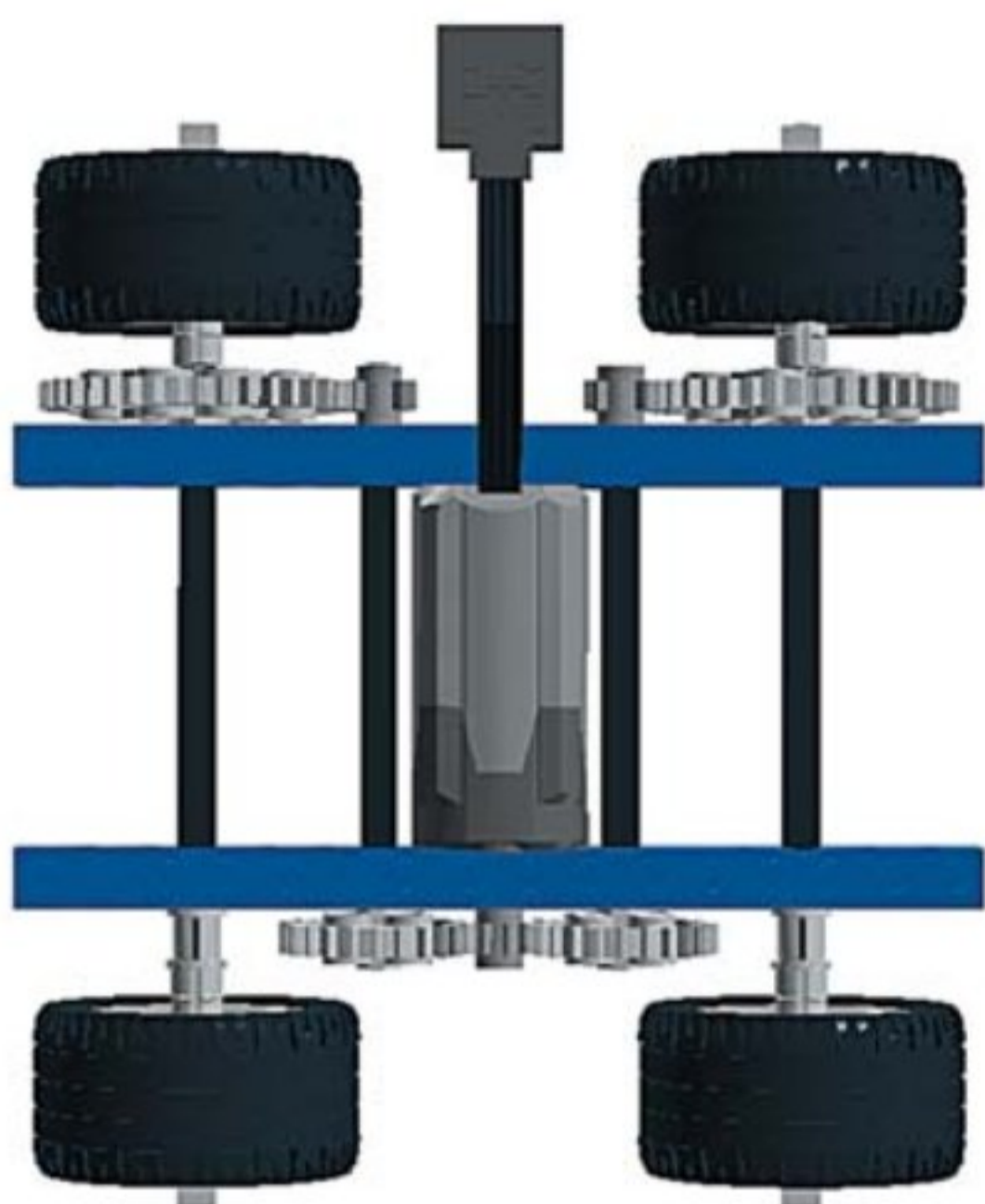
рая определяется формой тела. А при движении тела можно представить, что в ней сосредоточена вся его масса. Поскольку наши тележки симметричны относительно продольной оси, центр масс можно примерно найти, установив тележку на поперечную ось, например, линейку. В искомом положении тележка будет почти уравновешена, как весы (рис. 2.23). Ваша задача — сместить центр масс достаточно далеко от точки крепления каната и обеспечить максимальное сцепление задних колёс с поверхностью (рис. 2.24).

Во время проведения соревнований будут выявлены основные недостатки и вы сможете усовершенствовать конструкцию тележки.

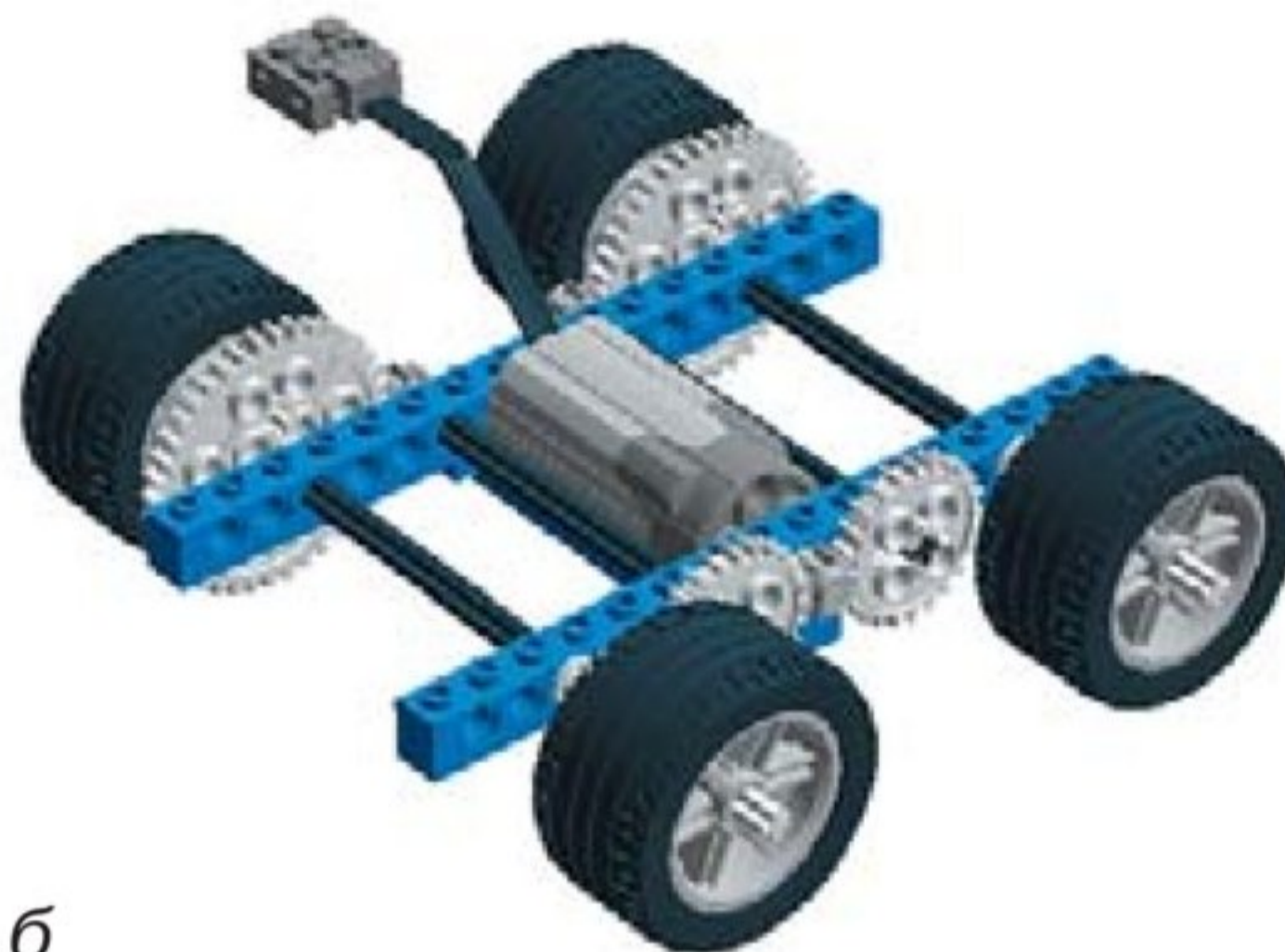
## Механическое сумо

Следующая наша игра с роботами будет посвящена бульдозерам (см. рис. 2.20, *а*). Как правило, с помощью своего ковша они двигают по земле огромные массы различных материалов. Используя тот же принцип, что и у бульдозера, проведите соревнования «механическое сумо». Хотя соревноваться будут простые механизмы с электродвигателями, можно условно назвать их роботами.

По правилам классического сумо двух роботов устанавливают в круге диаметром 77 см один строго напротив другого на расстоянии не менее 10 см и по команде начи-



а



б

**Рис. 2.24.** Полноприводная тележка с редуктором: а — вид сверху; б — вид сбоку

нают их движение навстречу. Один робот должен вытолкнуть другого из круга. В момент соприкосновения можно выяснить и крепость конструкции, и эффективность редуктора. Для большего сцепления имеет смысл построить тележку с полным приводом.

**Полный привод** — это конструкция транспортного средства, позволяющая передавать вращение от двигателя на все колёса. На рисунке 2.25 представлена конструкция основы полноприводной тележки с редуктором. В центре расположен электродвигатель.

Обратите внимание, что в этой конструкции нет паразитных шестерёнок, хотя с одной стороны расположены три зубчатых колеса подряд. Дело в том, что центральная 8-зубая шестерёнка является ведущей и передаёт вращение на две соседних 24-зубых.

Вторая ступень состоит из двух пар 8-зубой и 40-зубой шестерёнок. Таким образом, передаточное отношение для каждой пары колёс будет равно 15:1:

$$i = \frac{24}{8} \cdot \frac{40}{8} = \frac{3}{1} \cdot \frac{5}{1} = \frac{15}{1}.$$







**Рис. 2.25.** Самоходная силовая тележка в сборе

Обратите внимание на то, что передние и задние колёса при одинаковом размере должны вращаться с одинаковой скоростью, поэтому механические передачи от двигателя ко всем колёсам должны иметь одинаковое передаточное отношение. Если колёса за один промежуток времени будут проходить различный путь, тележка будет мешать сама себе, как сказочный Тянитолкай.

Полученную основу тележки можно назвать самоходным шасси.



**Шасси** — это совокупность механизмов и конструкций, обеспечивающих передачу вращения от двигателя к колёсам. Как правило, в конструкцию шасси входит рама или кузов.

Для того чтобы начать движение, требуется закрепить аккумуляторный блок. Это самая тяжёлая часть тележки. Желательно не поднимать его слишком высоко, чтобы избежать опрокидывания.

Тележка готова к движению, но не к поединку. Открытые колёса очень уязвимы к внешним воздействиям. Чего здесь не хватает? Верно, ковша. Или хотя бы бампера. Постройте его самостоятельно. Однако не слишком увлекайтесь: чтобы уравнивать соперников, в правилах «механического сумо» предлагается ограничить стартовые размеры роботов до  $15 \times 15$  см, а их массу до 750 г (рис. 2.26).

Можно заметить, что в некоторых случаях слишком большое передаточное отношение настолько замедляет тележку, что более быстрый соперник сдвигает её.

Подумайте, как можно понизить передаточное отношение тележки до 9:1, не перемещая осей, на которых



**Рис. 2.26.** Поединок в механическом сумо

закреплены колёса. Например, используйте паразитные шестерёнки. Что тогда изменится вместе с передаточным отношением?

**Запомните** ♦ Центр масс ♦ Полный привод ♦ Шасси



# 3

## Основы управления роботом

Вы уже знаете, что автомат работает по программе, заложенной в него человеком. Что же собой представляет эта программа? И как заложить её в автомат? Ответы на эти вопросы пришли с появлением компьютеров. Теперь главным управляющим устройством почти любого автомата и робота стал контроллер, который может и запоминать, и исполнять программу, давая команды двигателям и другим подключённым к нему устройствам.

## § 3.1. Контроллеры

### Какие бывают контроллеры

**Контроллер** — это электронное устройство управления. Контроллер похож на компьютер, но его главная особенность в том, что исполнять программы он должен не на экране компьютера, а в реальном мире (рис. 3.1).

Обычно контроллеры помещают в защитный корпус, но бывают и открытые контроллеры, которыми пользуются разработчики устройств. У контроллера всегда есть разъёмы, с помощью которых его подключают к внешним устройствам и источникам питания (рис. 3.2).

Основой каждого контроллера является **микроконтроллер** (рис. 3.3). Это сложная микросхема, размещённая на кристалле кремния. Микроконтроллер подобен

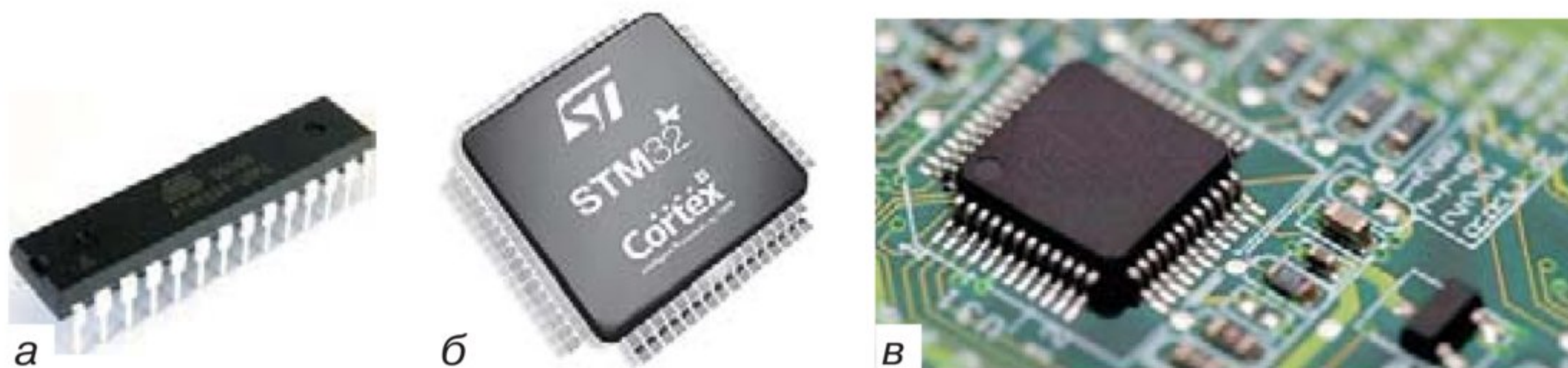


Рис. 3.1. Учебные контроллеры: а — «ТРИК»; б — NXT; в — EV3



Рис. 3.2. Примеры контроллеров: а — Arduino Uno; б — Raspberry Pi; в — MyRio





**Рис. 3.3.** а—в — микроконтроллеры

микропроцессору компьютера, но обладает меньшей мощностью и включает в себя дополнительные устройства: память, устройства ввода-вывода, таймеры и другие. Микроконтроллер устанавливается на специальной плате, с помощью разъемов которой соединяется с внешними устройствами.

## Программное обеспечение

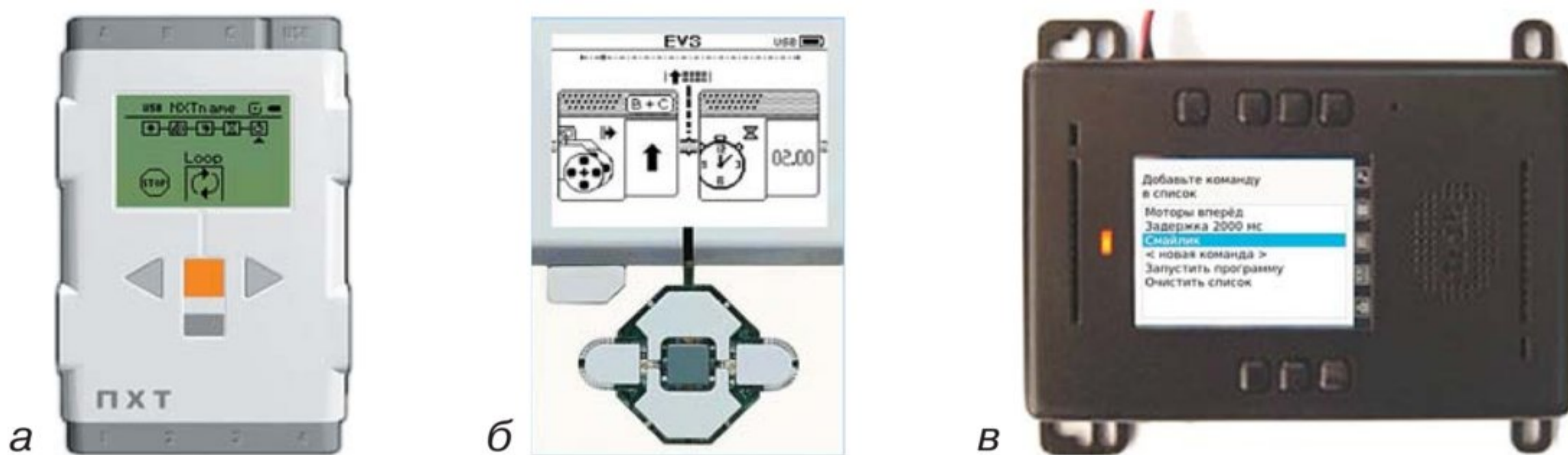
На уроках информатики вы знакомитесь с понятием *алгоритма* как последовательностью действий, приводящих от исходного состояния к конечному результату. Например, можно описать алгоритм деления в столбик.

Основная цель разработчика робота — управление им. Тогда результат выглядит не как число, а как некоторое ожидаемое поведение робота.

Как вы уже знаете, алгоритмы создаются для исполнителей и называются *программами*, если они написаны на специальном языке программирования. Исполнителем ваших алгоритмов будет робот. Для создания программ управления роботом вам придется воспользоваться одним из доступных языков, который будет понимать контроллер.

Чтобы запрограммировать контроллер, требуется специальное **программное обеспечение**, установленное на компьютер и позволяющее составить, отредактировать и загрузить в контроллер определённую программу. После установки программа может быть запущена из памяти контроллера.

Существуют два основных подхода к составлению программ для роботов: графический и текстовый. При графиче-



**Рис. 3.4.** Встроенные среды: а — NXT Program; б — Brick Program; в — «ТРИК»

ском подходе команды задают пиктограммами (картинками) и связывают линиями «проводов», а в текстовом — словами. И взрослые инженеры, и школьники используют оба подхода для решения различных задач.

У некоторых контроллеров (NXT, EV3, «ТРИК») есть встроенная графическая среда программирования, которая позволяет составить простую программу без помощи компьютера. Вы можете начать программировать робота, используя её возможности. Однако сложные и интересные программы можно составить только в специальной программной среде на компьютере (рис. 3.4).

В России создана универсальная среда программирования роботов *TRIK Studio*, которая позволяет создавать как графические, так и текстовые программы. С помощью *TRIK Studio* можно программировать контроллеры «ТРИК», NXT, EV3, а также виртуальных роботов, которые выполняют задания на экране компьютера в режиме 2D-модели, то есть на плоском поле. Эта среда распространяется свободно и работает на различных платформах: Windows, Mac OS, Linux. Вы сможете скачать её из Интернета и установить на свой домашний компьютер.

## Исполнительные устройства

Основные команды контроллер направляет на **исполнительные устройства**, чаще всего электродвигатели (моторы), которые производят некоторые физические действия





**Рис. 3.5.** Прямое управление моторами в разомкнутой системе

в окружающей среде. Если контроллер не получает информации о том, выполнена ли его команда, можно говорить о *разомкнутой системе управления* (рис. 3.5).

Такая система проста в построении, но имеет серьёзные ограничения. Например, вы запрограммировали робот-тележку на движение вперёд и запустили его на столе. Но он ничего «не знает» о длине стола, поэтому свалится. Чтобы научиться избегать таких ситуаций, рассмотрим более подробно команды действия и ожидания.

### Команды действия и ожидания

Любое действие в реальном мире характеризуют длительностью. Можно включить моторы, подождать некоторое время, а потом выключить. За это время тележка проедет какое-то расстояние. Такую последовательность действий надо «объяснить» контроллеру, чтобы он смог повторить её с моторами уже без участия человека. Для этого придётся применить два типа команд: *действия* и *ожидания* (табл. 3.1). На них построена любая программа управления роботами.

Например, алгоритм для управления перемещением двухмоторной тележки (рис. 3.6) будет выглядеть так.

Моторы вперёд  
 Ждать 2 секунды  
 Моторы стоп

Если через две секунды не выключить моторы, они продолжат работать и тележка будет ехать, пока не врежется



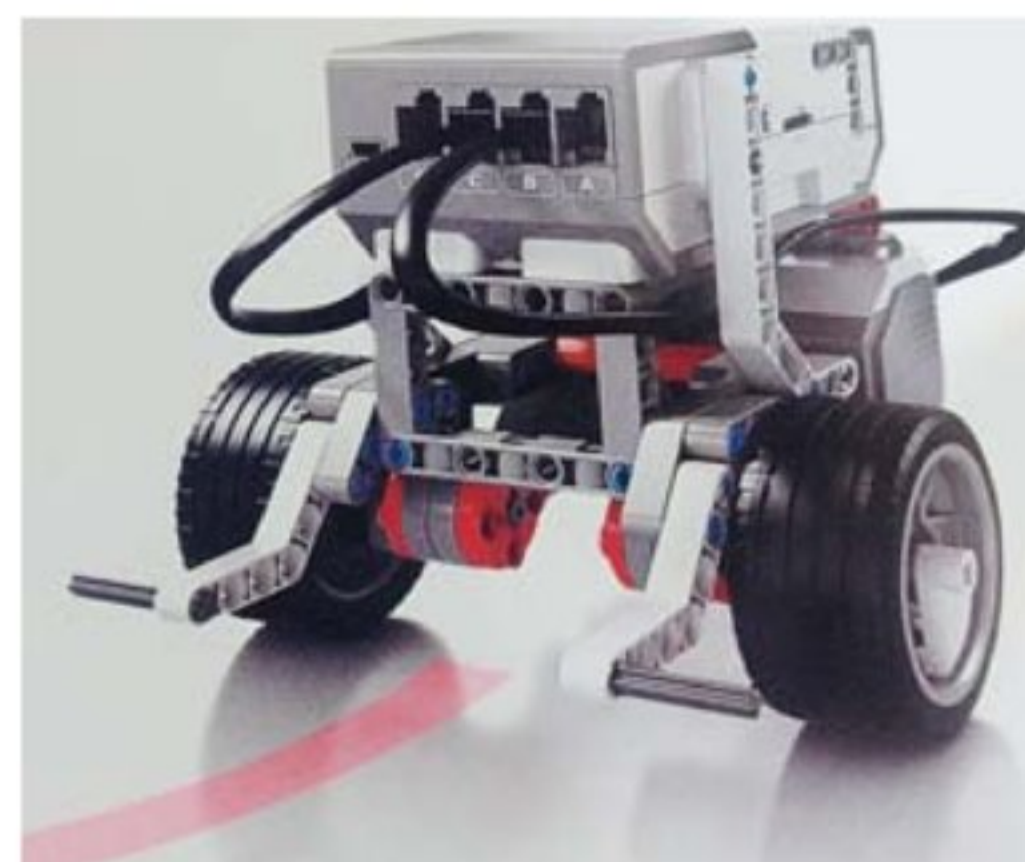
во что-нибудь. В некоторых контроллерах стоит защита: при завершении программы с моторов снимается напряжение и они плавно останавливаются.

Соберите двухмоторную тележку по инструкции, прилагающейся к конструктору, включите контроллер и составьте простейшую программу движения.

Заставьте тележку ехать вперёд две секунды и назад две секунды. Если всё сделано правильно, тележка вернётся на стартовую позицию и остановится. Полученный алгоритм можно сделать повторяющимся. Команды, расположенные между фигурными скобками, будут повторяться одна за другой, пока вы не выключите питание.

повторять бесконечно

```
{
  Моторы вперёд
  Ждать 2 секунды
  Моторы назад
  Ждать 2 секунды
}
```



**Рис. 3.6.** Перемещение двухмоторной тележки

**Таблица 3.1.** Характеристика команд действия и ожидания

Команды действия	Команды ожидания
Выполняются мгновенно	Не выполняют никаких физических действий
Сразу передают управление следующей команде	Останавливают следование программы до совершения определённого события
Их последствия ограничены только аналогичными командами и концом программы	Передают управление следующему блоку после наступления события





Следующая задача — поворот. Как правило, во встроенной среде программирования есть команды, позволяющие выполнять плавные повороты при движении вперёд или назад. Если четыре раза повторить набор действий с движением вперёд и поворотом на прямой угол, робот объедет квадрат и вернётся в исходную точку.

```
повторять 4 раза
{
  Моторы вперёд
  Ждать 2 секунды
  Поворот направо
  Ждать 1 секунду
}
Моторы стоп
```

Вполне возможно, что роботу не удастся сделать поворот на прямой угол, и он не вернётся строго в исходную точку. При программировании с помощью компьютера можно будет откорректировать значение угла поворота.

## Команды низкого и высокого уровня



**Низкий уровень команды** совсем не принижает её качество, а означает, что команда обращена непосредственно к оборудованию робота, например, включает моторы. Однако для их выключения потребуется другая команда.



**Высокий уровень** означает, что внутри команды содержится сложный алгоритм, заложенный туда инженерами-создателями среды программирования.

В этом алгоритме может происходить и включение моторов, и их выключение, и контроль количества оборотов мотора, и контроль синхронности и равномерности вращения.

Команда высокого уровня более сложная, чем команда низкого, но и менее гибкая.



Рассмотрим пример. Установите на робота датчик препятствия так, чтобы он был направлен вперёд по курсу его движения на достаточной высоте. Задача робота — путешествие по комнате. Двигаясь вперёд, он должен реагировать на препятствия, выполняя отъезд назад с разворотом.

```
повторять бесконечно
{
  Моторы вперёд
  Ждать препятствия
  Поворот назад направо
  Ждать 1 секунду
}
```

Программа составлена с помощью команд низкого уровня. Какие из них можно заменить на команды высокого уровня, а какие нельзя? Очевидно, что двигаться вперёд робот должен с помощью команды низкого уровня, иначе, проехав немного, он остановится и будет ждать появления препятствия. А вот выполнить поворот можно и с помощью команды высокого уровня.

## Датчики и обратная связь

**Датчик** — это чувствительное (сенсорное) устройство, передающее на контроллер информацию об окружающей среде (датчики могут реагировать на препятствие, свет, звук, температуру и другое). Используя датчики, можно усовершенствовать схему робота: контур управления замыкается. Датчики собирают информацию из окружающей среды и передают её на контроллер. На основании полученных данных программа, работающая в контроллере, принимает решение и посылает команды на моторы. Наличие датчиков даёт возможность построить систему с обратной связью. Результат её действия влияет на её работу и зависит от информации, передаваемой датчиками. Такую систему иначе называют *замкнутой системой управления*. Она





**Рис. 3.7.** Управление с обратной связью в замкнутой системе

устойчива к внешним помехам и при тщательной настройке может долго работать без сбоев (рис. 3.7).



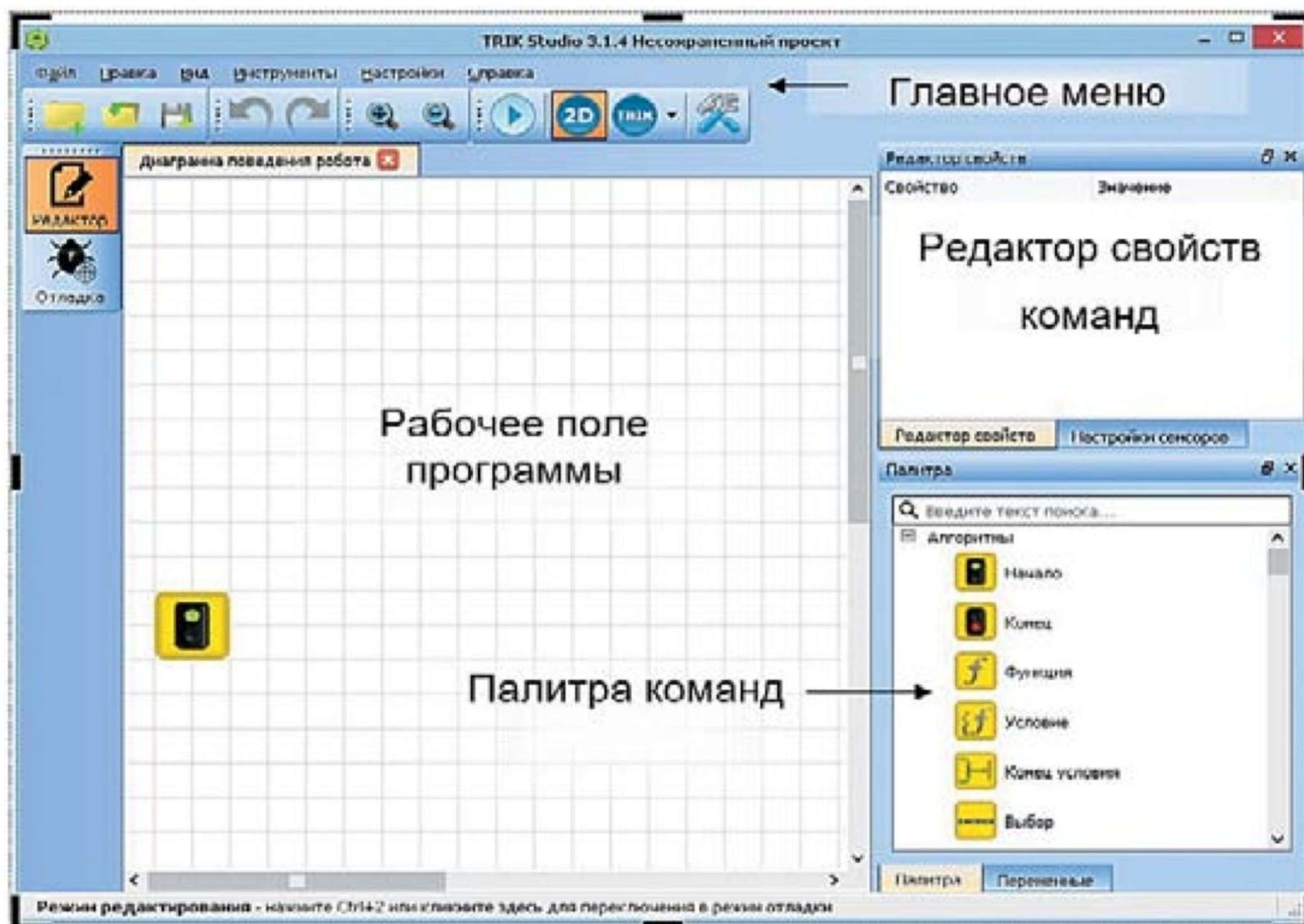
- Запомните** ♦ Контроллер ♦ Микроконтроллер  
 ♦ Программное обеспечение ♦ Исполнительные устройства  
 ♦ Команды действия и ожидания ♦ Команды низкого уровня  
 ♦ Команды высокого уровня ♦ Датчик

## § 3.2. Среда программирования роботов

### Программируем робота

Пришло время научиться программировать робота с помощью компьютера. Среда программирования роботов TRIK Studio позволяет составить программу, проверить её на виртуальной модели, исправить ошибки и загрузить в память реального робота. TRIK Studio поддерживает работу с конструкторами TRIK и LEGO® Mindstorms® NXT и EV3. Запустите программу TRIK Studio (рис. 3.8). Центральное поле — это рабочее поле программы. На нём составляют графические программы, которые в TRIK Studio называют *диаграммами*. В Палитре команд можно найти все доступные команды в виде пиктограмм. Первая пиктограмма — зелёный светофор — уже стоит на рабочем



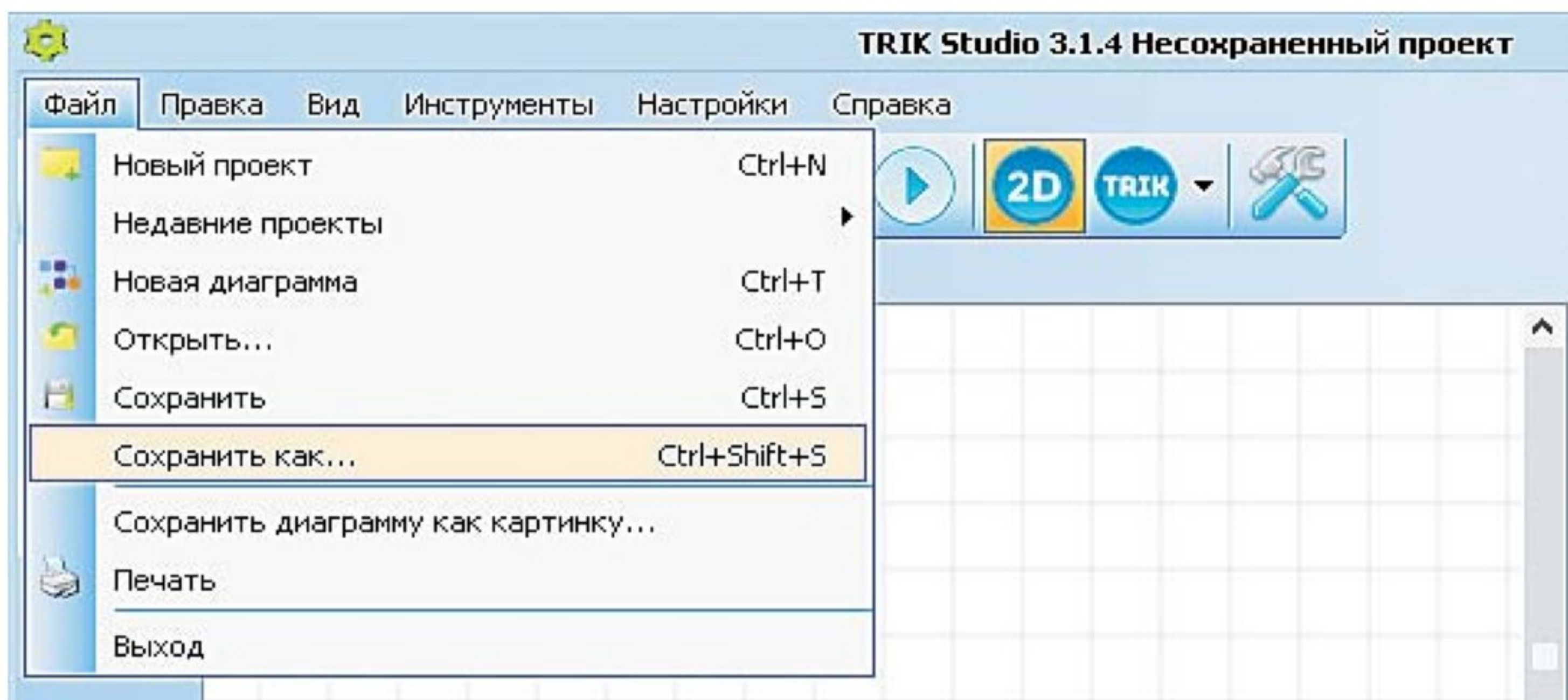


**Рис. 3.8.** Основное окно среды программирования TRIK Studio

уже стоит на рабочем поле, поскольку любая программа начинается с неё. В Редакторе свойств можно настроить команду, указав для неё нужные параметры исполнения.

В верхней части окна расположены сразу два меню: текстовое (Главное меню) и пиктографическое. В текстовом меню можно найти все необходимые пункты для настройки среды программирования и соединения с роботом. В пиктографическом меню располагаются пункты, которые часто используются.

В заголовке окна при старте среды написано «Несохранённый проект». Опытный программист сохраняет свою работу в самом начале. Так следует поступать и вам. Через пункт меню «Файл Сохранить как...» создайте папку и сохраните проект с именем «Привет.qrs». Расширение «.qrs» добавится автоматически. После сохранения



**Рис. 3.9.** Сохранение проекта

полное имя проекта появится в заголовке окна программы (рис. 3.9). Кстати, наши программы будут называться проектами, поскольку могут включать в себя несколько различных файлов, относящихся к одной общей задаче.

Итак, первой задачей будет появление изображения смайлика на экране виртуального контроллера. Для этого в графическом меню должна быть выделена кнопка 2D и выбран контроллер «ТРИК» (рис. 3.10).

Пиктограммы соединяют стрелочками с помощью курсора мыши, а в конце ставят красный светофор — завершение программы.

В программе применены две команды: действие и ожидание. Действие — вывод смайлика на экран. Ожидание — задержка 10 000 миллисекунд. Почему такое большое чис-



**Рис. 3.10.** Вывод смайлика на экран с задержкой 10 секунд

ло? Время для робота течёт быстрее, чем для нас с вами. Известно, что время реакции нервной системы человека на сигнал —  $\frac{1}{100}$  секунды. Робот может среагировать за  $\frac{1}{1000}$  секунды и быстрее. Этот интервал времени называют 1 миллисекунда и обозначают 1 мс.

$$1 \text{ мс} = \frac{1}{1000} \text{ с.}$$

Поэтому в программах применяют измерение времени в миллисекундах. Сколько секунд содержится в 10 000 миллисекунд?

Чтобы запустить программу, нажмите кнопку «Выполнить» в пиктографическом меню. Когда смайлик исчезнет, вернитесь в рабочее поле программы кнопкой «Редактор» (рис. 3.11).

В качестве упражнения составьте и запустите следующий алгоритм.

Показать грустный смайлик  
Подождать 3 секунды  
Показать весёлый смайлик  
Подождать 10 секунд

Сохраните программу с названием «Смайлик\_2».



Рис. 3.11. Запуск программы и возвращение в редактор



б

Редактор свойств	
Свойство	Значение
Вычислять	<input type="checkbox"/> Ложь
Текст	Привет, мир!
Обновить картинку	<input checked="" type="checkbox"/> Истина
X	1
Y	1

**Рис. 3.12.** Программа «Привет, мир!» (а) и редактор свойств (б)

Следующая задача с выводом текста на экран. Создайте новый проект и сохраните его с названием «Привет». Теперь вместо смайлика на верхней строке экрана появится надпись «Привет, мир!», которую вы введёте в специальное текстовое поле под пиктограммой или в окно редактора свойств (рис. 3.12). Все великие программисты начинают изучение среды программирования с этой надписи на экране.

Помимо текста вы заметили два свойства: X и Y. Скорее всего, вы догадались, что это координаты начала текста на плоском экране. Понятие координат точки на плоскости изучаются на уроках математики. Попробуйте изменить их так, чтобы надпись переместилась на центр строки, на центр экрана. Добавьте надпись на второй строке, например, «Меня зовут Вася». Результат сохраните в проекте под названием «Привет\_2».

## Робот разговаривает

Если у вас есть контроллер «ТРИК», то следующее действие можно выполнить, подключившись к нему. Соединение с контроллером может быть установлено по беспроводной сети Wi-Fi.

В контроллере «ТРИК» встроен специальный *синтезатор речи* — программа, которая может прочитать набор слов «металлическим голосом», то есть без интонации. Такая речь присуща современным роботам, но в будущем



**Рис. 3.13.** Использование синтезатора речи

с развитием искусственного интеллекта появится возможность синтезировать речь с эмоциональной окраской. Тогда роботы станут не только интересными, но и приятными собеседниками.

Обратите внимание, что команда «Сказать» тоже является командой действия. Она выполняется достаточно долго, но сразу передаёт управление следующему блоку. Поэтому, чтобы программа не закончилась до того, как робот произнесёт слова, необходима задержка. Результат сохраните в проекте под названием «Сказать\_Привет» (рис. 3.13). Попробуйте синтезировать более сложную фразу.

**Запомните** ♦ Диаграммы ♦ Синтезатор речи



## § 3.3. Управление мобильным роботом

Первый опыт программирования контроллера с помощью компьютера вы получили на предыдущем занятии. Следующий шаг — научиться управлять двухмоторной тележкой.

Двухмоторные мобильные роботы — самый распространённый тип роботов. Как правило, в них для построения шасси используют трёхколёсную схему, в которой два колеса являются ведущими, а третье — подвижной опорой, то есть свободно вращающимся роликом (как у компьютерного кресла), шариком в специальном держателе (рис. 3.14). Центр масс робота обычно смещают





**Рис. 3.14.** Трёхколёсная схема (а) и виды опорных колёс (б)

в сторону оси ведущих колёс для улучшения их сцепления с поверхностью и уменьшения проскальзывания. Однако размещать центр масс на ведущей оси нельзя, поскольку при резком изменении скорости тележка начнёт переворачиваться. Опорное колесо может быть как спереди, так и сзади. В двумерной модели тележки «ТРИК» опорное колесо расположено спереди.

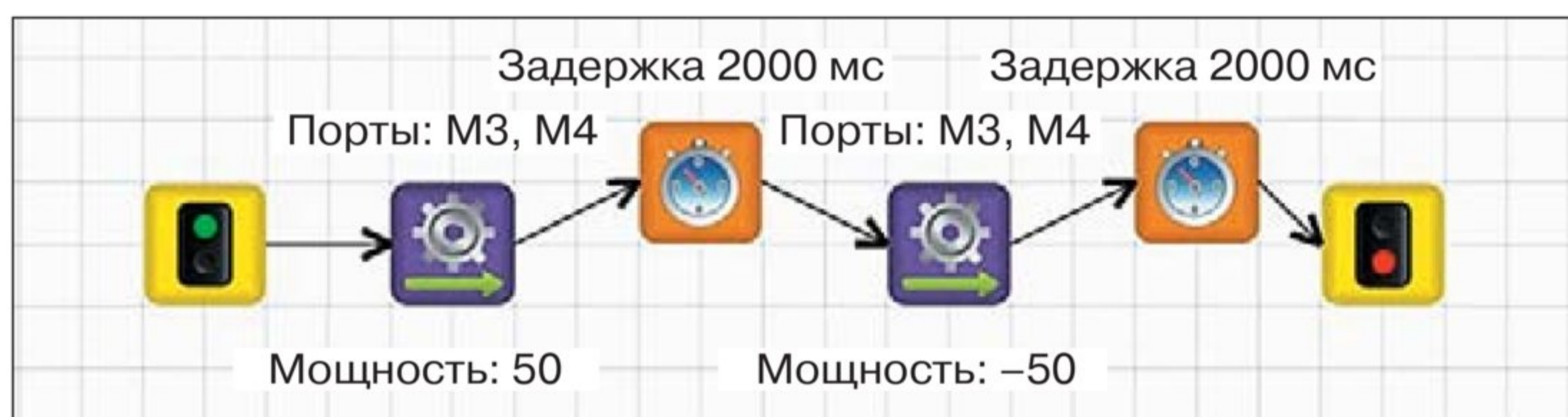
Познакомиться с основными принципами управления можно с помощью виртуальной модели на экране компьютера.

Используя навыки, полученные при программировании контроллера, научите тележку двигаться по экрану вперёд-назад. Все порты контроллера имеют свои обозначения. Каждый мотор, подключённый к контроллеру, получает имя соответствующего порта. Например, в контроллерах NXT и EV3 ведущие моторы тележки обычно подключают на порты В (левый) и С (правый), а в контроллере «ТРИК» — на порты М3 и М4. Соответствующие обозначения должны быть использованы и в программе (рис. 3.15).

Есть ещё один важный параметр, который следует указать в программе: мощность, подаваемая на мотор, в процентах от максимально возможной. Будем считать мощностью количество электроэнергии, подаваемой контроллером на мотор в единицу времени (подробнее вы узнаете об этом на уроках физики). На мотор подаётся электроэнергия, и в зависимости от её количества меняется его скорость. Например, значение 100% означает максимально возмож-



**Рис. 3.15.** Подключение моторов в двухмоторной тележке: а — на «ТРИК»; б — на EV3



**Рис. 3.16.** Программа для движения вперёд-назад

ную скорость при данном заряде батарей, а 50% означает, что моторы вращаются в 2 раза медленнее. Такой способ задания скорости связан с тем, что, как вы знаете, реальная скорость мотора обычно зависит от заряда батареи.

Если нужно задать движение назад, то скорость должна быть отрицательной, со знаком «минус»; например, «-100%» означает «ехать назад с максимальной скоростью». В этом случае можно говорить о том, что контроллер сам переключает «плюс» и «минус» батареи. Очевидно, значение скорости «0» указывает на остановку моторов. Таким образом, скорости моторов задают в диапазоне от -100% до 100% (рис. 3.16 и 3.17).



**Рис. 3.17.** Движение вперёд-назад в режиме двумерной модели

Сохраните программу под названием «вперёд-назад», запустите на исполнение в режиме двумерной модели и посмотрите результат. Тележка с контроллером проедет вперёд в течение 2 с, а затем вернётся в исходную позицию на красный крестик.

Теперь выполните поворот на месте на  $90^\circ$ . Для этого на левый и правый моторы подайте скорость с разными знаками: 50 и  $-50$ . Длительность поворота придётся уменьшить примерно до 1 с. Подберите такую длительность, чтобы робот поворачивался ровно на прямой угол. Обратите внимание, что тележка при повороте немного съезжает с начальной позиции. Это связано с тем, что ведущие колёса находятся в задней части корпуса (рис. 3.18, а).

В режиме отладки часто требуется вернуть виртуального робота в исходное положение. Для этого используйте кнопку «Домой» (рис. 3.18, б).

На следующем этапе научите робота ездить по квадрату. Вы должны указать точное значение времени для поворота на  $90^\circ$ . Набор этих действий известен вам из предыдущего параграфа. Для многократного повторения необходимо произвести «зацикливание», то есть перенаправить поток программы с последнего блока на первый, убрав красный светофор (рис. 3.19).

Какая из команд в программе лишняя? Вы научились делать поворот, включая моторы в разные стороны. Но если моторы уже вращаются вперёд, повторно включать один из них не требуется. В следующем примере команда на повторное включение мотора М3 отсутствует.



**Рис. 3.18.** Поворот на месте (а), кнопки «Домой» и «Запуск» (б)

Чтобы робот остановился после того как проедет по квадрату, следует повторить действия ровно 4 раза. Для этого потребуется специальный блок Цикл, в котором в качестве параметра указано число итераций (повторений). Чтобы обозначить пройденный путь на экране, в среде TRIK Studio реализована возможность рисовать виртуальным маркером. Для этого робот должен опустить маркер заданного цвета, а когда закончит рисовать, — поднять (рис. 3.20).


Теперь наш виртуальный робот готов к выполнению дорожной разметки, например, для места парковки автомобилей. Но использование таймера при выполнении



**Рис. 3.19.** Бесконечное движение по квадрату



**Рис. 3.20.** Робот рисует квадрат и останавливается



поворота всё равно не даёт желаемой точности: квадрат получается не идеальным. Кроме того, реальный робот при изменении заряда батареи будет выполнять повороты совсем не на то число градусов, которое вы запланировали. На следующих занятиях вы узнаете, как научить его выполнять точные повороты и не зависеть от заряда батареи.

## Проверьте себя

1. Как поведёт себя робот, если центр масс окажется на опорном колесе?
2. Составьте программу, по которой робот рисует зону парковки из 8 мест, а потом сам паркуется на одно из свободных мест.

## § 3.4. Знакомство с датчиками

На этом занятии вам предстоит научиться использовать датчики. Как вы помните, *датчик* — это чувствительный элемент, который передаёт на контроллер информацию об окружающей среде. Синоним слова датчик — сенсор. Датчики измеряют физические параметры окружающего мира и самого робота, а затем передают их на контроллер в виде электрических сигналов. Эти сигналы преобразуются в цифровые значения.

Диапазоны показаний датчиков могут различаться. В таблице 3.2 представлены некоторые характеристики датчиков различных конструкторов.

Показания каждого из датчиков можно увидеть на экране контроллера в режиме просмотра. Для этого необходимо, чтобы датчик был подключён на допустимый порт.

Составьте таблицу показаний датчика освещённости на поверхностях различных цветов: белого, чёрного, синего и красного. При этом направляйте датчик на поверхность под прямым углом с расстояния около 1 см (толщина пальца).

**Таблица 3.2.** Основные виды датчиков

Название	Значение параметра	Внешний вид
Датчик касания Touch Sensor	0 — отпущен 1 — нажат	
Датчик освещённости Light Sensor	0 ... 100 0 — полная темнота 100 — яркий свет	
Датчик цвета Color Sensor	0 ... 7 (EV3) 0 ... 17 (Hitechnic) 0 ... 16777216 (режим RGB)	
Ультразвуковой датчик расстояния Ultrasonic Sensor, SONAR	3...250 см 255 см — нет сигнала	
Инфракрасный датчик расстояния	0...100 (не всегда в сантиметрах)	
Гироскопический датчик скорости вращения Gyro Sensor	0...440°/с — скорость вращения, в режиме измерения угла: -32768 ... 32767°	
Датчик оборотов Encoder	-32768...32767° или -2 млрд ... 2 млрд градусов	
Видеомодуль	— Кадр 640 × 480 пикселей — смещение линии — координаты объекта — размер объекта и другое	

## Датчик расстояния

Принцип работы ультразвукового датчика расстояния основан на отражении звука от предметов подобно тому, как определяет расстояние летучая мышь. Обычно у такого датчика два глазка: один издаёт сигнал, другой принимает отражённый сигнал (эхо) (рис. 3.21). Это происходит очень часто, несколько сотен раз в секунду. Измерив время возврата сигнала  $t$  и зная скорость звука в воздухе  $v = 330$  м/с, расстояние  $S$  до объекта можно вычислить по формуле:  $S = v \cdot \frac{t}{2}$ . Примерно такой расчёт и происходит внутри датчика.

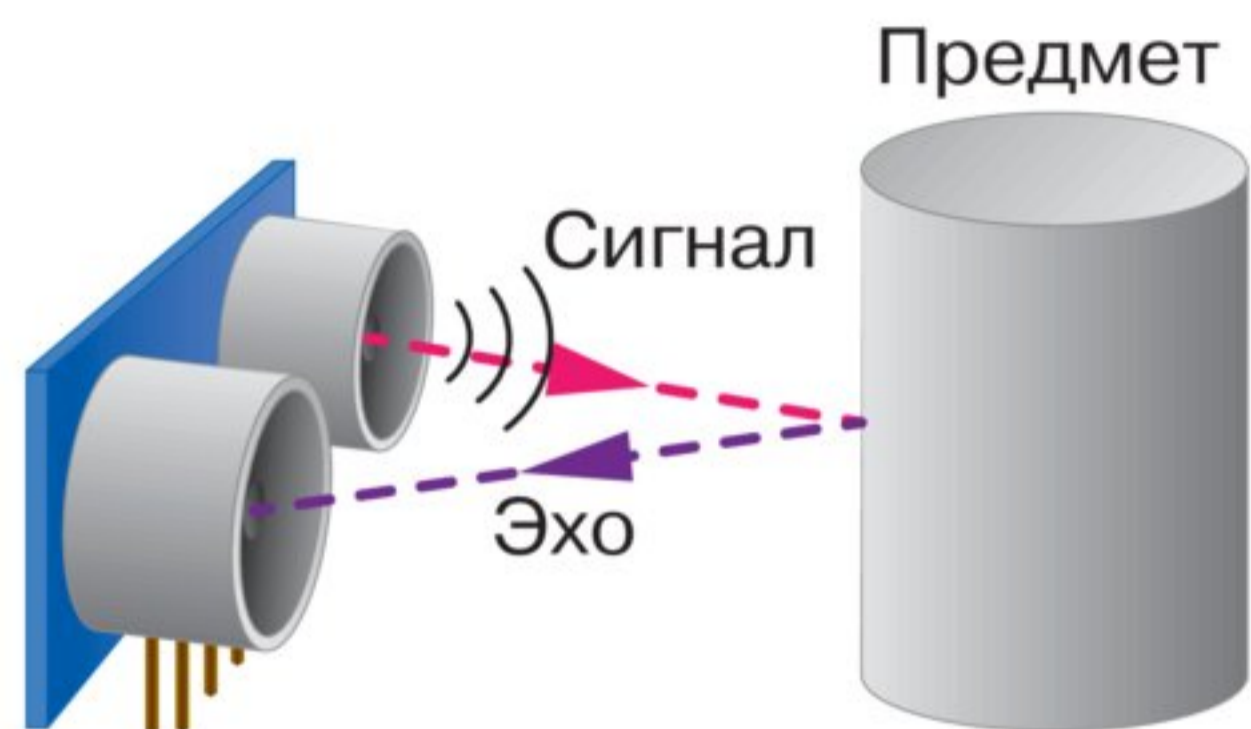


Рис. 3.21. Принцип работы ультразвукового датчика

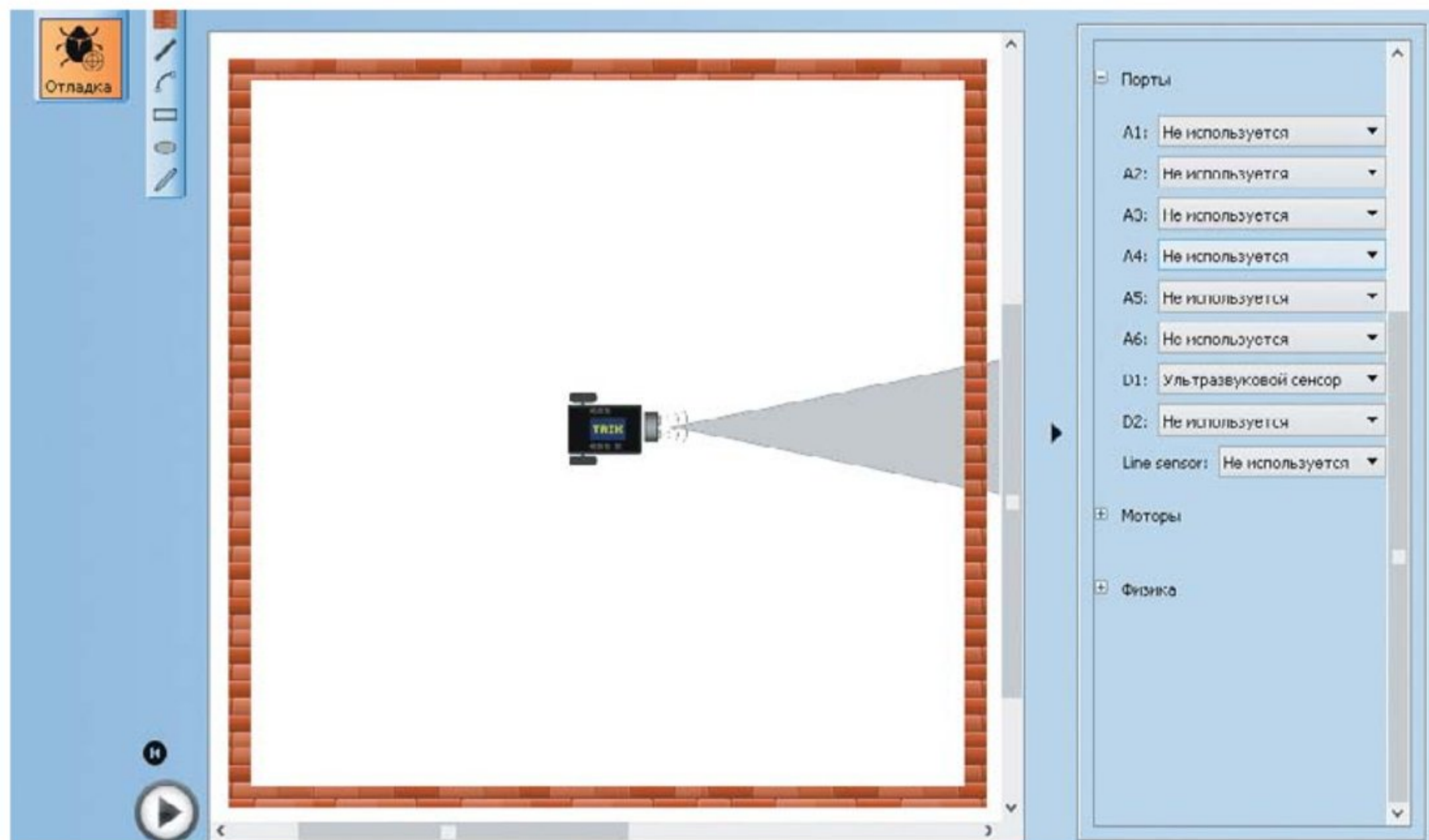


Рис. 3.22. Виртуальная комната



**Рис. 3.23.** Программа для путешествия по комнате

Установите на двухмоторный робот датчик расстояния так, чтобы он был направлен вперёд. С помощью TRIK Studio в настройках робота выберите ультразвуковой датчик на порту D1. Используя инструмент «Стена» постройте виртуальную комнату из четырёх стен, на которые будет реагировать робот (рис. 3.22). Составьте программу движения робота по комнате (рис. 3.23) и запустите её в режиме 2D модели.

Робот будет двигаться вперёд, пока не увидит стену на расстоянии 35 см. Затем отъезжает назад в течение 0,5 с и выполняет поворот на месте в течение 1,2 с. Благодаря этому робот обследует практически всю комнату целиком.

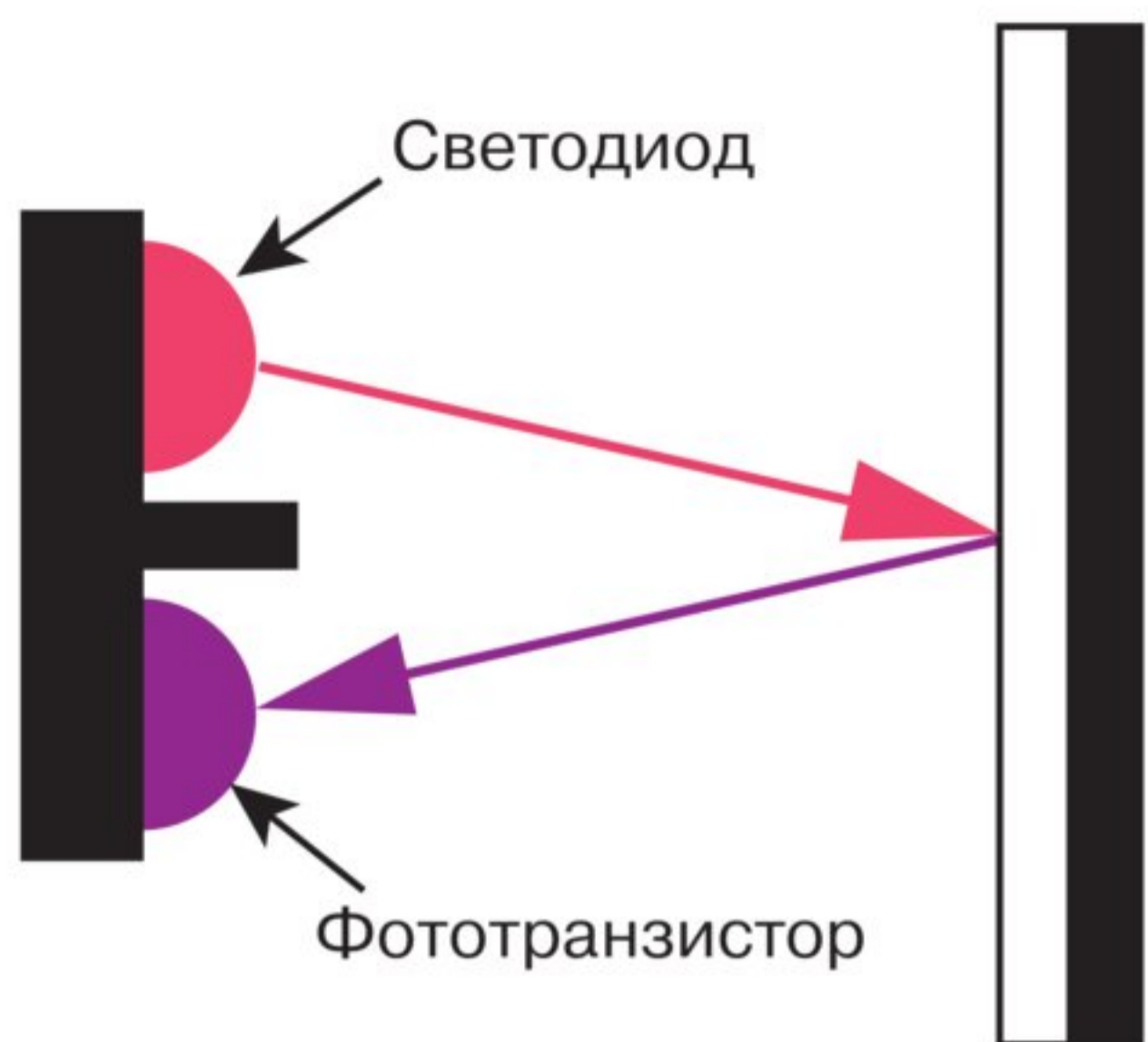
Загрузите программу в память реального робота и проанализируйте его поведение в комнате.

Постройте самостоятельно аналогичный алгоритм с датчиком касания. В каких ситуациях он может дать сбой? Проверьте его на практике.

## Датчик освещённости

Датчик освещённости работает в режиме приёма отражённого сигнала. Только измеряет он не время, а яркость. Специальный светодиод (по сути фонарик) светит на из-





**Рис. 3.24.** Принцип работы датчика освещённости

меряемую поверхность, а отражённый свет попадает на *фототранзистор* — специальный чувствительный элемент, который преобразует свет в электрический сигнал. Что происходит дальше, вы знаете — сигнал поступает в контроллер.

Чтобы датчик хорошо работал, расстояние до поверхности должно быть не меньше 0,5 см и не больше 5 см. У различных датчиков эти характеристики могут различаться (рис. 3.24).

Аналогично предыдущей задаче можно решить задачу движения внутри круга, ограниченного чёрной линией. Для определения границы круга роботу следует заменить датчик расстояния на датчик освещённости. Это необходимо сделать и в программе, и в настройках робота. В параметре «Считанное значение» укажите 45% — примерную границу освещённости между чёрным и белым цветами. В остальном программа остаётся практически такой же.



**Рис. 3.25.** Программа для движения внутри круга

Робот начинает движение вперёд и едет, пока датчик не покажет значение меньше 45% освещённости. Это значит, что датчик начал наезжать на чёрную линию (показания датчика меняются постепенно). Затем робот отъезжает назад, делает поворот, и всё повторяется (рис. 3.25).

Круг нарисуйте самостоятельно с помощью инструмента «Эллипс» и увеличьте ширину линии во всплывающем окне примерно до 15 пикселей (рис. 3.26).

В следующей задаче не нужно переставлять датчик освещённости, но придётся изменить алгоритм: робот поедет по линии. Нарисуйте её в виртуальном мире с помощью инструмента «стилус» и установите ширину около 15 пикселей. Принцип движения вдоль границы чёрной линии и белого поля таков. Робот выполняет плавный поворот вокруг одного из колёс, пока датчик не заедет на чёр-

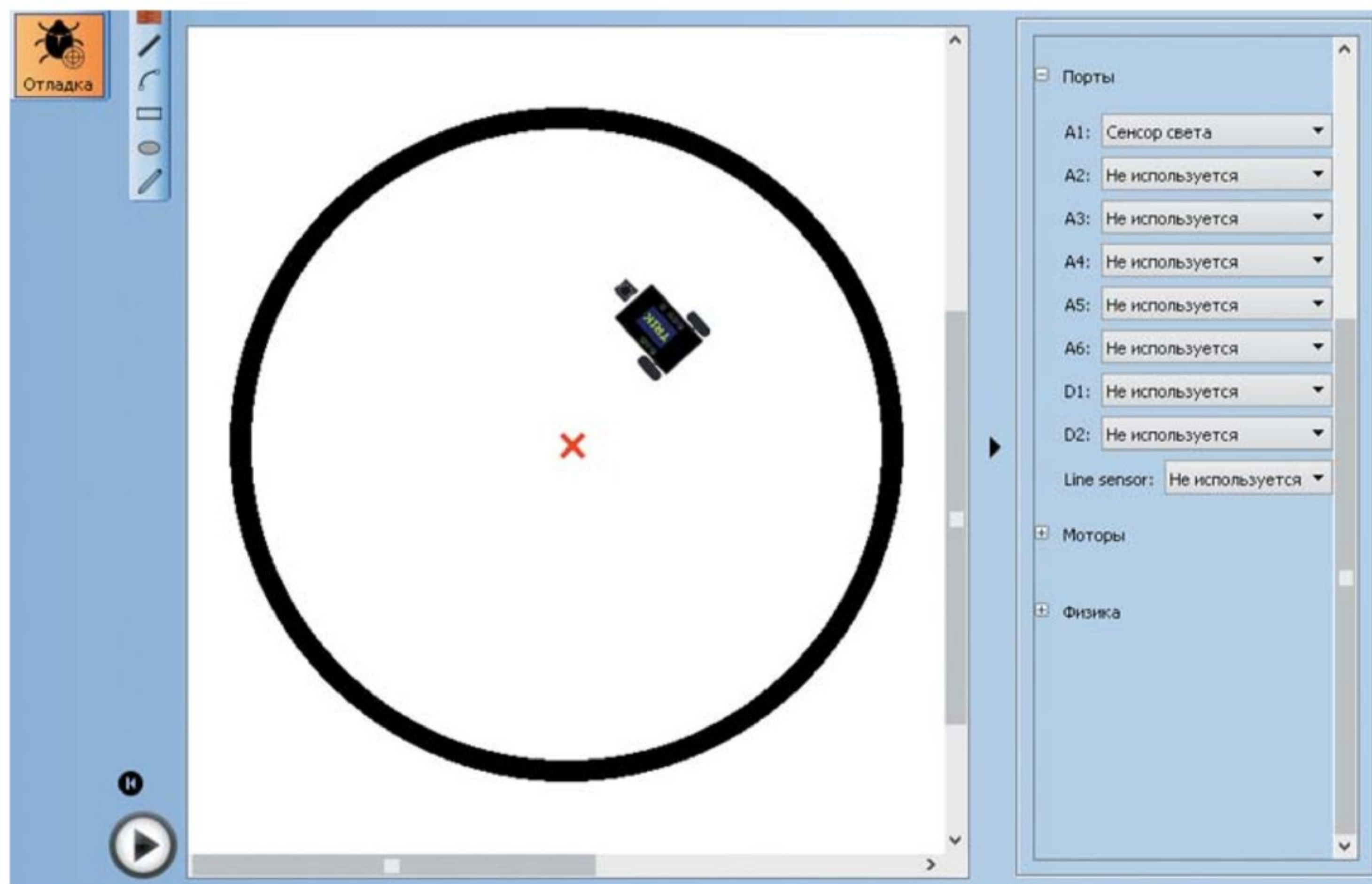
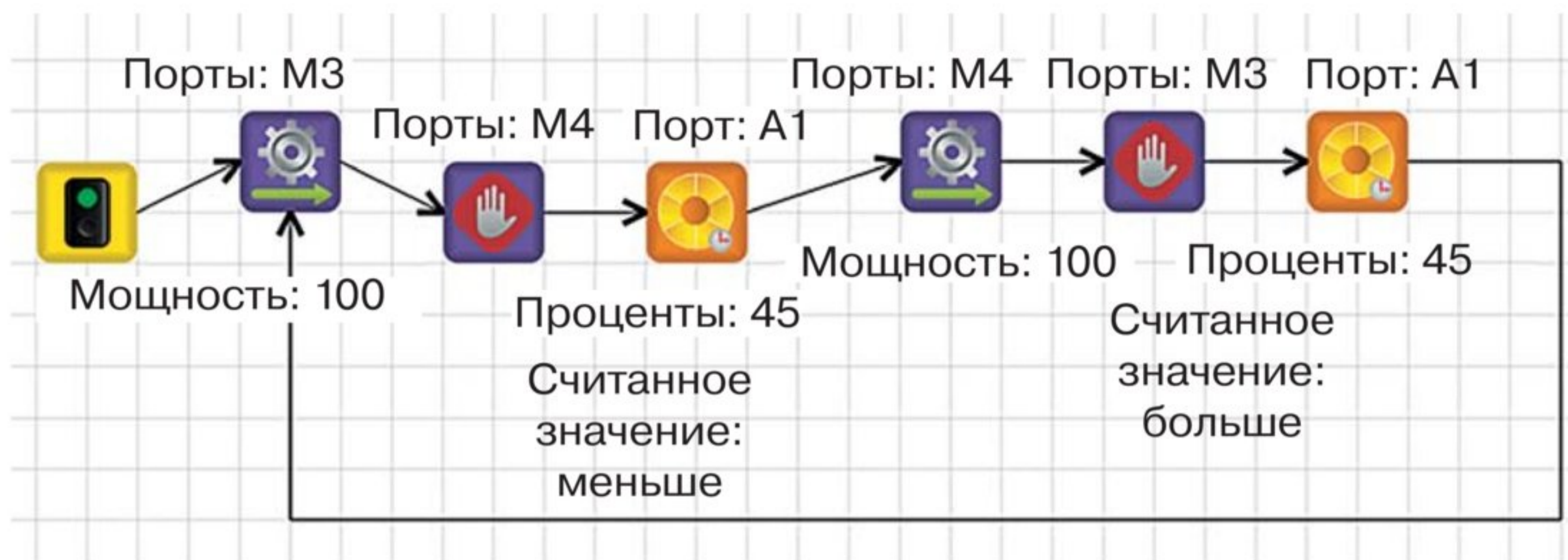
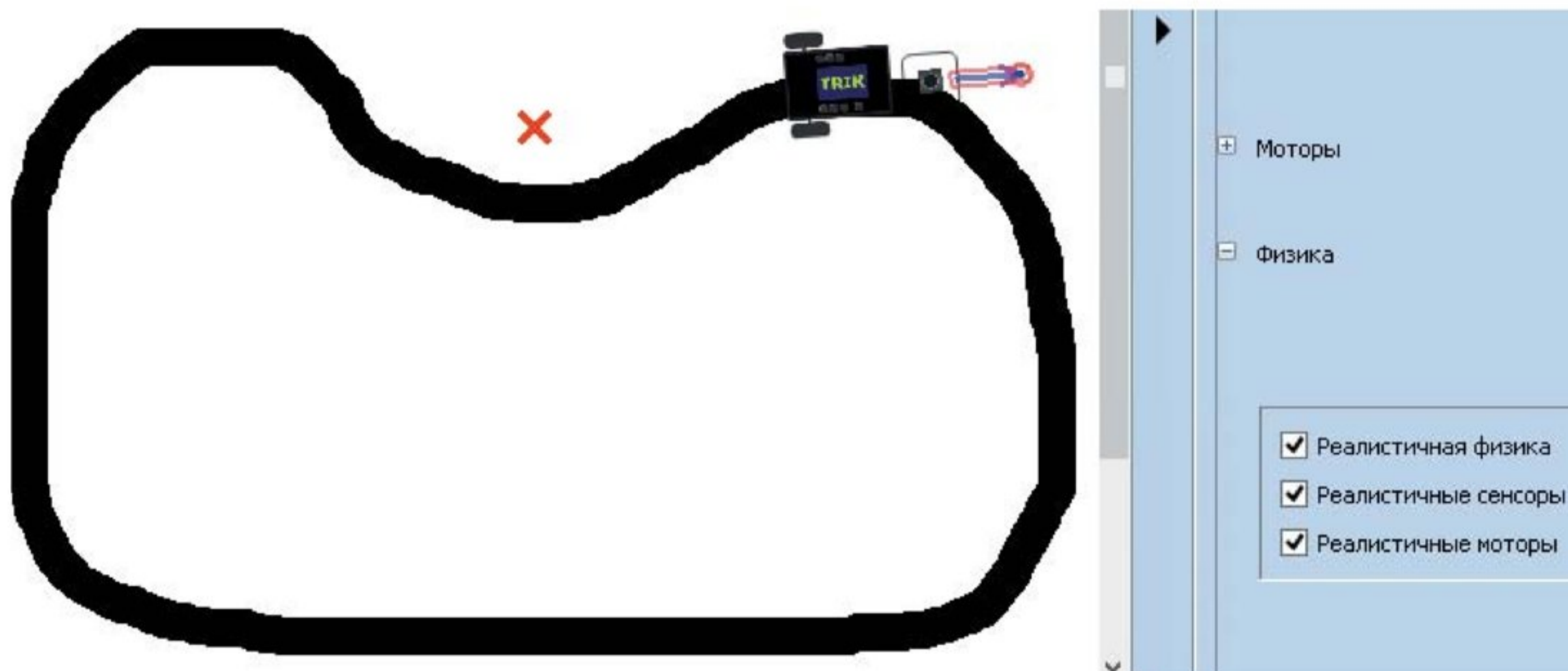


Рис. 3.26. Робот внутри виртуального круга



**Рис. 3.27.** Программа для движения по линии



**Рис. 3.28.** Виртуальный робот едет по линии

ную линию. Затем он выполняет плавный поворот вокруг другого колеса, пока датчик не вернётся на белое поле. Чтобы робот не сбивался, нельзя давать датчику полностью пересекать линию (рис. 3.27 и 3.28).

В реальном мире робот будет двигаться, активно колеблясь на линии. В TRIK Studio есть возможность сделать виртуальную модель похожей на реальный мир. Для этого раскройте вкладку «Физика» и выберите все пункты:

реалистичная физика, реалистичные сенсоры, реалистичные моторы. Робот начал себя вести уже не так идеально. Теперь подумайте, как можно избавиться от колебаний при движении?

Попробуйте увеличить скорость робота. Можно ли научить его ехать быстрее, не останавливая моторы?

**Запомните** ♦ Датчик расстояния ♦ Датчик освещённости



# 4

## Возвратно-поступательное движение

### § 4.1. От мультипликатора до маятника Капицы

Вы уже познакомились с механической передачей, её видами и основными характеристиками. Чтобы пойти дальше, сначала повторим пройденный материал.

**Механическая передача** — это механизм, преобразующий и передающий механическое движение. Чаще всего с его помощью передаётся вращение.


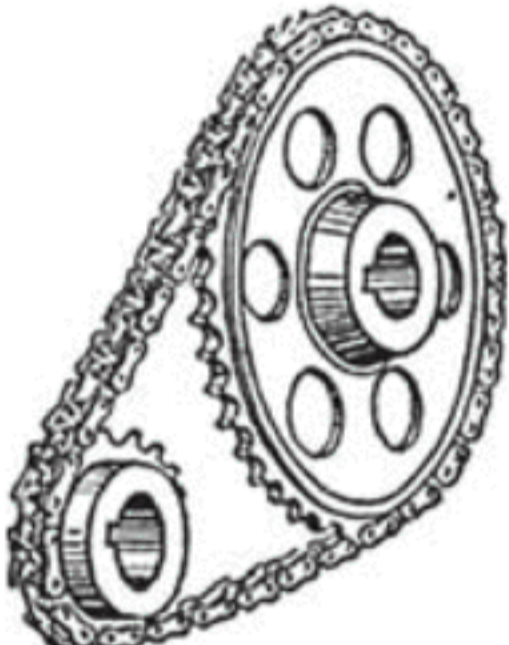
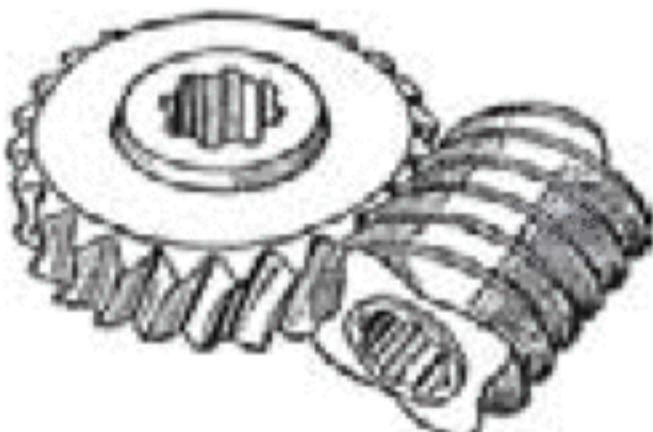
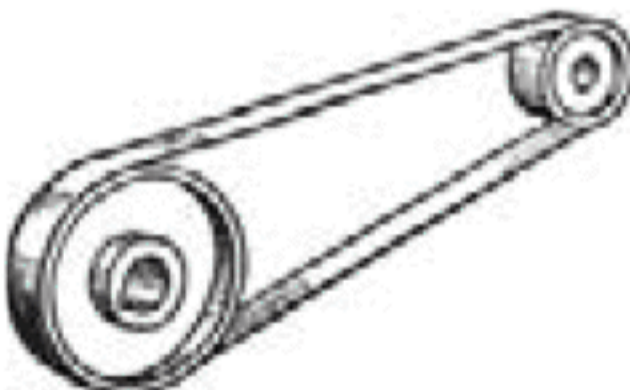
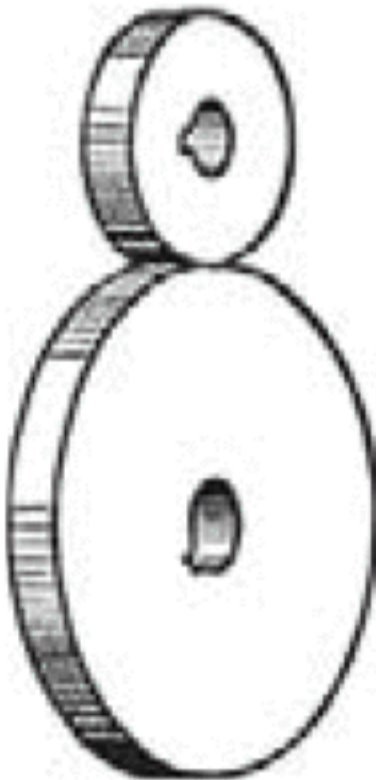
Для расчёта передаточного отношения механической передачи используются математические формулы, показывающие отношение угловых скоростей вращения ведущего и ведомого валов (табл. 4.1). Эта величина обратно пропорциональна отношению их диаметров или количеству зубцов на шестерёнках:

$$i = \frac{\omega_1}{\omega_2} = \frac{D_2}{D_1} = \frac{z_2}{z_1}.$$

Передаточное отношение  $i$  показывает, во сколько раз изменились развиваемая сила и скорость вращения. При  $i > 1$  передача называется понижающей, а механизм, реализующий её, — **редуктором**. При  $i < 1$  передача называется повышающей, а механизм — **мультипликатором**.

Редукторы чаще применяются в окружающих нас механизмах. Как правило, рабочая

**Таблица. 4.1** Расчёт передаточных отношений различных видов механической передачи

Передача	Изображение	Расчёт передаточного отношения	Обозначения
Зубчатая прямая		$i = \frac{z_2}{z_1}$	Количество зубцов на шестернях: $z_1$ — на ведущей, $z_2$ — на ведомой
Цепная		$i = \frac{z_2}{z_1}$	Количество зубцов на звёздочках: $z_1$ — на ведущей, $z_2$ — на ведомой
Червячная		$i = \frac{1}{z_1}$	$z_1$ — количество зубцов на ведомой шестерне; один заход червяка соответствует одному зубу
Ремённая		$i = \frac{D_2}{D_1}$	Диаметр шкива: $D_1$ — ведущего, $D_2$ — ведомого
Фрикционная		$i = \frac{D_2}{D_1}$	Диаметр колеса: $D_1$ — ведущего, $D_2$ — ведомого



скорость электродвигателей значительно превосходит потребности большинства бытовых устройств, поэтому они комплектуются редукторами, которые не только замедляют вращение, но и создают выигрыш в силе.

Используя мультипликатор, можно значительно увеличить скорость вращения ведомой оси. Ограничения в скорости возникнут только по причине трения. Например, используя мультипликатор, можно придать достаточно большую скорость волчку, вращающемуся на полу.

С детства вы помните замечательную игрушку — *волчок*: круглый маховик-колесо, закреплённый на валу, может долго вращаться, не падая за счёт «неведомой» силы (рис. 4.1). И конечно, хочется продлить этот процесс, придав максимальную скорость вращения при старте. Удобнее всего это сделать с помощью мультипликатора.

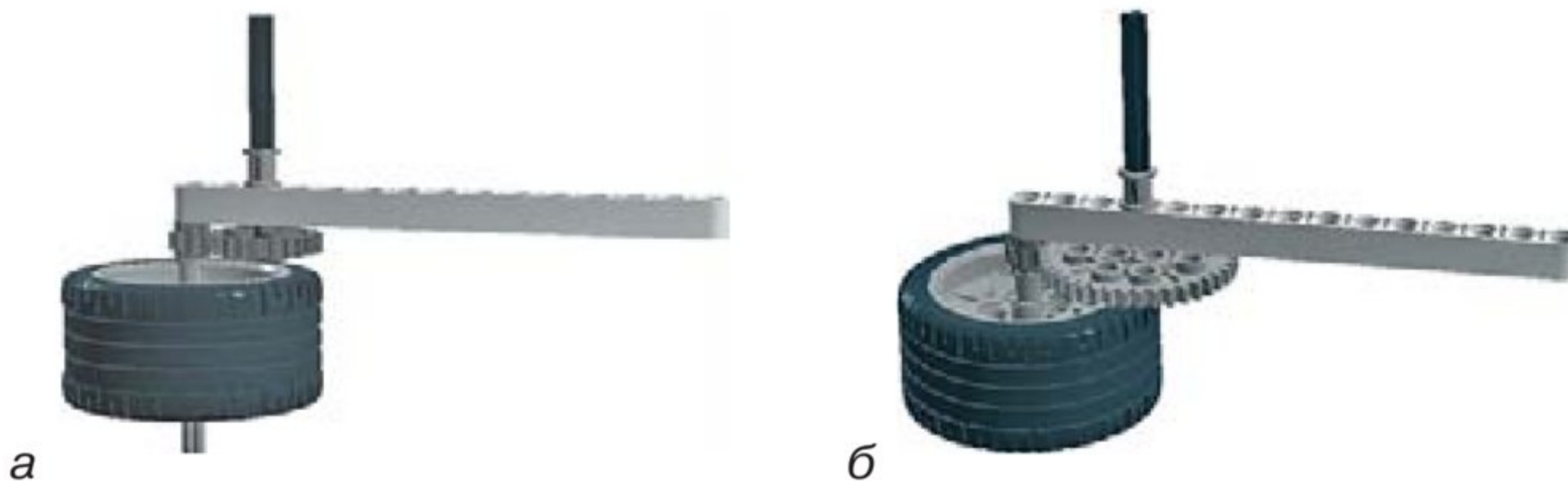
Установив малую шестерню на ось волчка, а большую — на пусковой механизм, получаем мультипликатор,

который разъединяется сразу после приобретения волчком нужной скорости (рис. 4.2). Определяем передаточное отношение, поделив количество зубцов ведомой шестерни на количество зубцов ведущей.

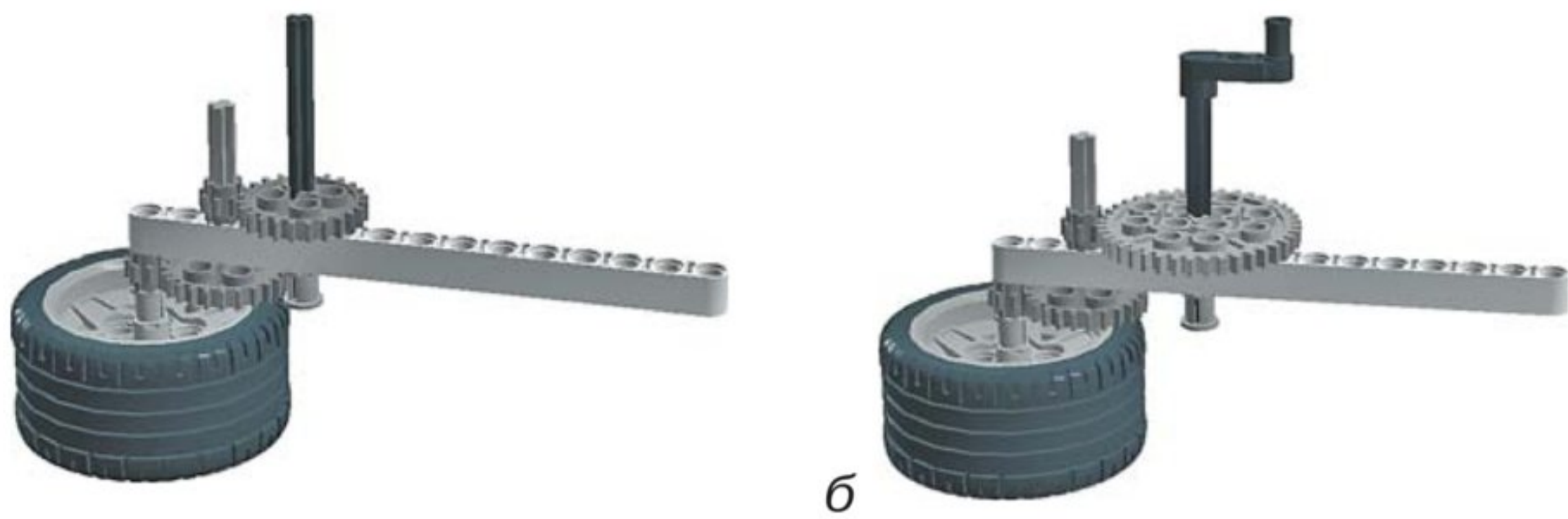
Чтобы ещё больше увеличить скорость вращения, необходимо либо найти очень большую веду-



**Рис. 4.1.** Волчки: *а* — для запуска рукой; *б* — пусковым механизмом



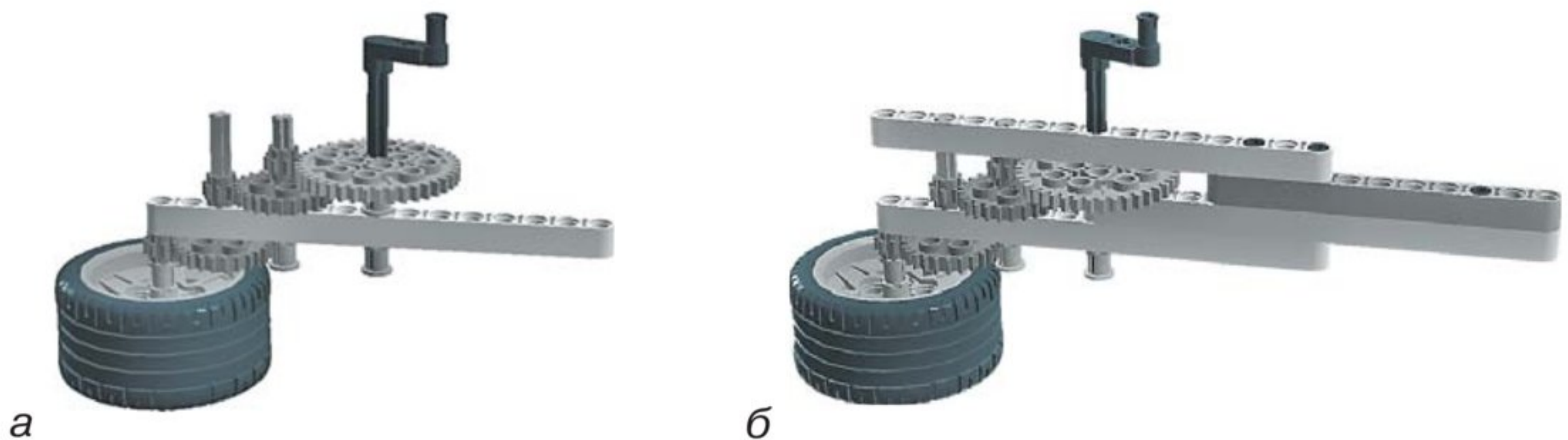
**Рис. 4.2.** Одноступенчатые пусковые механизмы: *а* — с передаточным отношением 1:3; *б* — с передаточным отношением 1:5



**Рис. 4.3.** Двухступенчатые пусковые механизмы с рычагом: *а* — с передаточным отношением 1:9; *б* — с передаточным отношением 1:15

щую шестерёнку, либо создать вторую ступень передачи. При этом для увеличения скорости вращения волчка от запускающего человека потребуется дополнительное усилие на старте механизма. Чтобы облегчить эту задачу, на ведущую ось добавим рычаг, который поможет взять быстрый разгон (рис. 4.3).

Однако расширение пускового механизма до трёх ступеней (рис. 4.4) может привести к проскакиванию зубчиков шестерёнок, что вызывает характерный треск. В таком состоянии механизм оставлять нельзя: в скором времени треск приведёт к быстрому износу зубчатых колёс! В результате шестерёнки останутся совсем без зубцов. Поэтому для постоянного надёжного зацепления оси соприкасающихся шестерёнок должны быть закреплены с двух сторон, особенно при большой нагрузке.



**Рис. 4.4.** Конструкция трёхступенчатого пускового механизма: *а* — неправильная; *б* — правильная





## Проверьте себя

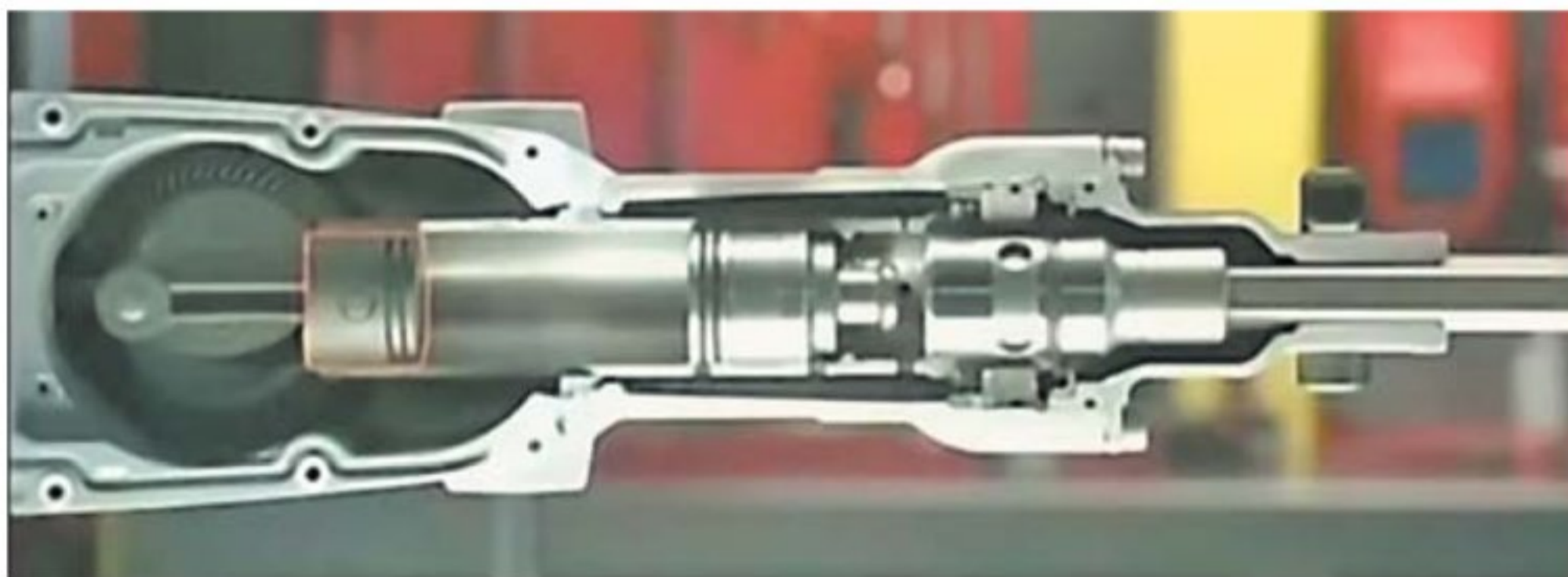
1. Проведите соревнования, чей волчок продержится дольше. Подумайте, как можно усовершенствовать волчок.
2. Как угловые скорости соприкасающихся шестерёнок связаны с количеством зубцов на них?
3. От чего зависит передаточное отношение ремённой и фрикционной передач?
4. Сколько зубцов поворачивает червяк за один заход?
5. В каких случаях в механизмах применяется мультипликатор?
6. Постройте запускающий механизм для волчка с передачей  $1:27$  и надёжно закрепите конструкцию, чтобы избежать проскивания зубцов шестерёнок.



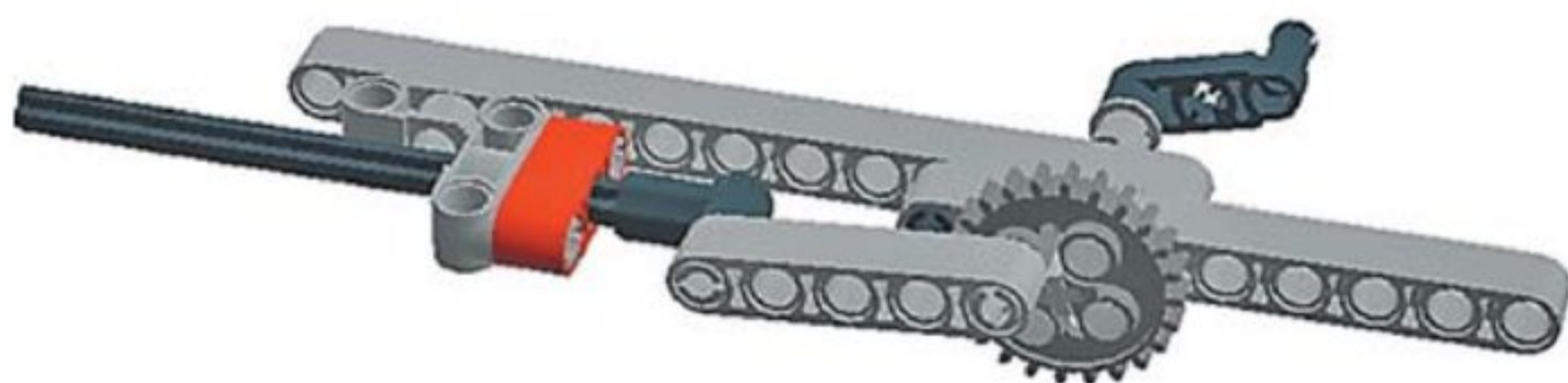
**Запомните** ♦ Механическая передача ♦ Редуктор  
♦ Мультипликатор

## § 4.2. Возвратно-поступательное движение, маятник Капицы

Мультипликатор с высокой скоростью вращения выходного вала можно использовать для создания высокочастотных вибраций, например, небольшого отбойного молотка (рис. 4.5). Для этого потребуются заставить двигаться ось не по кругу, а вперёд–назад.



**Рис. 4.5.** Электрический отбойный молоток: внутри размещены двигатель и специальный кривошипно-шатунный механизм



**Рис. 4.6.** Самодельный кривошипно-шатунный механизм



**Рис. 4.7.** Самодельный отбойный молоток с передачей 1:15

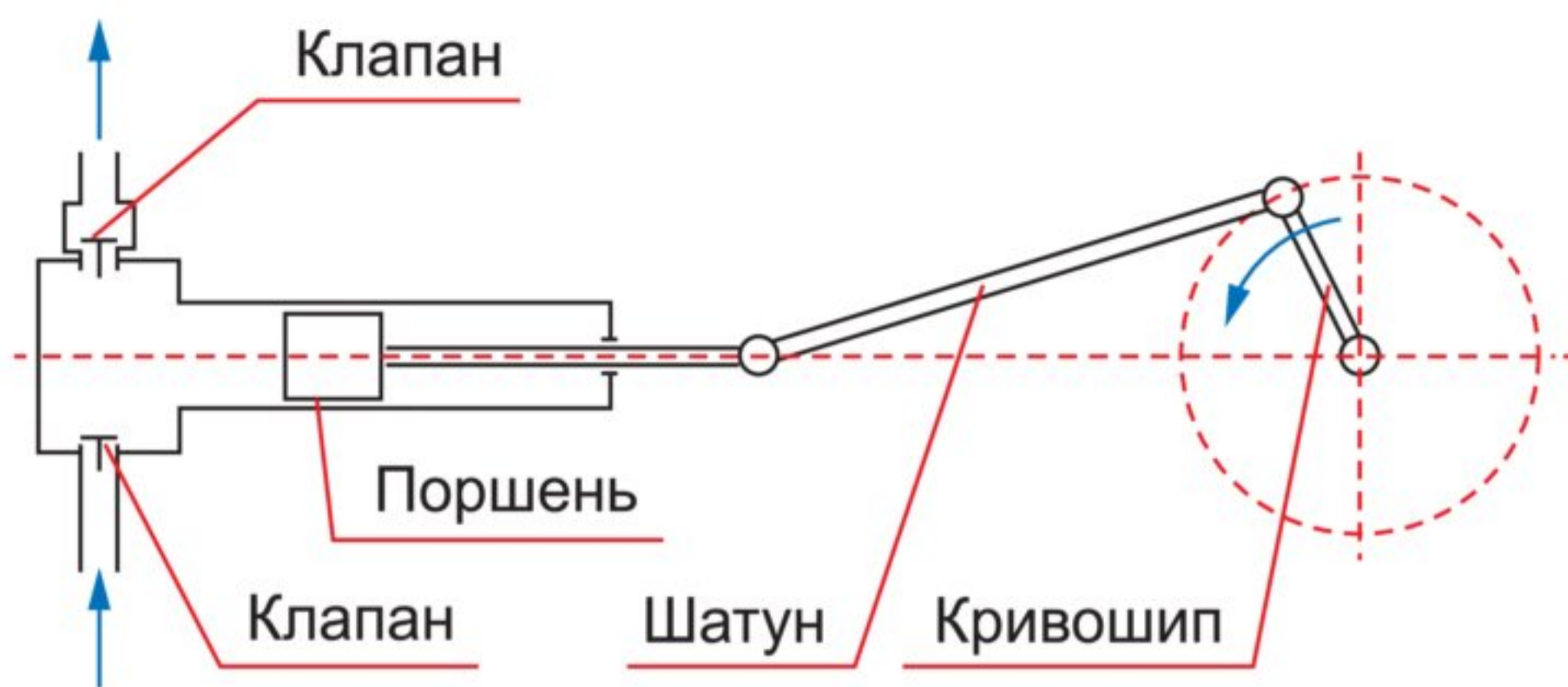
Преобразовать вращение в возвратно-поступательное движение можно с помощью **кривошипно-шатунного механизма**.

Кривошип — это звено механизма, вращающееся вокруг неподвижной оси. Шатун — звено, соединяющее кривошип и ось. Постройте упрощённую конструкцию, в которой обычная ось будет проходить через отверстия в паре зафиксированных деталей. Кривошипом станет штифт, закреплённый в одном из отверстий шестерёнки, а шатуном — небольшая гладкая балка. Вращая шестерню, вы получите возвратно-поступательное движение оси. Обратите внимание, что в конструкторе для подвижных соединений используются серые гладкие штифты (рис. 4.6).

Теперь объединим кривошипно-шатунный механизм с мультипликатором и электромотором. Получим игрушечный отбойный молоток (рис. 4.7). Регулируя передаточное отношение мультипликатора, можно изменять частоту вибраций. У настоящего отбойного молотка эта частота немногим больше 30 ударов в секунду. При передаточном отношении  $i = 1/15$  и скорости мотора примерно 2 оборота в секунду можно получить аналогичный результат.

Часто кривошипно-шатунный механизм работает в совокупности с поршнем: либо в специальном насосе, либо



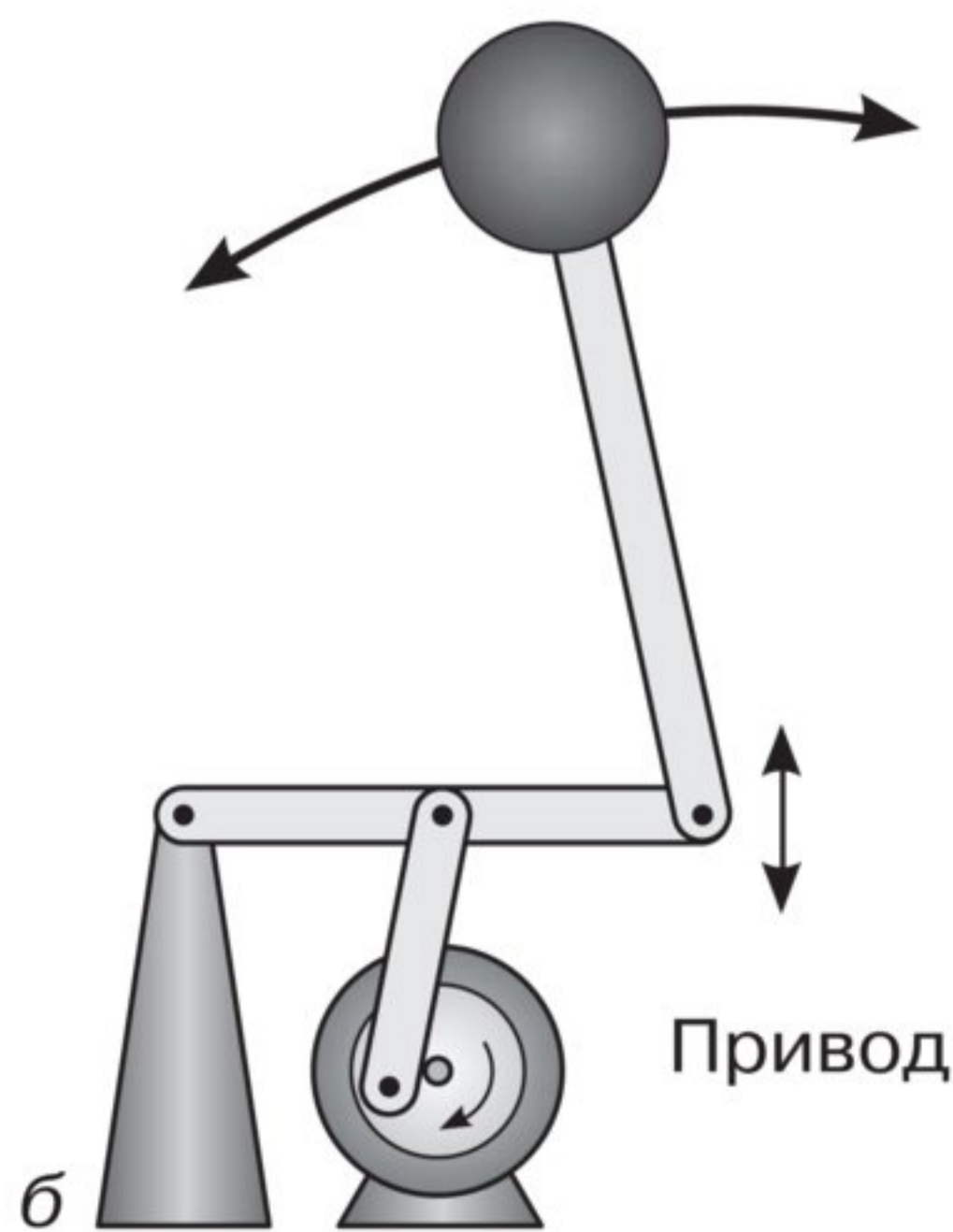
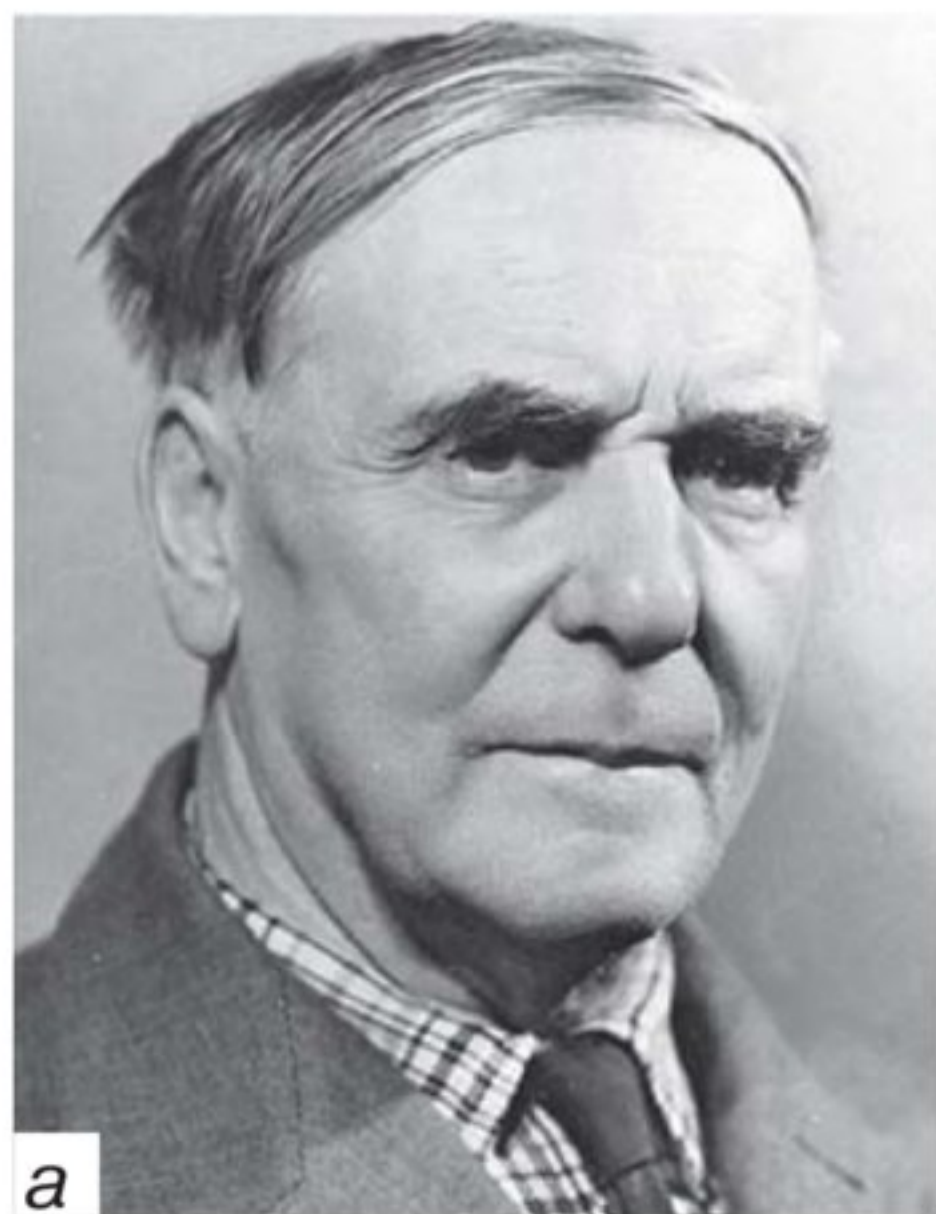


**Рис. 4.8.** Устройство простейшего поршневого насоса

в двигателе внутреннего сгорания. В первом случае движение преобразуется из вращательного в поступательное, во втором — наоборот (рис. 4.8).

Великий русский учёный *Пётр Леонидович Капица* (1894–1984), лауреат Нобелевской премии по физике, построил маятник, который принимает неустойчивое вертикальное положение за счёт вибрации в точке подвеса (рис. 4.9). Он получил название *маятник Капицы* и стал одним из прообразов целой науки, кибернетической физики.

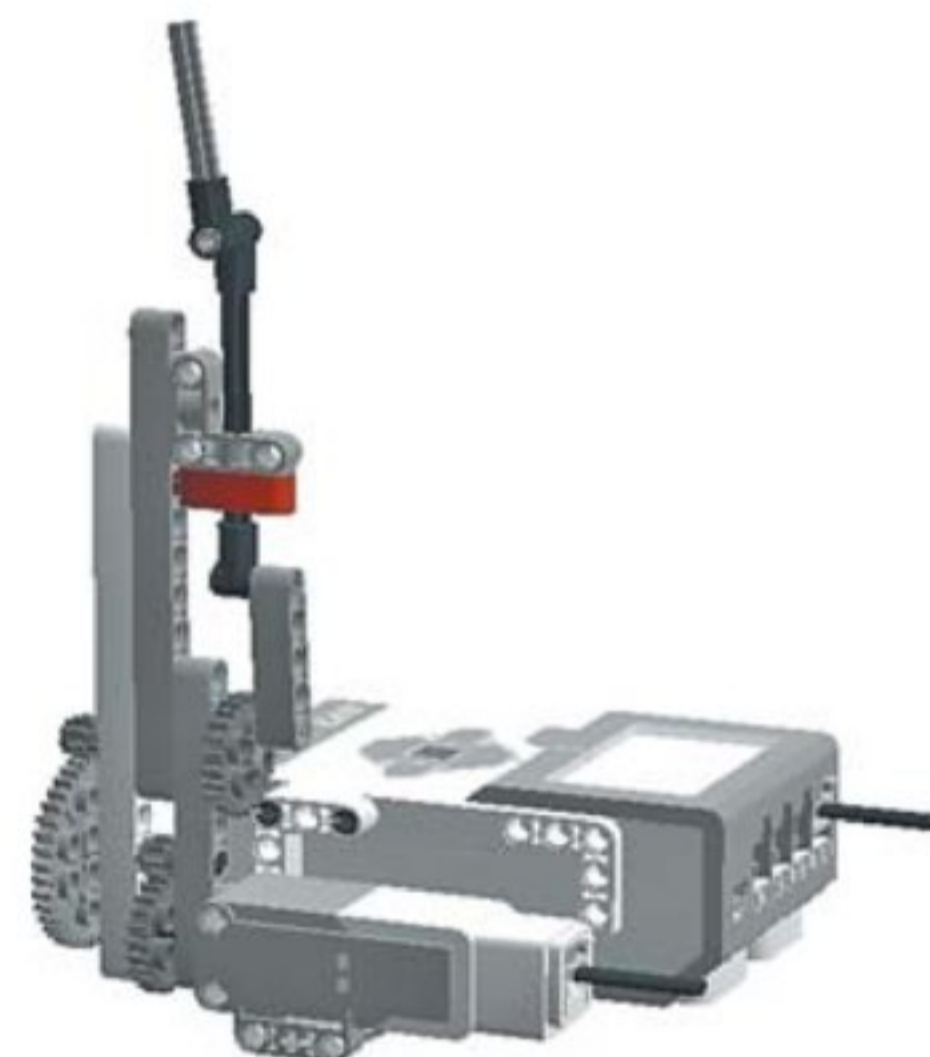
Эта наука изучает физические системы кибернетическими методами, то есть описывает управляемые физические эффекты. И эффекты эти, как правило, весьма необычны. В частности, учёные доказали, что для стабилизации



**Рис. 4.9.** П. Л. Капица (а) и его маятник (б)

маятника Капицы в вертикальном положении можно подобрать такую частоту вибраций, что амплитуда колебаний может быть сколь угодно мала. Это значит, что удерживать маятник в устойчивом вертикальном положении можно без видимых усилий. Вместе с другими открытиями это может привести к появлению вокруг нас прямоходящих роботов и множества других интересных устройств.

Маятник Капицы можно собрать самостоятельно (рис. 4.10).



**Рис. 4.10.** Самодельный маятник Капицы

## Проверьте себя


1. Как в кривошипно-шатунном механизме вращательное движение преобразуется в возвратно-поступательное?
2. Какой мультипликатор надо построить, чтобы при скорости мотора 300 об/мин получить скорость вибрации отбойного молоточка 45 ударов в секунду?
3. Приведите примеры преобразования вращательного движения в поступательное (без возврата).
4. Установите собранный отбойный молоточек в вертикальном положении и закрепите в верхней точке подвижной оси небольшой свободно вращающийся маятник. Регулируя его длину, можно довольно быстро достичь эффекта стабилизации. Постарайтесь отрегулировать систему так, чтобы стабилизировался маятник наибольшей длины.

**Запомните** ♦ Кривошипно-шатунный механизм  
♦ Маятник Капицы

### Это интересно!

Ударный механизм настоящего пневматического отбойного молотка не соприкасается ни с поршнем, ни с шатуном. Удар возникает за счёт работы сжатого воздуха в специальном цилиндре, причём резец отбойника всегда остаётся неподвижным относительно его корпуса. Но ударные волны,

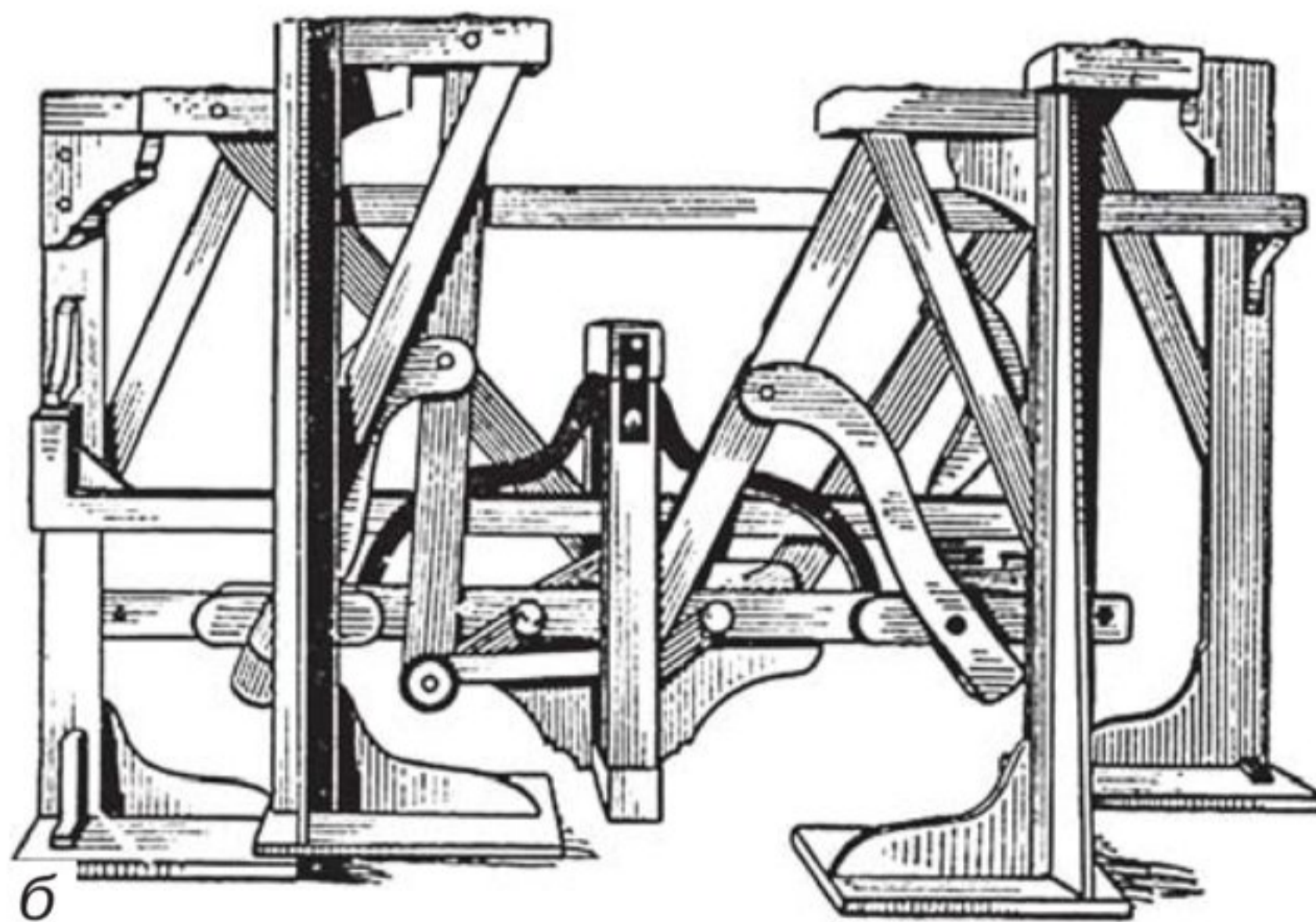
которые распространяются в стали со скоростью 18 000 км/час, разрушают асфальт и бетон, с которыми соприкасается отбойный механизм.

 Первый маятник Капицы был сделан самим Петром Леонидовичем из мотора от швейной машинки. Вы тоже можете построить такой маятник, например, из старой электробритвы, сняв с неё лезвия и установив самодельный стержень с маятником.

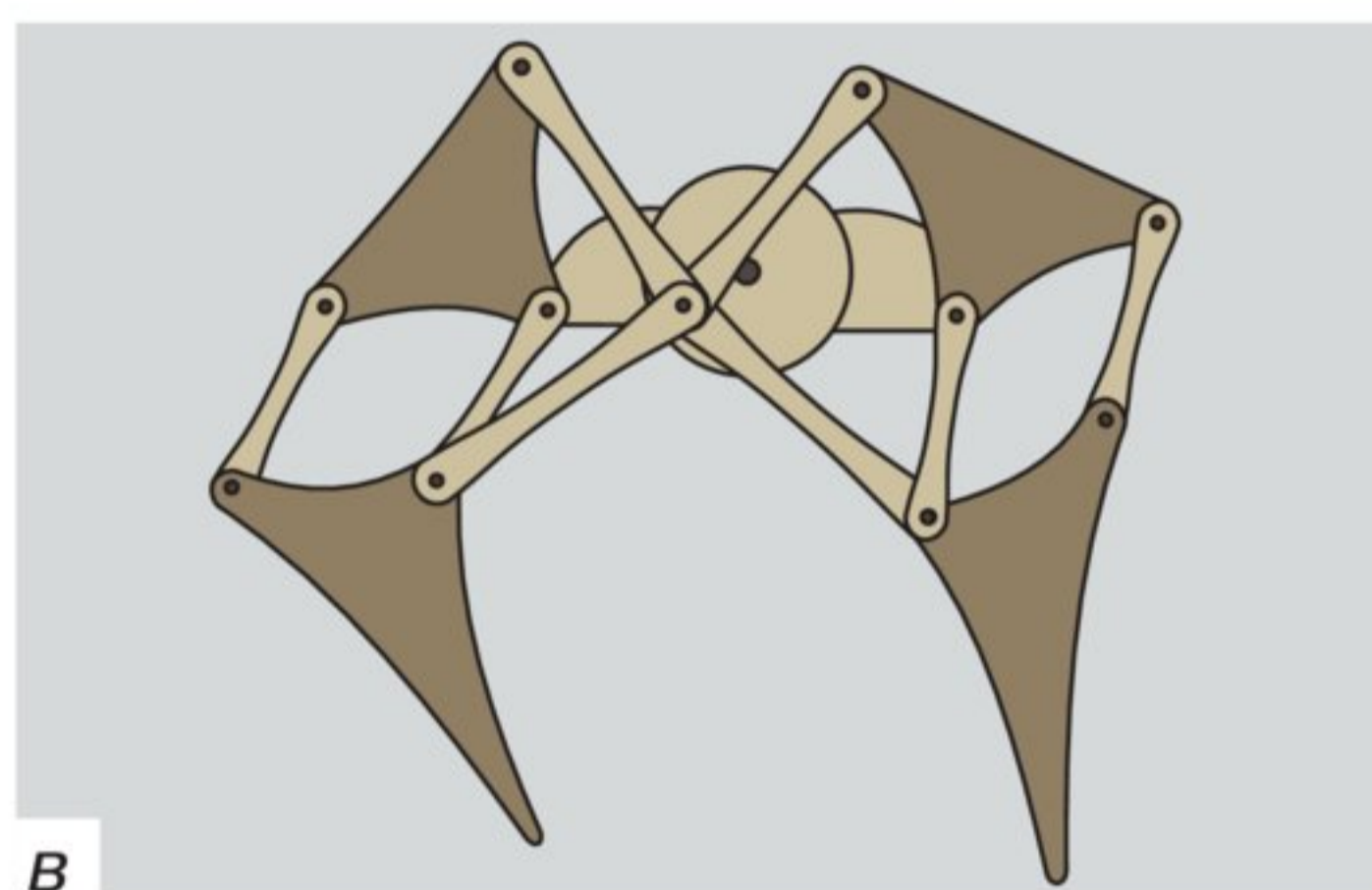
### § 4.3. Шагающие роботы

В природе колесо практически нигде не встречается, это изобретение человека. Для перемещения по поверхности земли у большинства живых существ есть ноги — это самый удобный, хоть и не самый быстрый, механизм. При замене человека роботом часто возникает ситуация, когда колёса малоэффективны и требуются шагающие конструкции. В технике уже давно применяются шагающие механизмы, например, шагающие экскаваторы.

Одним из первых таких механизмов стала «стопоходящая машина». Её изобрёл великий русский математик *Пафнутий Львович Чебышёв* (1821–1894). С помощью



**Рис. 4.11.** П. Л. Чебышёв (а) и его «стопоходящая машина» (б)



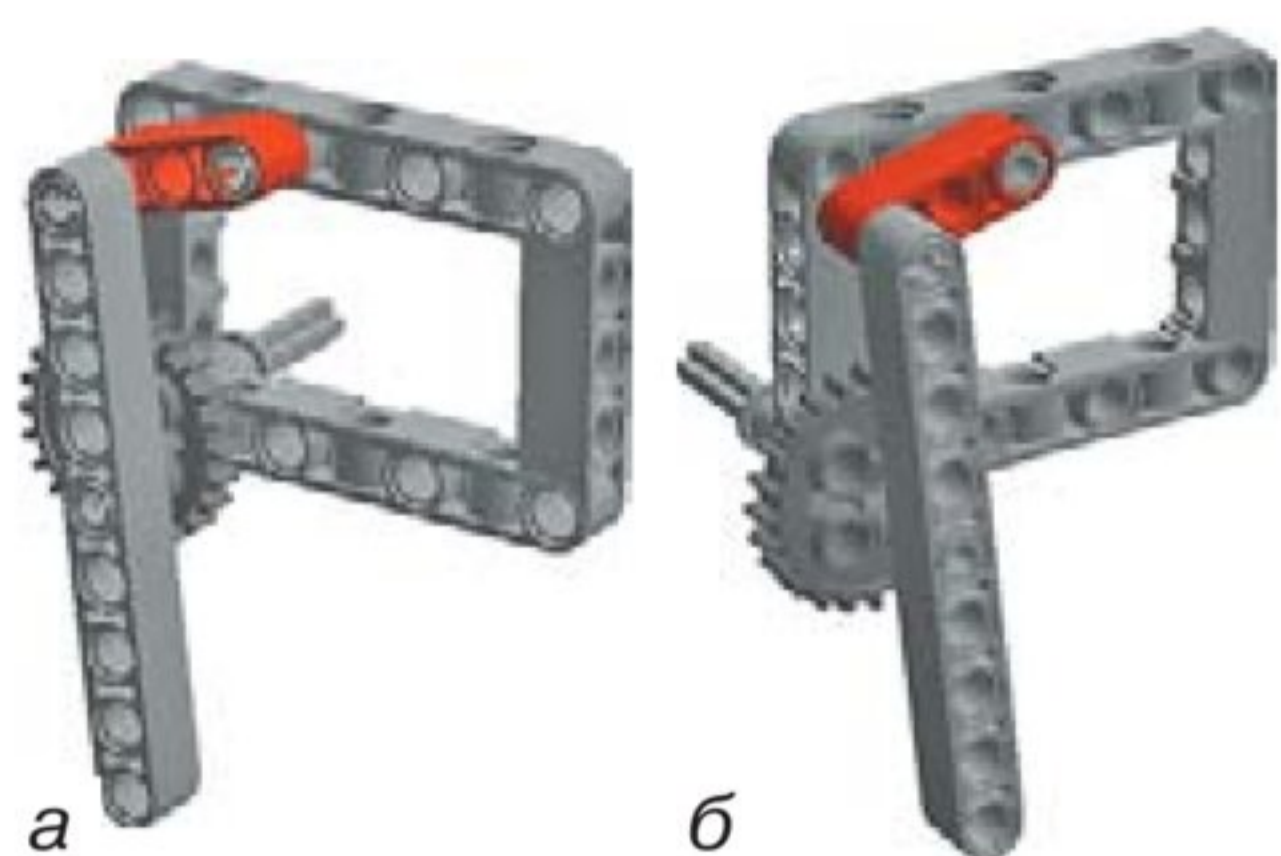
**Рис. 4.12.** а—в — шагающие механизмы Тео Янсена

**Рис. 4.13.** Шагающий велосипед



системы шарниров вращение двигателя преобразуется в плавное и равномерное движение стопыхода (рис. 4.11).

Современный художник и скульптор *Тео Янсен* из Голландии строит удивительные шагающие механизмы, которые могут перемещаться под воздействием ветра и напоминают скелеты гигантских животных (рис. 4.12). Существует даже шагающий велосипед, созданный на основе механизма Тео Янсена (рис. 4.13).



**Рис. 4.14.** а—б — «нога» на основе кривошипно-шатунного механизма



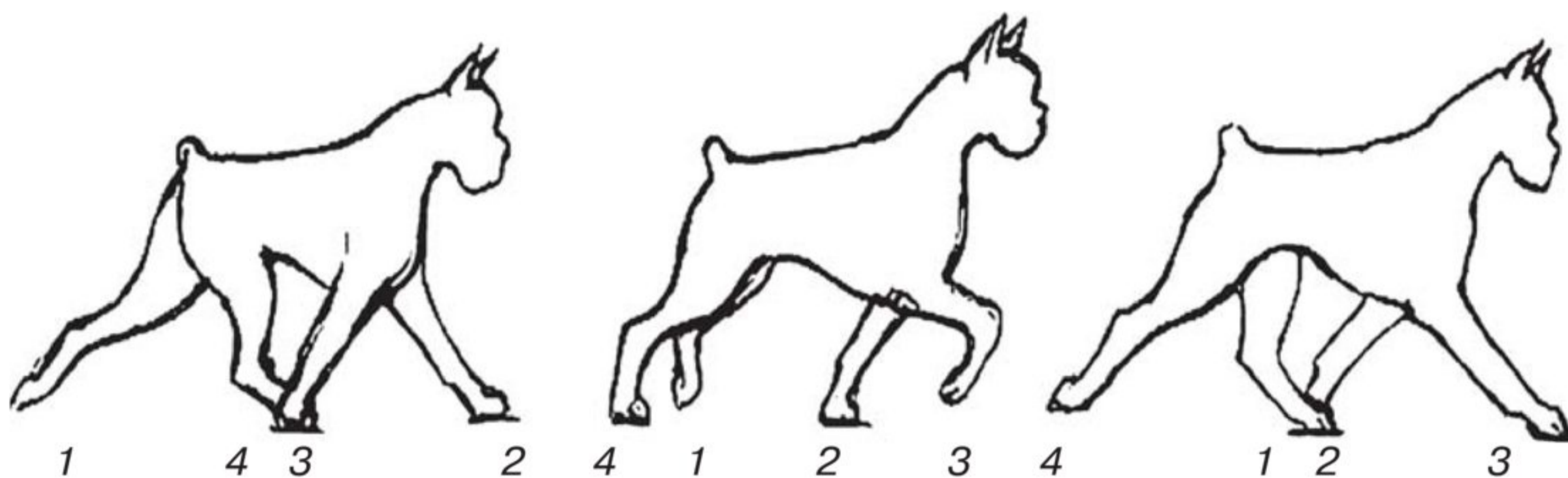
**Рис. 4.15.** Так могут выглядеть шагающие роботы

В основе движения шагающего робота лежит преобразование вращательного движения в поступательное. Для этого проще всего применить кривошипно-шатунный механизм, который вы изучили ранее. Шатун исполнит роль ноги, а кривошип обеспечит движение. Для задания траектории движения точки опоры потребуется добавить дополнительное сочленение наподобие колена (рис. 4.14).

Создать робота, шагающего на двух точках опоры, — непростая задача, поскольку удержание равновесия требует применения очень сложной механики и знания математики. Гораздо проще собрать четвероногого робота (рис. 4.15). Механизм на четырёх точках опоры устойчив, как табурет.

Для правильного размещения кривошипных механизмов (ног) надо понять, как ходит четвероногое существо. На рисунке ноги собаки пронумерованы: 1 — левая задняя, 2 — левая передняя, 3 — правая передняя, 4 — правая задняя (рис. 4.16). Обратите внимание, что соседние ноги всегда двигаются в противоположных направлениях относительно корпуса собаки: когда одна идёт вперёд, вторая остаётся сзади.

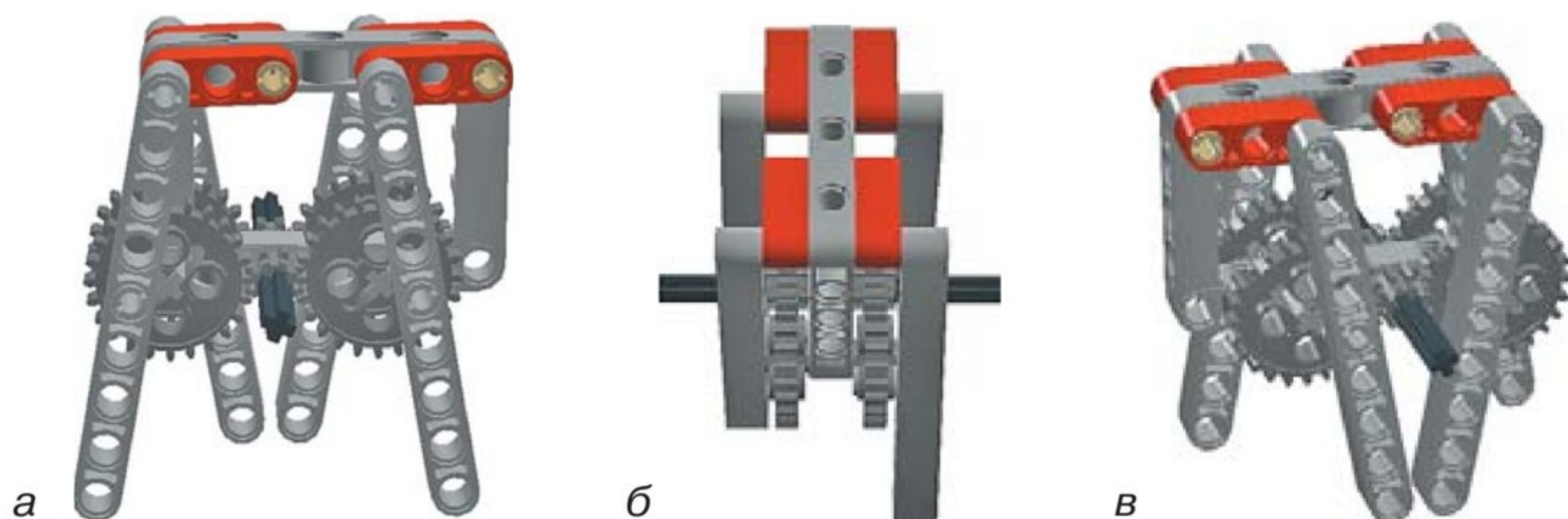
Значит, кривошип соседней ноги следует устанавливать в положении  $180^\circ$ , то есть штифты вставляются в диаметрально противоположные отверстия соседних шестерёнок. Легко заметить, что на рисунке ноги с нечётными номерами двигаются синхронно, так же, как и ноги с чётными номерами (рис. 4.17 и 4.18).



**Рис. 4.16.** Перемещение четвероногого существа

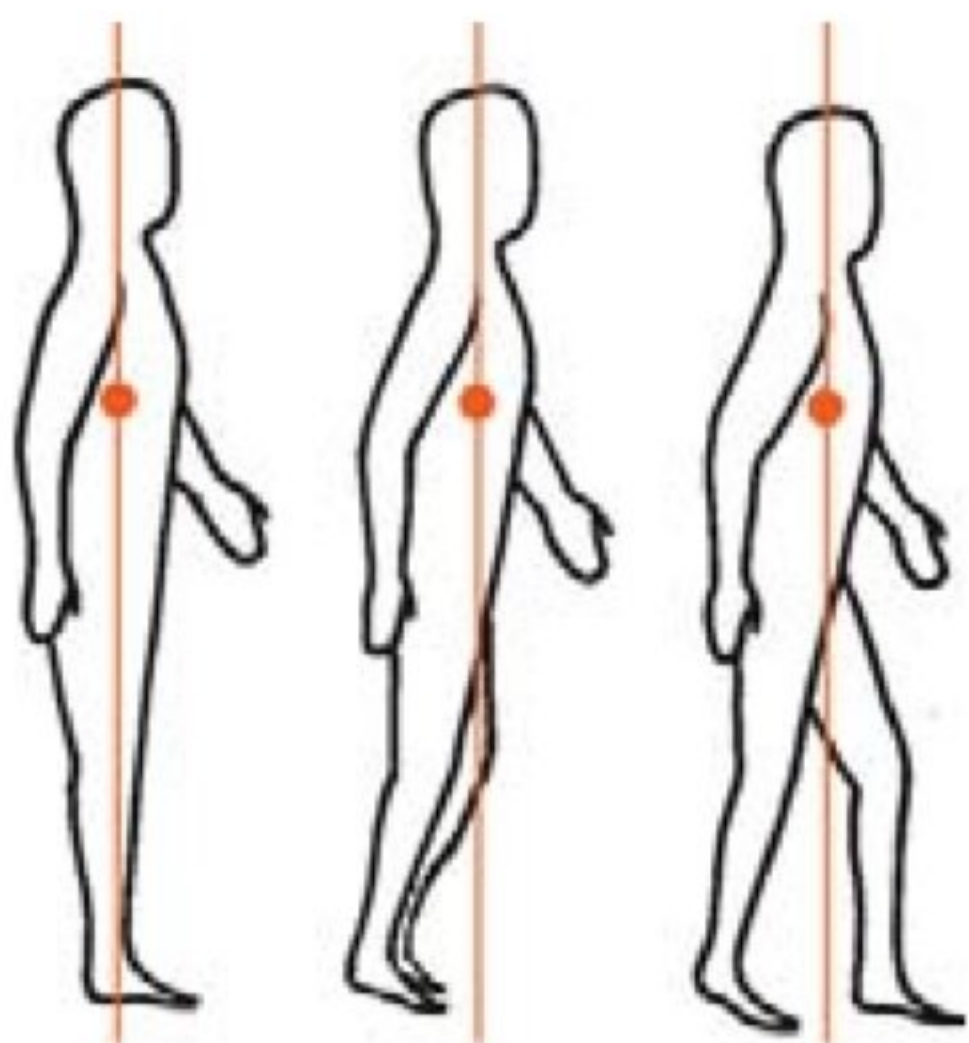


**Рис. 4.17.** Соседние ноги двигаются с разницей в  $180^\circ$  за счёт разной установки кривошипов



**Рис. 4.18.** а—в — наглядная модель четвероногого шагающего механизма с осью для вращения рукой





**Рис. 4.19.** Ходьба — это падение

Ходьба — это падение (рис. 4.19). Человек немного наклоняет корпус и начинает как бы падать вперёд, но в последний момент выставляет ногу. Звучит странно, но именно так мы и ходим. Иными словами, при ходьбе необходимо постоянно переносить центр масс в направлении движения. В тот момент, когда центр масс переносится за крайнюю точку опоры конечности, начинается падение. Когда нога выставляется вперёд, равновесие восстанавливается.

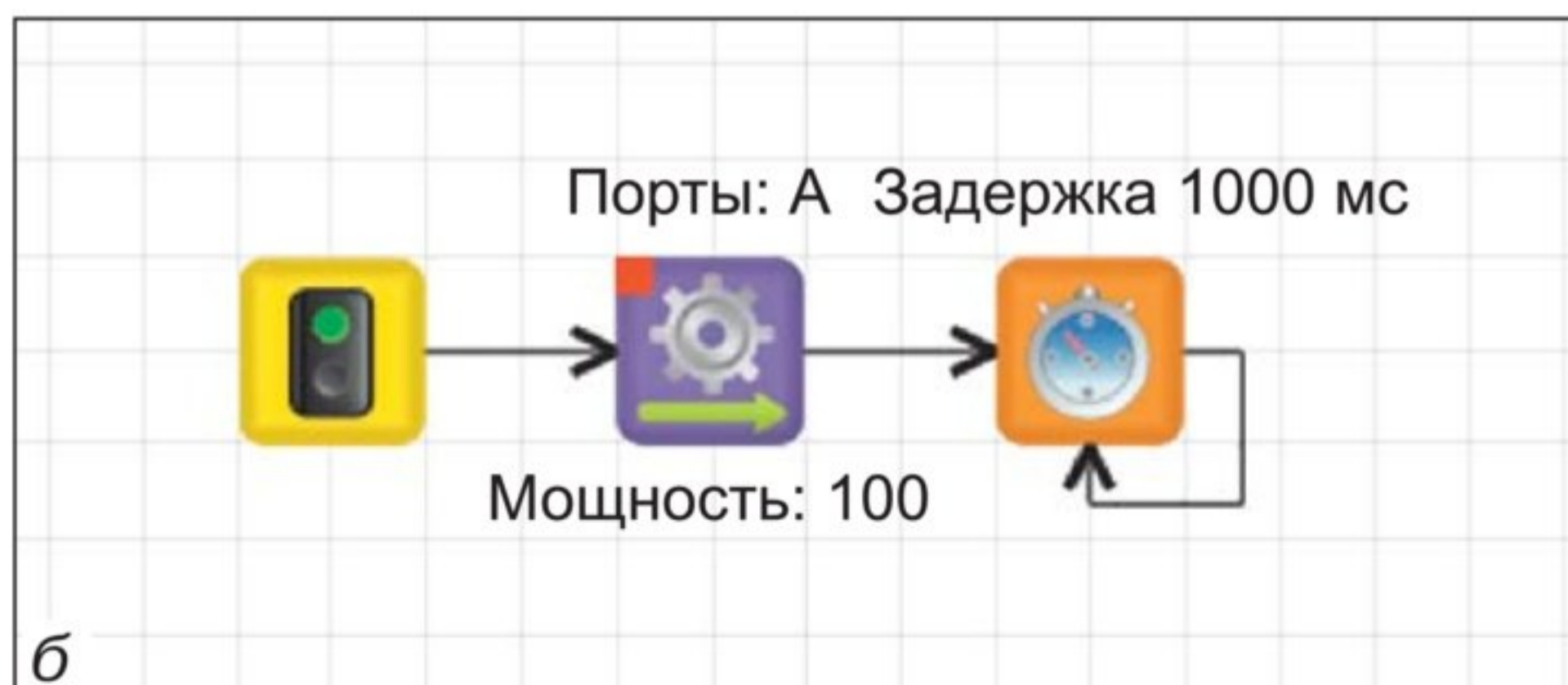


## Проверьте себя

1. В каких областях человеческой деятельности могут применяться шагающие роботы?
2. Как преобразуется вращение мотора в возвратно-поступательное движение конечности?
3. Как расположены у четвероногого существа пары конечностей, которые при ходьбе двигаются синхронно: спереди, сзади, по диагонали, сбоку?
4. Возможно ли синхронное движение других пар конечностей во время прыжков или бега? Ответ поясните.
5. Постройте шагающий механизм с электродвигателем и блоком питания (можно с контроллером), смещая центр масс в направлении движения (рис. 4.20). Если все части системы будут построены правильно, по программе «Мотор вперёд» (или «Мотор



а




б


**Рис. 4.20.** Шагающий механизм (а) и программа «Мотор вперёд» (б)



- назад») робот начнёт движение с достаточно большой скоростью. Проведите соревнования, кто придёт к финишу быстрее.
- Придумайте «сапожки» для четвероногого робота, которые улучшат его движение.
  - Какие элементы можно добавить в конструкцию, чтобы конечности и шестерёнки не откреплялись даже при больших нагрузках?
  - Увеличьте скорость движения робота за счёт механической передачи.

### Это интересно!

 Четвероногий шагающий робот из Китая, названный «Ходок № 1», в 2015 году поставил рекорд по пройденной дистанции на одном заряде аккумуляторной батареи. За 54 часа 30 минут он прошагал путь в 134 км, сделав около 360 тысяч шагов.

 Самый быстрый шагающий робот под названием «Гепард», сделанный американскими учёными, смог развить скорость 28,9 км/ч. Его механика имеет много общего с самым быстрым животным на планете. В частности, робот сгибает и разгибает спину, что увеличивает силу толчка конечностями.

# 5

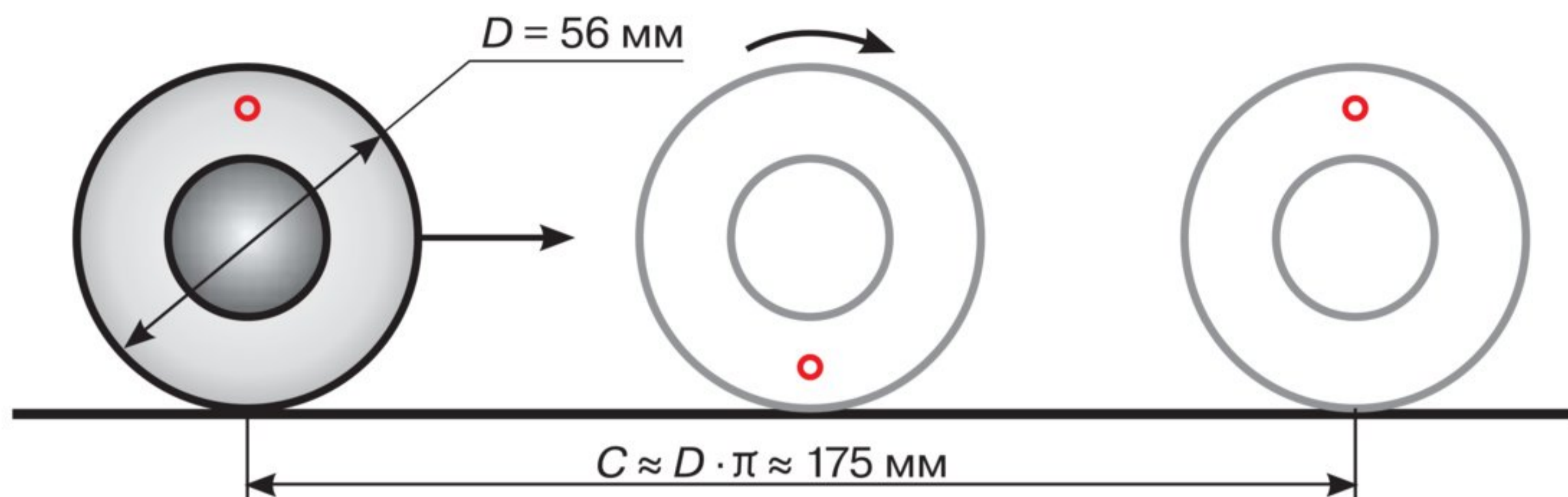
## Управление движением робота

### § 5.1. Точные перемещения

Вы уже знаете, что скорость вращения мотора зависит от напряжения батареи. Это значит, что при изменении заряда моторы за одно и то же время будут совершать разное число оборотов. Тогда почти все программы перемещений придётся корректировать, изменяя временные задержки. Но существует метод, позволяющий сделать перемещения робота независимыми от заряда батарей — с помощью датчиков оборотов *энкóдеров*. На небольшой скорости мотора точное перемещение по градусам на энкодере практически неизменно при любом заряде батареи.

Как проехать заданное расстояние, используя встроенные датчики оборотов? Очевидно, нужно знать, какое расстояние колесо проезжает за один полный оборот, а также за один градус поворота (рис. 5.1). Допустим, диаметр колеса  $D = 56$  мм. Чтобы найти длину окружности колеса, нужно его диаметр умножить на число  $\pi$ . Тогда за один оборот колесо проедет расстояние  $C$ , равное  $D \cdot 3,14 \approx 175$  мм. Можно для простоты расчётов округлить это число до 180 мм. Тогда за один градус поворота энкодера колесо будет проезжать расстояние  $180/360 = 0,5$  мм. Нетрудно посчитать, что 1 м робот проедет примерно за  $2000^\circ$  поворота ко-





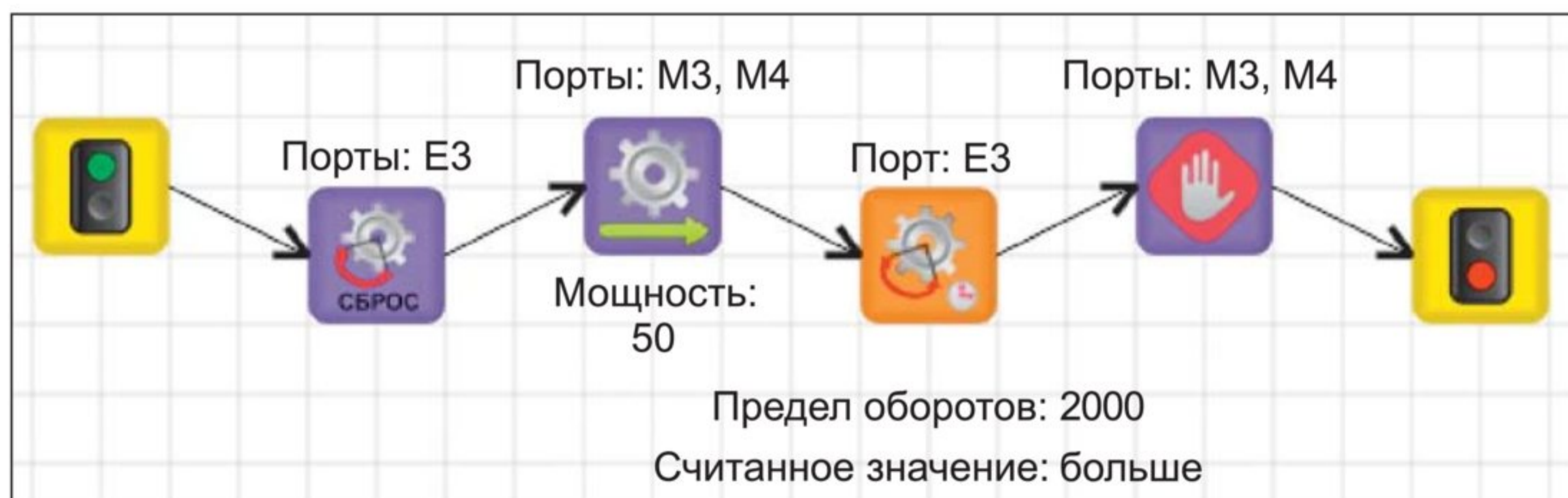
**Рис. 5.1.** Колесо за один полный оборот проезжает путь, равный длине его окружности

леса:  $0,5 \cdot 2000 = 1000$  мм. Найдите более точное значение самостоятельно.

Составим программу, по которой робот проедет 1 м и остановится. Для начала отсчёта потребуется обнулить энкодер, то есть сбросить показания датчика оборотов. После включения моторов стоит команда ожидания показания энкодера. Она сработает, когда мотор повернётся на  $2000^\circ$  или больше (рис. 5.2). Обратите внимание, что скорость надо ставить небольшую, иначе может произойти проскальзывание колёс и точность перемещения понизится.

Что нужно изменить в программе, чтобы робот проехал 1 м назад и остановился?

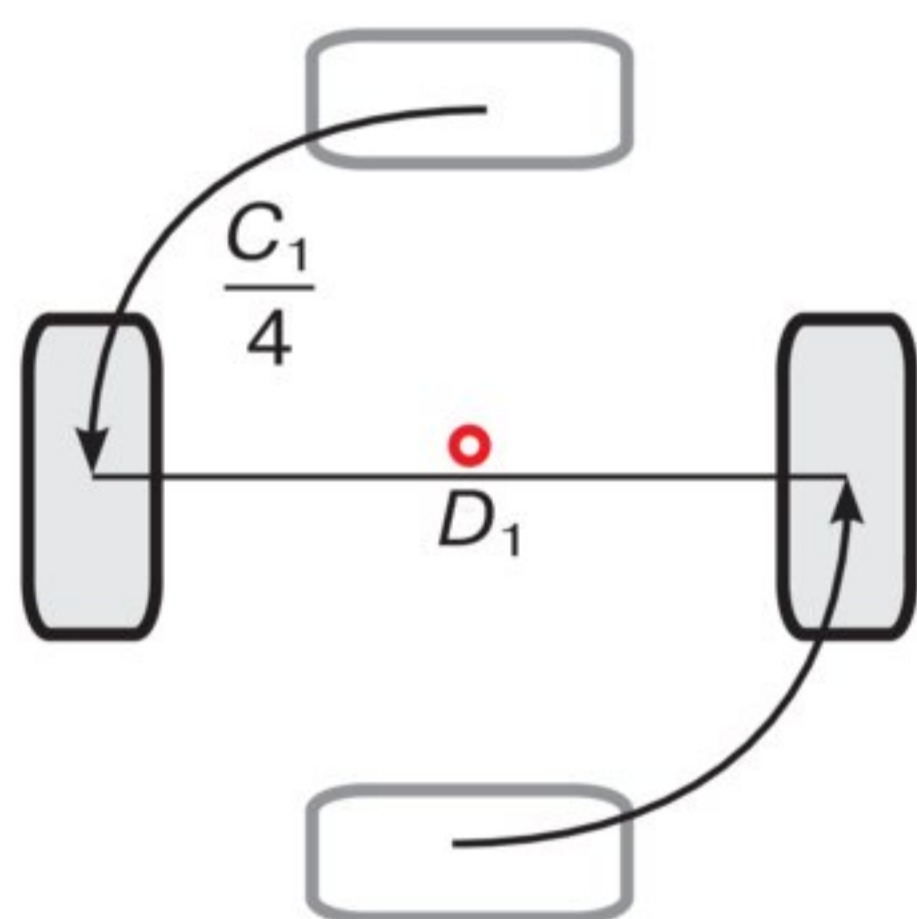
Следующая задача — точный поворот корпуса тележки на  $90^\circ$  налево. При повороте всего робота придётся



**Рис. 5.2.** Робот проедет  $2000^\circ$  по энкодеру и остановится



**Рис. 5.3.** Колея транспортного средства (а) и колея на дороге (б)



**Рис. 5.4.** Траектории движения колёс при повороте тележки

учитывать не только диаметр колеса, но и колею. В обычной жизни под колёй понимают углубление от колёс на дороге, но в технике колея — это расстояние между центральными линиями колёс или гусениц (рис. 5.3).

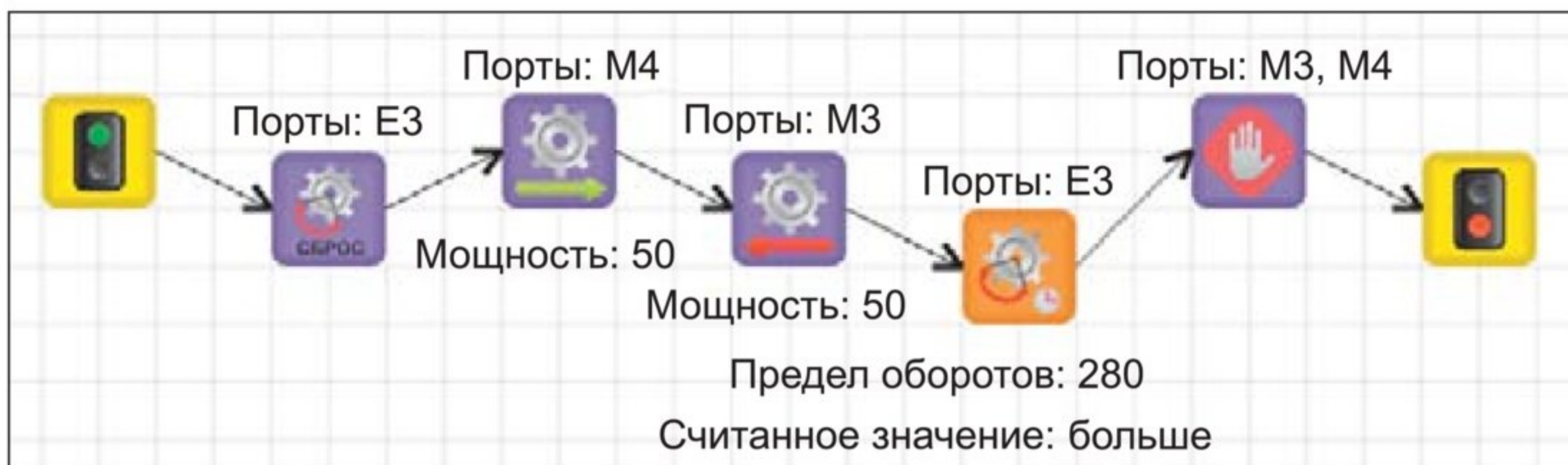
Чтобы выполнить точный поворот корпуса тележки, нужно представить, по какой траектории будут двигаться колёса (рис. 5.4). Предположительно, это будут дуги окружности диаметра  $D_1$ , равного ширине колеи. Полная длина этой окружности  $C_1 = D_1 \cdot 3,14$ . Поскольку длина дуги определяется

в градусах, то расчёт значений для энкодера можно провести по следующей формуле:

$$E = \frac{C_1 \cdot \frac{90^\circ}{360^\circ}}{C} \cdot 360^\circ = \frac{C_1}{C} \cdot 90^\circ = \frac{D_1}{D} \cdot 90^\circ.$$

При колее  $D_1 = 168$  мм можно получить значение поворота энкодера на  $270^\circ$ . Но ширина тележки может отличаться и, как правило, точное значение приходится подбирать вручную.

Для изменения направления поворота потребуются внести несколько изменений в программу: поменять направления вращения моторов и знак ожидаемых значений



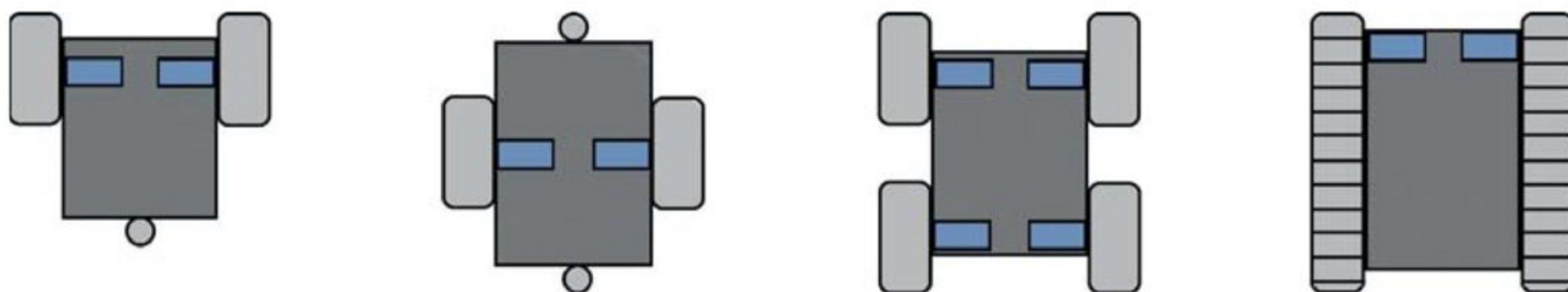
**Рис. 5.5.** Поворот тележки налево на  $90^\circ$



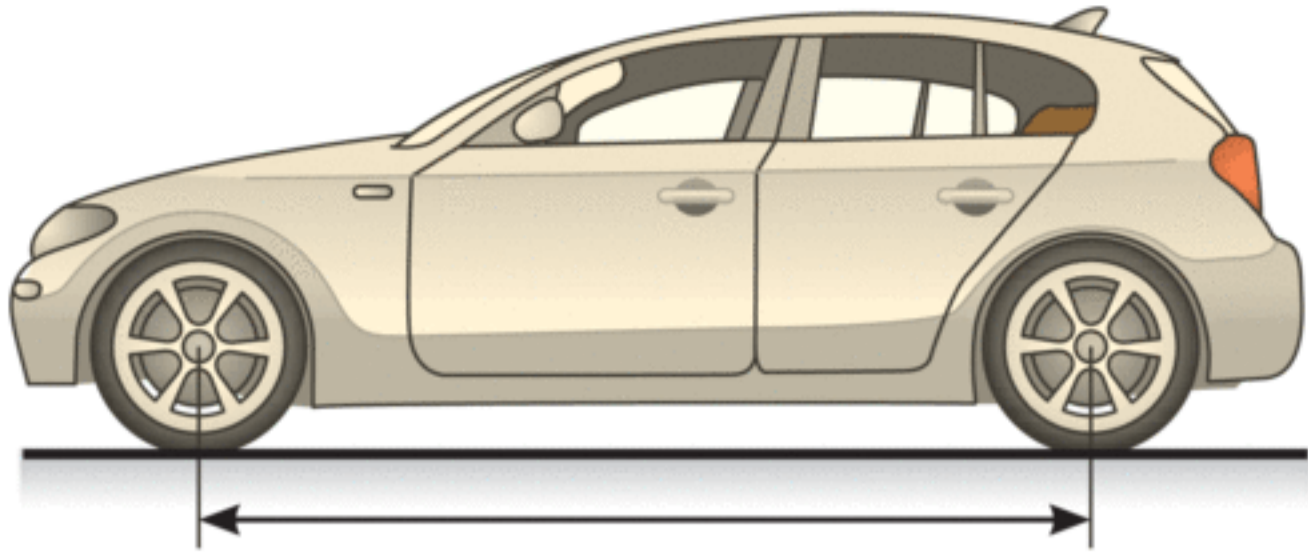
**Рис. 5.6.** Поворот тележки направо на  $90^\circ$

в градусах на энкодере (рис. 5.5 и 5.6). Поскольку мотор, энкодер которого используется в программе, будет вращаться назад, то ожидаемое число градусов должно быть отрицательным.

Поворот на месте может выполнить далеко не любое транспортное средство. На рисунке 5.7 приведены примеры шасси, которые способны выполнить такой поворот.



**Рис. 5.7.** Шасси с моторами, способное выполнить корректный поворот на месте



**Рис. 5.8.** Колёсная база транспортного средства

Для этого транспортное средство должно обладать как минимум двумя независимыми моторами с каждой стороны и соответствующим шасси — колёсным или гусеничным.

Для корректного поворота на месте четырёхколёсной тележки существенную роль

играет колёсная база — расстояние между центрами колёс передней и задней осей (рис. 5.8). Колесная база не должна быть длиннее колеи. Кроме того, тележка должна обладать полным приводом: если одна пара колёс будет ведущими, а вторая пара — свободно вращающимися, то эта вторая пара при повороте будет просто волочиться по поверхности. Это создаст боковое трение шин о поверхность, которое помешает корректному повороту.



## Проверьте себя

1. Почему длительность поворота надёжнее измерять с помощью энкодера, чем по таймеру?
2. Как, зная диаметр колеса  $D$ , определить путь, пройденный за  $1^\circ$  поворота?
3. Каков смысл слова «колея»?
4. Определите примерную длину пути, пройденного левым колесом тележки с колеёй  $D_1 = 20$  см, при полном обороте вокруг неподвижного правого колеса.
5. Постройте алгоритм движения тележки по волнистой линии и проверьте его на двумерной модели.
6. Постройте алгоритм для игры кегельринг, используя энкодеры.
7. Постройте двухмоторную четырёхколёсную тележку с полным приводом и отладьте на ней повороты на прямой угол ( $90^\circ$ ).
8. Постройте двухколесную передачу с двумя роликовыми опорами и отладьте на ней поворот на прямой угол. Что должно быть расположено чуть выше — ролики или ведущие колёса?



## Запомните ♦ Энкодер

## Это интересно!

💡 «Танковая» схема шасси обеспечивает гораздо большую манёвренность и проходимость, чем у обычного автомобиля. Такая схема используется у некоторых тягловых машин, например, бульдозеров. Кроме того, машины-погрузчики, работающие в условиях ограниченного пространства, могут быть оборудованы четырьмя колёсами с полным приводом. Главная особенность такой схемы шасси — возможность поворота на месте.

💡 Трёхопорные шасси распространены у лёгких и средних самолётов. У них количество колёс на одной опоре может быть увеличено для большей надёжности и лучшего сцепления с поверхностью.

## § 5.2. Путешествие в лабиринте

Роботам приходится существовать в мире людей, который повсеместно ограждён стенами: в помещениях и на улицах городов, в подвалах и катакомбах (рис. 5.9). Ориентироваться в таких лабиринтах — очень важная задача. Некото-



Рис. 5.9. Городские лабиринты





**Рис. 5.10.** Полигон для обхода лабиринта

рые лабиринты роботу заранее известны, в других он может оказаться впервые.

Для начала следует научить робота проезжать путь, который известен заранее. Большинство лабиринтов имеют прямоугольную структуру, поэтому здесь пригодится умение поворачиваться на месте на  $90^\circ$ .

На занятиях робототехникой обычно используют небольшой лабиринт с квадратными ячейками со стороной 30 см и высотой стенок 10 см (рис. 5.10).

Построим робота, который сможет свободно поворачиваться в такой ячейке, не касаясь стенок.




Для перемещения робота в лабиринте потребуется определить три основных действия: движение вперёд, направо, налево. Каждое действие состоит из следующих простых операций:

- ◆ обнуление датчика оборотов (энкодера) на одном из моторов;
- ◆ включение моторов на движение в определённых направлениях;
- ◆ ожидание значения энкодера;
- ◆ остановка моторов.

Обратите внимание, что при повороте налево левый мотор  $B$  вращается назад и на датчике оборотов появляются отрицательные значения. Копируя команды для этих трёх действий, можно построить программу прохождения роботом известного лабиринта (табл. 5.1).




Однако при отладке программы быстро станет понятно, что от робота необходимо каким-то образом получать информацию о выполнении им очередного действия. Для этого удобно использовать различные звуковые сигналы после выполнения каждой команды, пока робот стоит на месте. В некоторых роботах есть возможность выбрать заранее записанные сигналы или установить частоту их

**Таблица 5.1.** Основные действия робота в лабиринте

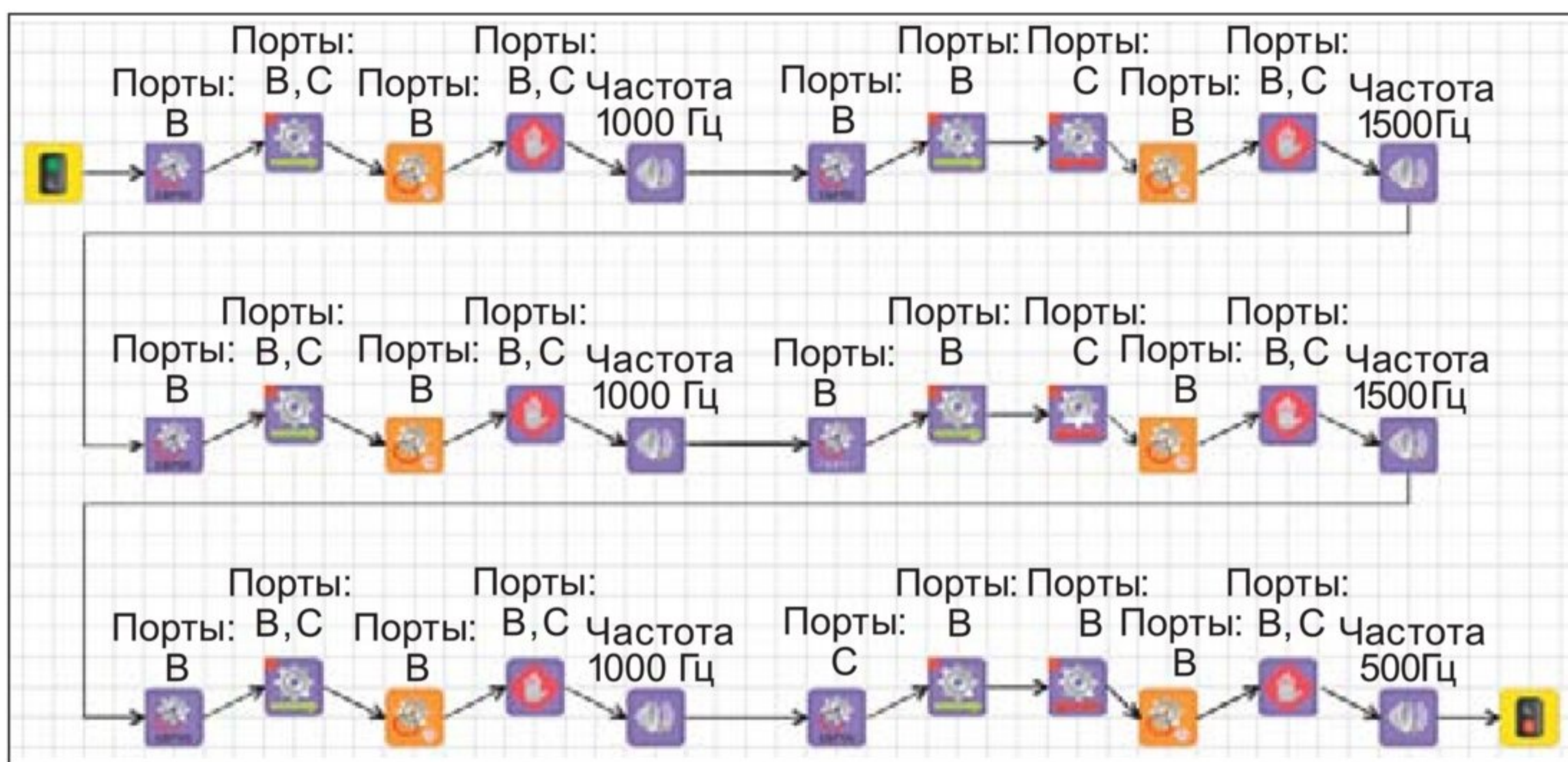
Команда	Описание	Программа
Вперёд	Робот проезжает вперёд одну ячейку лабиринта и останавливается	 <p>Порты: В, С      Порты: В, С</p> <p>Порты: В      Порты: В</p> <p>Мощность: 50</p> <p>Предел оборотов: 500</p> <p>Считанное значение: больше</p>
Направо	Робот поворачивается на месте направо на 90° и останавливается	 <p>Порты: В      Порты: С      Порты: В, С</p> <p>Порты: В      Порты: В</p> <p>Мощность: 50</p> <p>Предел оборотов: 250</p> <p>Считанное значение: больше</p>
Налево	Робот поворачивается на месте налево на 90° и останавливается	 <p>Порты: В      Порты: В      Порты: В, С</p> <p>Порты: С      Порты: В</p> <p>Мощность: 50</p> <p>Предел оборотов: -250</p> <p>Считанное значение: меньше</p>

звучания. Известно, что звук — это вибрация некоторой мембраны. У человека вибрируют голосовые связки, а у робота — детали динамика. В музыке каждой ноте соответствует определённая частота. Например, нота «ля» первой октавы имеет частоту 440 Гц (Герц), то есть 440 колебаний в секунду. При увеличении частоты звук становится выше, при уменьшении — ниже. Таким образом, нота «ля» второй октавы имеет частоту 880 Гц. Все остальные

**Таблица 5.2.** Звуковые сигналы после выполнения действий

Команда	Фрагмент программы
Вперёд	 <p>Порты: В, С Порты: В Порты: В, С Частота: 1000 Гц Мощность: 50 Предел оборотов: 500 Считанное значение: больше Громкость: 100% Длительность: 1000 мс Ждать завершения: Истина</p>
Направо	 <p>Порты: В Порты: В Порты: С Порты: В, С Частота: 1500 Гц Мощность: 50 Предел оборотов: 250 Считанное значение: больше Громкость: 100% Длительность: 1000 мс Ждать завершения: Истина</p>
Налево	 <p>Порты: В Порты: С Порты: В Порт: В Порты: В, С Частота: 500 Гц Мощность: 50 Предел оборотов: -250 Считанное значение: меньше Громкость: 100% Длительность: 1000 мс Ждать завершения: Истина</p>

ноты имеют дробные значения частоты. Используя эти знания, установите такие частоты сигналов, чтобы робот издавал чистые звуки «ля» на различных октавах или комбинации из этих звуков (табл. 5.2).



**Рис. 5.11.** Пример программы для прохождения части лабиринта

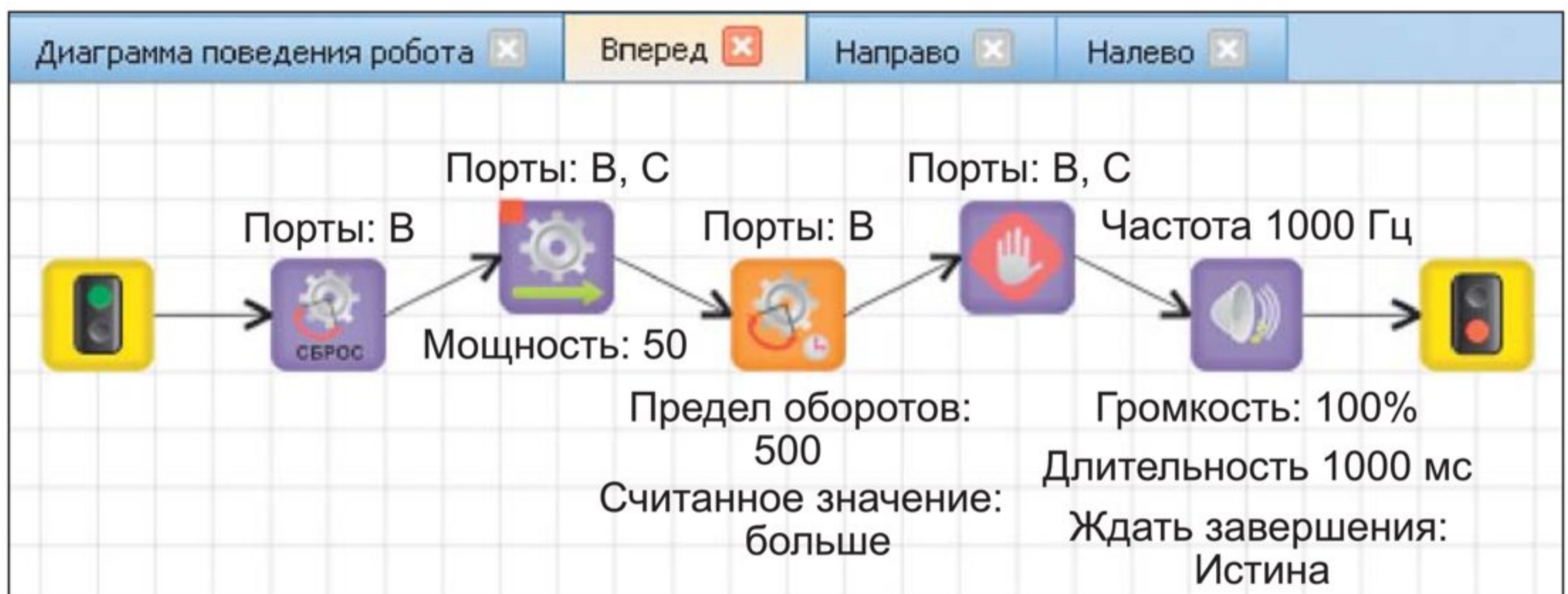
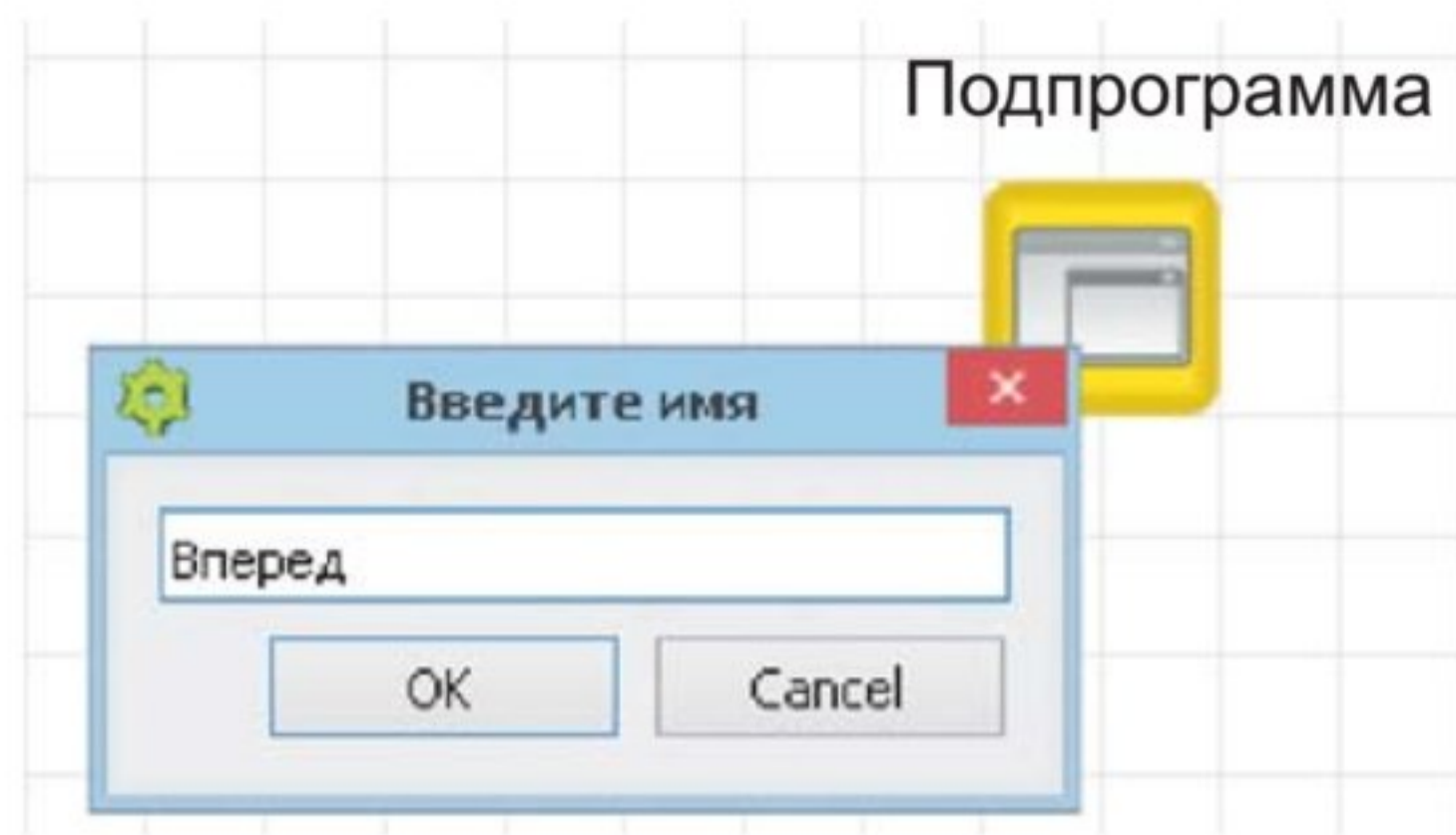
Постройте программу для прохождения лабиринта в вашем классе, устанавливая блоки в цепочку (рис. 5.11).

Надо признать, что получается очень длинно. Можно ли сократить программу, заменив повторяющиеся фрагменты? Для этого существует специальная алгоритмическая конструкция, которая называется подпрограмма. **Подпрограмма** — это вспомогательный алгоритм, записанный отдельно от основной программы и вызываемый по имени.

Для каждого из базовых действий в лабиринте создайте отдельные подпрограммы и назовите их соответствующим образом. Двойным щелчком по значку созданной подпрограммы можно открыть отдельную вкладку для неё. На этой вкладке подпрограмма выглядит как обычная программа со светофором начала и конца (рис. 5.12).

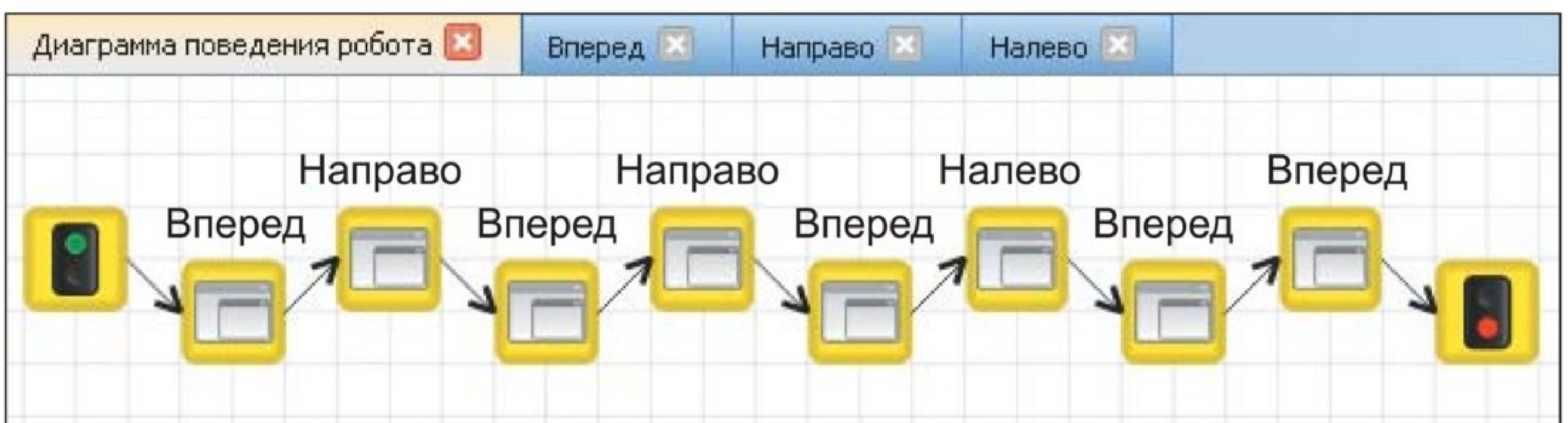
Когда все подпрограммы на своих вкладках будут готовы, составьте основную программу из нескольких вызовов подпрограмм (рис. 5.13). Обратите внимание, что подпрограммы выполняются только после вызова и каждую подпрограмму можно вызвать несколько раз.



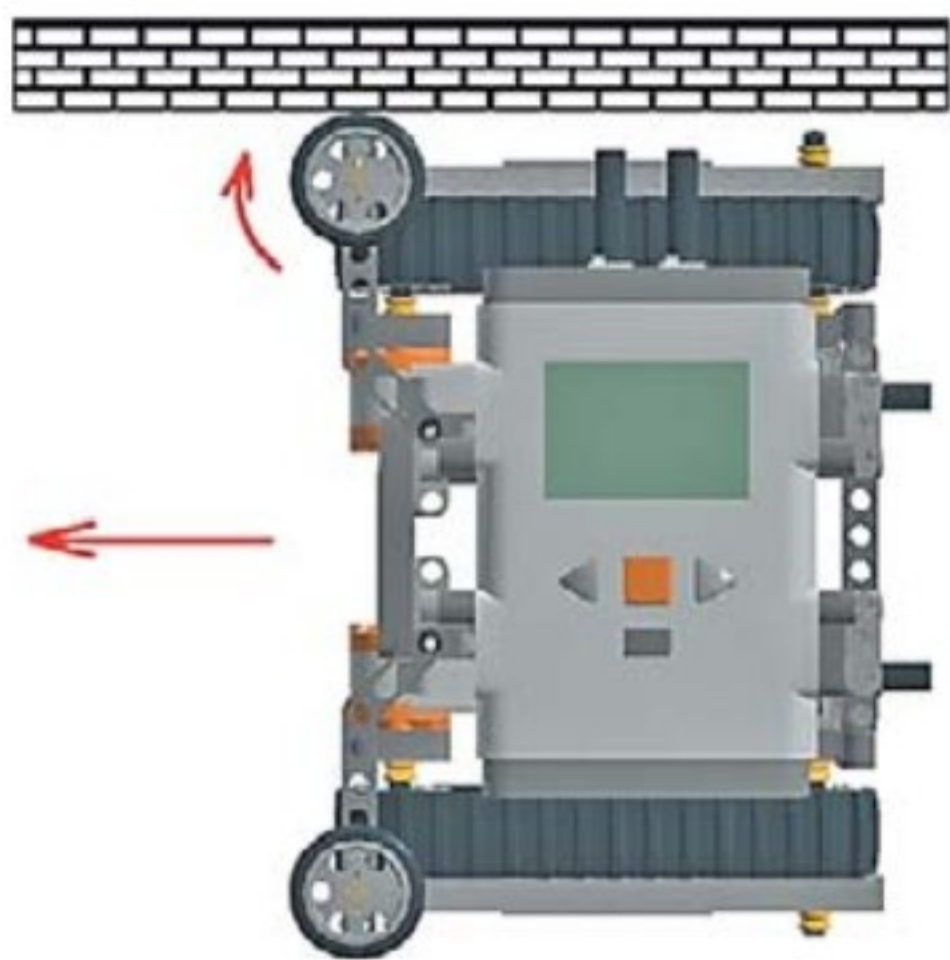


**Рис. 5.12.** Создание подпрограммы «Вперёд»

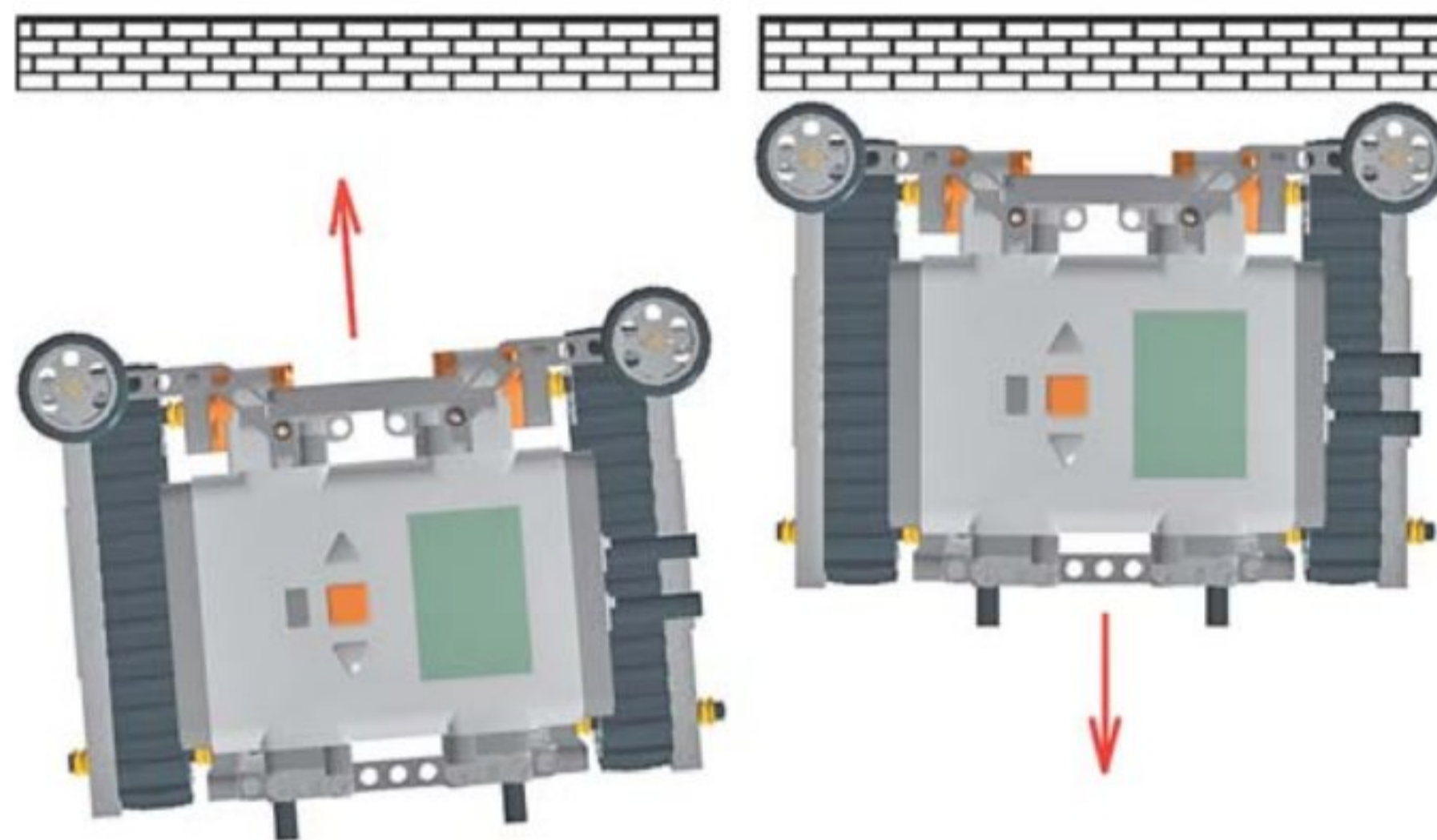
В процессе выполнения программы неизбежно накапливается ошибка при выполнении поворотов и робот рано или поздно может зацепиться за угол или стену. Отчасти



**Рис. 5.13.** Прохождение лабиринта с использованием подпрограмм



**Рис. 5.14.** Выравнивание при движении вдоль стены

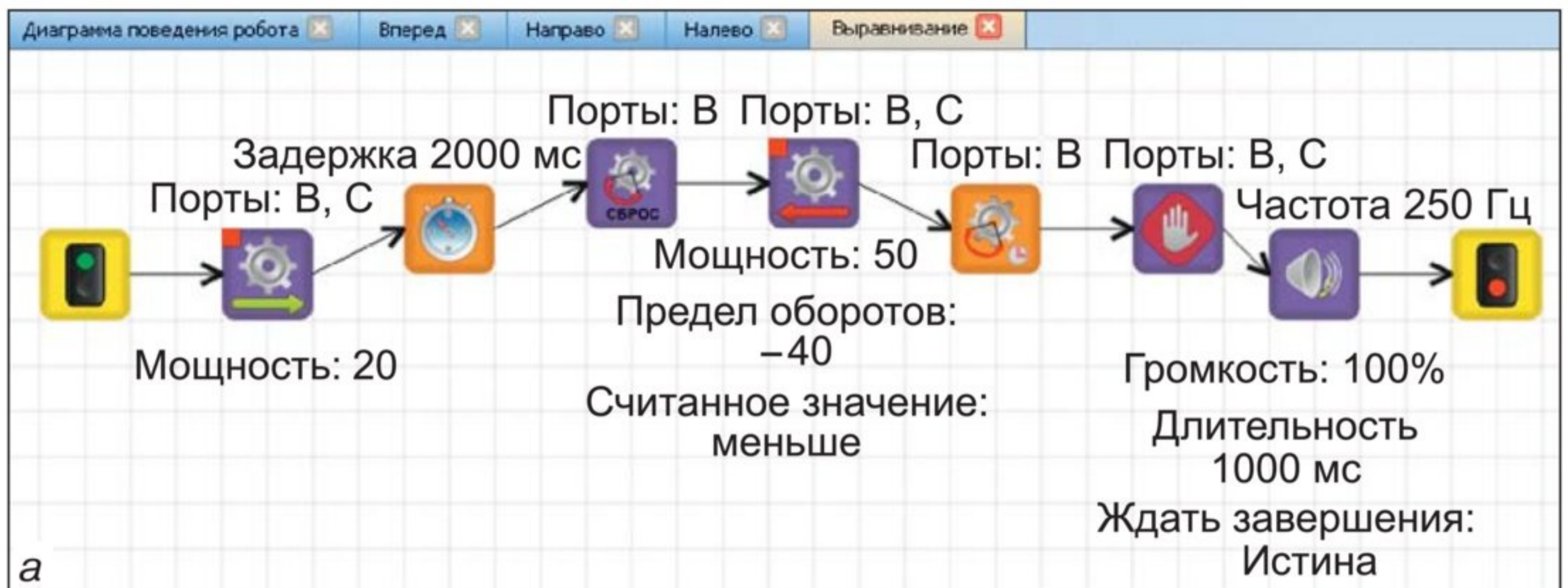


**Рис. 5.15.** Выравнивание робота по стене спереди

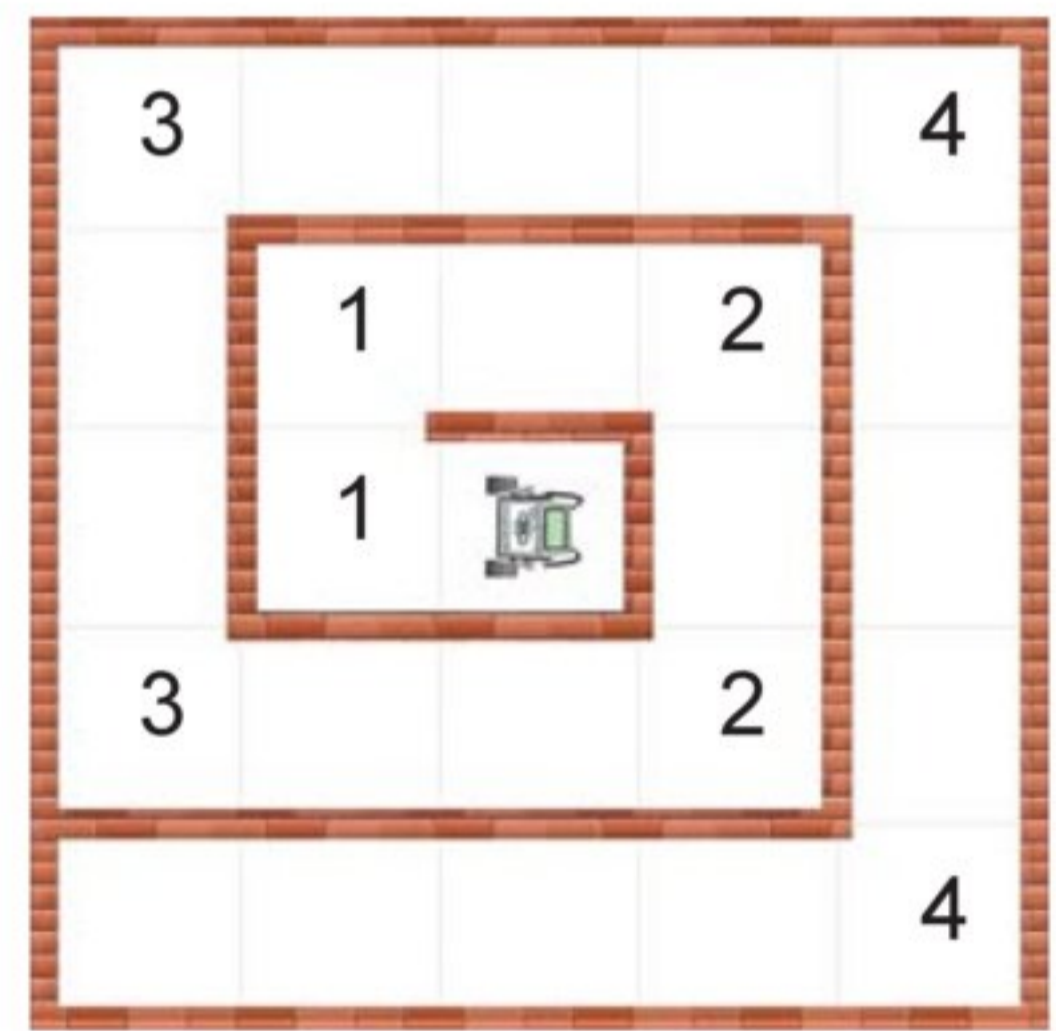
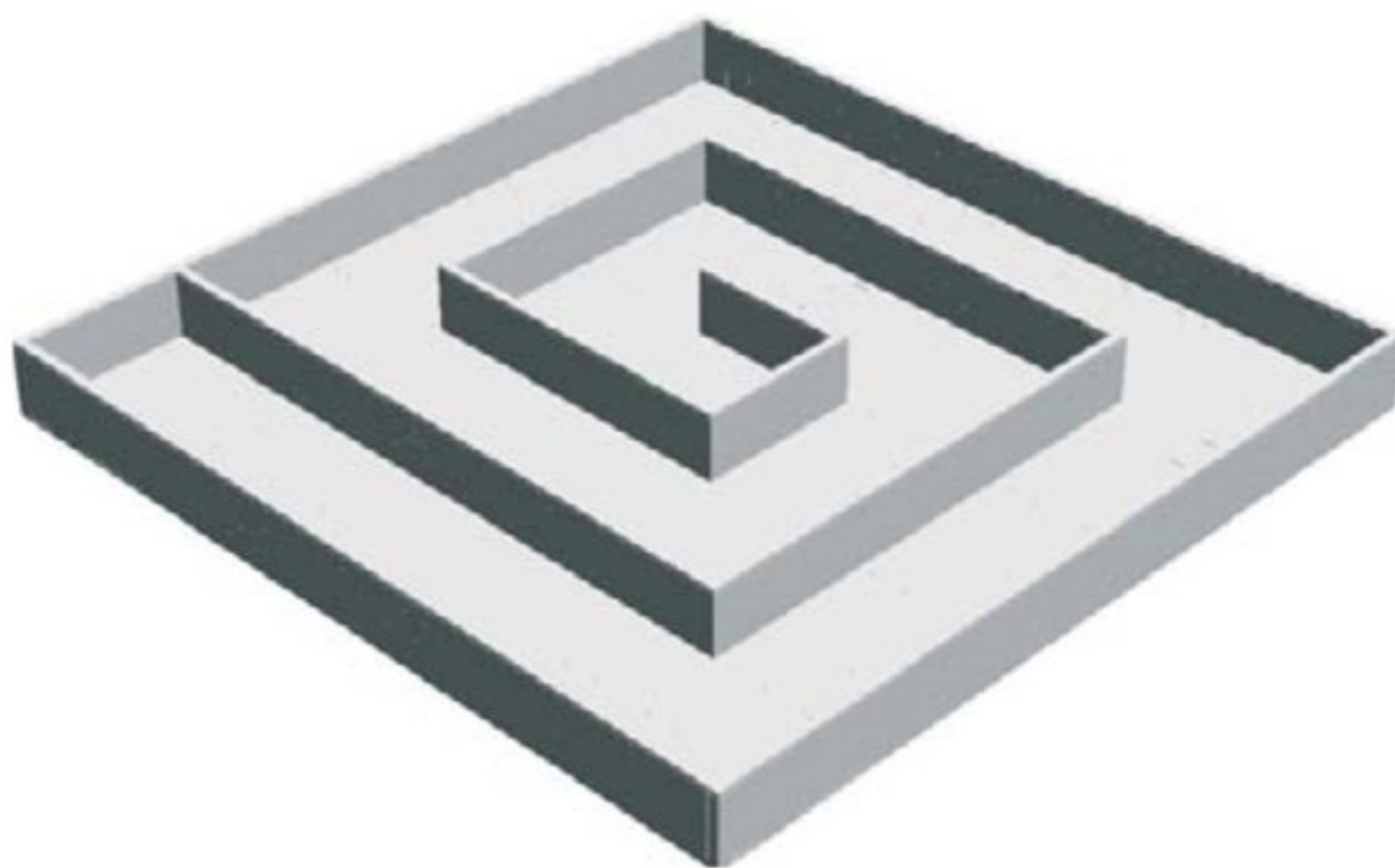
эту проблему можно решить конструкторскими методами. Например, установить на роботе усики с горизонтально расположенными колёсами, которые свободно вращаются при соприкосновении со стеной. Следует соблюсти правило: эти колёса должны быть размещены в передней части робота на достаточно большом расстоянии от ведущих колёс. Это обеспечит необходимое выравнивание при движении вдоль стены (рис. 5.14).

Другой способ выравнивания — сближение со стенами в тупиках. Для этого роботу надо заранее знать, что перед ним находится стена. Для выравнивания роботу потребуется широкий бампер, которым он будет упираться в стены в моменты сближения (рис. 5.15). Подпрограмма выравнивания может быть вызвана несколько раз в процессе выполнения программы, как правило, после подпрограммы проезда вперёд (рис. 5.16).

Следующей интересной задачей будет выход из спиралевидного лабиринта. Особенность его коридоров заключается в том, что через каждые два прохода длина коридора увеличивается на одну клетку. Значит, можно придумать алгоритм, по которому длина проезда робота будет рассчитываться автоматически (рис. 5.17).



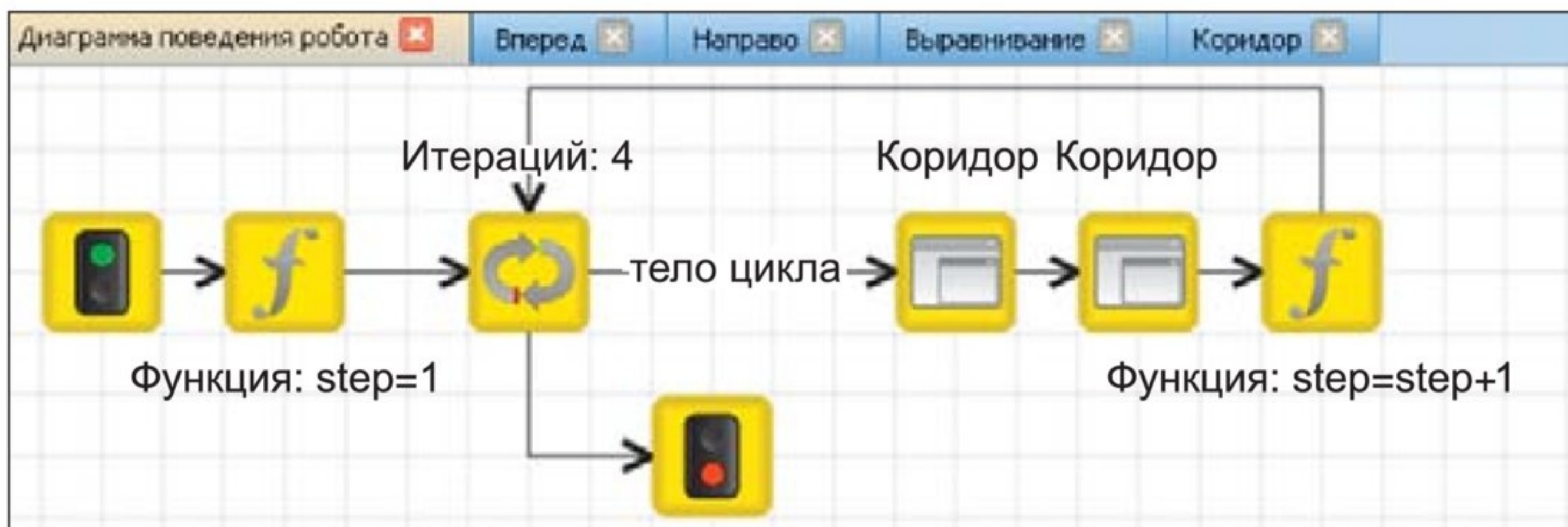
**Рис. 5.16.** Подпрограмма выравнивания (а) и её вызовы в основной программе (б)



**Рис. 5.17.** Спиралевидный лабиринт и количество шагов в каждом коридоре

Для подсчёта количества клеток на каждом этапе требуется переменная *step*. **Переменная** — это ячейка памяти с именем, значение которой можно изменить. На-





**Рис. 5.18.** Программа для прохождения лабиринта по спирали




**Рис. 5.19.** В подпрограмме «Коридор» используют значение переменной *step*

чальное значение переменной *step* будет 1 — робот проедет по коридору одну ячейку (рис. 5.18).

Подпрограмма «Коридор» будет вызываться дважды в цикле из 4 повторений (итераций). После каждого второго вызова значение переменной *step* увеличивается на 1. В подпрограмме «Коридор» используется значение этой переменной, чтобы определить, сколько раз надо вызвать подпрограмму «Вперёд» (рис. 5.19).





Выполняя эту программу, робот не проедет последний коридор. Добавьте команду в нужное место так, чтобы робот остановился в последней ячейке.



## Проверьте себя

1. Какие три команды робот должен уметь выполнять в лабиринте?
2. Как в лабиринте можно обойтись двумя командами?
3. Из каких простых действий состоит выполнение каждой команды?
4. Зачем нужны звуковые сигналы при отладке программы в лабиринте?
5. Какая польза от подпрограмм?
6. Какой механизм позволяет выравнивать робота при движении вдоль стены?
7. Какая конструкция и алгоритм позволяют выравнивать робота по стене перед ним?
8. Создайте программу, по которой робот должен добраться до центра спиралевидного лабиринта, начиная движение снаружи. Используйте подпрограммы «Вперёд», «Направо», «Налево».
9. Разработайте подпрограмму для выравнивания по стене за «спиной» робота. Соберите бампер, благодаря которому гусеницы или колёса робота не наезжают на стену.



**Запомните** ♦ Подпрограмма ♦ Переменная

## § 5.3. Правило правой руки

Всегда можно найти выход из незнакомого лабиринта, если этот выход существует. В мифе о Тесее древнегреческий герой использовал нить из клубка, полученного им от царевны Ариадны, по которой нашёл выход из лабиринта Минотавра (рис. 5.20). Роботу, чтобы обойти незнакомый лабиринт, потребуются два датчика расстояния и правило правой руки (рис. 5.21). Суть правила: если человек попал в незнакомый лабиринт, то, двигаясь так, чтобы правая рука всегда касалась стены, он сможет найти выход. Как же сформулировать это правило для робота?



**Рис. 5.20.** Развалины лабиринта Минотавра на острове Крит



**Рис. 5.21.** Робот с двумя датчиками расстояния

Для описания правила правой руки можно воспользоваться языком, похожим на язык исполнителя «Робот» из среды программирования «Исполнители»<sup>1</sup>.

```
ПОВТОРЯЙ
{
  ЕСЛИ (справа_свободно)
  {
    направо;
    вперёд;
  }
  ИНАЧЕ
  ЕСЛИ (впереди_свободно)
  вперёд;
  ИНАЧЕ
  налево;
}
```

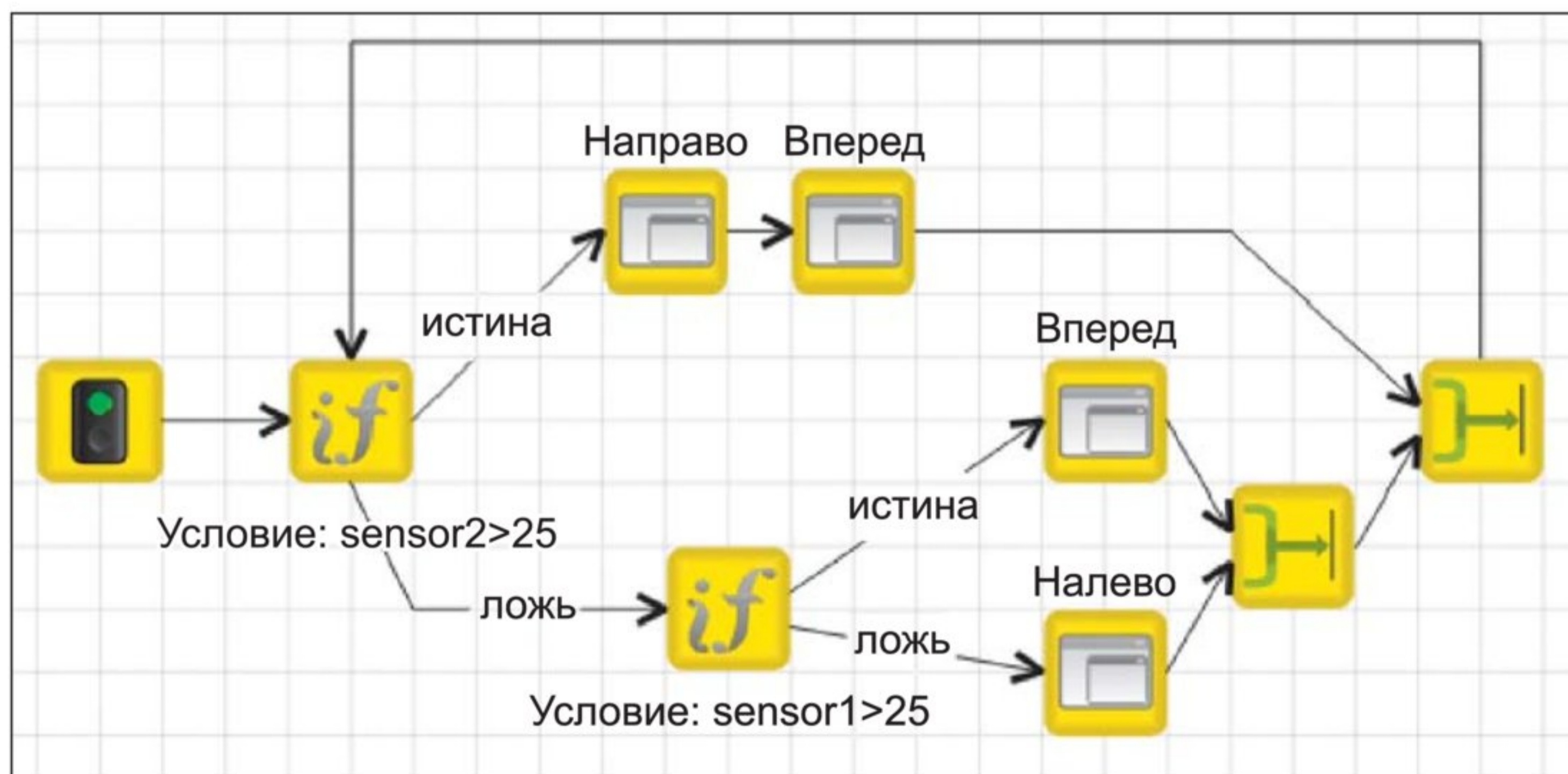
<sup>1</sup> Среду «Исполнители» можно бесплатно загрузить с сайта <http://kpolyakov.spb.ru>

Фигурные скобки означают, что все команды, находящиеся между ними, относятся к одному смысловому блоку. Обнаружив проход справа, робот должен не только повернуться, но и заехать в него. При отсутствии прохода справа робот проверяет наличие прохода впереди и движется в него, а при его отсутствии поворачивает налево.

В соответствии с алгоритмом обхода лабиринта по правилу правой руки составьте программу в среде TRIK Studio. Используйте готовые подпрограммы «Вперёд», «Направо» и «Налево» (рис. 5.22).

При построении программы считаем, что передний датчик подключён на первый порт, а боковой справа — на второй. Если расстояние до стены больше 25 см, считаем, что перед датчиком стены нет и есть как минимум одна свободная ячейка помимо той, на которой стоит робот.

Конечно, двигаясь по правилу правой руки, робот проходит гораздо больший путь. Он «заглядывает» во все тупики. В отличие от человека, он не может посмотреть на лабиринт сверху и точно определить, куда нужно ехать. Да и любой из нас, попав в тёмные катакомбы без карты, будет действовать точно так же.



**Рис. 5.22.** Программа для обхода лабиринта по «правилу правой руки»

Если существует выход из лабиринта, расположенный вдоль внешней стены, то по правилу правой руки робот найдёт его. Однако нет гарантий, что все действия будут выполнены идеально и он не зацепится за какой-нибудь неудобный угол или выступ. Чтобы застраховаться от подобных неудач, необходимо вновь научить робота при движении выравниваться по стенам.

Для определения наличия стены перед роботом в незнакомом лабиринте можно использовать датчик расстояния в передней части робота (см. рис. 5.15). Обнаружив стену перед собой, робот может медленно подъехать к ней до упора по времени и затем отъехать на стандартное расстояние по энкодерам. Обратите внимание, что подъезд к стене осуществляется именно по времени, а не по энкодерам: при взаимодействии со стеной моторы могут оказаться заблокированными и датчики оборотов не сработают.

Подпрограмма выравнивания может быть добавлена в конце подпрограмм «Вперёд» и «Налево», поскольку при повороте направо по правилу правой руки робот должен проехать вперёд одну ячейку. Кроме того, после поворота налево можно добавить выравнивание по стене сзади (рис. 5.23 и 5.24).

Когда проход лабиринта отлажен, можно заняться увеличением скорости. Для этого в первую очередь потребуются сократить все временные задержки.

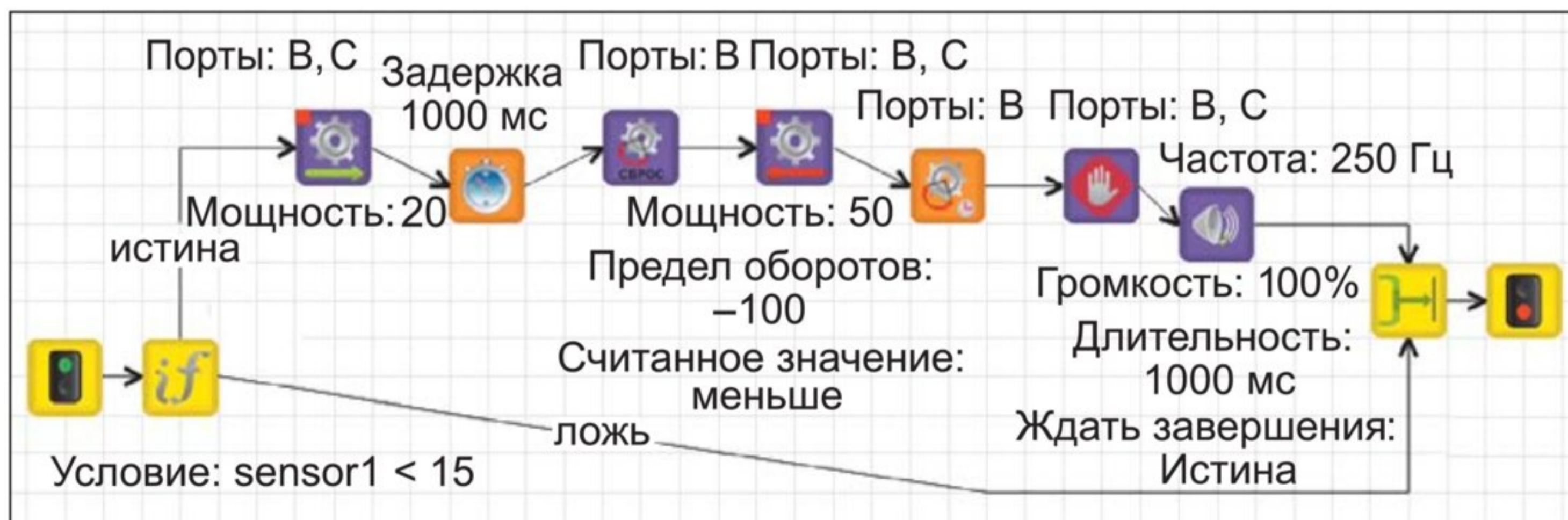


Рис. 5.23. Подпрограмма выравнивания с проверкой наличия стены

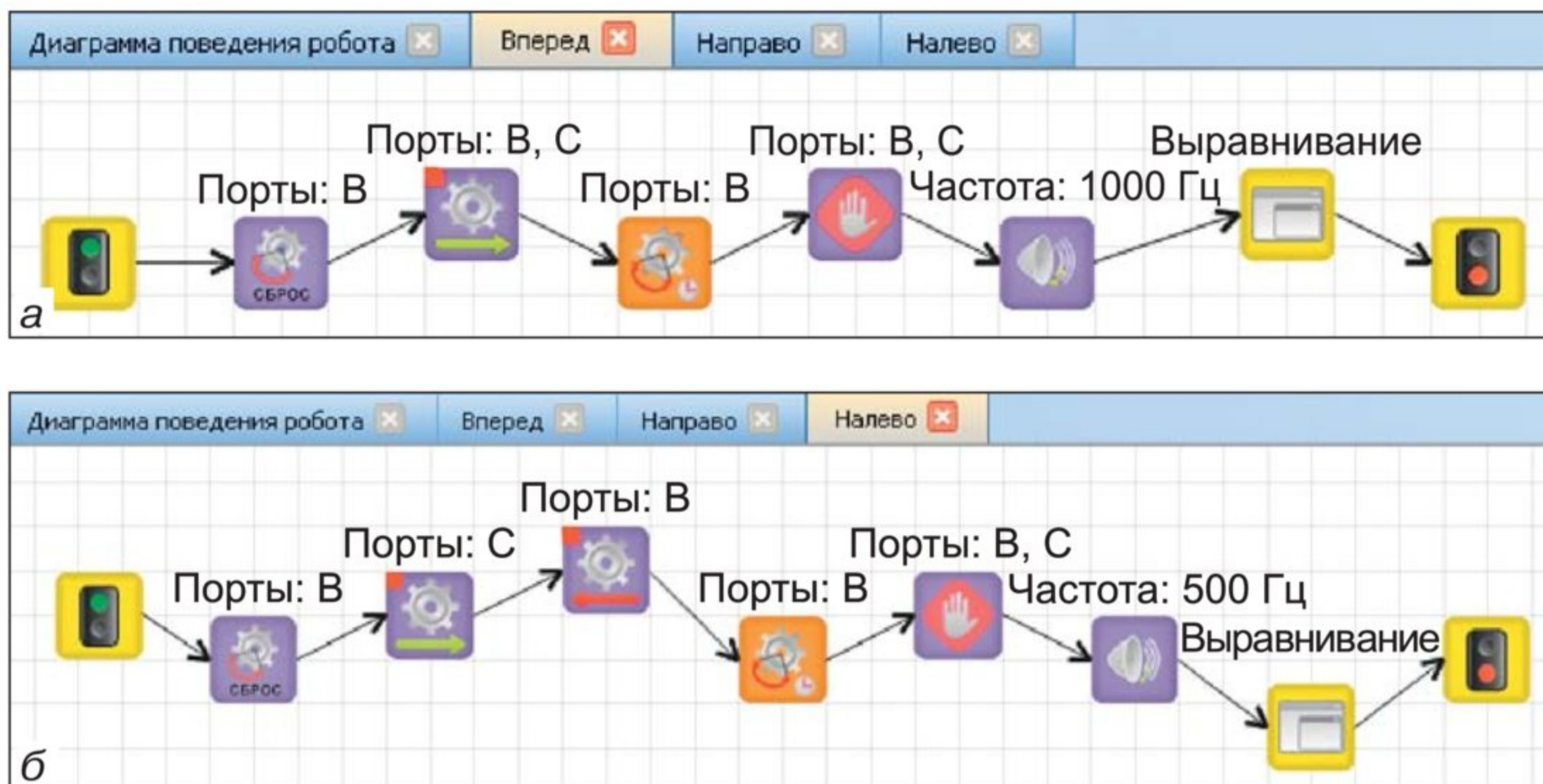


Рис. 5.24. а—б — вызовы подпрограммы выравнивания

Усовершенствуйте все подпрограммы так, чтобы остановка со звуковым сигналом производилась только при отсутствии стены перед датчиком расстояния либо после выравнивания.


Во всём мире уже не одно десятилетие проводят среди школьников и студентов соревнования по робототехнике с прохождением лабиринта. Как правило, заранее лабиринт неизвестен, но роботу даётся возможность исследовать его, чтобы найти кратчайший маршрут. В будущем вы узнаете, как с применением сложных алгоритмов можно научить робота решать эту задачу.

### Проверьте себя

1. Попробуйте сформулировать правило левой руки. В каких ситуациях оно может быть применено?
2. Приведите пример лабиринта, который нельзя обойти полностью по правилу правой руки.
3. Приведите пример лабиринта и стартового положения робота, при котором он никогда не сможет найти выход по правилу правой руки.

4. В конце какой подпрограммы можно безошибочно добавить выравнивание по стене сзади за роботом?
5. Измените конструкцию и программу так, чтобы робот проходил лабиринт по правилу левой руки.
6. Замените передний датчик расстояния на датчик касания и пройдите лабиринт по правилу правой руки.
7. Постройте робота только с одним датчиком расстояния и двумя моторами и составьте программу, по которой он найдёт выход из лабиринта по правилу правой руки.

### Это интересно!

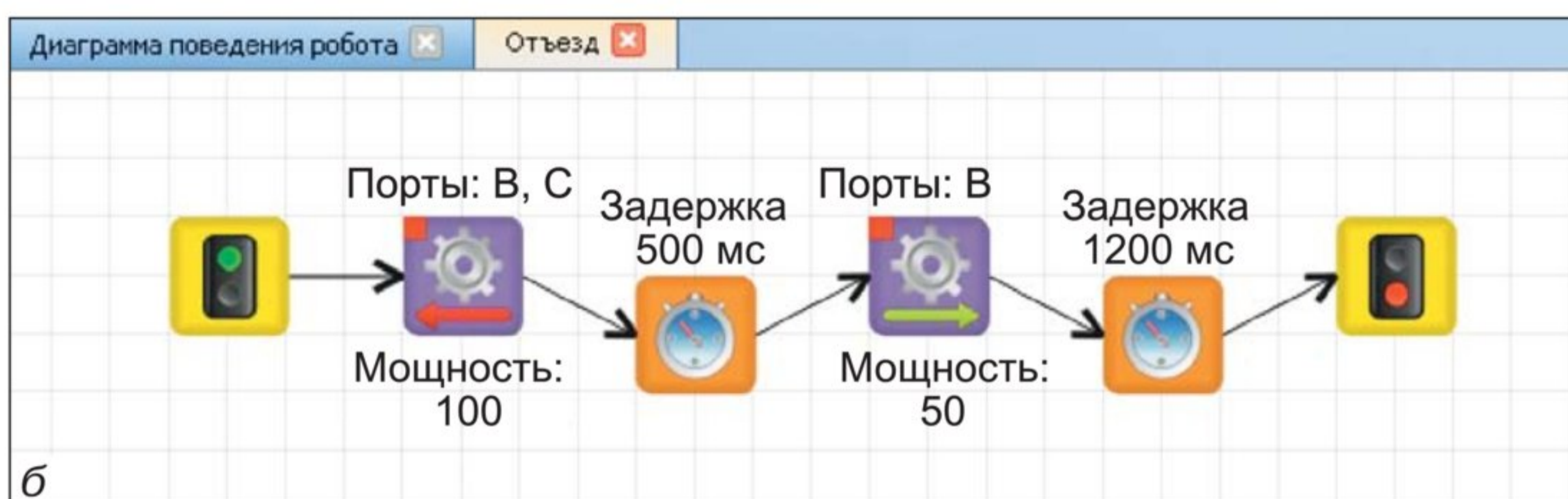
 В 1952 году американский учёный и математик *Клод Шеннон* разработал обучаемую механическую мышь, которая могла находить выход из лабиринта. Многие идеи Шеннона легли в основу современной информатики и робототехники.



## § 5.4. Защита от застреваний

На примере лабиринта вы увидели, что робот часто попадает в затруднительные или, как говорят, нештатные ситуации. Но далеко не всегда в алгоритме предусмотрена соответствующая защита. Однако каждый робот должен заботиться о своей безопасности. Поэтому необходимо предусмотреть возможность автоматической реакции на непредвиденные обстоятельства.

В 3 главе вы строили робота, который путешествовал по комнате. Это простая задача, и повторить её не составляет труда: в цикле робот движется вперёд, пока не увидит препятствие, затем отъезжает назад с поворотом (рис. 5.25). Для краткости поместите отъезд назад с поворотом в подпрограмму.



**Рис. 5.25.** Программа для путешествия по комнате (а) и подпрограмма «Отъезд» (б)

Но что произойдёт, если робот перестанет реагировать на появление препятствия и застрянет у ножки стула или на домашних тапочках (рис. 5.26)? Следует предусмотреть такую ситуацию в алгоритме движения. В этом деле помогут параллельные задачи.

**Параллельные задачи** — это независимые процессы, которые выполняются одновременно на одном контроллере, разделяя между собой его ресурсы. То есть, программа робота может содержать несколько частей (задач), которые будут запущены параллельно, независимо друг от друга. До сих пор в наших примерах все части программы выполнялись последовательно, одна за другой. При параллельном программировании, как правило, есть основная задача, а команды остальных задач выполняются в моменты появления задержек в основной. Поэтому в парал-

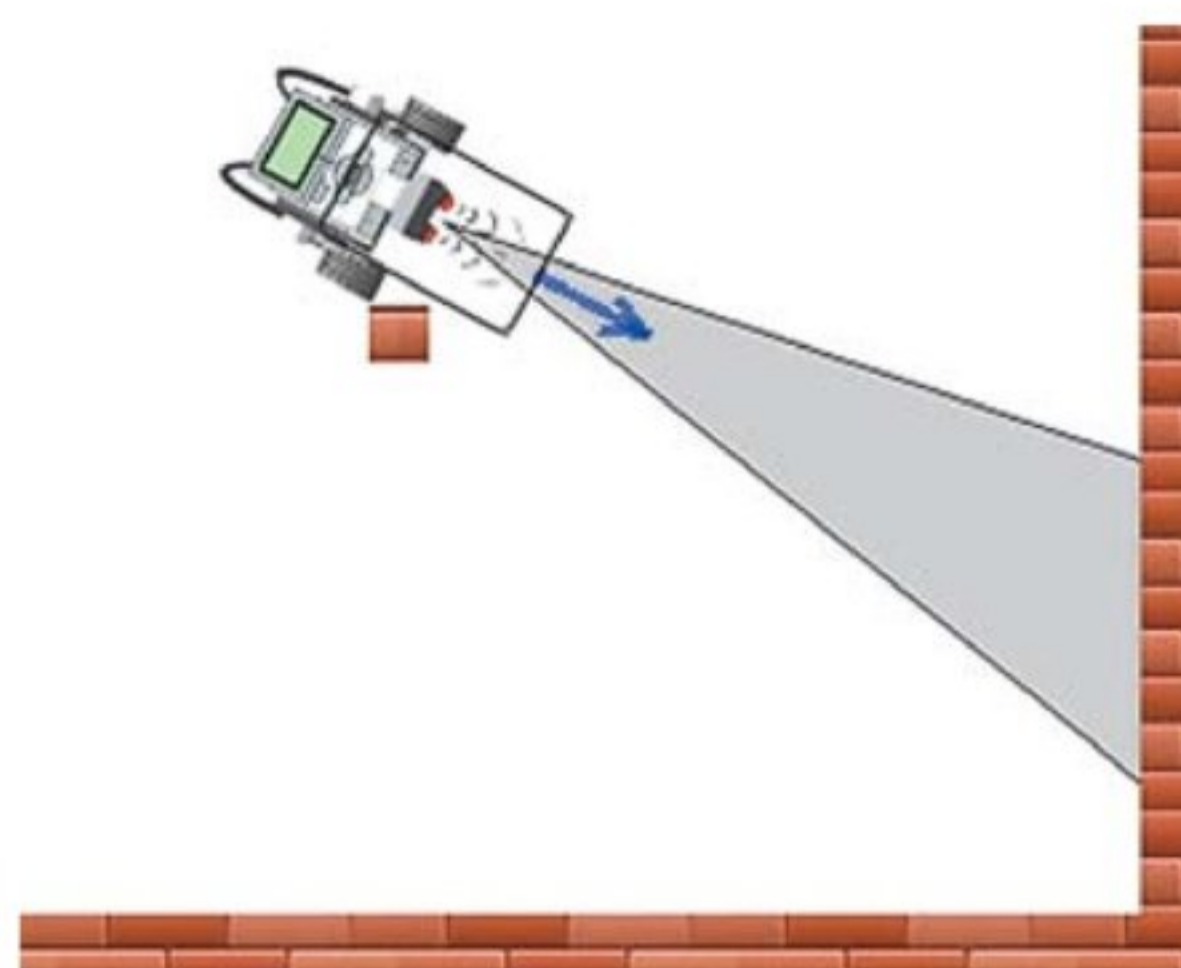
лельных задачах для робота не следует делать циклы без задержек.

Простейший пример параллельных действий — пение и танец одновременно. Или рисование одной рукой треугольника, а другой — квадрата. Это непростое упражнение для человека. Для робота оно не составит труда.

Для реализации защиты от застреваний можно использовать сторожевой таймер. Эта функция напоминает диспетчера охранной компании, который каждый час принимает дежурный звонок от сторожа на охраняемом объекте. Если сторож не позвонил через час, диспетчер поднимает тревогу и на объект немедленно выезжает отряд быстрого реагирования. В нашем случае отсутствие звонка от сторожа — это отсутствие реакции датчика расстояния на какой-либо объект. А вместо отряда быстрого реагирования будет вызываться подпрограмма «Отъезд». Но для начала надо научиться засекаать время. Пусть не час, а хотя бы 10 секунд.

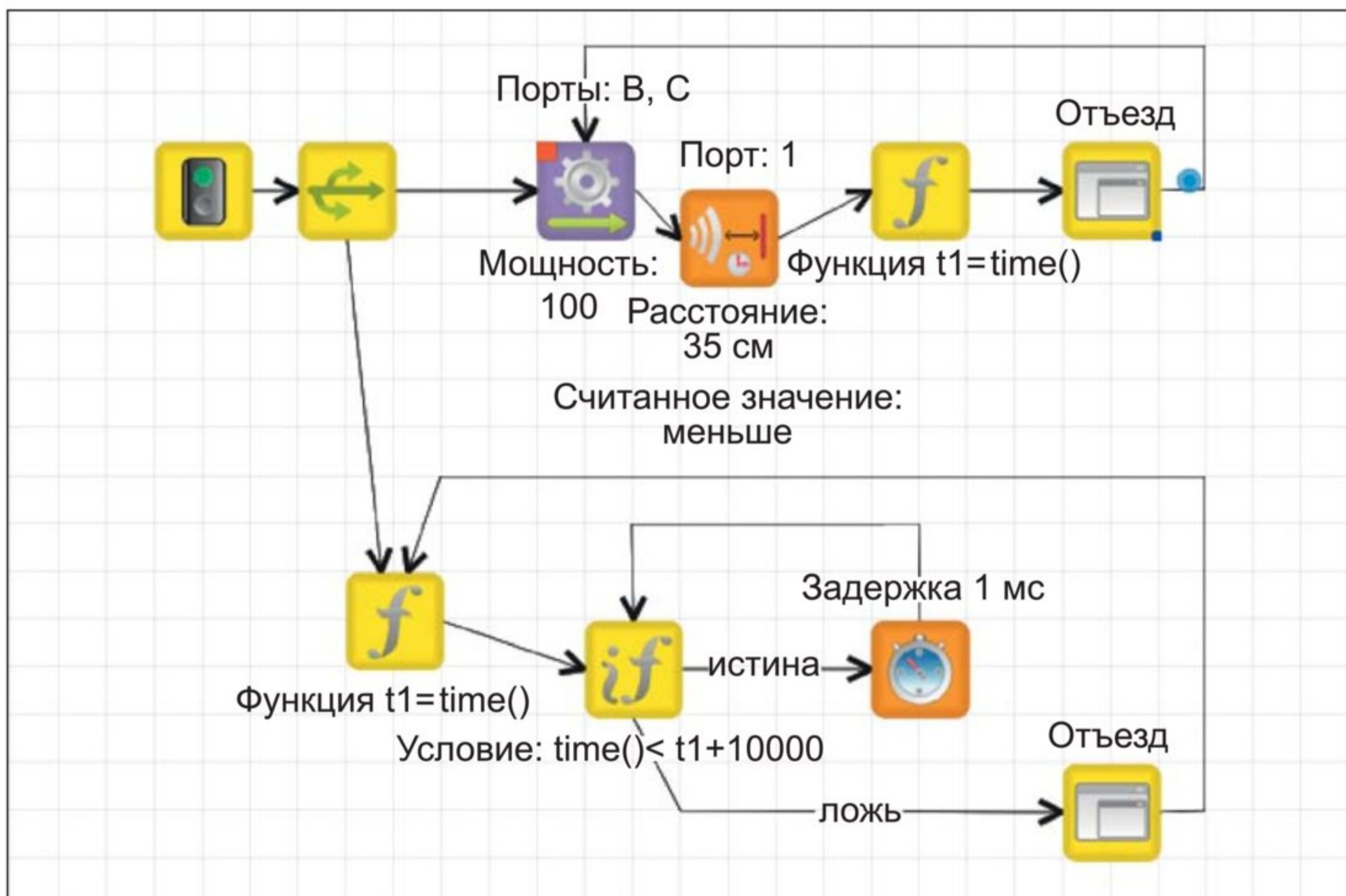
Внутри включённого контроллера всегда работают часы, которые позволяют отмерять время в миллисекундах, например, с момента запуска программы. За это отвечает функция *time()*, которую можно использовать для контроля реакции на объекты. Функция — это подпрограмма, которая может возвращать какое-либо значение. Каждый раз, встречая объект, робот записывает в переменную *t1* значение времени с начала работы программы.

То же самое происходит и в начале параллельной задачи, но следующим действием робот ожидает, когда пройдёт 10 секунд, во время которых ни разу не срабатывает обновление переменной *t1*, и только затем вызывает подпрограмму «Отъезд», если этого не произошло (рис. 5.27).



**Рис. 5.26.** Робот может застрять на препятствии, которое не заметит датчик расстояния



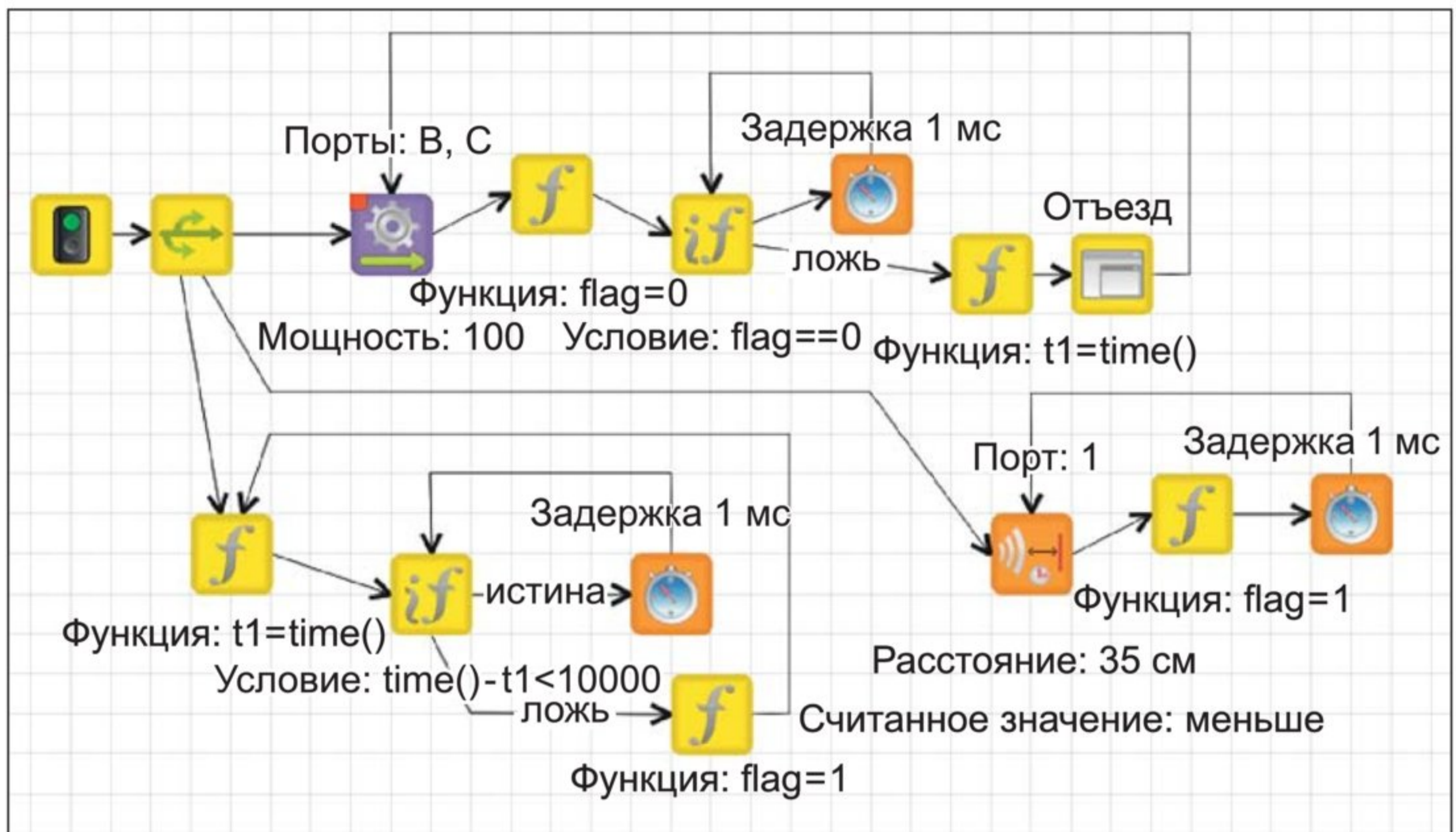


**Рис. 5.27.** Программа с защитой от застреваний

Эта программа будет работать, но возможна странная ситуация. Представьте себе, что сработала защита от застреваний, робот начал выполнять отъезд и вдруг увидел объект. В основной задаче сразу будет вызвана подпрограмма «Отъезд», которая и так уже работает. В этом случае возможна коллизия — наложение двух противоречащих друг другу команд на управление моторами. Робот может начать вести себя неадекватно. Как это предотвратить?

Достаточно ввести переменную-флаг, с помощью которой будет обеспечен вызов подпрограммы «Отъезд» либо при появлении объекта на пути, либо при срабатывании сторожевого таймера (рис. 5.28).

Исходя из полученного опыта сформулируем правило: из параллельных задач нельзя обращаться к одним и тем же устройствам робота (моторам и датчикам). Это равно-



**Рис. 5.28.** Программа с защитой от застреваний без коллизий


сильно тому, чтобы дать команду одной рукой рисовать и треугольник, и квадрат одновременно. В результате не получится ни того, ни другого.

## Проверьте себя

1. Как выполняются параллельные задачи на одном контроллере?
2. Зачем нужен сторожевой таймер?
3. По какой причине может произойти застревание робота?
4. Почему нельзя обращаться к одному устройству из параллельных задач?
5. Добавьте к конструкции робота бампер с датчиком касания, а в программу — обработку сигнала нажатия на кнопку датчика с вызовом подпрограммы «Отъезд».
6. Усовершенствуйте программу для поиска выхода из лабиринта, добавив в неё защиту от застреваний.


**Запомните** ♦ Параллельные задачи

### Это интересно!

 В современных контроллерах аппаратная система «сторожевой таймер» используется для защиты от зависаний. Таймер периодически сбрасывается контролирующей системой. Если сброс не произошёл, происходит перезагрузка системы и программа выполняется сначала.

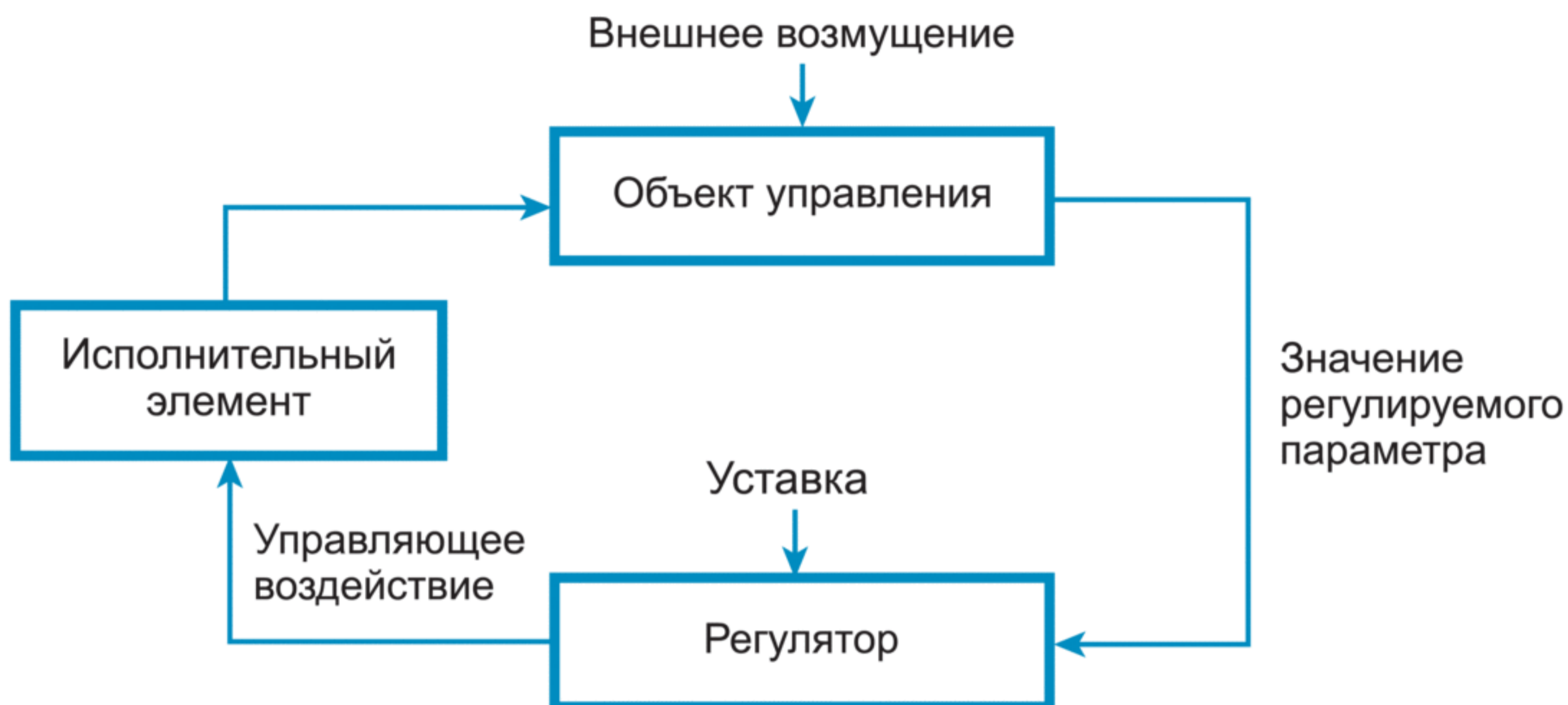
## § 5.5. Простейшие регуляторы

Одной из важнейших задач, решаемых в робототехнике, является управление поведением системы. Это относится ко всем устройствам — от мобильного робота до космического манипулятора. Правильное поведение предполагает не только точное исполнение всех команд, но и нужную реакцию на различные отклонения и помехи, вызываемые внешними причинами. Для управления системами при возникающих отклонениях разработан целый класс устройств, называемых регуляторами.

 **Регулятор** обеспечивает желаемое поведение системы, следит за функционированием системы, постоянно анализирует её состояние и вырабатывает определённое управляющее воздействие. В состав регулятора входят чувствительный элемент (датчик), исполнительный элемент (мотор) и контроллер или другое устройство, реализующее порядок (алгоритм) управления (рис. 5.29). Благодаря тому, что регулятор в виде обратной связи постоянно получает информацию об изменении состояния системы, становится возможным обеспечить качественное управление. Понятия «регулирование» и «управление» при этом будем считать синонимами.

При построении регуляторов всегда необходимо определить цель управления. В технике применяют специальный термин «уставка», который показывает желаемое значение регулируемого параметра системы.

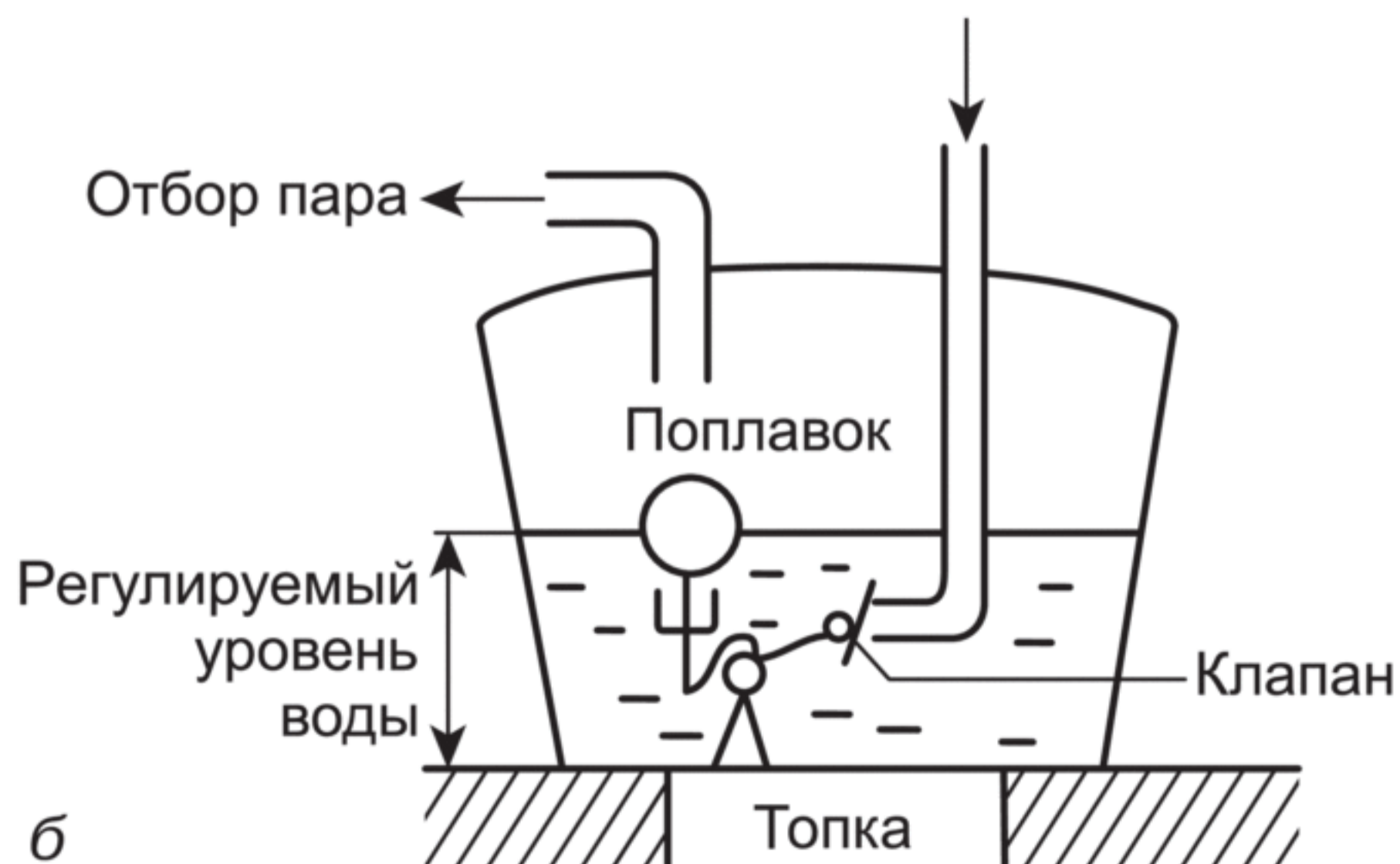
Термин «регулятор» пришёл из теории автоматического управления. Существуют релейные, пропорциональные,



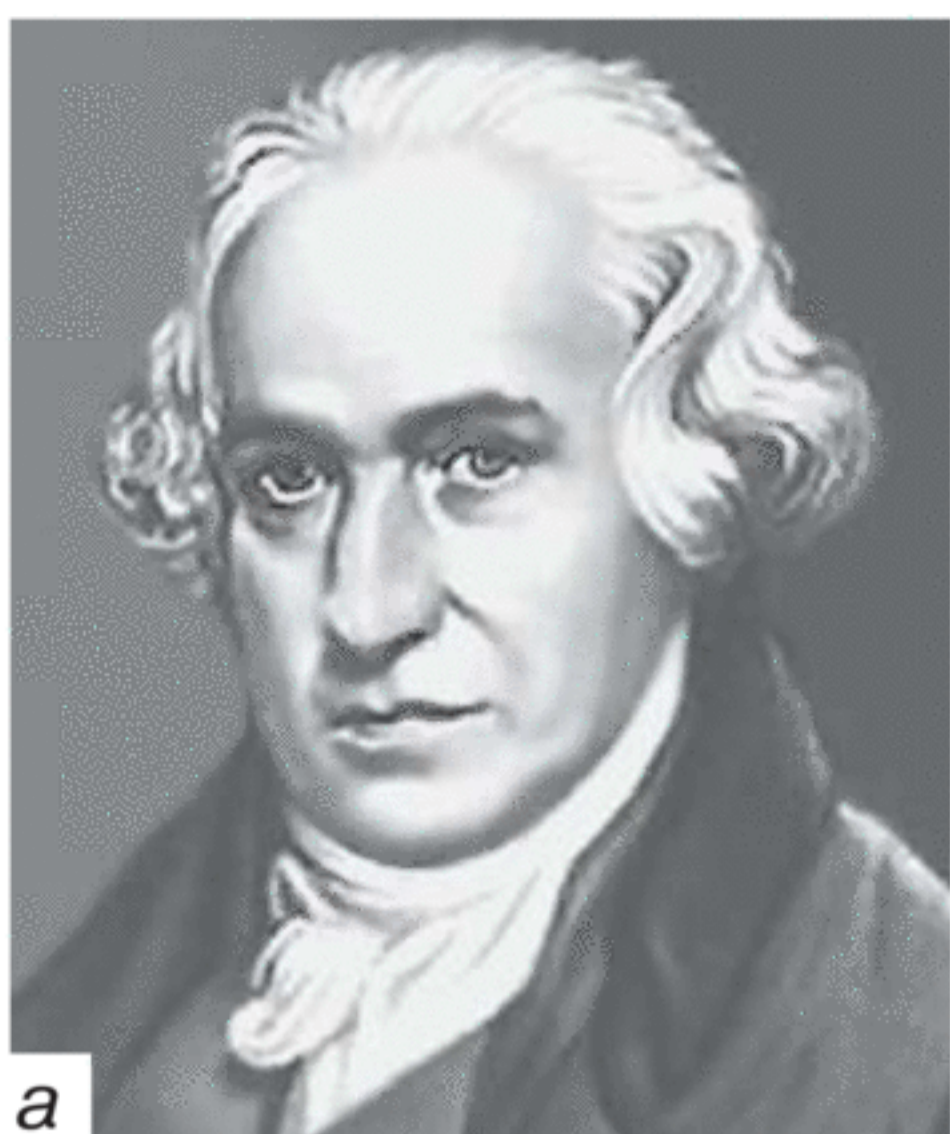
**Рис. 5.29.** Обобщённая схема регулятора

дифференциальные, интегральные регуляторы и их различные сочетания. Определённые регуляторы действуют не только в технике, но и в различных областях человеческой деятельности, например, в экономике и политике.

Первый регулятор промышленного назначения был разработан и внедрён известным русским инженером *И.И.Ползуновым* ещё в 1765 году. Это поплавковый регулятор уровня воды в барабане парового котла (рис. 5.30).



**Рис. 5.30.** И. И. Ползунов (а) и его поплавковый регулятор (б)



а



б

**Рис. 5.31.** Джеймс Уатт (а) и его центробежный регулятор (б)

Поплавок на шарнире, опускаясь вместе с уровнем воды в баке, открывает клапан для подачи воды. Изобретение регулятора для парового котла положило начало автоматизации котельных установок. А в те времена в мире уже работали сотни тысяч паровых машин, обеспечивая промышленное производство. Интересно отметить, что поплавковый регулятор до сих пор широко используется в конструкциях сливных бачков для унитазов.

Другой, не менее распространённый регулятор для паровых машин, изобрёл шотландский инженер *Джеймс Уатт* в 1788 году. Это устройство, контролирующее скорость вращения вала паровой машины (рис. 5.31). Это было важное изобретение, поскольку из-за потери контроля над скоростью вращения валов на производствах часто гибли рабочие.

Регуляторы повсеместно встречаются в повседневной жизни. Электрический обогреватель имеет регулятор температуры, мобильный телефон — регулятор яркости экрана. Летящий самолёт поддерживают в стабильном состоянии сразу несколько регуляторов. Можно назвать много других примеров. Давайте попробуем разобраться, как работает регулятор, и построим свой собственный.

Рассмотрим задачу, связанную с *управлением углом поворота мотора*: необходимо поднять балку на оси в поло-

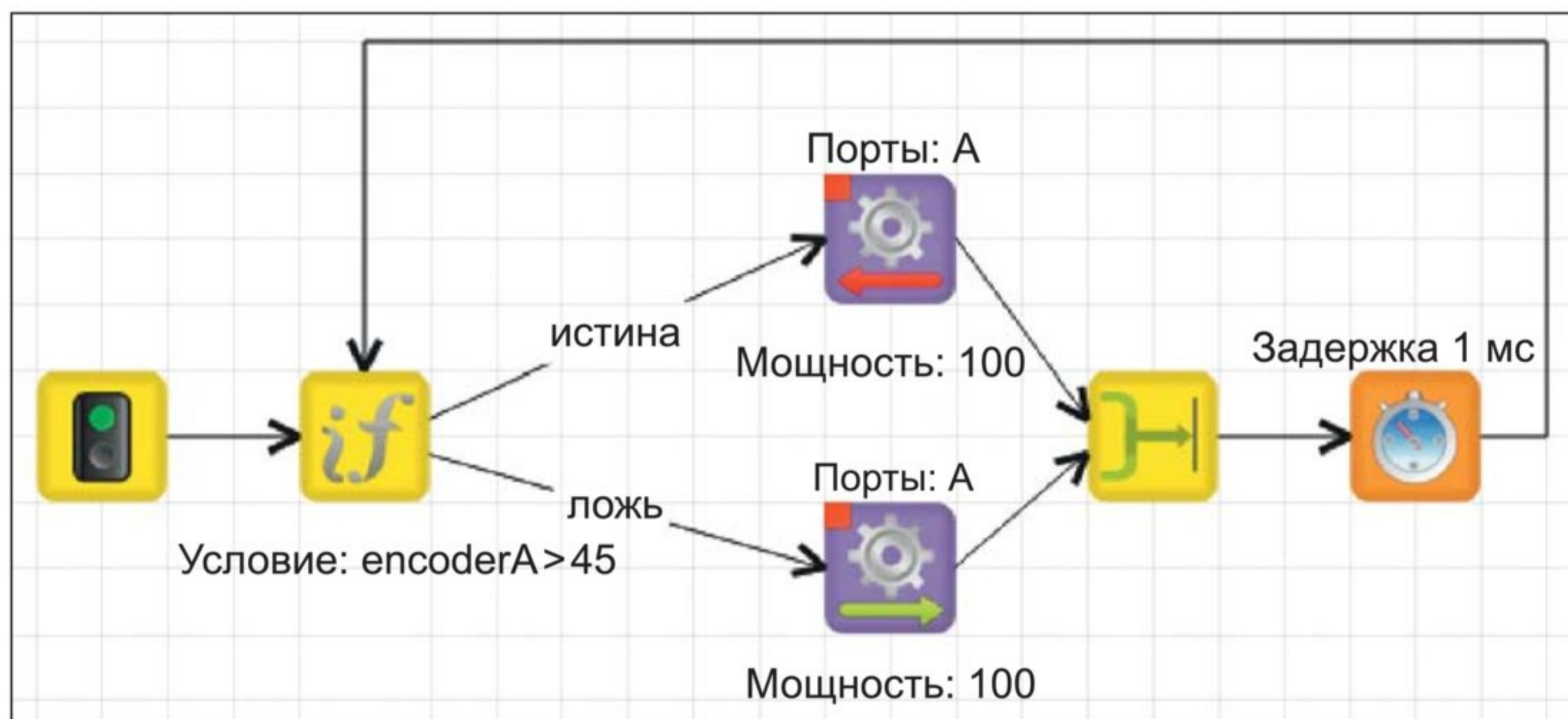
жение  $45^\circ$  и зафиксировать его (рис. 5.32). При этом, в случае возможных отклонений, мотор должен всегда возвращать рычаг в указанное положение.

Решение представляется очевидным: если показания датчика оборотов меньше 45, надо направить мотор вперёд, в других случаях — направить мотор назад. Проверку энкодера следует проводить с высокой частотой, например, 1000 раз в секунду. Таким образом, мы рассматриваем всего два состояния датчика:  $encoderA > 45$  и  $encoderA \leq 45$  — и два состояния мотора: «полный вперёд» и «полный назад». Такой регулятор называется *релейным двухпозиционным*.

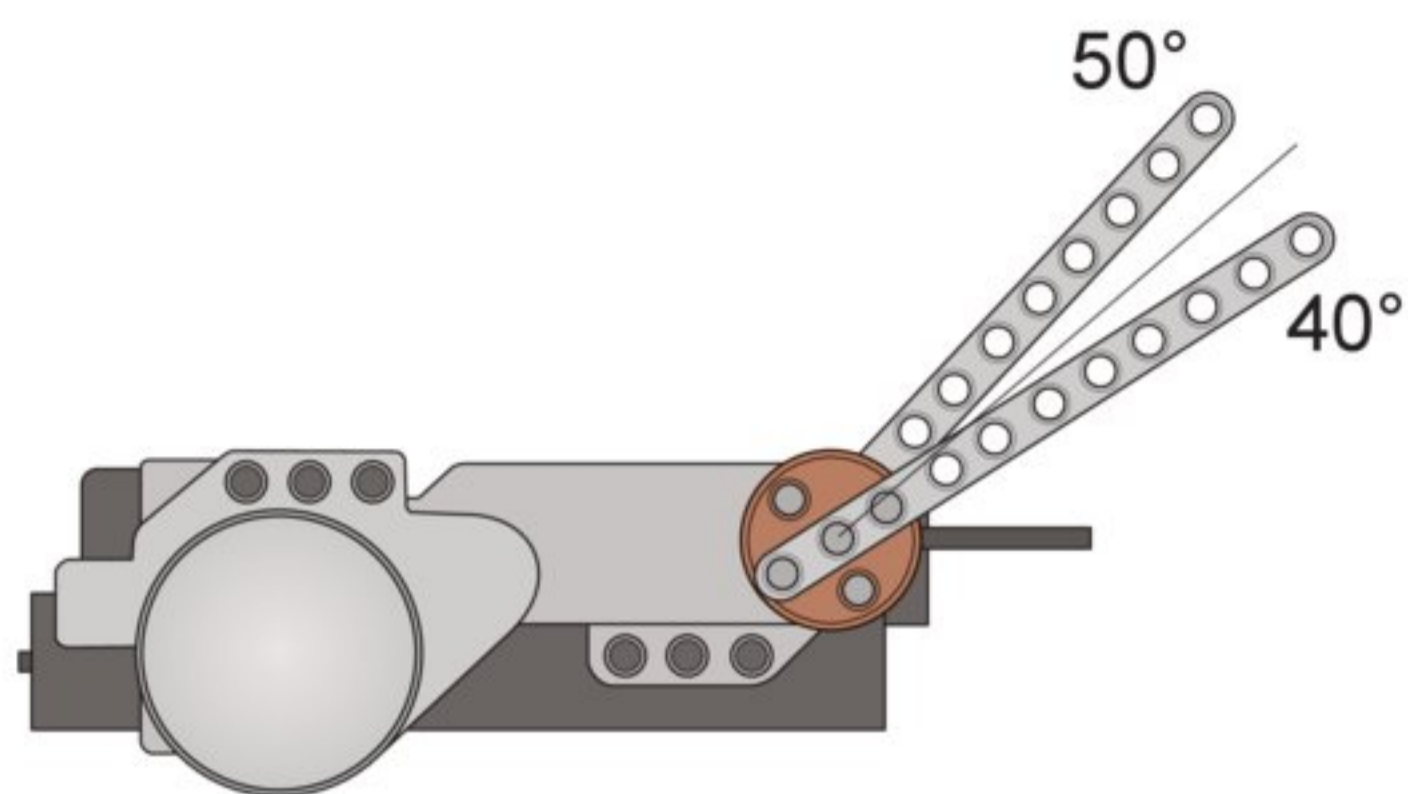
Релейный регулятор изменяет управляющее воздействие скачком при переходе через пороговое значение (рис. 5.33).



**Рис. 5.32.** Задача: зафиксировать мотор в положении  $45^\circ$



**Рис. 5.33.** Программа релейного двухпозиционного регулятора для управления положением мотора

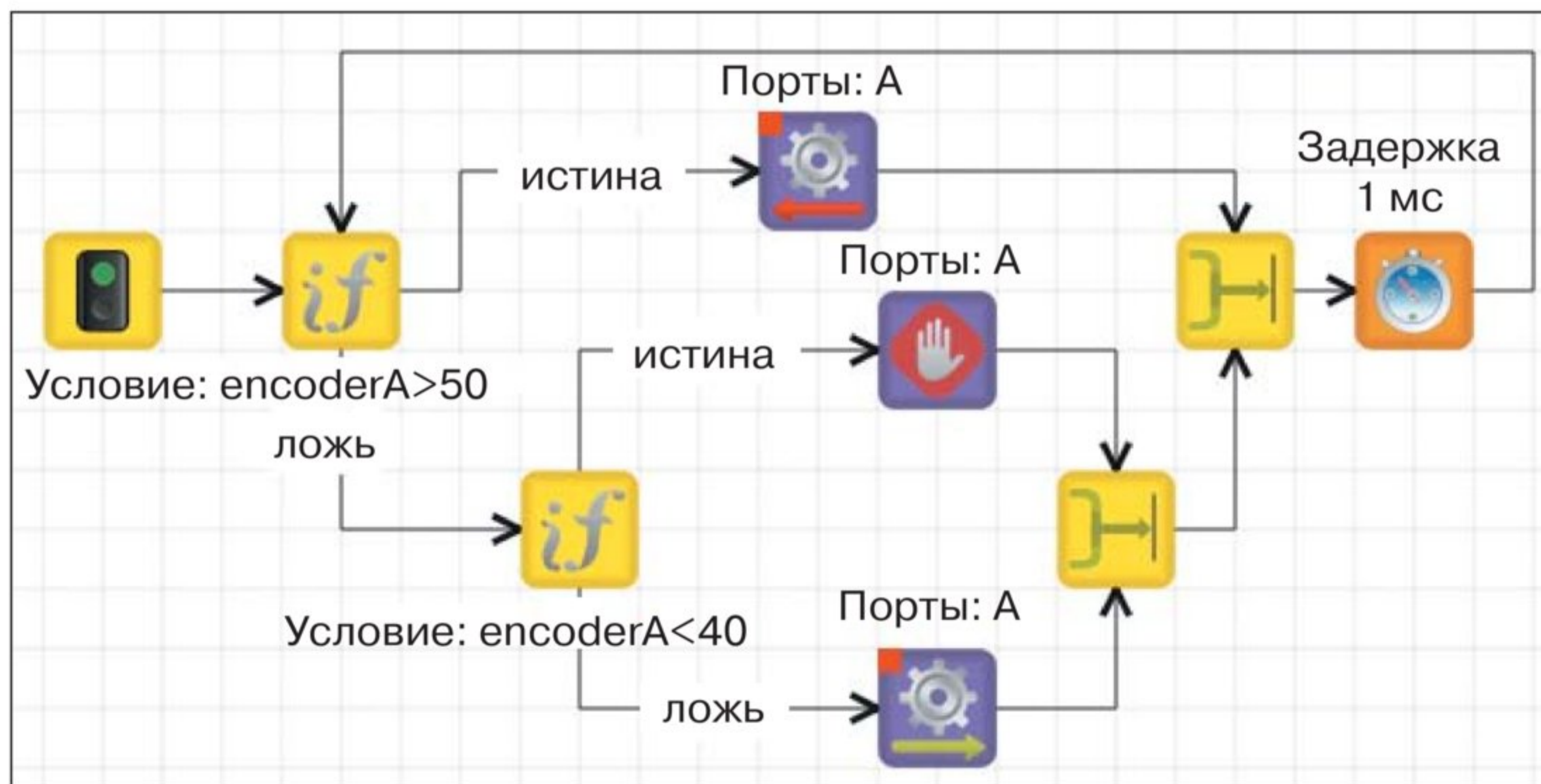


**Рис. 5.34.** Диапазон для остановки мотора

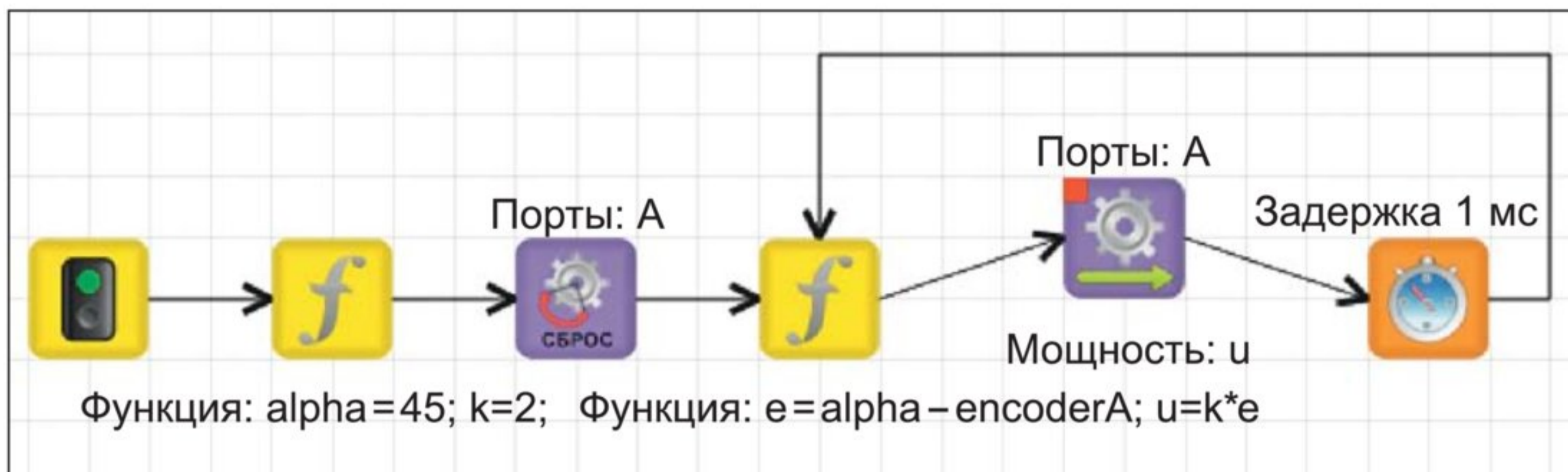
При запуске регулятора в работу можно увидеть, что скачкообразное изменение скорости моторов приводит к возникновению относительных колебаний вокруг желаемого значения. Такие колебания называются *перерегулированием*. Можно сократить перерегулирование, если понизить скорость мотора, однако это ослабит систему в целом. Таким образом, с помощью двухпозиционного регулятора практически невозможно зафиксировать мотор в заданном положении. Возможно, с этим лучше справится трёхпозиционный регулятор?

Установим окрестности угла  $45^\circ$  в диапазоне  $\pm 5^\circ$ . Пусть мотор будет останавливаться в моменты, когда энкодер попадает в диапазон от  $40^\circ$  до  $50^\circ$ . Это и будет «третья позиция» для регулятора (рис. 5.34 и 5.35).

Установим окрестности угла  $45^\circ$  в диапазоне  $\pm 5^\circ$ . Пусть мотор будет останавливаться в моменты, когда энкодер попадает в диапазон от  $40^\circ$  до  $50^\circ$ . Это и будет «третья позиция» для регулятора (рис. 5.34 и 5.35).



**Рис. 5.35.** Программа для трёхпозиционного релейного регулятора



**Рис. 5.36.** Программа для пропорционального регулятора

Однако при запуске такого алгоритма скорее всего не удастся зафиксировать мотор в заданном диапазоне. Можно понизить скорость, расширить диапазон, но всё это полумеры. Как же быть? Оказывается, существует более эффективное решение, которое позволит зафиксировать мотор точно в заданном положении. Это достигается применением *пропорционального регулятора* (рис. 5.36).

Пропорциональный регулятор изменяет управляющее воздействие пропорционально отклонению системы от желаемого состояния. Запомним новые термины.

**Уставка** ( $x_0$ ) — желаемое значение регулируемой величины, цель управления.

**Невязка**, отклонение, динамическая ошибка ( $e$ , *error*) — разность между уставкой и текущим значением регулируемой величины ( $x$ )

$$e = x_0 - x.$$

**Коэффициент усиления регулятора** ( $k$ ) — множитель, который увеличивает малые значения отклонения регулируемой величины до ощутимого управляющего воздействия ( $u$ ).

$$u = k \cdot e.$$

Под управлением пропорционального регулятора мотор работает совсем неинтересно: быстро принимает желаемое положение и даже не пытается создать колебания. Это вызвано тем, что регулятор хорошо настроен. Измените







значение уставки  $\alpha$  на другое значение (например, 180, 360 или  $-180$ ) и посмотрите, как будет себя вести регулятор.

Попробуйте изменить коэффициент усиления: сделайте его очень маленьким, например, 0,1 или очень большим, например, 100. Что изменилось? Найдите оптимальный коэффициент.

Вы увидели одно из замечательных изобретений человечества в действии. Пропорциональный регулятор (П-регулятор) является одним из самых простых и распространённых в технике. И вам предстоит ещё не раз им воспользоваться.



## Проверьте себя

1. Из каких элементов состоит регулятор?
2. Чем механический регулятор отличается от регулятора, работающего под управлением контроллера?
3. Почему релейный двухпозиционный регулятор так называется?
4. В каком случае имеет смысл строить многопозиционный регулятор?
5. По какой причине возникает перерегулирование?
6. Что такое уставка?
7. Назовите синонимы к термину «невязка».
8. Для чего нужен коэффициент усиления?
9. Постройте систему управления положением шлагбаума: когда появляется объект, шлагбаум открывается, когда объект отъезжает, шлагбаум закрывается. Для управления положением шлагбаума используйте П-регулятор в параллельной задаче.
10. Постройте регулятор, который обеспечит подъём и удержание подвешенного груза на балке, закреплённой на моторе. Подберите коэффициент, при котором балка не будет значительно отклоняться от заданного положения.
11. Постройте систему управления поворотами единственного датчика расстояния на работе в лабиринте с помощью П-регулятора и параллельной задачи.



**Запомните** ♦ Регулятор ♦ Уставка ♦ Невязка  
♦ Коэффициент усиления регулятора

## § 5.6. Следование по линии. Релейный и пропорциональный регуляторы

Задача следования по линии — одна из самых популярных в робототехнике. Существует множество алгоритмов и систем управления для обеспечения автоматического движения транспортного средства по заданной траектории. В самолётах, поездах и некоторых автомобилях уже используются автопилоты, которые позволяют полностью или частично заменить человека (рис. 5.37).

Разумеется, замена человека в условиях дороги — это сложная задача, требующая применения искусственного интеллекта. Будем решать её на доступном уровне постепенно, шаг за шагом.

**Релейный двухпозиционный регулятор.** Решаем задачу: робот должен двигаться по границе чёрного и белого, используя один датчик освещённости. Чтобы робот не сбивался с пути, подготовим трассу в виде достаточно широкой линии, например, 50 мм.

Построим двухмоторную тележку с датчиком освещённости, направленным вниз и закреплённым на высоте 10 мм. Диапазон чувствительности разных датчиков может колебаться от 3 до 50 мм. Следует учитывать возможные неровности полигона.

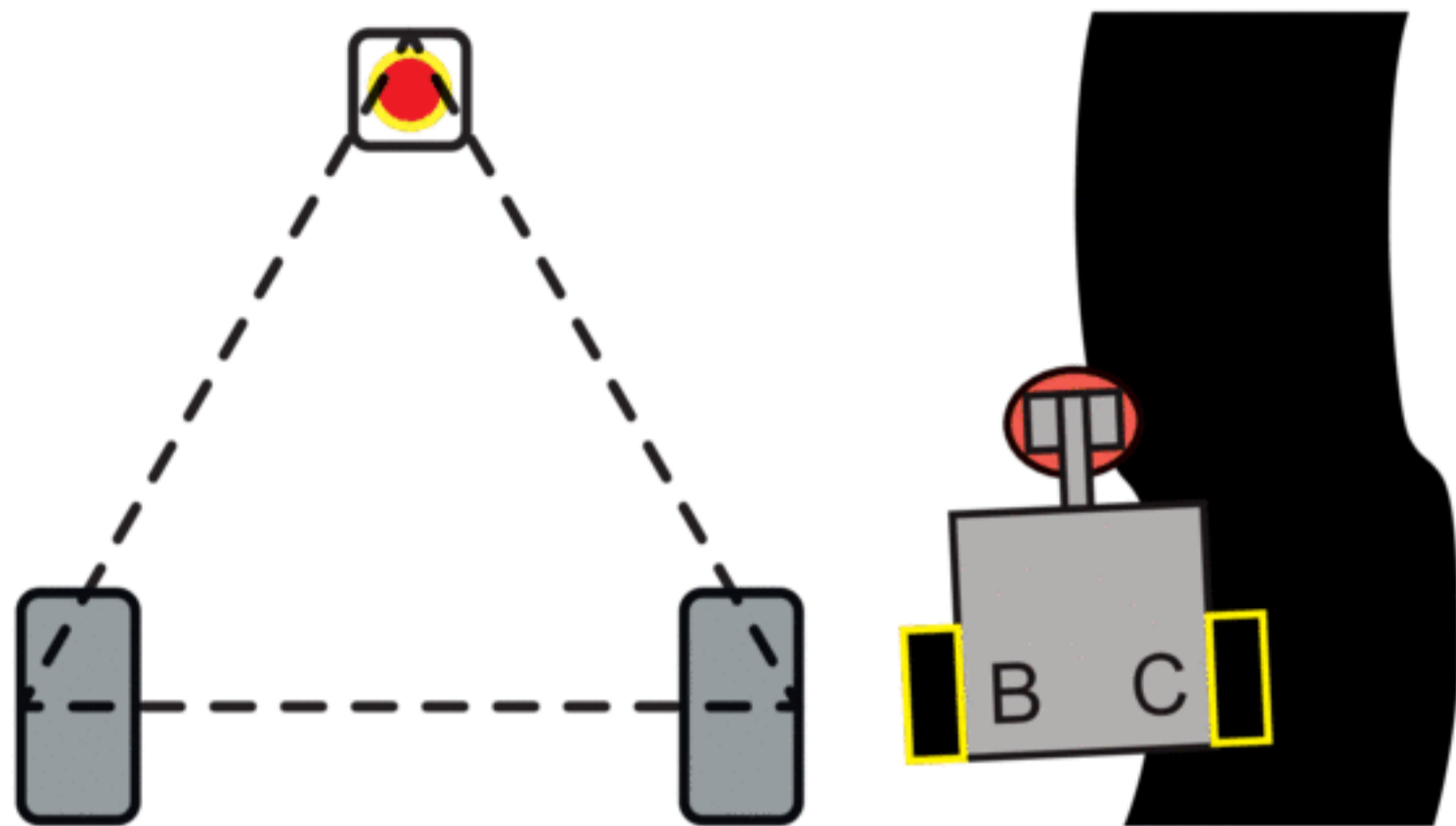
Лучше всего, если точки опоры колёс и датчик образуют равносторонний треугольник — это оптимальная конструкция для решения задачи (рис. 5.38 и 5.39).

Для построения алгоритма необходимо определить показания датчика освещённости на чёрном и белом поле. Используйте для этого режим просмотра показаний датчиков, который установлен

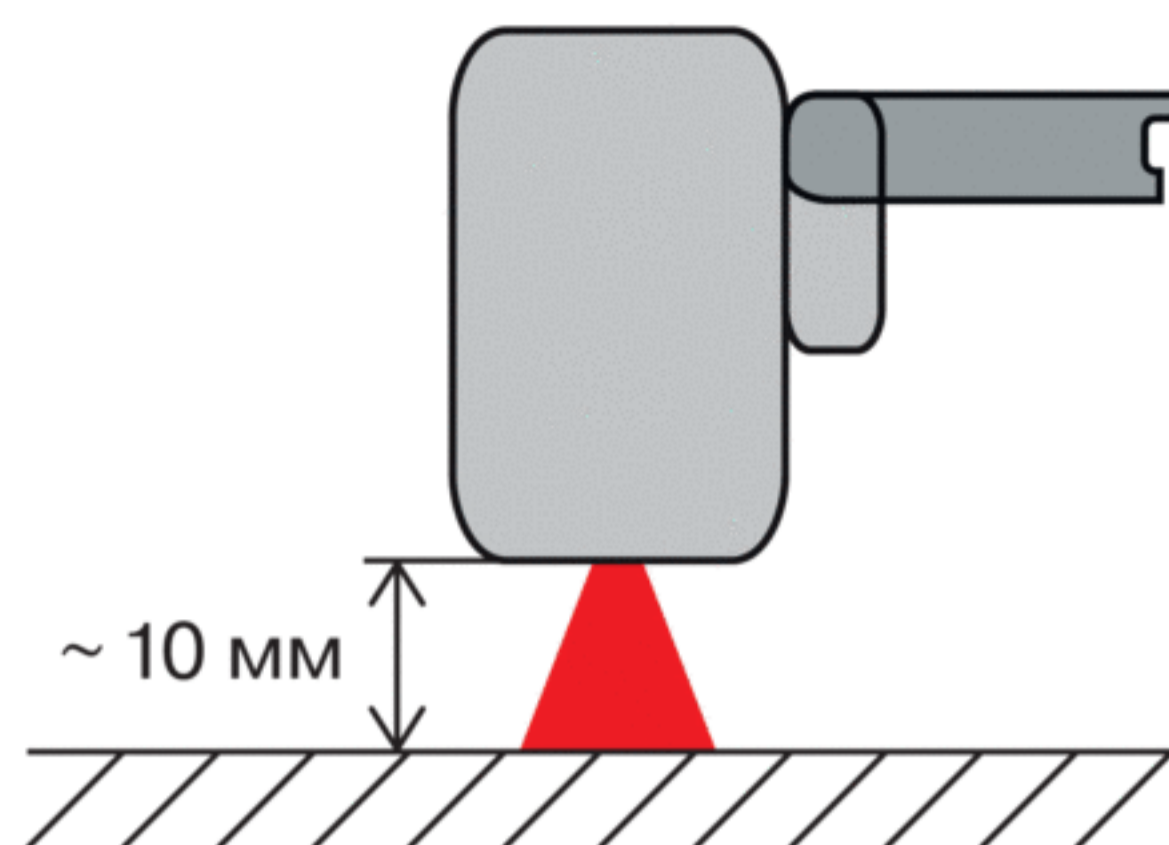


Рис. 5.37. Использование автопилота в автомобиле





**Рис. 5.38.** Задача следования по границе чёрного и белого



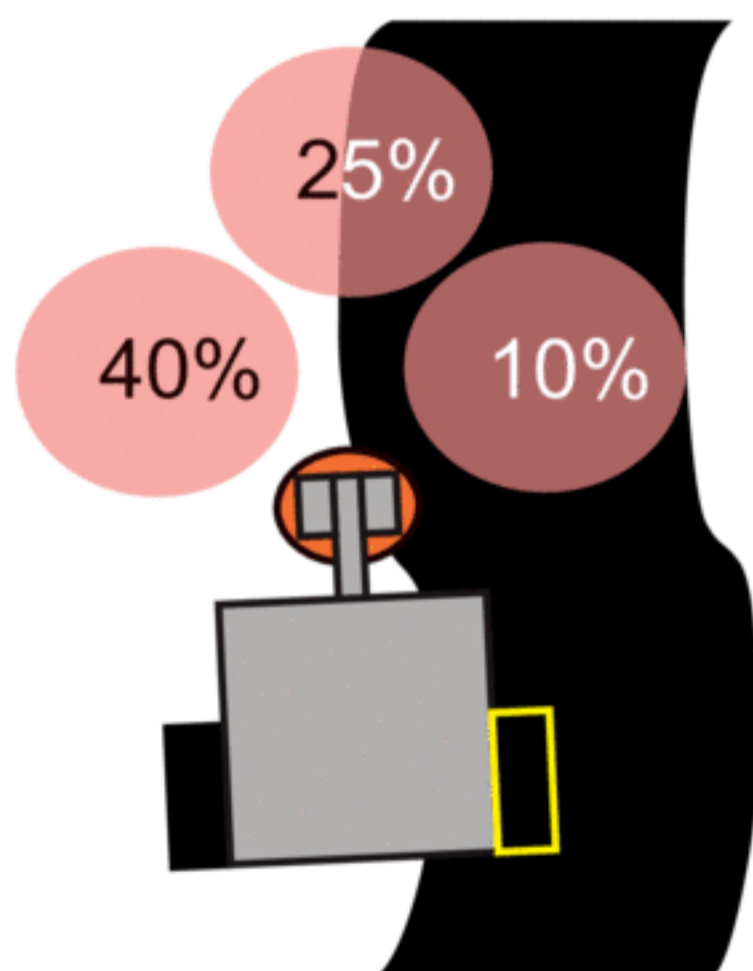
**Рис. 5.39.** Схема компоновки робота и высоты расположения датчика

в вашем контроллере. Среднее арифметическое полученных чисел — это пороговое значение для релейного регулятора. Допустим, на белом поле датчик показывает 40% освещённости, а на чёрном — 10%. Тогда на условном сером цвете, который расположен на границе, датчик покажет  $\frac{10 + 40}{2} = 25\%$ . Оценка пороговых показаний датчика называется *калибровкой* (рис. 5.40).

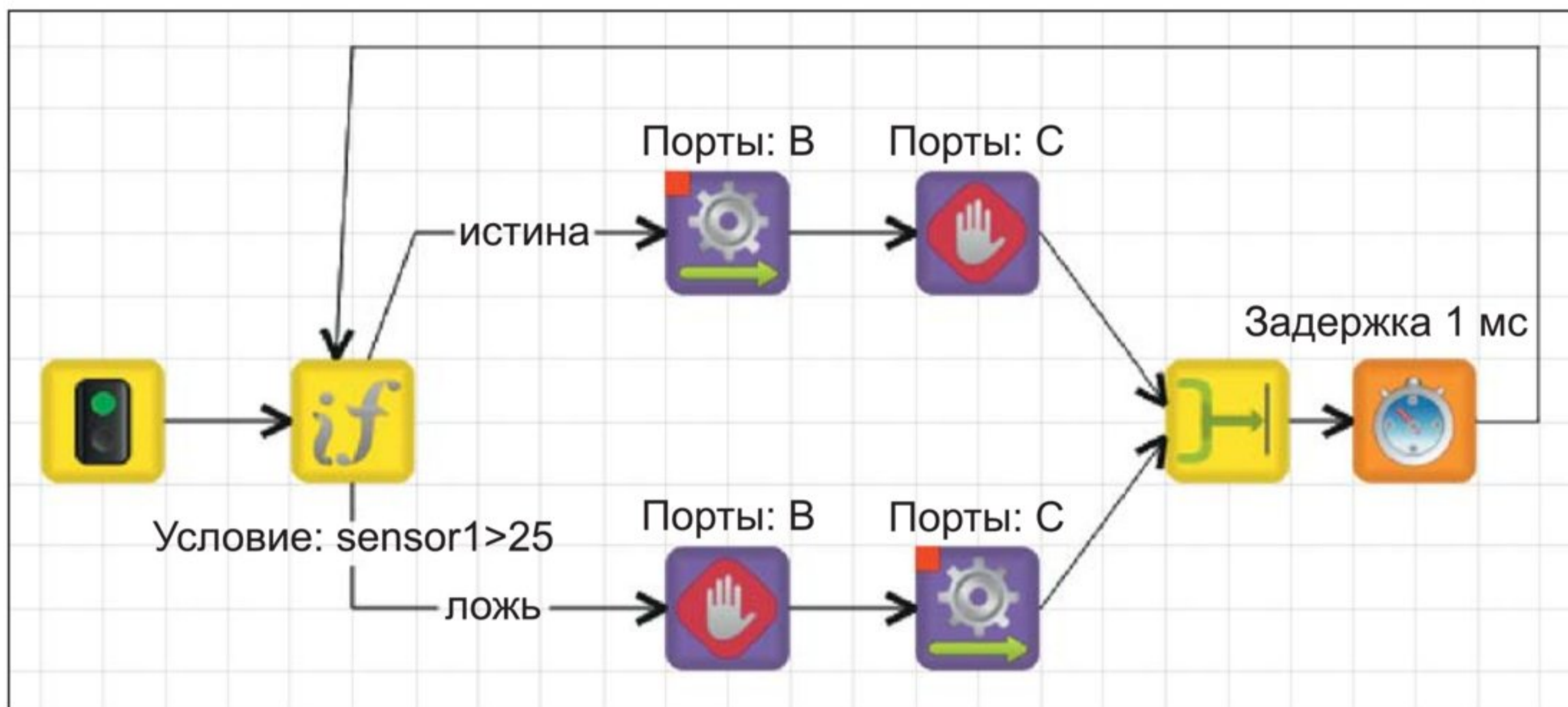
$$grey = \frac{white + black}{2}.$$

Обратите внимание, что показания датчика освещённости необходимо считывать, когда робот полностью стоит на поле в таком положении, в котором он будет ехать. Если немного поднять или опустить датчик, показания могут сильно измениться.

Найденное значение освещённости на границе используется в программе движения (рис. 5.41). Если показания датчика выше порогового значения 25%, то робот заехал на белое поле, если ниже — на чёрное. Останавливая один из моторов, робот плавно переезжает на другое поле.



**Рис. 5.40.** Калибровка датчика освещённости по белому и чёрному цветам

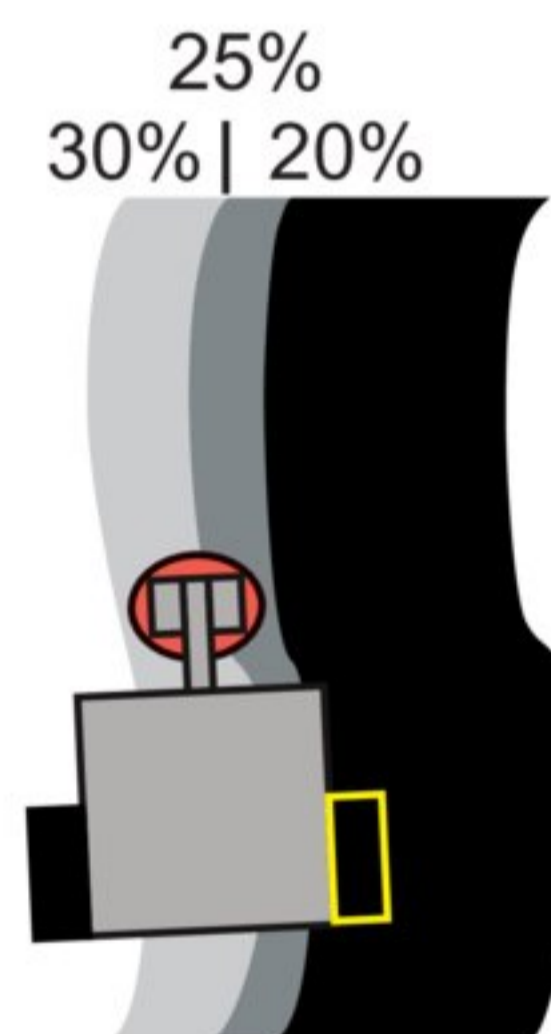


**Рис. 5.41.** Программа движения по границе чёрного и белого на релейном регуляторе

Запустив работа с этим алгоритмом, вы столкнётесь со знакомым поведением релейного двухпозиционного регулятора — появятся колебания. Колебательные движения составляют суть этого алгоритма: робот никогда не движется прямо, а только поворачивает в сторону белого либо в сторону чёрного поля. Подумайте, как, изменяя скорость моторов, добиться более плавного движения по линии?

Конечно, самый простой способ — понизить скорость того мотора, который работает в данный момент. Однако можно сделать движение более плавным, наоборот, повысив общую скорость движения. Как же этого достичь?

**Релейный трёхпозиционный регулятор.** Попробуем решить эту же задачу с помощью трёхпозиционного релейного регулятора. Для этого рассмотрим окрестность значения серого цвета  $\pm 5\%$  (рис. 5.42). Попадая датчиком в эту область (от 20 до 30% освещённости), робот будет



**Рис. 5.42.** Окрестность серого цвета



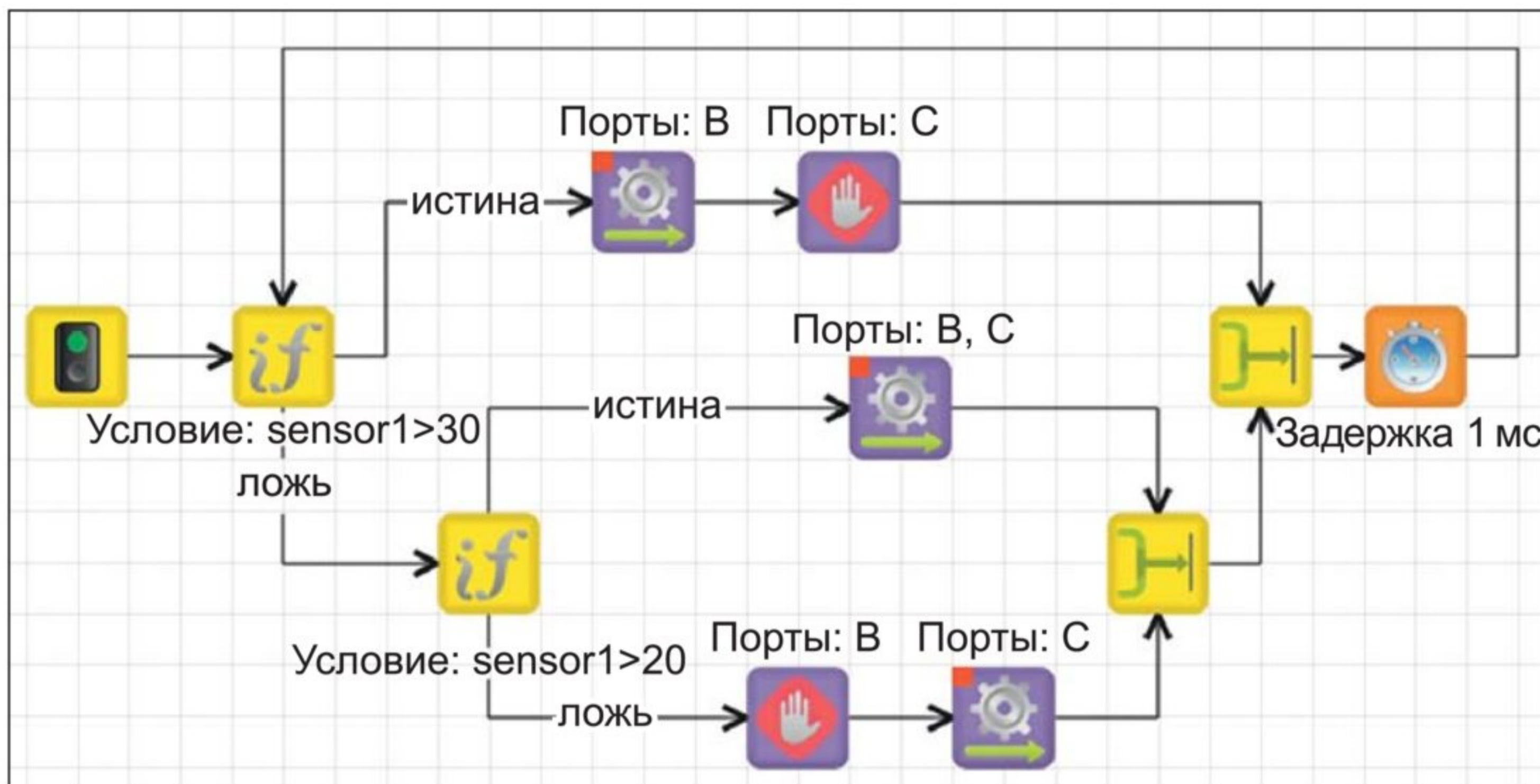


Рис. 5.43. Программа для трёхпозиционного регулятора

двигаться прямолинейно, а выходя за её пределы, поворачивать в нужную сторону. Это сделает его движение более плавным. При необходимости можно расширить зону окрестности, но значения её границ не должны слишком близко подходить к показаниям датчика на белом и чёрном полях (рис. 5.43).



**Пропорциональный регулятор.** Сделав всё возможное для сглаживания колебаний робота, вы подготовились к пониманию задачи пропорционального регулятора для движения по линии.



Рис. 5.44. Что видит человек и робот

Реализуя его алгоритм, следует представлять, что робот со своим единственным датчиком освещённости довольно «близорук». Там, где человек видит чёткую границу между чёрным и белым полями, робот «видит» плавный постепенный переход по различным оттенкам серого (рис. 5.44).

Это означает, что, перемещаясь с датчиком поперёк границы, робот с точностью до одного процента фикси-

рует изменение освещённости. Зависит это от соотношения показателей белого и чёрного полей, попадающих в пятно отражённого света. Диапазон значений может достигать несколько десятков процентов, и это даёт возможность построить полноценный регулятор по отклонению от значения на сером. Ниже приведены расчёты управляющего воздействия пропорционального регулятора на моторы, которые по умолчанию вращаются со скоростью  $v$ .

$$e = grey - sensor1,$$

$$u = k \cdot e,$$

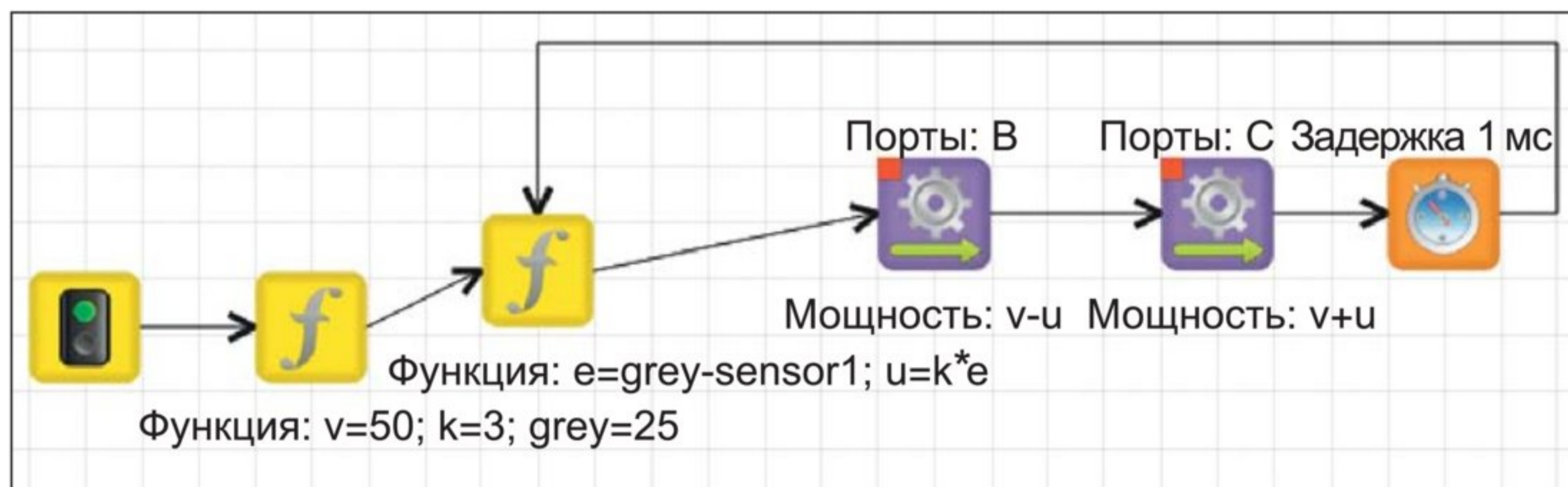
$$motorB = v - u,$$

$$motorC = v + u.$$

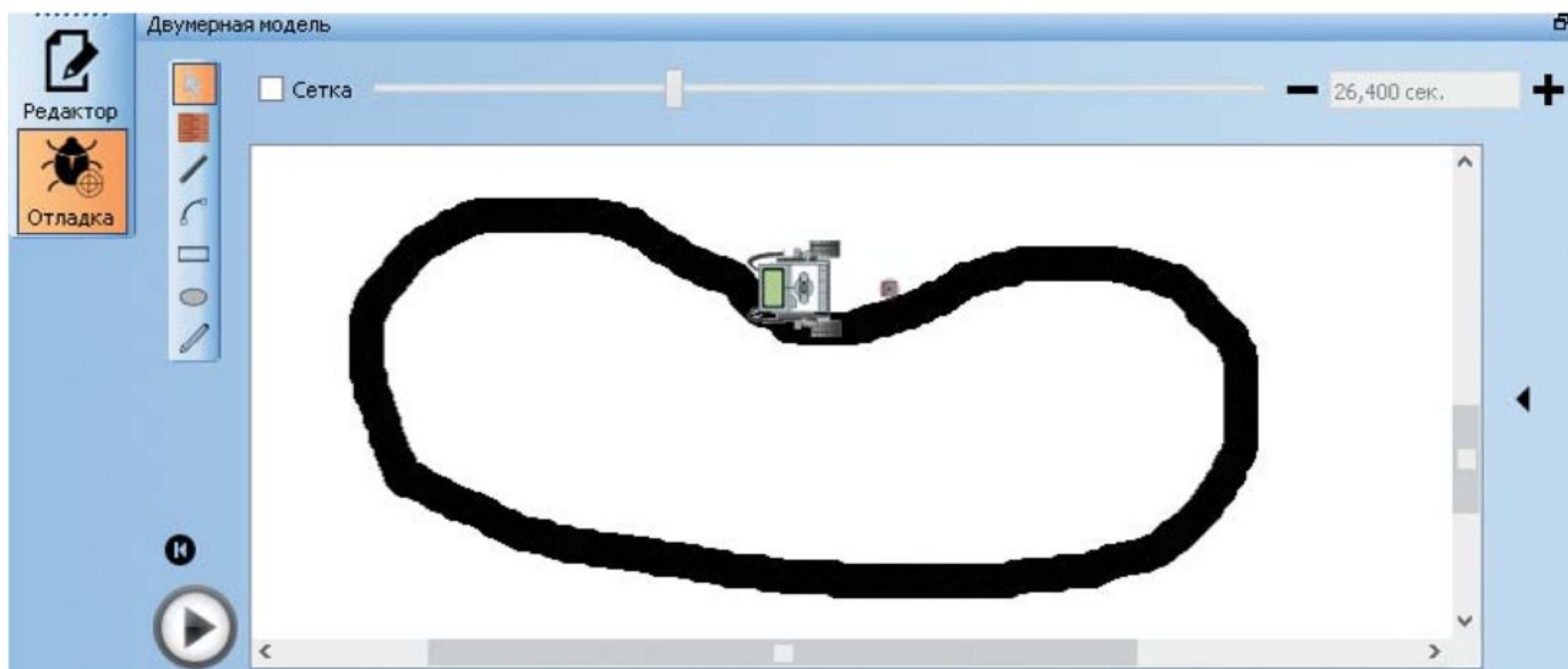
Эти действия помещаются в цикл с задержкой 1 мс и обеспечивают движение по линии. Нежелательные колебательные движения можно гасить, понижая значение коэффициента усиления регулятора (рис. 5.45).

Для проверки работоспособности программ без конструктора нарисуйте линию в режиме отладки TRIK Studio (рис. 5.46) и добавьте роботу датчик освещённости, установленный по схеме равностороннего треугольника.

Возникает вопрос: как рассчитать коэффициент усиления регулятора? Очевидно, его величина должна быть такова, чтобы обеспечить выполнение задачи, не допуская перерегулирования. На значение коэффициента влияют



**Рис. 5.45.** Программа для пропорционального регулятора для следования по линии



**Рис. 5.46.** Отладка следования по линии в двумерной модели

многие характеристики, в том числе и физические параметры робота: масса, размер, расположение центра масс, сцепление колёс с поверхностью, скорость, диапазон значений отклонения и другие. Использовать все эти параметры для определения коэффициента — сложная задача. Пока для следования по линии коэффициент усиления сделаем зависимым от текущей скорости и максимально возможной невязки (отклонения регулируемого параметра от уставки).

Допустим, базовая скорость робота равна  $v$  при максимальной  $v_{\max}$  и минимальной  $v_{\min}$ . Минимальное значение скорости влияет на крутизну поворотов, которые выполняет робот: на широкой линии они достаточно плавные, а на узкой — резкие. Предполагаем, что при переходе границы скорости произойдёт её автоматическое ограничение. Надо найти максимальную величину управляющего воздействия  $u_{\max}$ , необходимого, чтобы получить предельное значение скорости на моторе. Им станет наибольшее из чисел  $v_{\max} - v$  и  $v - v_{\min}$ .

Далее необходимо найти величину максимальной ошибки  $e_{\max}$  как половину разности между показаниями датчика на белом и чёрном поле:

$$e_{\max} = \frac{white - black}{2}.$$



Отсюда можно найти ориентировочное значение коэффициента усиления  $k$ :

$$k = \frac{u_{\max}}{e_{\max}}.$$

Рассмотрим пример.

Исходные данные:

$$v = 50,$$

$$v_{\max} = 100,$$

$$v_{\min} = 0,$$

$$white = 50,$$

$$black = 10.$$

Решение:

$$u_{\max} = 100 - 50 = 50,$$

$$e_{\max} = \frac{50 - 10}{2} = 20,$$

$$k = \frac{50}{20} = 2,5.$$

Ответ: коэффициент усиления  $k = 2,5$ .

При настройке конкретного робота коэффициент усиления скорее всего придётся изменить. Но его стартовое значение перед началом настройки удобно рассчитать по формулам, приведённым выше. Такой же подход используют при конструировании, например, автомобилей. Сначала прототип конструкции изготавливают по расчётным характеристикам, затем доводят в ходе натуральных испытаний.

## Проверьте себя

1. Как находить пороговое значение для движения по границе чёрного и белого?
2. Как понизить колебательные движения при движении по границе чёрного и белого на релейном регуляторе?
3. Какие пороговые значения указываются в трёхпозиционном регуляторе?
4. Как влияет увеличение коэффициента усиления пропорционального регулятора на качество движения по границе чёрного и белого?







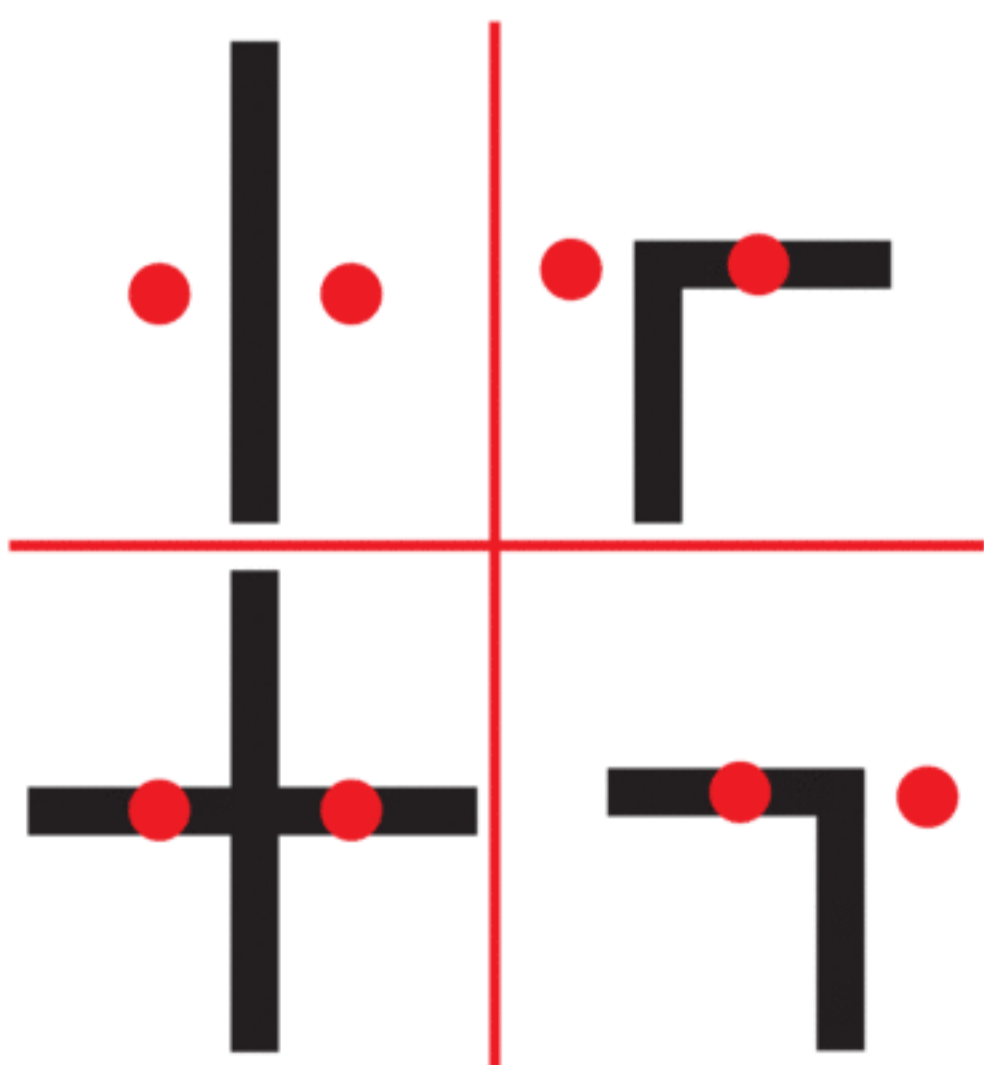
5. Какие параметры робота влияют на значение коэффициента усиления пропорционального регулятора?
6. Что требуется изменить в алгоритме при изменении высоты положения датчика освещённости?
7. В чём заключается правило равностороннего треугольника для робота на линии?
8. Увеличьте скорость робота до 100% и обеспечьте проезд по границе чёрного и белого, настраивая коэффициент усиления.
9. Постройте робота с большими колёсами и попробуйте проехать по линии на различных регуляторах с максимальной скоростью.
10. Используя секундомер, определите время проезда по линии и сообщите преподавателю лучший результат вашего робота.



**Запомните** ♦ Релейный двухпозиционный регулятор ♦ Релейный трёхпозиционный регулятор ♦ Пропорциональный регулятор

## § 5.7. Следование по линии с двумя датчиками. Калибровка

Выполнить следование по линии с помощью одного датчика возможно, но не всегда это обеспечивает желаемую стабильность. При небольшом сбое робот начинает вращаться на месте в поисках линии. Кроме того, с помощью



**Рис. 5.47.** Варианты расположения двух датчиков над линией

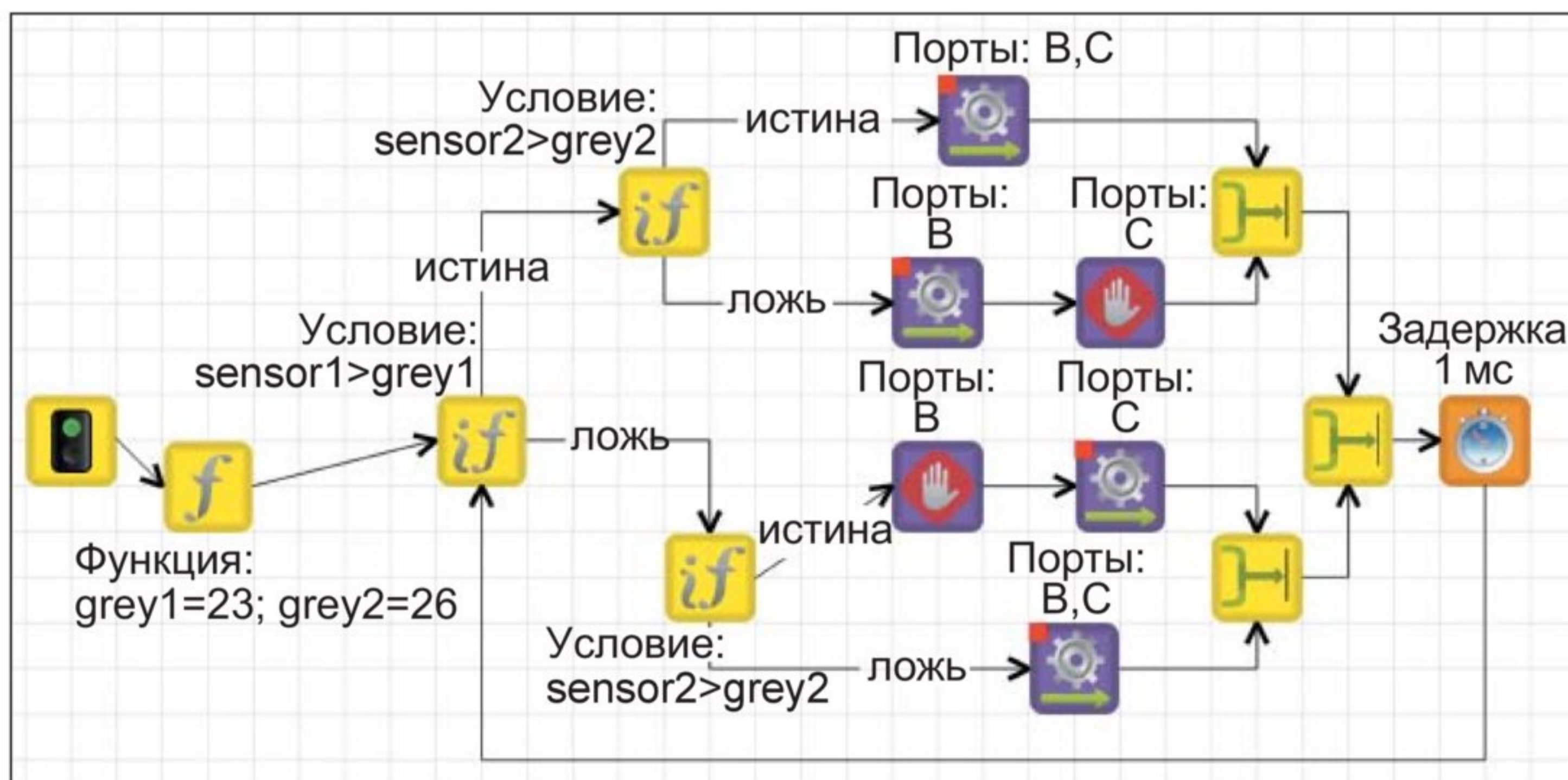
одного датчика затруднительно определить перекрёсток. Более надёжное решение — следование по линии с помощью двух датчиков освещённости (рис. 5.47). Датчики размещают по двум сторонам от линии, ширина которой 30–50 мм. Предполагая, что на линии могут быть перекрёстки, можно построить четыре основных варианта положения датчиков. Не забывайте правило равностороннего треугольника и требуемую высоту расположения датчиков над поверхностью — около 10 мм.

Для того, чтобы выяснить, над каким цветом находятся датчики, необходимо для каждого из них определить пороговое значение серого:  $grey1$  и  $grey2$ . Это можно сделать в режиме просмотра показаний датчиков, поочередно устанавливая работа датчиками на белое и на чёрное поле. Для каждого датчика необходимо найти своё пороговое значение, поскольку они могут существенно различаться.

$$grey1 = \frac{white1 + black1}{2},$$

$$grey2 = \frac{white2 + black2}{2}.$$

На основе пороговых значений постройте четырёхпозиционный релейный регулятор для движения по линии с помощью двух датчиков освещённости (рис. 5.48). Для этого потребуется три ветвления. В первом ветвлении показания левого датчика  $sensor1$  сравнивают с его значениями, полученными на сером  $grey1$ , которые задаются до цикла. Во втором и третьем ветвлениях то же выполняют со вторым датчиком  $sensor2$  и значением  $grey2$ .



**Рис. 5.48.** Программа четырёхпозиционного регулятора для робота с двумя датчиками освещённости



В предыдущем задании вы произвели калибровку вручную до загрузки программы в память робота. Но как быть, если робота с той же программой понадобится запустить при другой освещённости, или если конструкция робота была изменена и датчики оказались на другой высоте? Тогда лучше, чтобы робот произвёл калибровку сам (рис. 5.49). Самый простой способ — из показаний на белом вычесть некоторое число так, чтобы полученное значение было близко к показаниям на сером цвете. Это число принимаем как половину разности между показаниями на белом и на чёрном.

$$\begin{aligned} grey &= white - \frac{white - black}{2} = \\ &= white - \frac{white}{2} + \frac{black}{2} = \\ &= \frac{white}{2} + \frac{black}{2} = \frac{white + black}{2}. \end{aligned}$$

Например, при показаниях на чёрном 10 и на белом 40 разность будет 15. Это приблизительная калибровка, поскольку во время неё у робота отсутствуют показания датчика на чёрном.

В момент запуска программы очень важно установить робота так, чтобы датчики располагались именно *над белым полем*, иначе расчёт будет неверен и робот может сбиться с линии.

*Калибровка со считыванием показаний на чёрном и на белом.* При изменении высоты расположения датчиков диапазон их показаний может существенно изме-

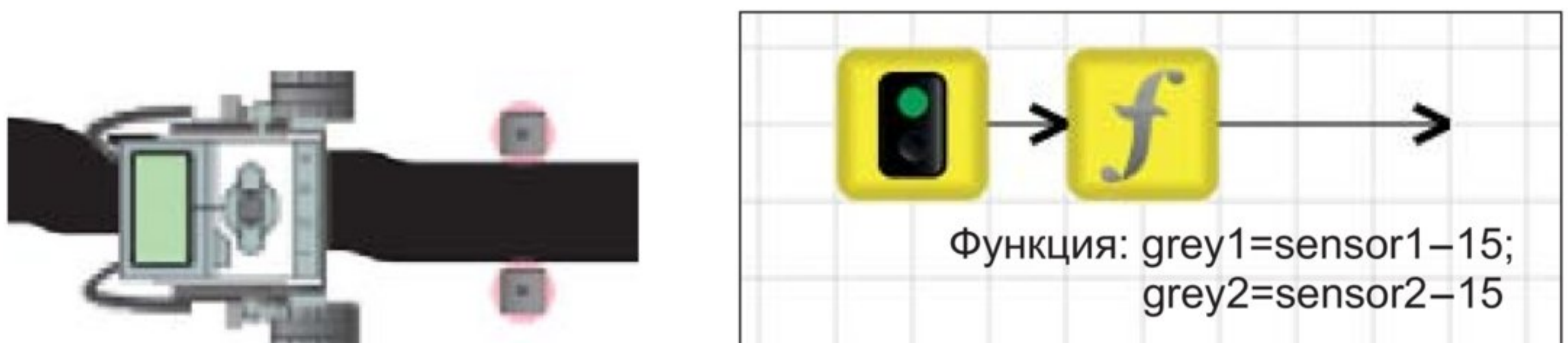
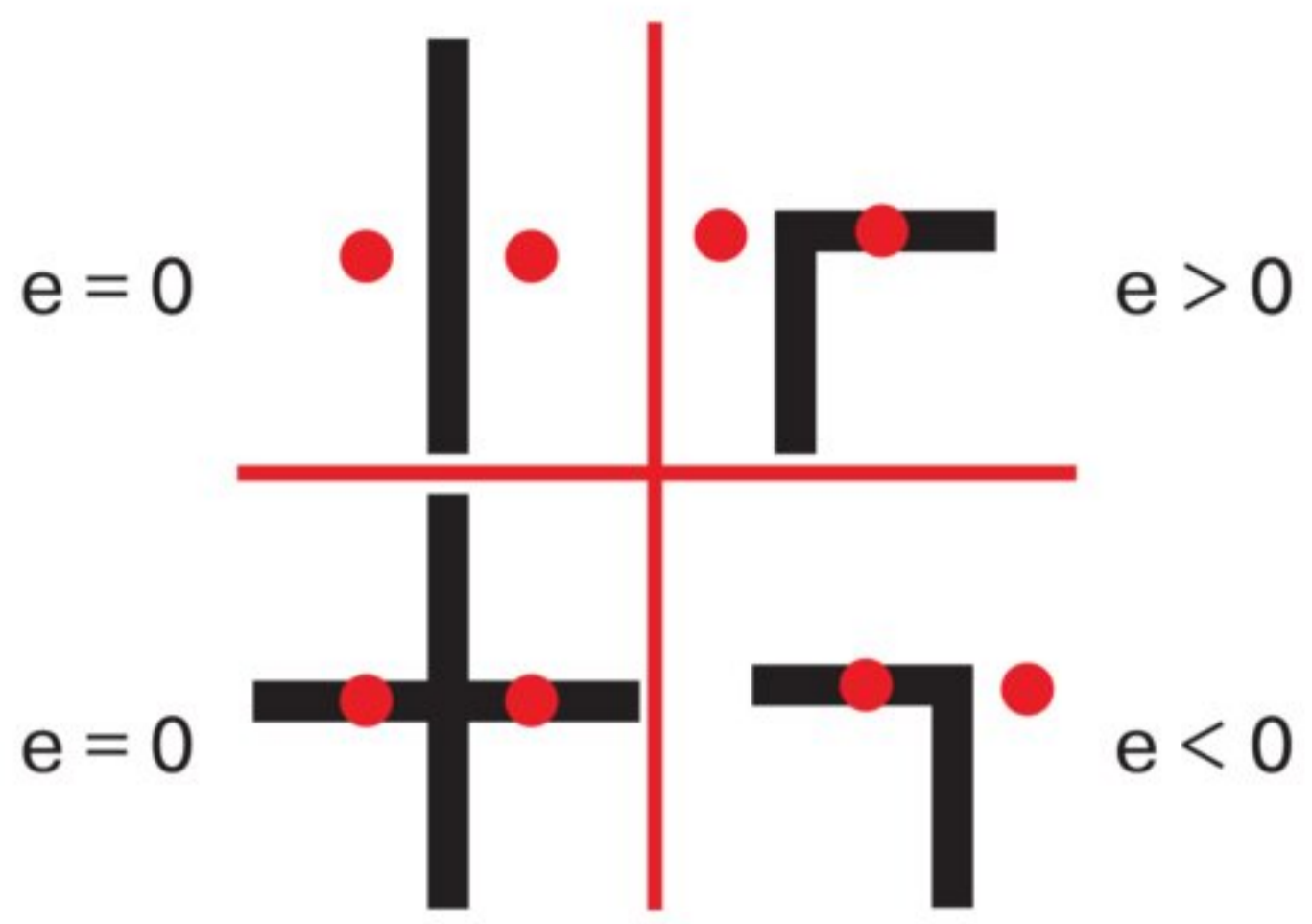


Рис. 5.49. Быстрая калибровка датчиков на белом цвете





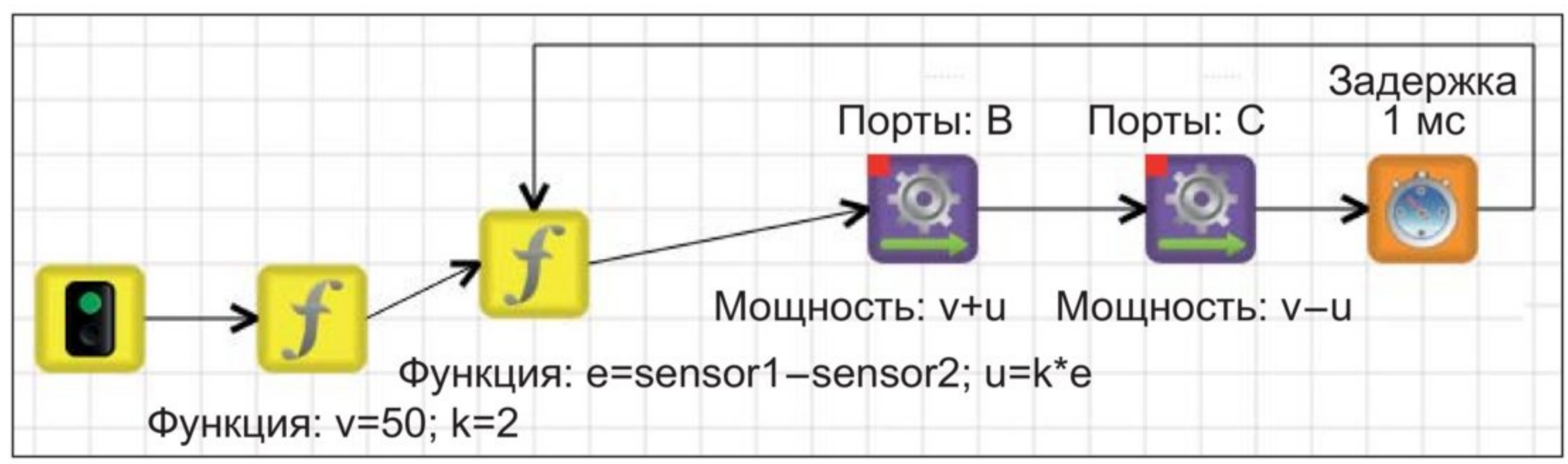
**Рис. 5.51.** Разность показаний датчиков можно принять за ошибку

полем) показания совпадают, то есть их разность равна нулю. В двух других случаях разность показаний либо положительная, либо отрицательная.

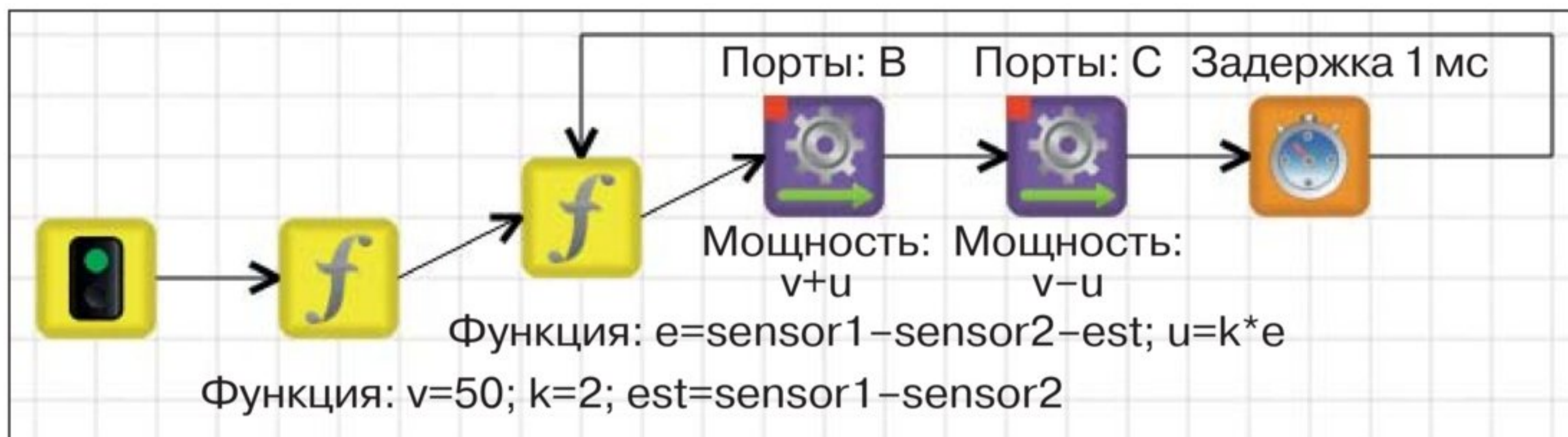
Значит, для построения пропорциональной зависимости за ошибку можно принять разность показаний датчиков. Программа будет очень похожа на программу пропорционального регулятора для одного датчика (рис. 5.52).

Как изменится коэффициент усиления по сравнению с регулятором для одного датчика? Какой коэффициент будет оптимальным для двух датчиков?

С правильно подобранным коэффициентом робот будет двигаться по линии достаточно стабильно. Но если установить робота на белое поле без чёрной линии, в большинстве случаев движение не будет прямолинейным. В чём причина? Почему получается, что разность показаний



**Рис. 5.52.** Программа следования по линии с двумя датчиками освещённости на П-регуляторе



**Рис. 5.53.** Программа пропорционального регулятора с вычислением статической ошибки на старте

датчиков на белом поле не равна нулю? Дело в том, что в предыдущем случае не учтена разница показаний датчиков, которая случается из-за погрешности изготовления. Разность показаний датчиков на однотонной поверхности и создаёт дополнительную ошибку. Будем называть её *статической*.

$$est = sensor1 - sensor2.$$

Это тоже некоторая калибровка, которую можно выполнять на старте. В момент её проведения оба датчика должны быть установлены строго над белым полем.

*Динамическая* ошибка возникает в процессе считывания цвета поверхности тоже как разность показаний датчиков. Только теперь для повышения точности измерений необходимо вычесть из неё статическую ошибку (рис. 5.53).

$$e = sensor1 - sensor2 - est.$$

## Проверьте себя

1. На какой высоте над полем могут быть размещены датчики освещённости? Назовите наименьшую и наибольшую высоту их установки. Как меняются при этом показания над чёрным и над белым полем?
2. Что показывают датчики и как ведут себя двигатели в каждой позиции релейного четырёхпозиционного регулятора?
3. Почему при вычислении ошибки пропорционального регулятора по двум датчикам можно обойтись без показаний датчиков на сером поле?





4. Объясните, почему возникают статическая и динамическая ошибки для двух датчиков освещённости.
5. В чём суть калибровки датчиков понижением показаний на белом поле?
6. Расскажите порядок калибровки датчиков с поиском среднего арифметического.
7. Постройте алгоритм для калибровки одного датчика при движении по границе чёрного и белого цветов.
8. Увеличьте скорость робота до 100% и обеспечьте его движение по линии, настраивая коэффициент усиления.
9. Постройте робота с большими колёсами и попробуйте проехать по линии на различных регуляторах с максимальной скоростью.
10. Используя секундомер, определите время проезда робота по линии и сообщите преподавателю его лучший результат.

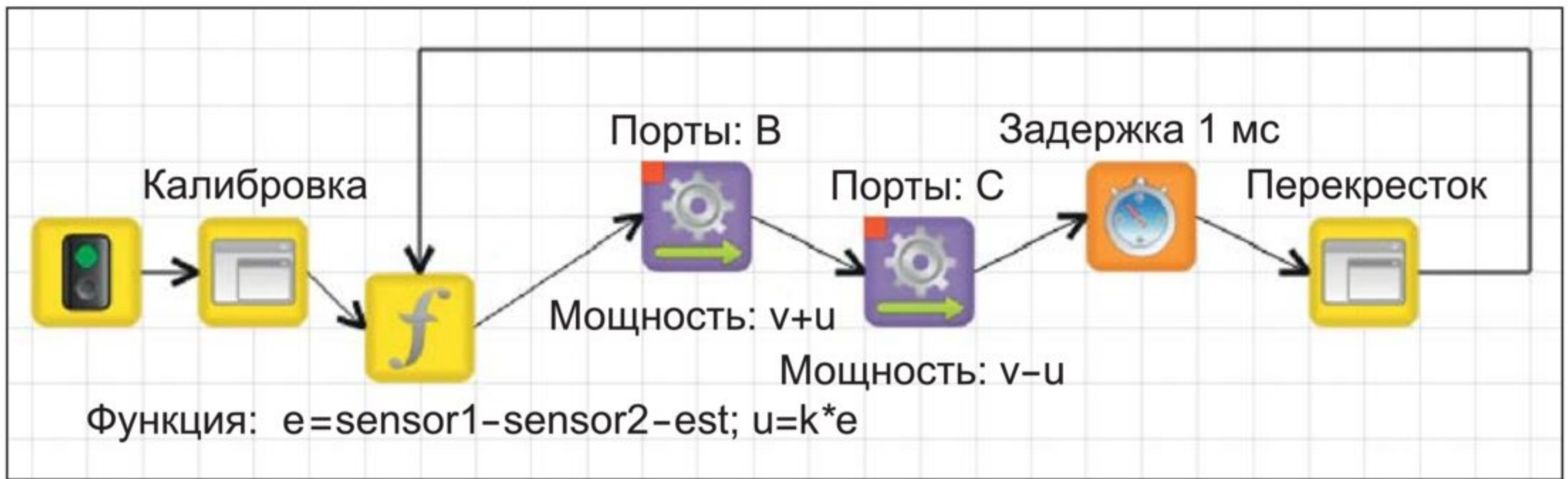
## § 5.8. Определение перекрёстков и действия на них

Следующая задача — *реакция на перекрёстки*. Ситуацию, когда два датчика освещённости оказываются над чёрной линией, легко распознать с помощью так называемого *сложного условия*. Для создания таких условий в информатике применяются логические операции: «И», «ИЛИ», «НЕ». Поставив логическое «И» между двумя условиями, получим сложное условие, по которому должны выполняться одновременно оба простых.

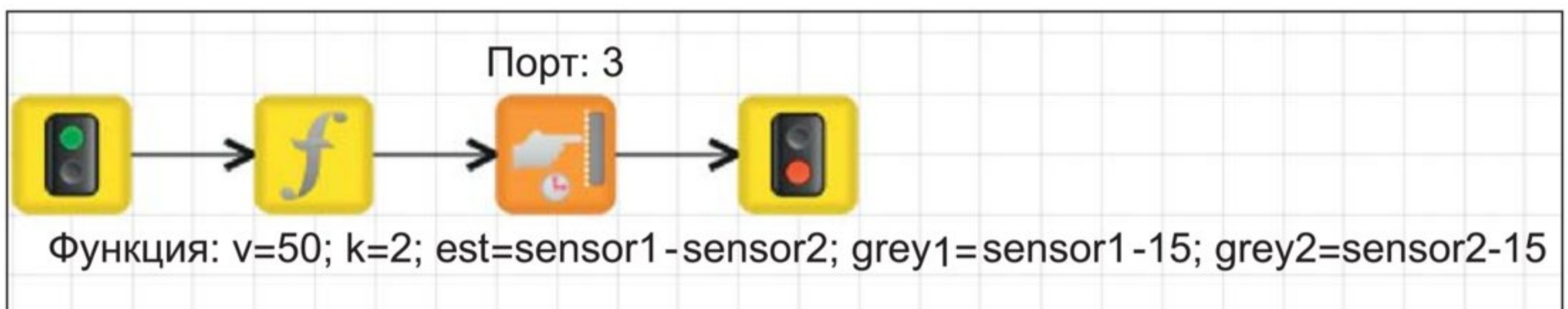
первый\_датчик\_на\_черном И второй\_датчик\_на\_черном  
 $sensor1 < grey1 \text{ and } sensor2 < grey2$

Применяя сложное условие, можно получить проверку наличия перекрёстка и издать на нём звуковой сигнал. Ветвление с этим условием добавляется в цикл движения робота (рис. 5.54). Для компактности следует оформить проверку перекрёстка и калибровку датчиков в виде отдельных подпрограмм (рис. 5.55 и 5.56).

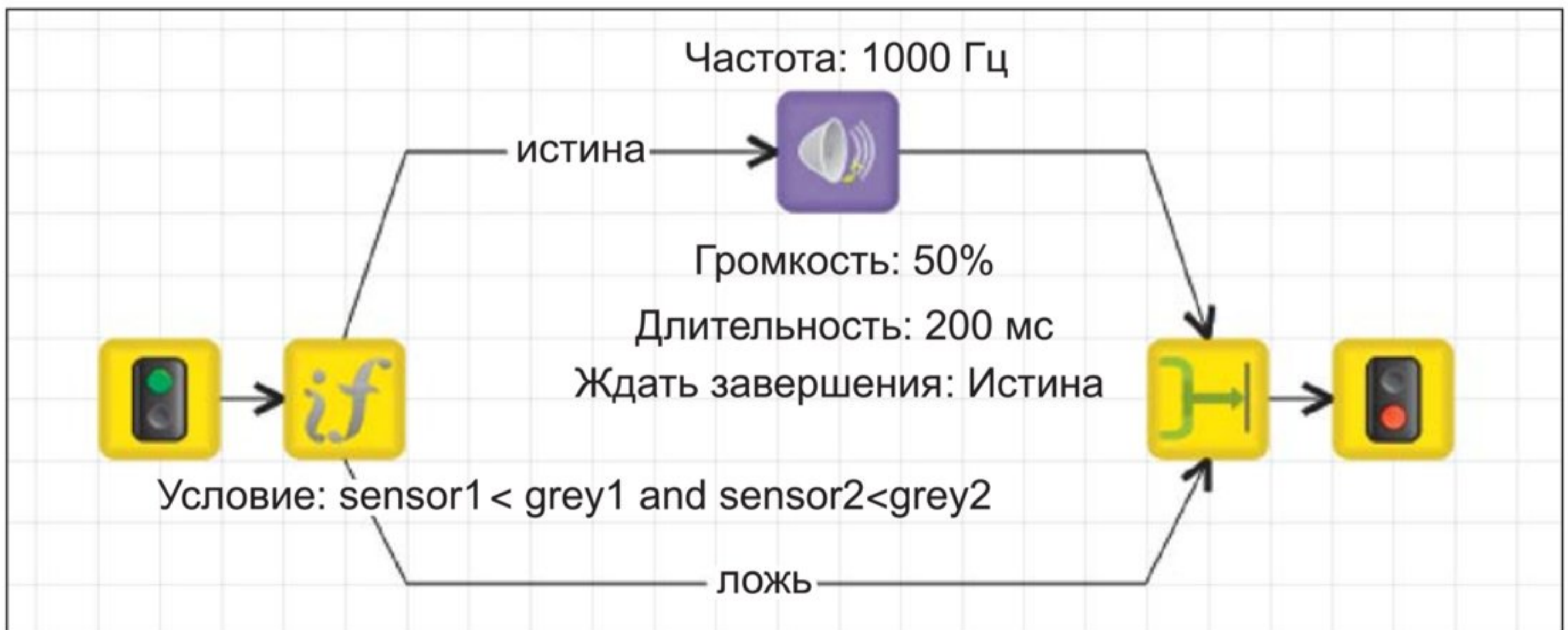
При этом необходимо учесть то, что робот движется со скоростью около 30 см/с и линию перекрёстка шириной



**Рис. 5.54.** Следование по линии с проверкой перекрёстков



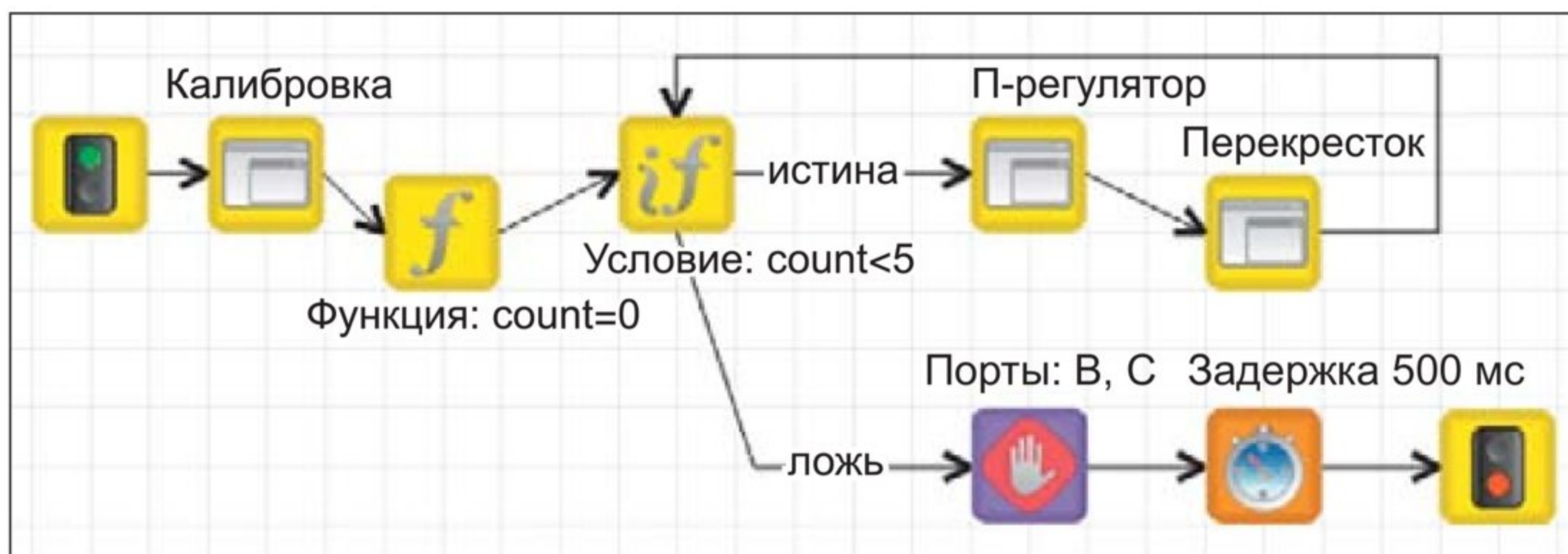
**Рис. 5.55.** Подпрограмма для калибровки



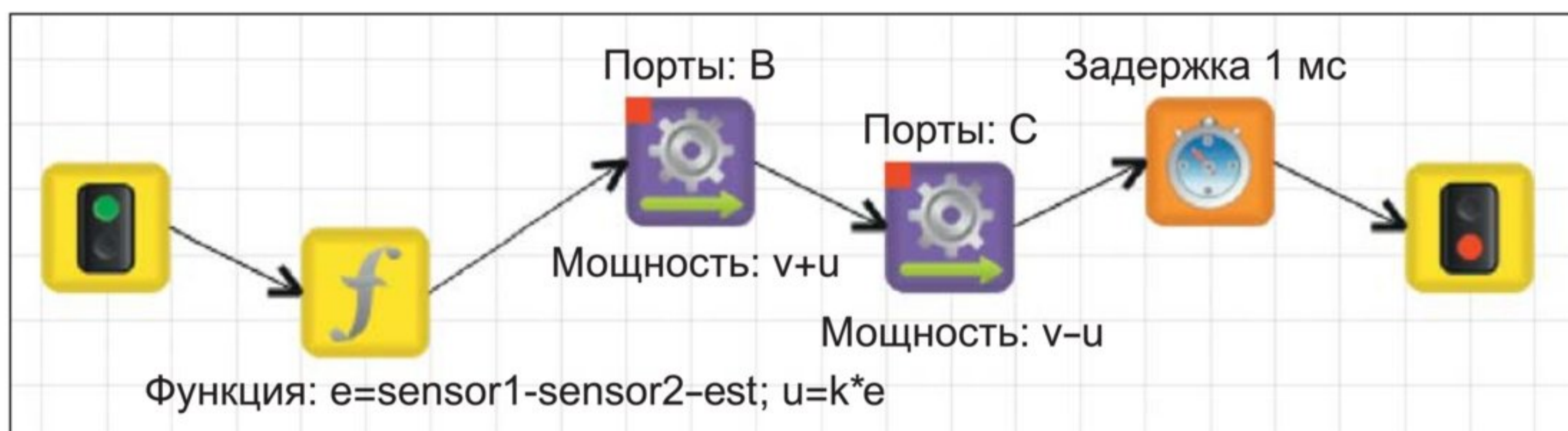
**Рис. 5.56.** Подпрограмма для проверки перекрёстка

около 3 см он проедет примерно за 0,1 с. Если не будет установлена задержка, робот начнёт безостановочно издавать звуковые сигналы, поскольку за один проезд перекрёстка он обнаружит его около 100 раз. Если поставить





**Рис. 5.57.** После пятого перекрёстка программа завершается

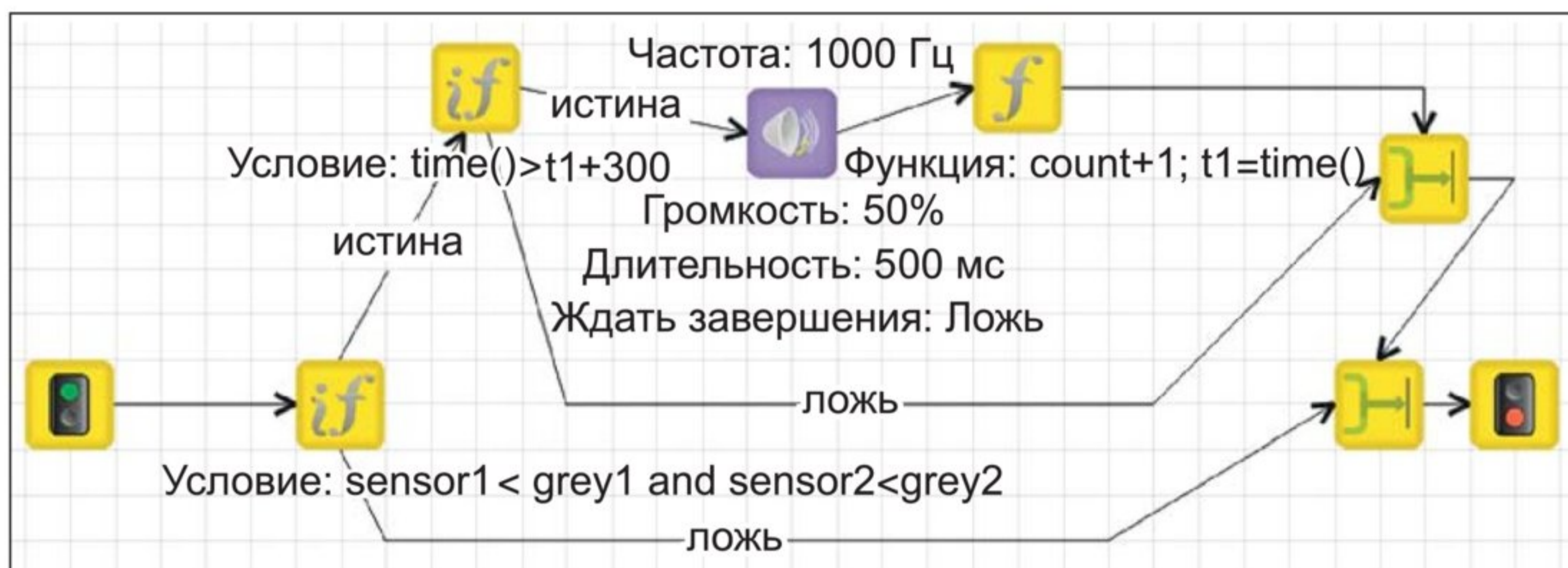


**Рис. 5.58.** Подпрограмма для основной части пропорционального регулятора

слишком большую задержку — робот собьётся с линии. Поэтому нужно найти «золотую середину». В примере на рисунке 5.56 это 0,2 с.

Чтобы *посчитать перекрёстки* и остановиться после очередного, потребуется переменная-счётчик и цикл с условием. На каждом шаге счётчик будет увеличиваться на 1 и, когда достигнет нужного значения, цикл движения завершится (рис. 5.57 и 5.58).

Надёжное исполнение этого алгоритма достигается в случае, если обеспечить стабильный проезд перекрёстка. Один из способов состоит в том, чтобы использовать таймер и не выставлять ожидание окончания звукового сигнала, который будет звучать в фоновом режиме. При каждом обнаружении перекрёстка робот фиксирует

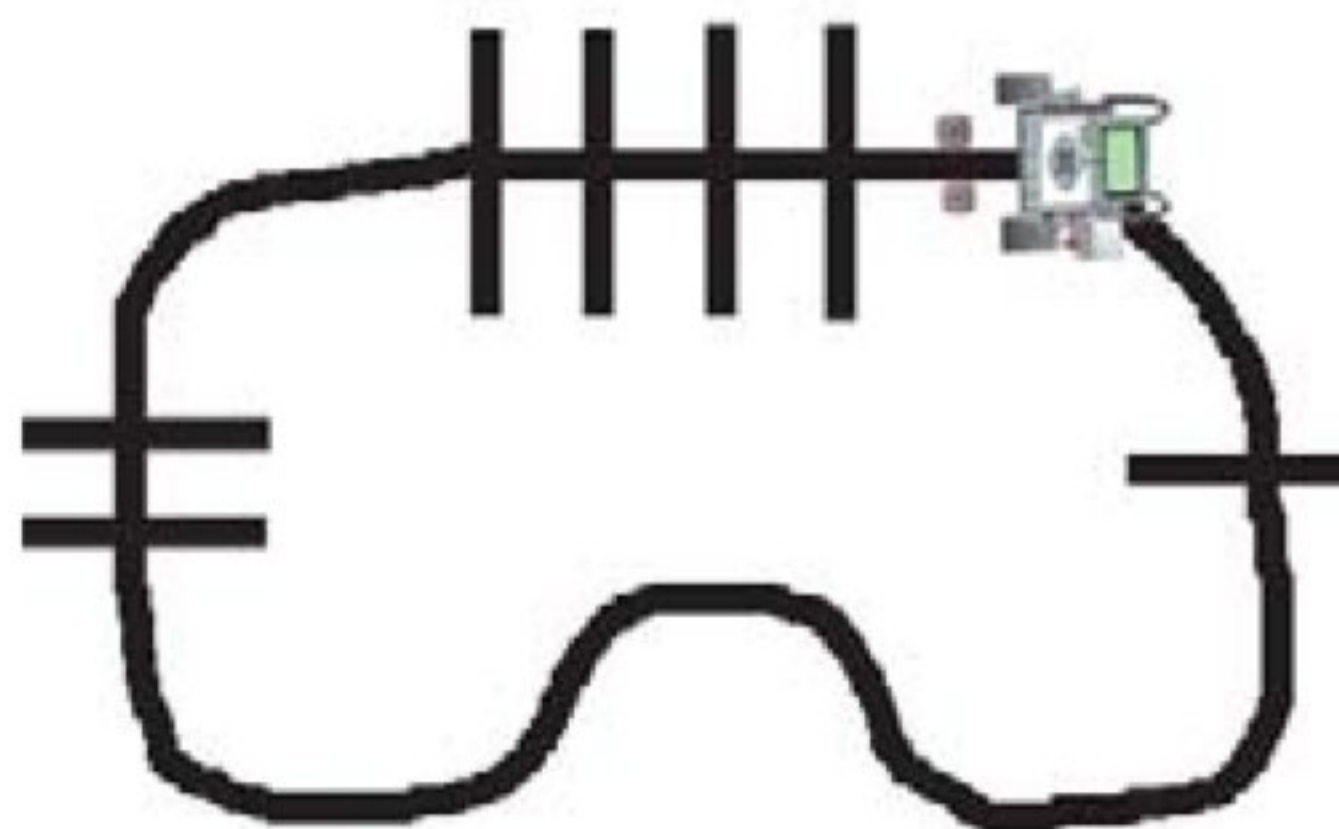


**Рис. 5.59.** Усовершенствованный проезд перекрёстка

текущее время и не воспринимает следующего, пока не пройдёт 0,2 с. Но при этом робот продолжает движение по линии. Поэтому в добавление к этому следует указать начальное значение  $t1 = time()$  вместе с обнулением  $count$  (рис. 5.59). На рисунке 5.60 показано движение по перекрёсткам в TRIK Studio.

*Действия на перекрестках.* Компания *Kiva Robotics* создала для работы на складах комплекс роботов, которые перемещаются по заданным направляющим и доставляют полки с нужными товарами. То есть не человек ходит по складу в поисках нужного товара, а товар сам приезжает к нему. В ключевых точках расположены специальные метки, по которым роботы определяют, в каком месте склада они находятся в данный момент. Сотни роботов одновременно двигаются по складу под управлением единого вычислительного центра и никогда не сталкиваются (рис. 5.61).

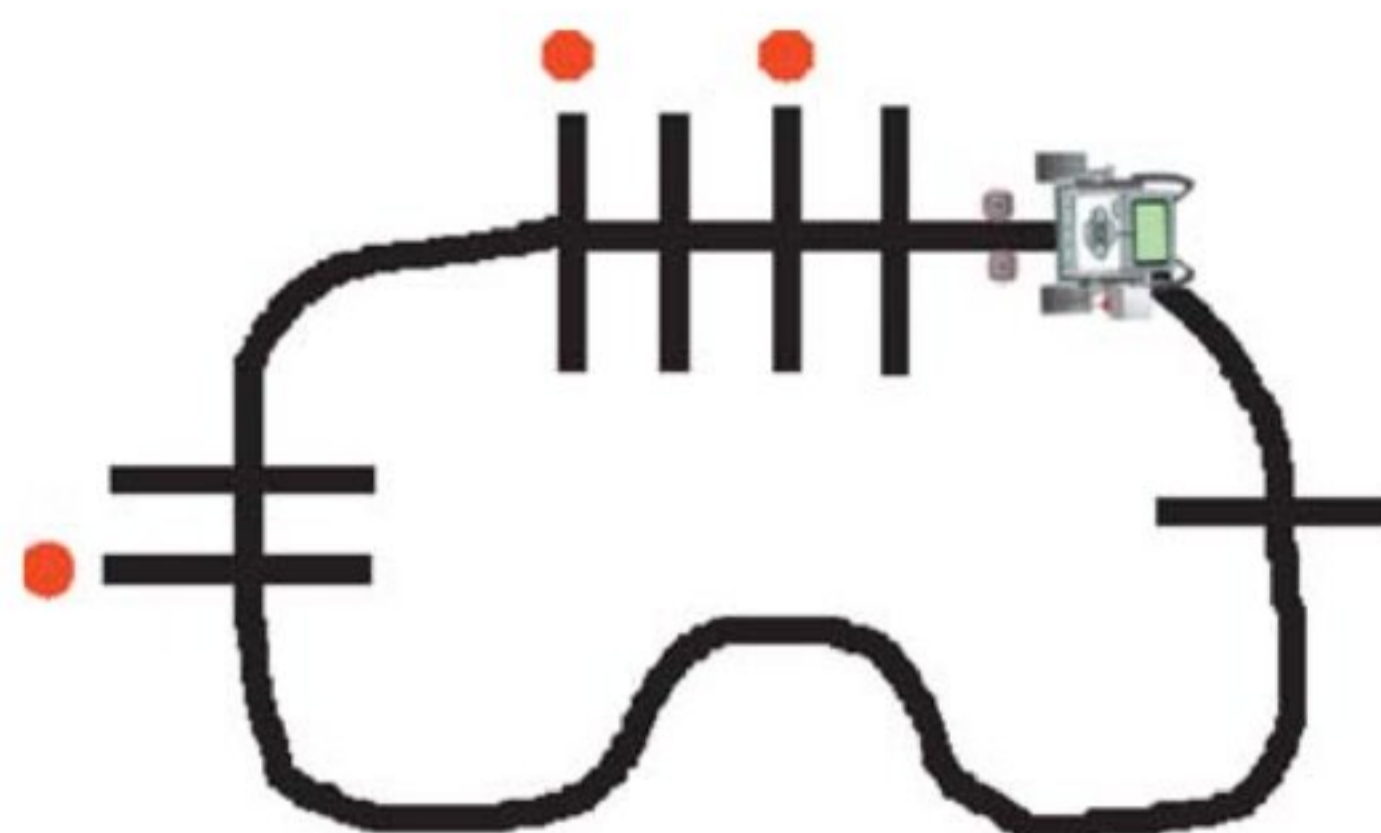
Такую совершенную систему создали учёные и инженеры. Вы тоже можете сделать робота, который выполняет



**Рис. 5.60.** Движение по перекрёсткам в TRIK Studio

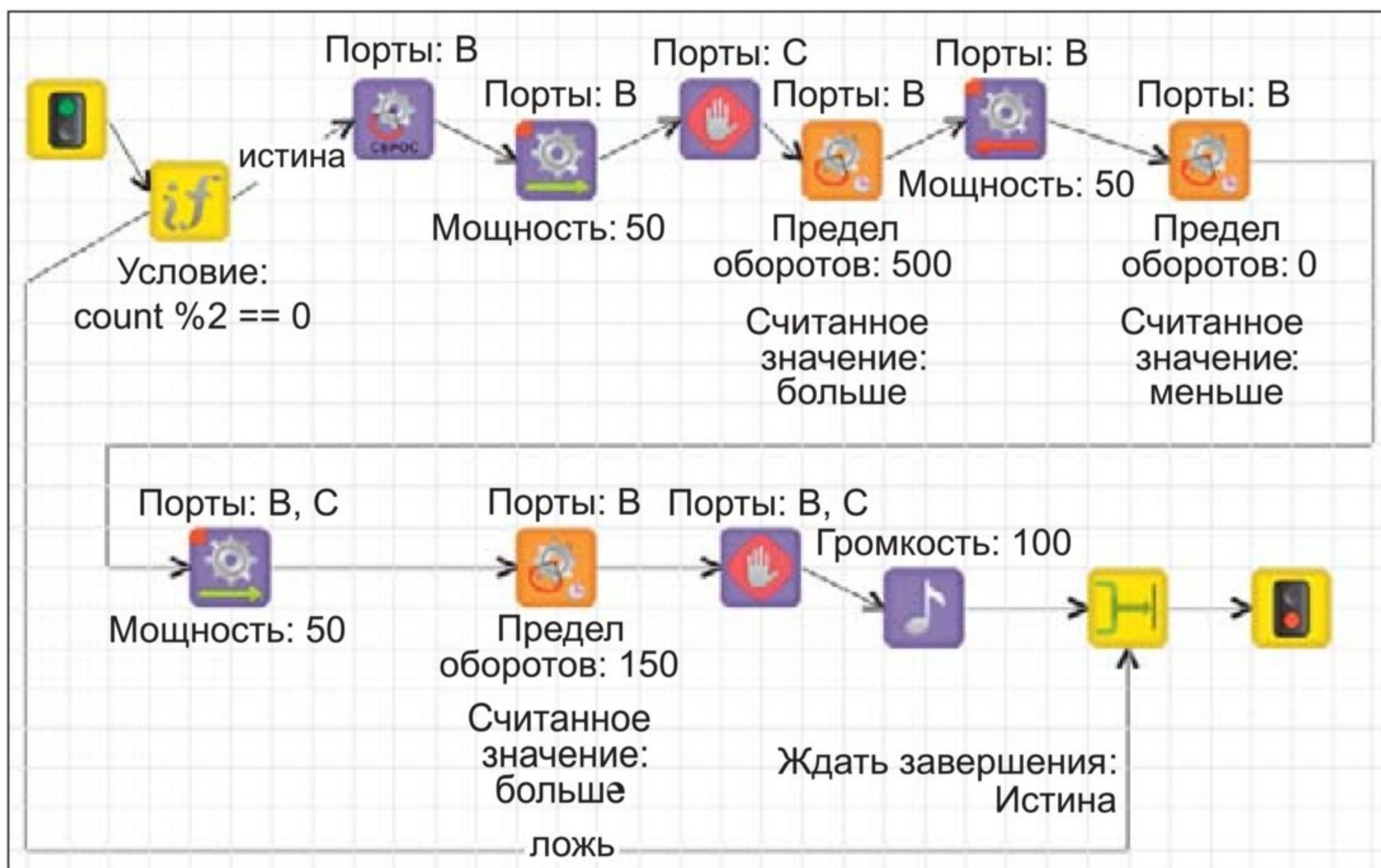


**Рис. 5.61.** Складские роботы перемещают полки с товарами



**Рис. 5.62.** Объекты на перекрёстках

определённые задачи, ориентируясь по перекрёсткам. Например, на некоторых перекрёстках находятся объекты, которые надо переставить в другое место (рис. 5.62). Тогда при заезде на перекрёсток робот выполняет проверку его номера и принимает решение о выполнении нужных



**Рис. 5.63.** Подпрограмма «Объект» сдвигает объекты, расположенные справа на каждом чётном перекрёстке



## § 5.9. Безаварийное движение

Современное транспортное средство снабжено специальными приборами, позволяющими контролировать дистанцию и сзади, и спереди (рис. 5.65). Как правило, это ультразвуковые датчики, которые измеряют расстояние от нескольких сантиметров до нескольких метров. Такие системы используют не только при парковке, но и для автоматизированного контроля дистанции на трассе.

Обычно датчики размещаются на бамперах транспортного средства. Система «Парктроник» сообщает о приближении бампера к препятствию с помощью звукового сигнала, и в некоторых случаях визуально (рис. 5.66). Благодаря этой системе человек может предотвратить столкновение. Но ещё больше возможностей существует у автопилота —

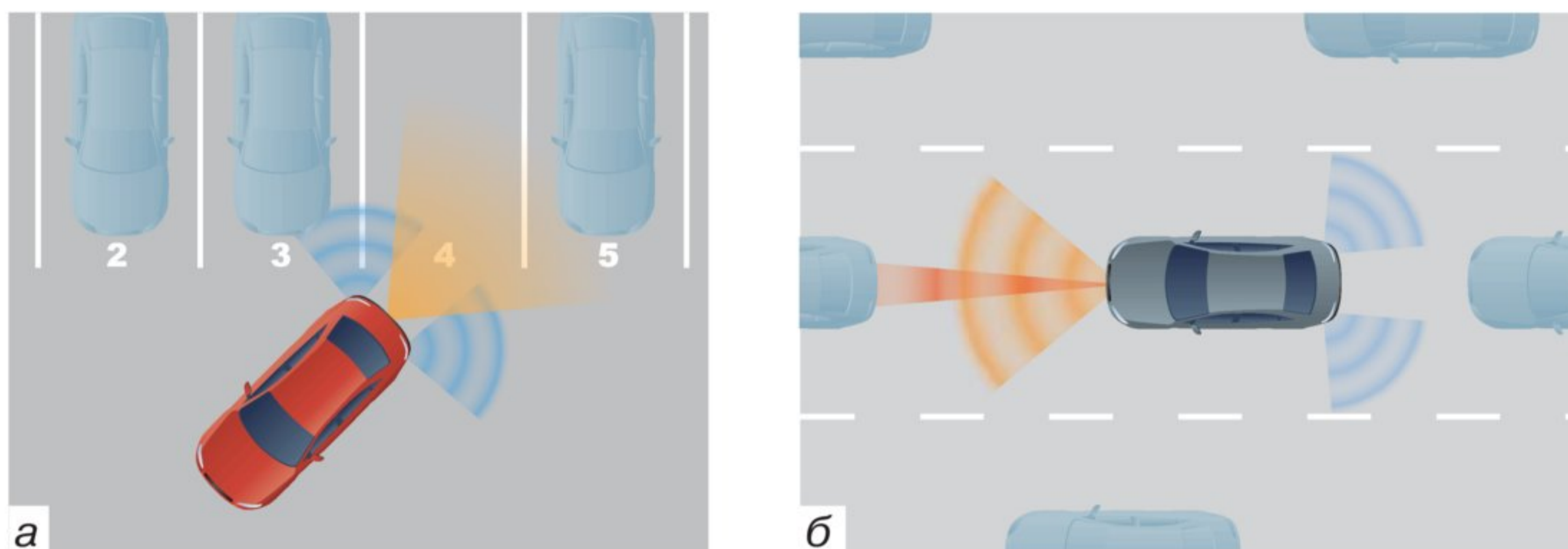


Рис. 5.65. Системы контроля дистанции при парковке (а) и на трассе (б)

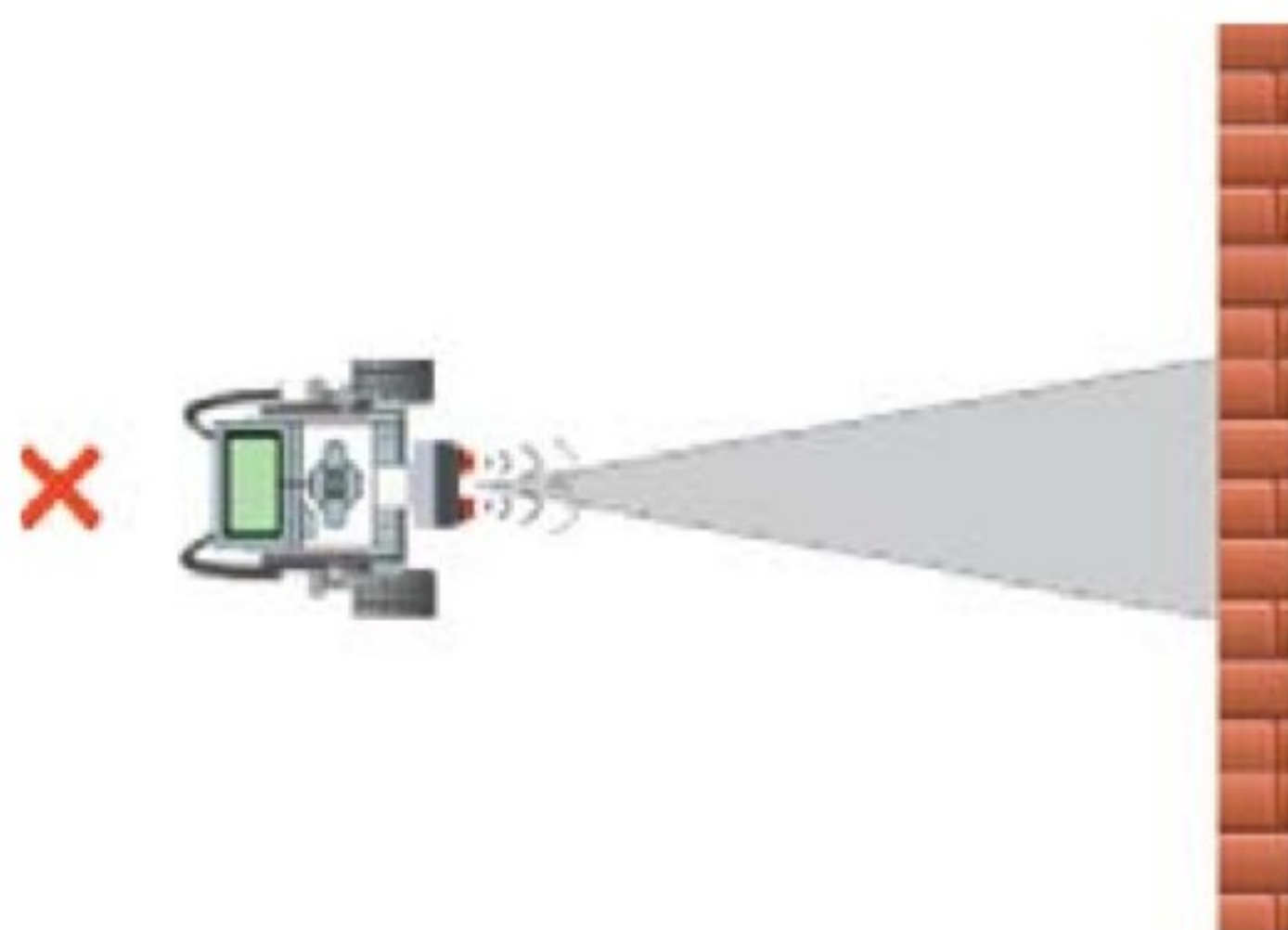


Рис. 5.66. Датчики расстояния (а) и сигналы системы (б)

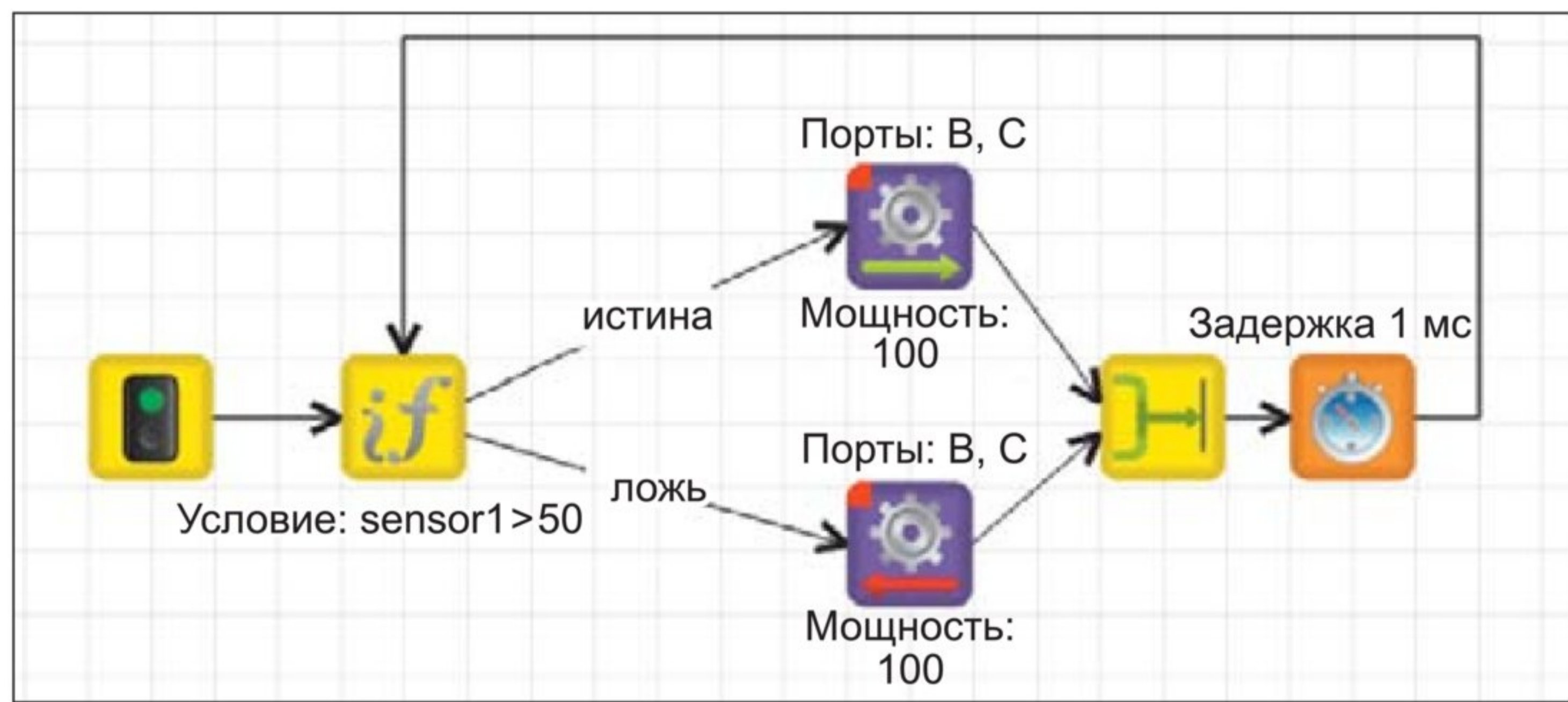
системы автоматического управления автомобилем.

*Контроль расстояния до объекта.* В стандартном робототехническом наборе имеется всего один датчик расстояния. Его использование даёт возможность избегать столкновений робота с препятствиями перед ним (рис. 5.67). Сначала научимся контролировать расстояние до объекта для тележки, которая движется прямолинейно. Для этого достаточно собрать одномоторную тележку или оба мотора двухмоторной тележки запрограммировать на вращение с одинаковой скоростью. Постройте релейный регулятор для выполнения этой задачи (рис. 5.68).

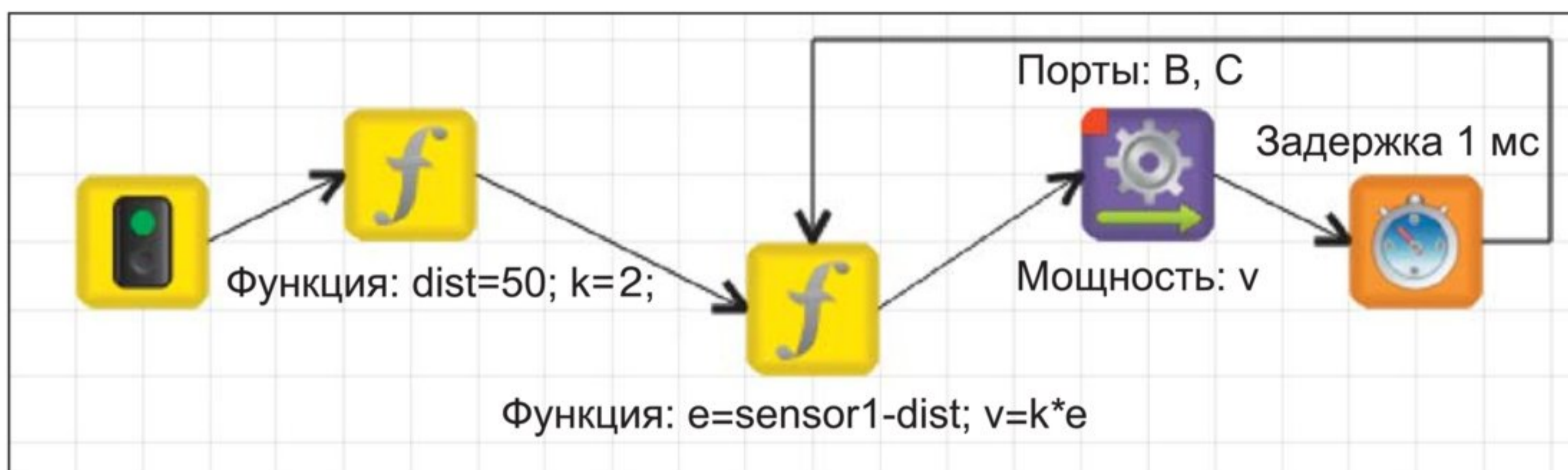
И с реальным роботом, и в виртуальной модели релейный регулятор повторяет своё поведение: система входит в режим колебаний. Очевидно, оптимальным решением



**Рис. 5.67.** Робот с датчиком расстояния располагается напротив стены



**Рис. 5.68.** Программа релейного регулятора с контролем расстояния до стены



**Рис. 5.69.** Программа пропорционального регулятора с контролем расстояния до стены

будет пропорциональный регулятор (рис. 5.69). Управляющее воздействие обозначим как скорость  $v$ .

Особенно интересным будет поведение роботов, если их поставить один за другим так, чтобы между их колёсами разместить длинную рейку (рельс), за которую не будут задевать корпуса роботов (рис. 5.70).

Управляя положением первого робота с помощью листа бумаги, можно с задержкой управлять всеми остальными.

*Контроль дистанции при движении по линии.* Вы научились контролировать скорость робота, теперь следует совместить контроль скорости с движением по линии. Получится двойной пропорциональный регулятор, в котором за скорость отвечает датчик расстояния  $sensor1$ , а за направление движения — датчик освещённости  $sensor2$ :

$$v = (sensor1 - dist) \cdot k_1,$$

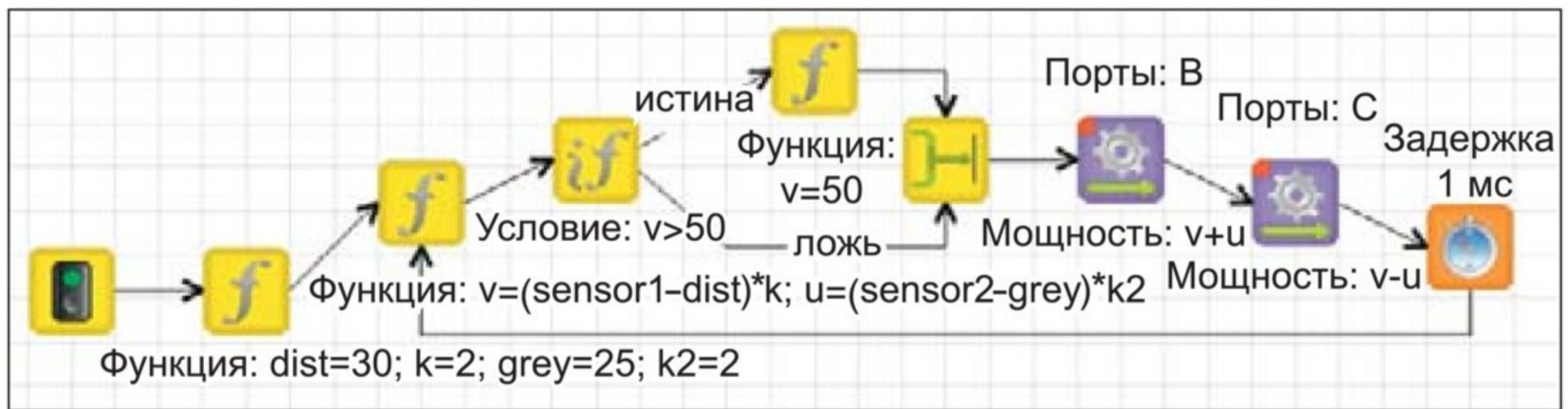
$$u = (sensor2 - grey) \cdot k_2,$$

$$motorB = v + u,$$

$$motorC = v - u.$$



**Рис. 5.70.** «Состав» из роботов на рельсе с контролем расстояния



**Рис. 5.71.** Программа совмещённого регулятора для движения по линии с контролем скорости

Однако для совместимости регуляторов потребуется ввести ограничение скорости. Вы знаете, что максимальные показания датчика расстояния могут достигать 255 см, а предельная скорость мотора — 100. Тогда почти при всех возможных значениях управляющего воздействия и сумма  $v + u$ , и разность  $v - u$  могут превысить число 100. Это означает, что робот будет просто двигаться вперёд с максимальной скоростью. Значит, следует ввести ограничение скорости (рис. 5.71).

Запустив программу, вы убедитесь, что, останавливаясь вблизи объекта, тележка может сильно колебаться на линии. Вы также помните, что коэффициент усиления регулятора зависит от скорости. В данном случае при торможении коэффициент следует снизить. Как вариант — ввести следующую зависимость, которая будет рассчитываться в цикле:

$$k_2 = k_{\min} + (k_{\max} - k_{\min}) \cdot \frac{v}{v_{\max}},$$

где  $k_{\min}$  — это коэффициент при остановке,  $k_{\max}$  — коэффициент при максимальной скорости  $v_{\max}$ .

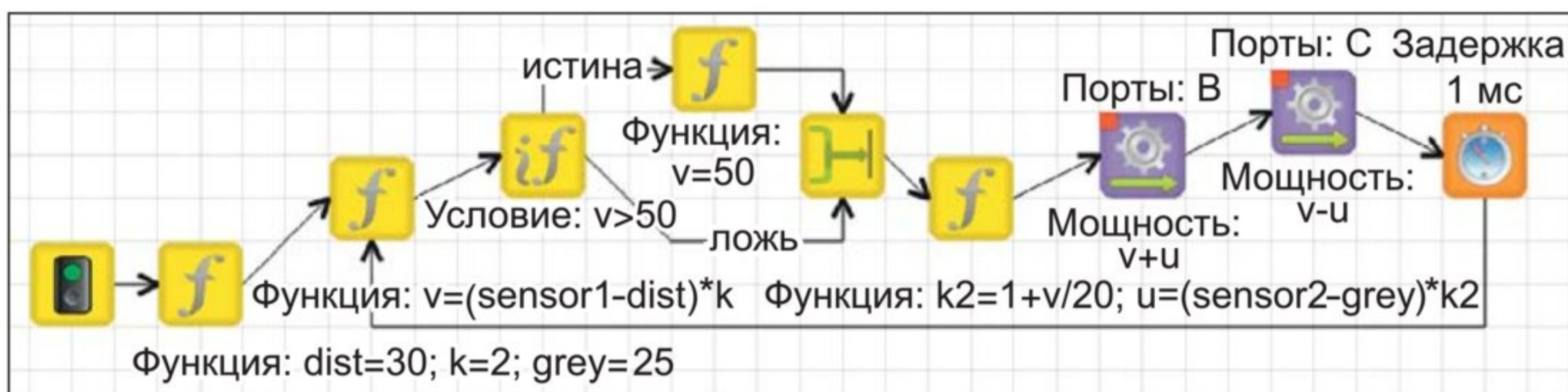
Например,

$$k_{\min} = 1, \quad k_{\max} = 6, \quad v_{\max} = 100.$$

Тогда получаем

$$k_2 = 1 + (6 - 1) \cdot \frac{v}{100} = 1 + \frac{v}{20}.$$





**Рис. 5.72.** Программа с изменением последовательности действий



**Рис. 5.73.** Безаварийное движение по линии

Каждые 20% скорости будут добавлять 1 к минимальному коэффициенту  $k_{\min}$ . В этом случае придётся изменить последовательность действий: сначала определение скорости  $v$ , затем определение коэффициента усиления  $k_2$  и, наконец, определение управляющего воздействия  $u$  (рис. 5.72).

Теперь мы подошли к самой интересной части. Подготовив

в классе несколько тележек с контролем скорости, запустите их, расположив на одной линии. Если всё сделано правильно, тележки будут двигаться без сбоев и столкновений (рис. 5.73).



## Проверьте себя

1. Какие датчики используются в современных автомобилях для контроля дистанции? Где они расположены?
2. Что является ошибкой при контроле дистанции до объекта? На что влияет эта ошибка?
3. Как совместить контроль дистанции и управляющее воздействие для движения по линии?
4. Как определяется коэффициент для движения по линии с учётом текущей скорости?
5. Постройте систему безаварийного движения по линии с помощью двух датчиков освещённости и одного датчика расстояния на каждом роботе.

6. Постройте систему движения по перекрёсткам, при которой на каждом перекрёстке каждый робот должен остановиться на несколько секунд.

### Это интересно!

По улицам городов уже давно ездят роботизированные беспилотные автомобили, которые делают панорамные снимки через каждые несколько метров для интернет-картографии. Для наблюдения за дорогой используются лазерные сканеры и компьютерное зрение. Скорость таких автомобилей, как правило, невелика — около 5 км/ч. На этой скорости совсем не сложно избежать столкновений с любыми объектами. И хотя небольшие аварии всё же случаются, они приводят к совершенствованию программного обеспечения автопилота.

## § 5.10. Объекты на линии

Транспортному средству, движущемуся в автоматическом режиме, необходимо не только контролировать дистанцию, но и выполнять различные манёвры: разворот, объезд, обгон и другие (рис. 5.73). В простейшем случае для принятия решения о совершении соответствующего манёвра достаточно показаний датчика расстояния.

Допустим, на траектории движения робота обнаружено препятствие. Необходимо выполнить плавное торможение, на расстоянии 20 см остановиться и подать звуковой сигнал. При этом пропорциональный регулятор нельзя настраивать на расстояние 20 см, поскольку робот может остановиться на расстоянии



Рис. 5.73. Объезд препятствия на дороге

21–22 см и не сдвинуться с места. Причина в том, что на малых отклонениях роботу не хватает управляющего воздействия на моторы для продолжения движения. Настройте регулятор на расстояние 10 см, тогда последние сантиметры робот проедет на небольшой, но достаточной скорости (рис. 5.74).

Для краткости поместите регулятор для движения по линии с контролем скорости в подпрограмму движения (рис. 5.75). Переменная *dist* задаёт уставку регулятора скорости 10 см. Вторая подпрограмма обеспечит реакцию на объект (рис. 5.76).

Вторая задача, которую следует решить, — разворот на линии. Если робот смог выполнить подпрограмму реакции на объект со звуковым сигналом, то у него получится и остальное. Разворот на 180° не составит труда (рис. 5.77).

Третья задача — объезд. Она самая сложная. Траекторию объезда можно выбрать самостоятельно: это будет дуга, прямоугольник или даже треугольник (с основанием на линии движения). Во всех случаях в начале движения необходимо выполнить поворот: на 90° в случае прямоугольника или полукруга и на меньший угол в случае треугольника (рис. 5.78).

Во всех случаях объезд должен завершаться ожиданием момента, когда датчик освещённости пересечёт линию с последующим поворотом. После поворота датчик должен оказаться с той стороны от линии, с которой он

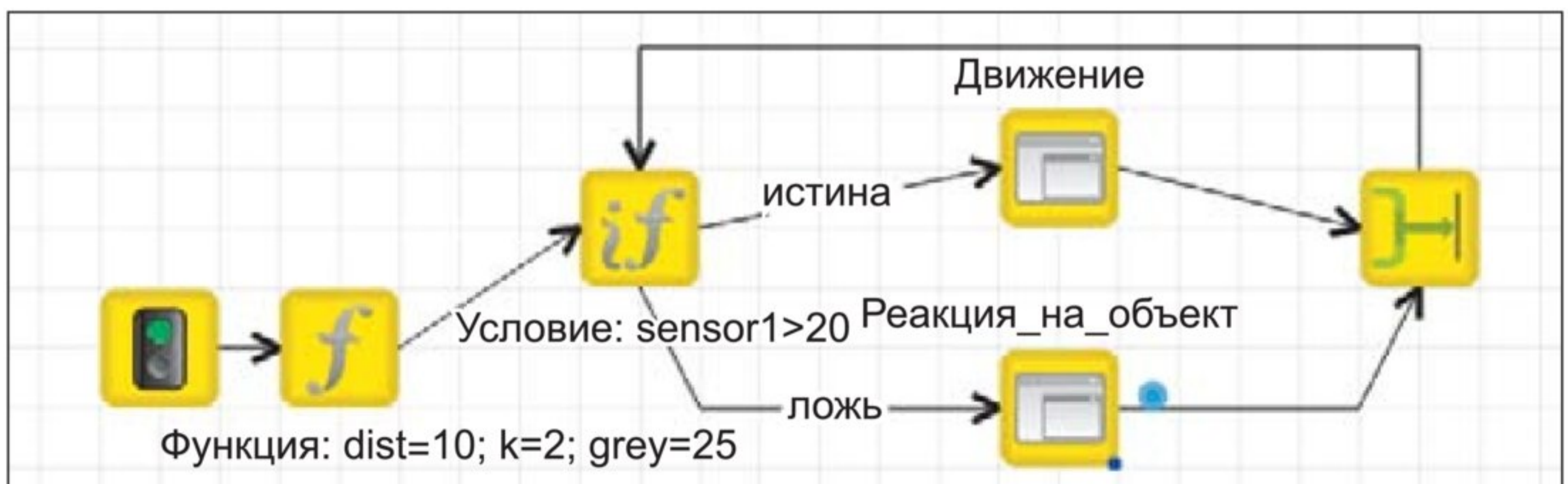
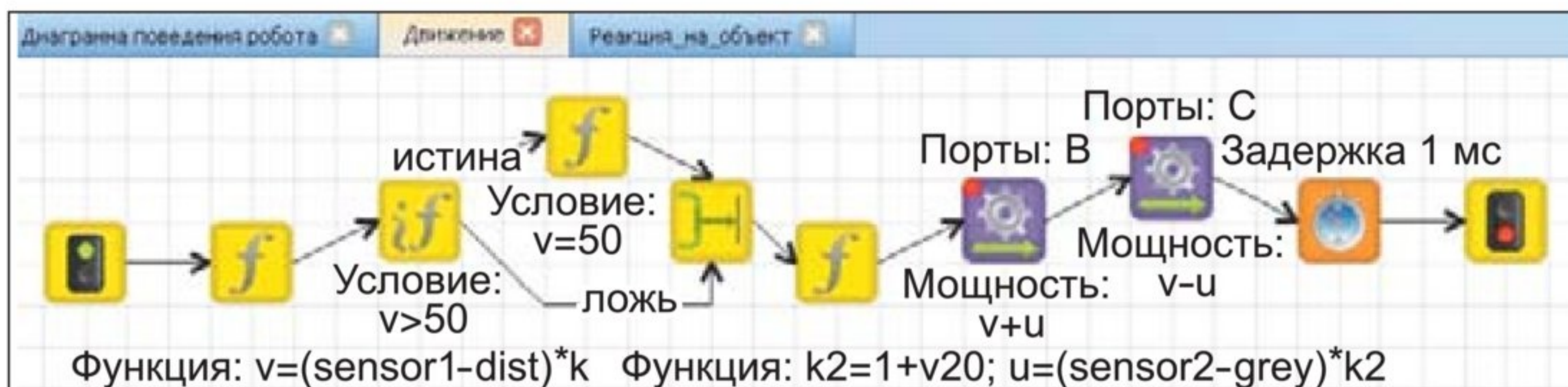
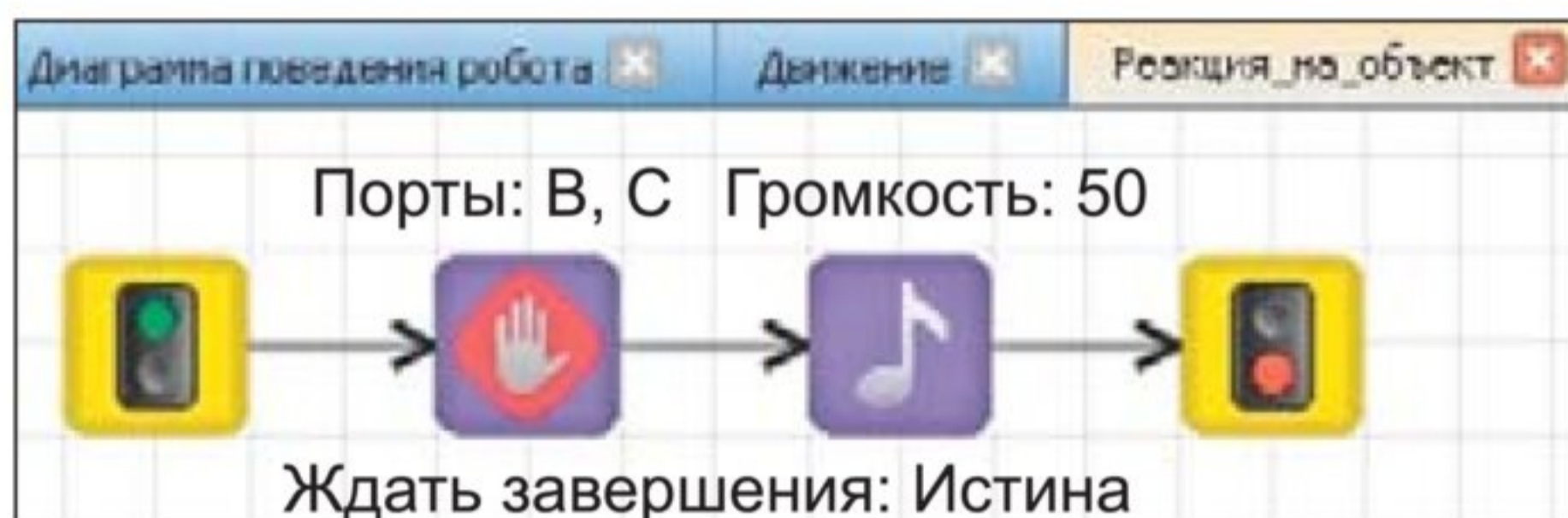


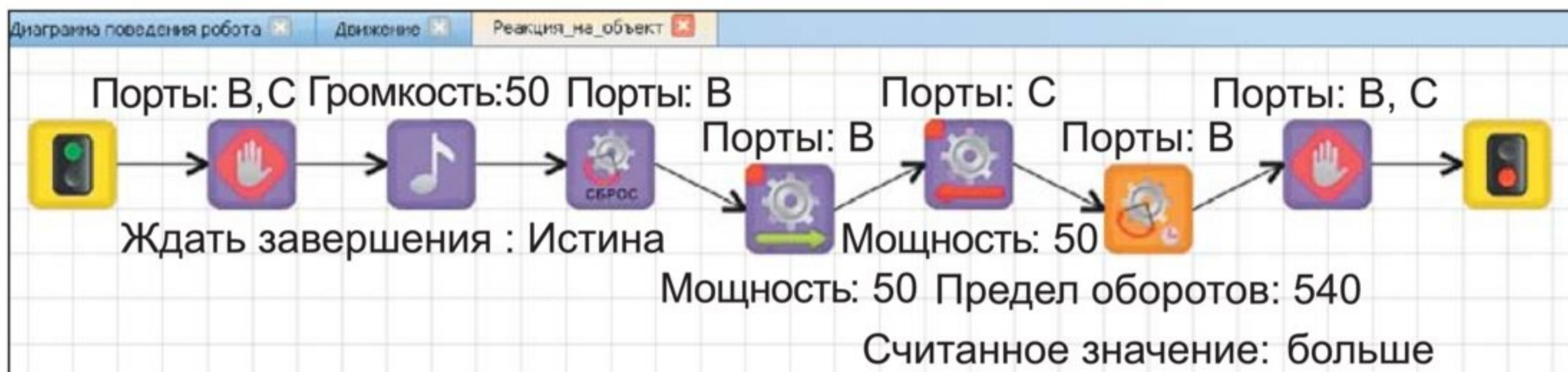
Рис. 5.74. Программа движения по линии с реакцией на объект



**Рис. 5.75.** Подпрограмма движения по линии с контролем скорости



**Рис. 5.76.** Подпрограмма реакции на объект

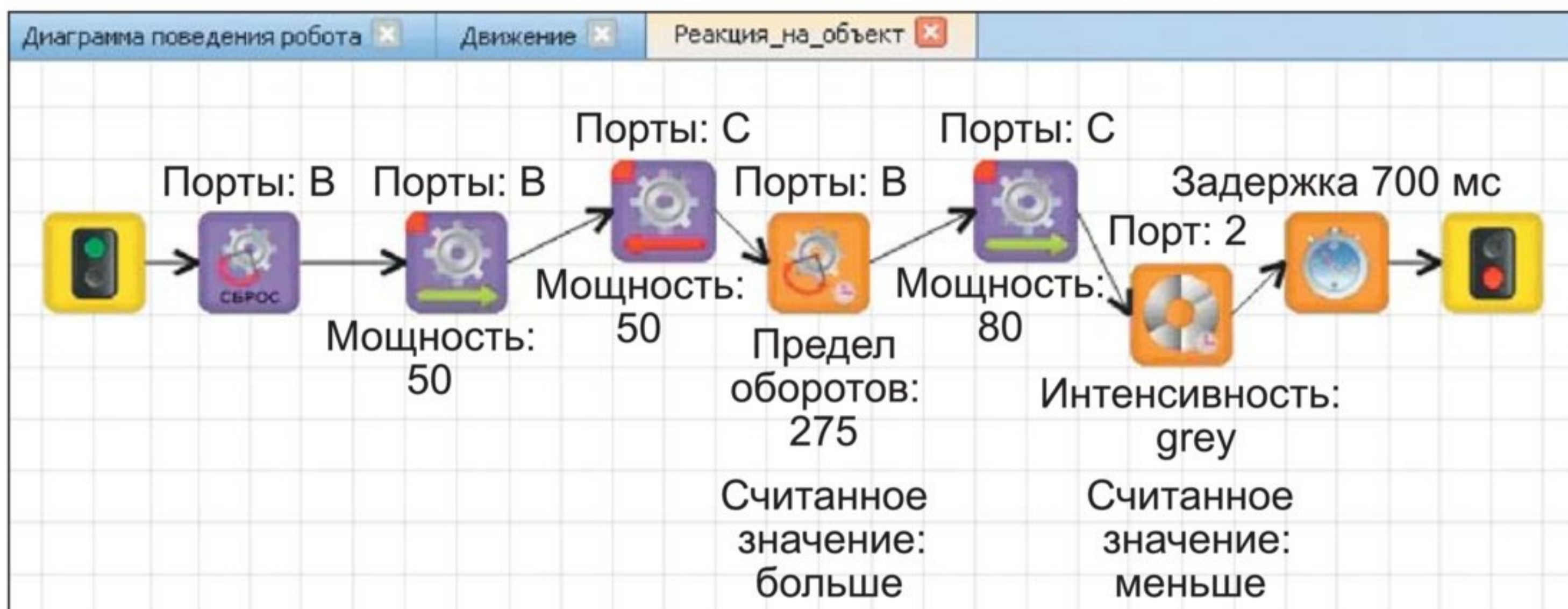


**Рис. 5.77.** Подпрограмма реакции на объект с разворотом



**Рис. 5.78.** Возможные траектории объезда препятствия

находится при движении по линии, чтобы регулятор мог продолжить свою работу. Для этого может потребоваться выполнить небольшой проезд вперёд после нахождения



**Рис. 5.79.** Подпрограмма для объезда препятствия по дуге

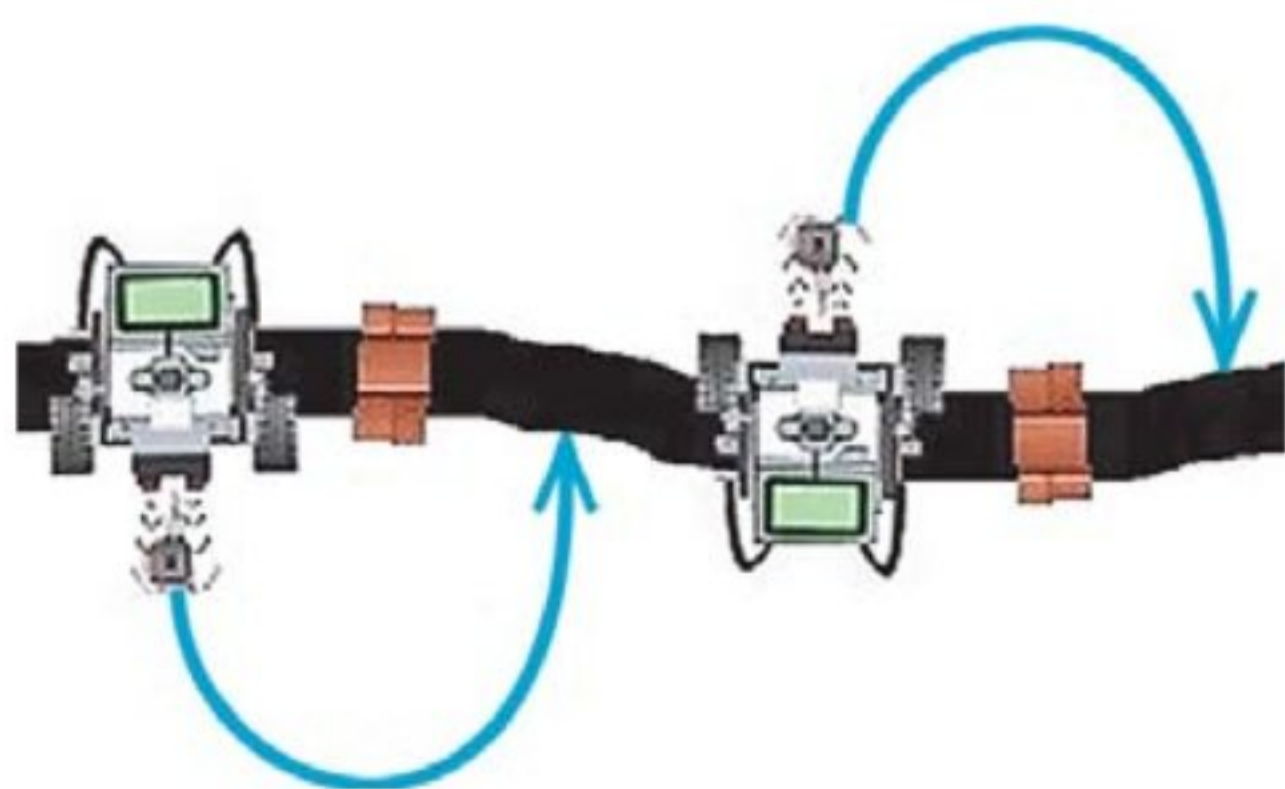
линии. Если всё сделано правильно, то и последний поворот выполняется на минимальное число градусов: при правильном положении датчика его завершит регулятор для движения по линии (рис. 5.79).

Для решения такой задачи на соревнованиях по робототехнике в России существует дисциплина «Слалом». Её особенность в том, что робот должен последовательно объезжать препятствия на линии то слева, то справа (рис. 5.80).

Эта задача может быть решена с помощью специальной переменной — флага, которая чередует свой знак. При положительном значении флага выполняется объезд справа, при отрицательном — объезд слева. По окончании объезда знак переменной изменяется. Очевидно, что в самом начале программы необходимо задать значение переменной  $flag = 1$ .

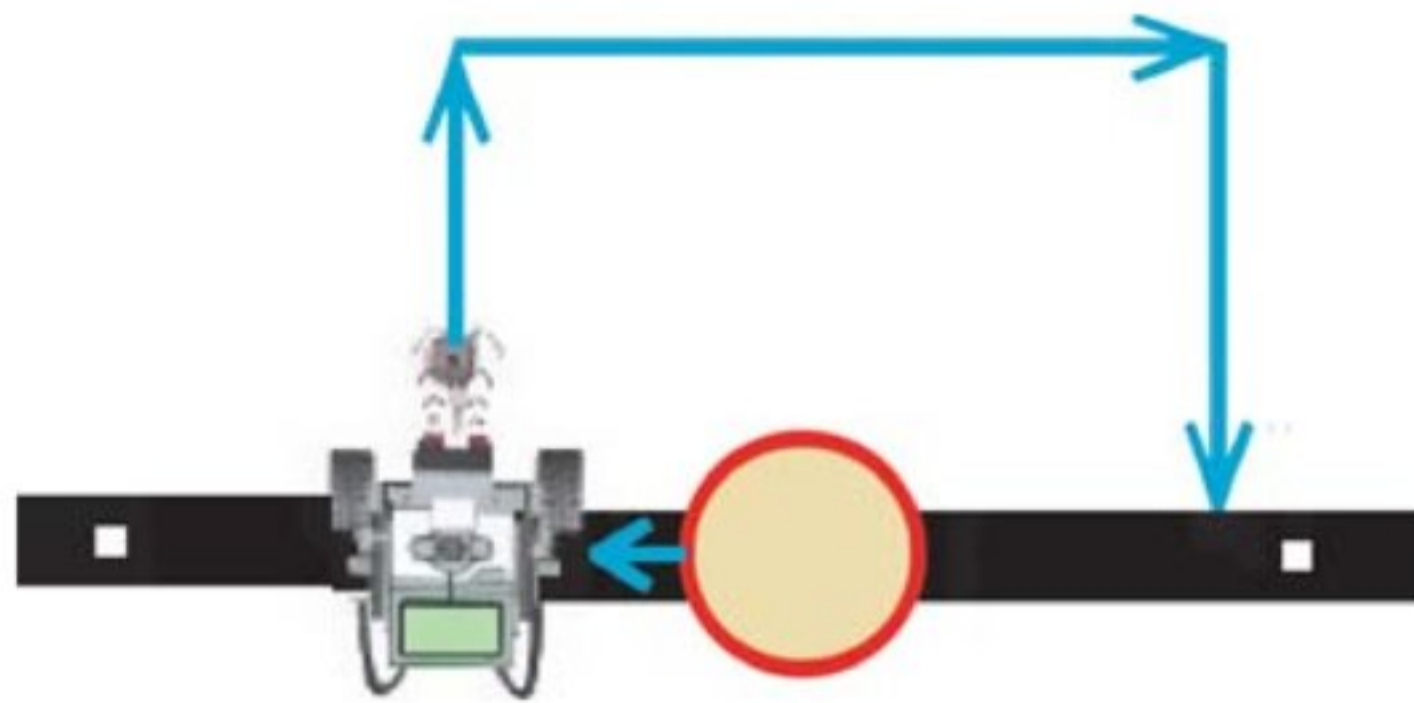
Для решения этой задачи создайте две подпрограммы — «Объезд справа» и «Объезд слева» (рис. 5.81), а в подпрограмме «Реакция на объект» постройте ветвление (рис. 5.82).

Объезд справа уже реализован, а вот над объездом слева



**Рис. 5.80.** Объезд объектов в режиме чередования





**Рис. 5.83.** Объезд помехи, движущейся по линии между метками

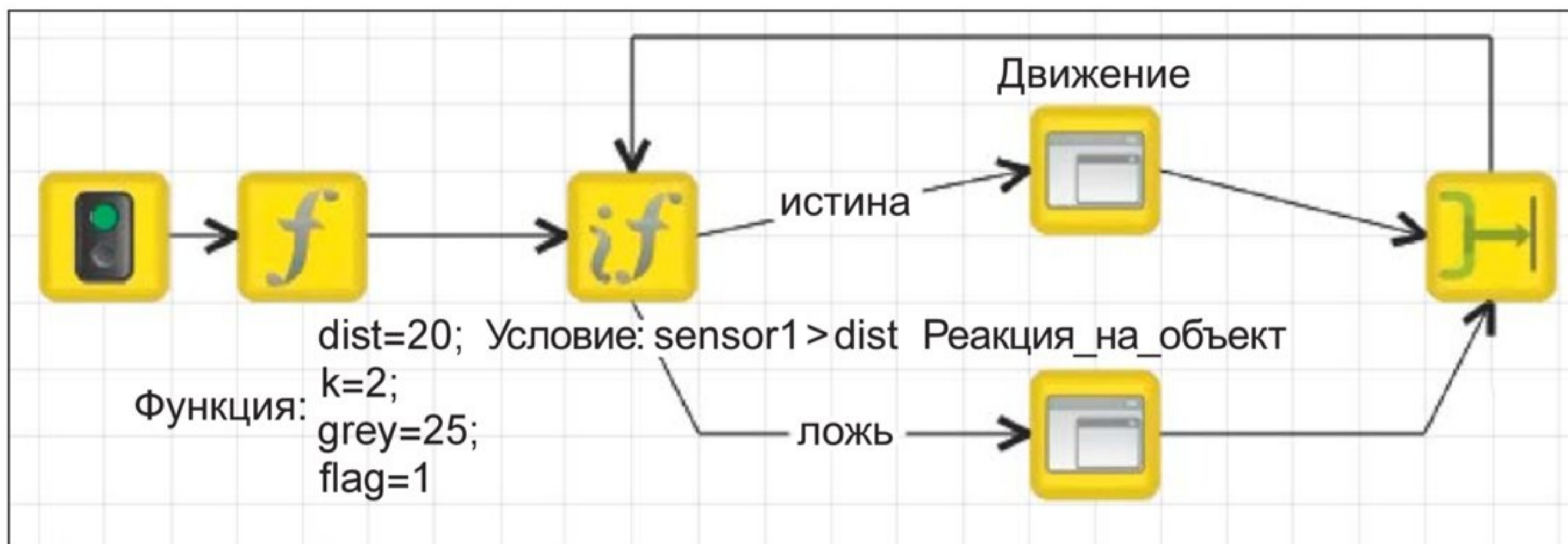


**Рис. 5.84.** Ультразвуковой радар

дуть вслепую, то есть необходимо хотя бы приблизительно контролировать, где находится помеха. Во избежание столкновения робот должен иметь возможность развить скорость, существенно превышающую скорость движения помехи. Простейший вариант — объезд помехи в момент её приближения. На обычной дороге такое можно увидеть разве что в боевике, в котором главный герой несётся на автомобиле по встречной полосе. Но в нашем случае это задача, решаемая с помощью простой математики.

Как оценить скорость помехи? Для этого потребуется сделать как минимум два замера расстояния и затем разность расстояний разделить на длительность промежутка времени между замерами. По этому принципу работает ультразвуковой радар инспектора дорожно-патрульной службы (рис. 5.84). Но если датчик установлен на роботе, придётся также учесть и его скорость движения.

Оказывается, алгоритм можно построить, не измеряя скорость специально. Для этого следует воспользоваться особенностью работы пропорционального регулятора по расстоянию, на малых отклонениях которого робот не может достичь цели. То есть, если помеха стоит на месте, а роботу необходимо к нему подъехать на заданное расстояние, скорее всего, он не доедет 1–2 см. Если помеха уезжает, робот будет медленно двигаться за ней. Если же помеха движется в сторону робота, это компенсирует недостающее отклонение в работе регулятора, и робот может среагировать на объект (рис. 5.85). Объезд помехи на встречном



**Рис. 5.85.** Программа объезда помехи, движущейся навстречу

движении более выгоден: скорее всего в конце объезда помеха будет далеко от робота. Но этого не скажешь об объезде при движении помехи в попутном направлении.

Остается только добавить, что в состязании «Дорога» объезд разрешён с любой стороны, но на поворотах линия становится пунктирной. Поэтому для проезда потребуется как минимум два датчика освещённости.

## Проверьте себя

1. Почему пропорциональный регулятор для контроля дистанции не всегда достигает цели? Как можно решить эту проблему?
2. С какого действия начинается объезд объекта?
3. Какие траектории можно использовать для объезда объекта?
4. Что должен ожидать робот при возврате на линию?
5. С какой стороны от линии должен оказаться датчик по окончании объезда объекта?
6. Составьте программу, по которой будет определена скорость движущегося по линии робота.
7. Используя три датчика освещённости, постройте и запрограммируйте робота-помеху.
8. Постройте алгоритм для состязания «Слалом» с двумя датчиками освещённости.
9. Постройте алгоритм для состязания «Дорога» с двумя датчиками освещённости.





## § 5.11. Транспортировка предметов



**Рис. 5.86.** Робот-транспортировщик радиоактивных отходов

Главное назначение роботов — помощь человеку. Сложную или монотонную работу робот выполняет лучше, чем человек. Одна из таких работ — транспортировка предметов. Она может встретиться везде — на дороге, на складе, в общественном месте, при строительстве, при добыче полезных ископаемых, на опасном производстве (рис. 5.86).

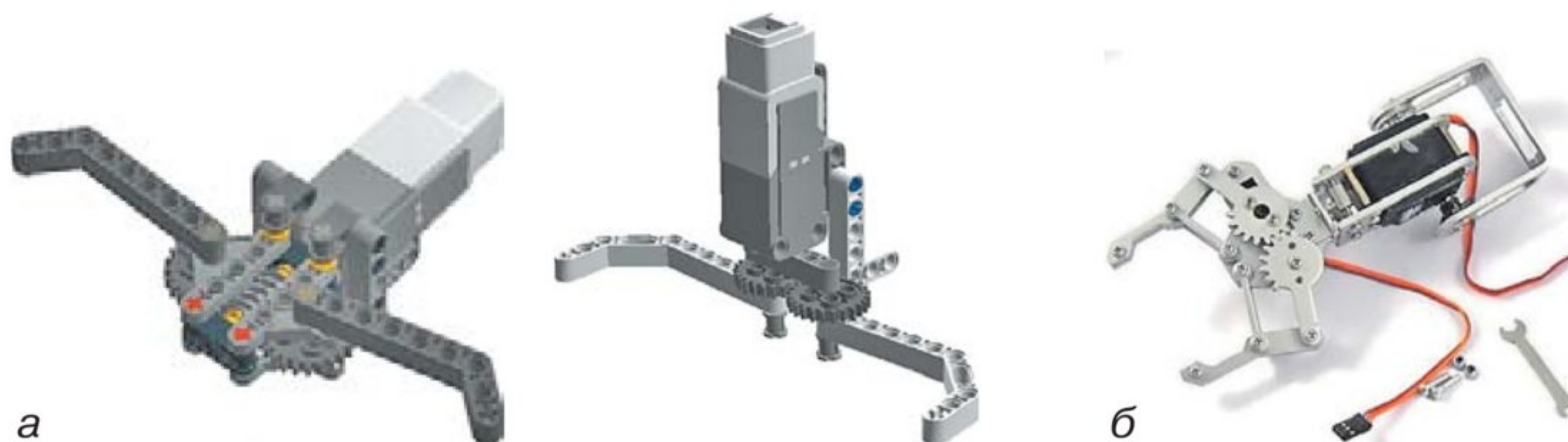
Для робота эту задачу можно разбить на несколько этапов, входящих в два основных режима: поиск и транспортировка.

1. Поиск объекта.
2. Захват объекта.
3. Транспортировка объекта к месту назначения.
4. Отпускание объекта.

Если объектов несколько, то робот может выполнять задачу в циклическом режиме: завершив транспортировку одного объекта, он сразу переходит в режим поиска другого.

Для транспортировки объекта потребуется создать новую конструкцию для робота — **захват**. Значит, будет использован третий двигатель. В некоторых робототехнических конструкторах для захвата используется такой же двигатель, что и для перемещения, в других — специальный серводвигатель с контролем положения (рис. 5.87).

Особенностью приведённых на рисунке захватов является наличие вращающихся в разных направлениях одинаковых шестерёнок, которые соприкасаются либо напрямую, либо через червячный механизм (рис. 5.88). Захват на червячном редукторе является наиболее прочным: он не ослабляется даже при отключении моторов. Это связано с большим передаточным отношением  $i = \frac{24}{1}$ . Однако



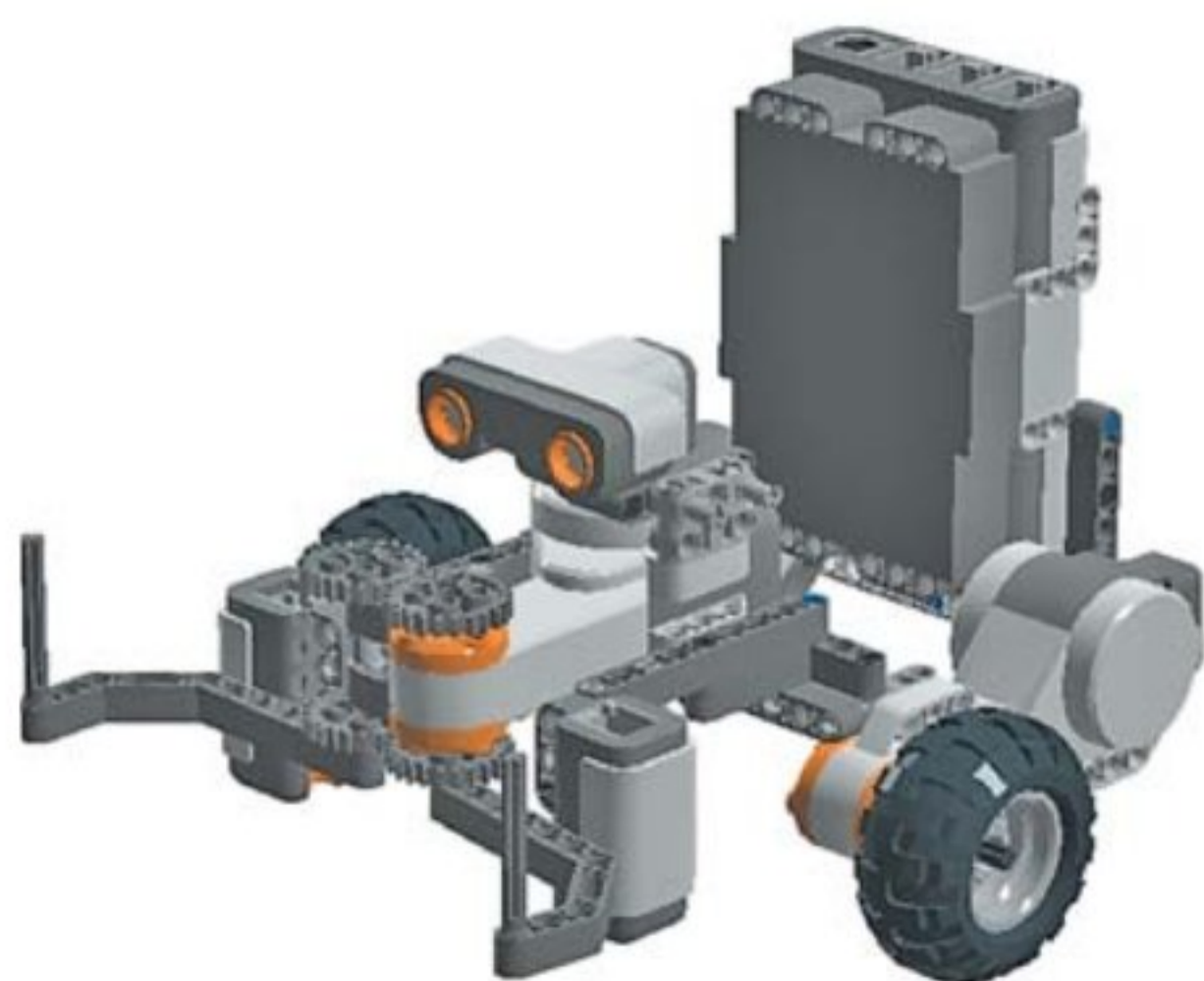
**Рис. 5.87.** Захваты на основе моторов EV3 (а) и на основе серводвигателя (б)

**Рис. 5.88.** Расположение шестерёнок и клешней в захвате

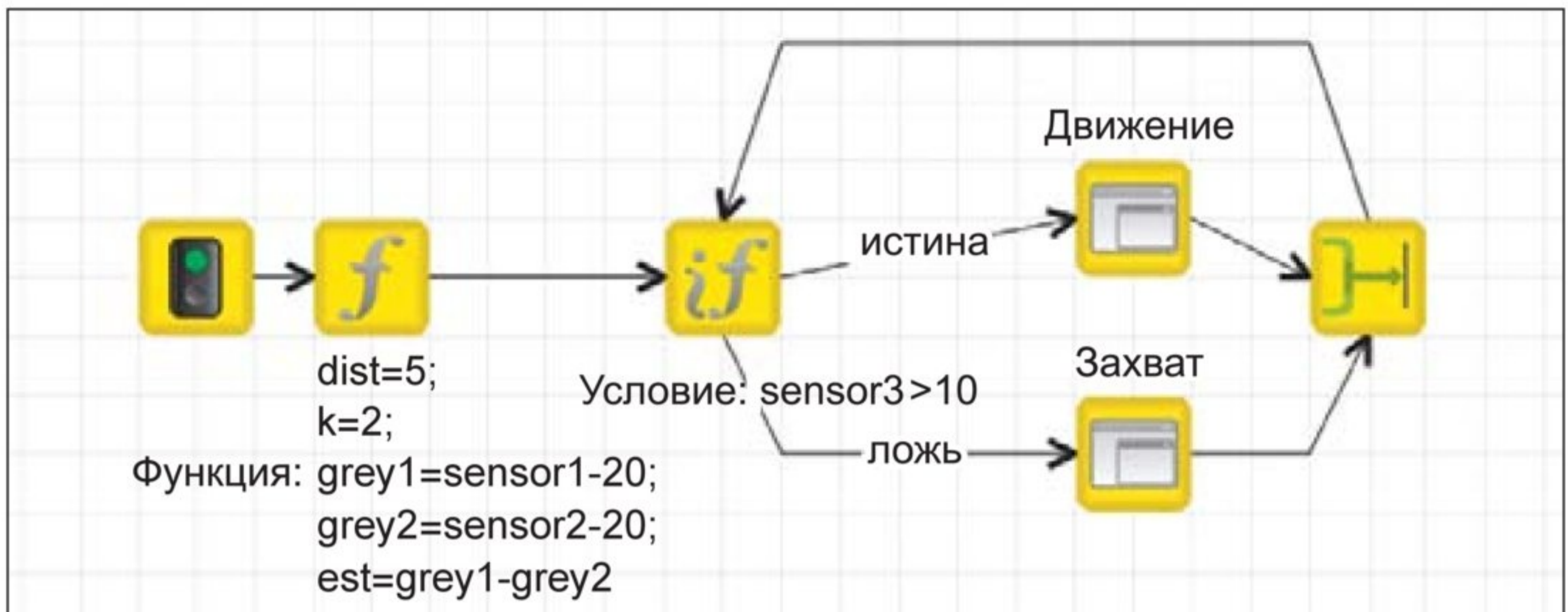


на таком захвате падает чувствительность и скорость (для поворота на  $90^\circ$  требуется около 1 секунды). Впрочем, при обращении со стационарными предметами высокая скорость не требуется.

Постройте робота с захватом, который может двигаться по линии с двумя датчиками освещённости (рис. 5.89). Для определения объекта робот будет использовать датчик расстояния.



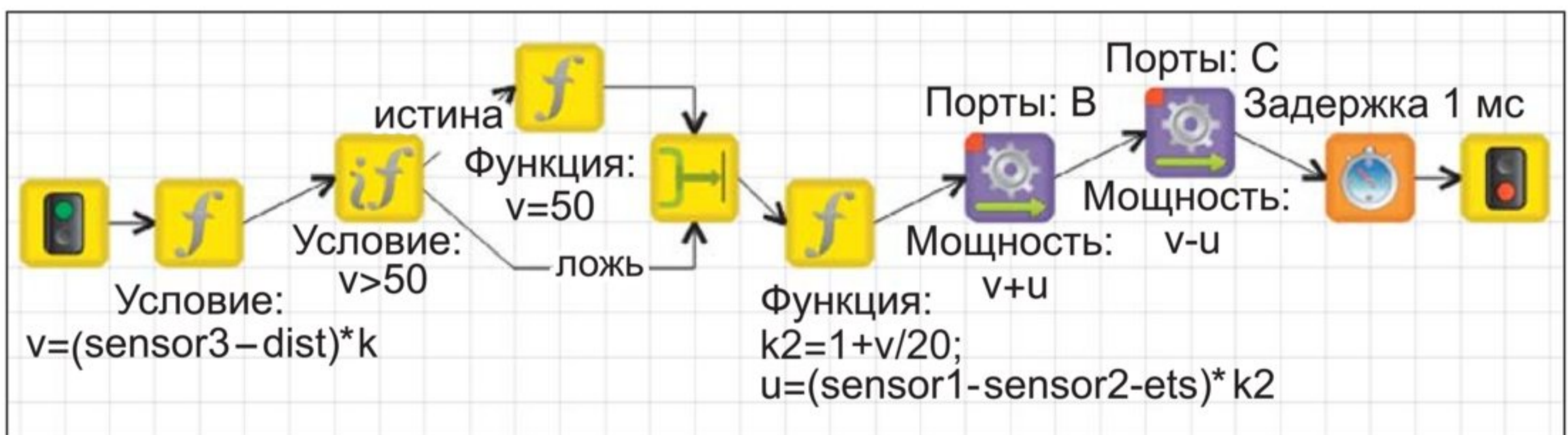
**Рис. 5.89.** Варианты мобильного робота с захватом



**Рис. 5.90.** Программа движения по линии с захватом объекта

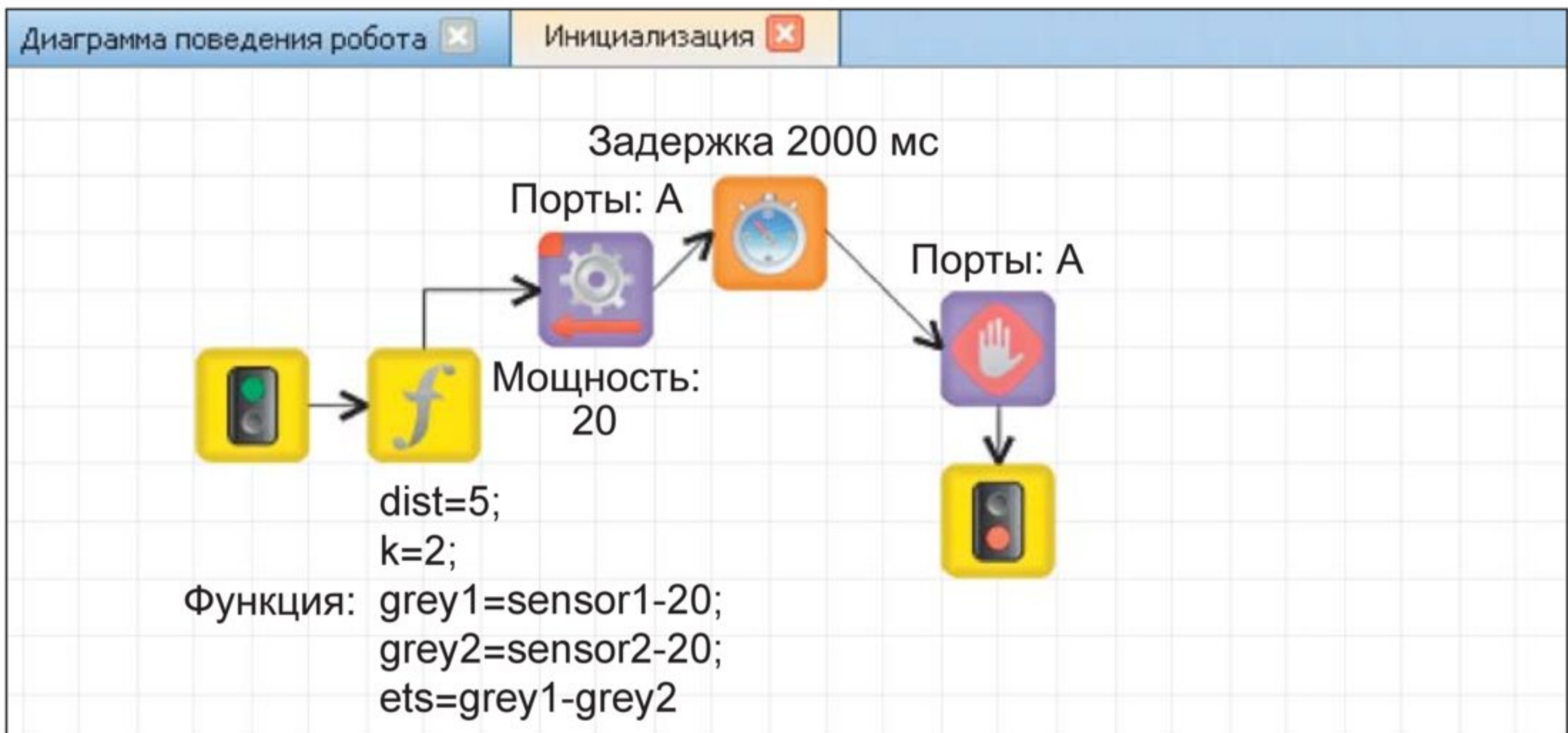
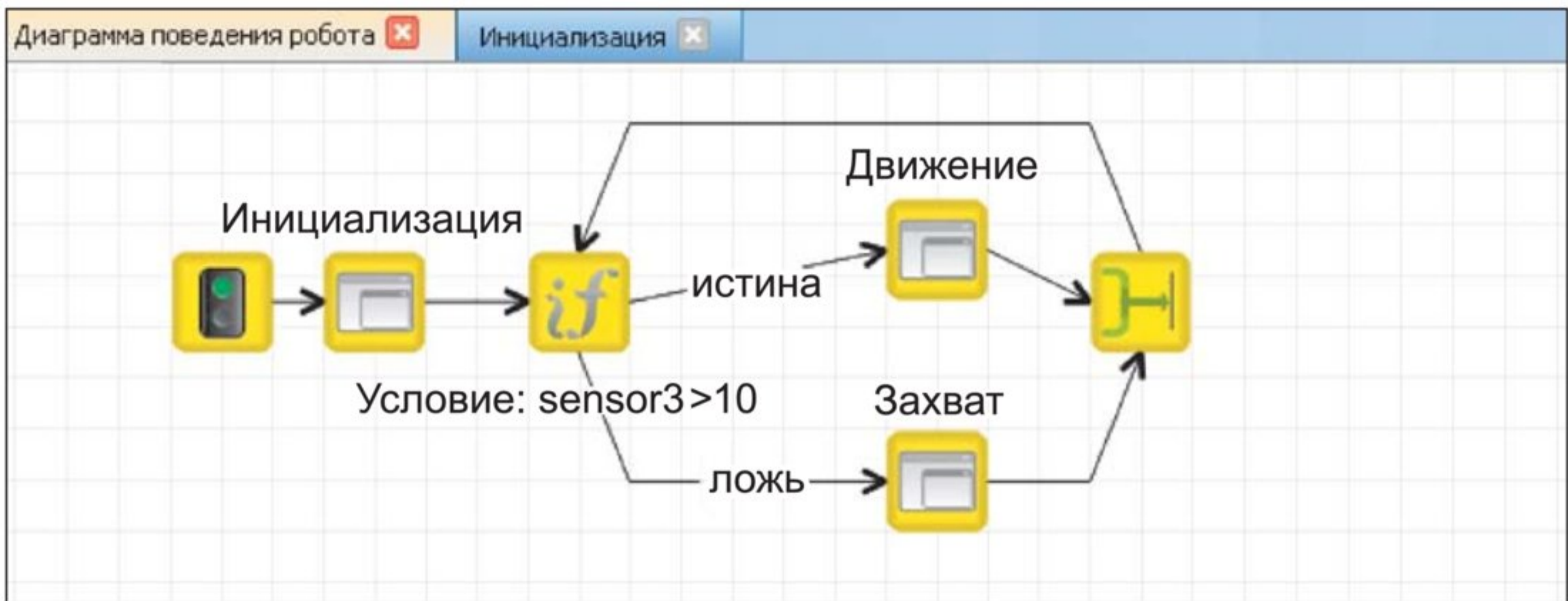


**Рис. 5.91.** Подпрограмма захвата объекта



**Рис. 5.92.** Подпрограмма движения с двумя датчиками освещённости

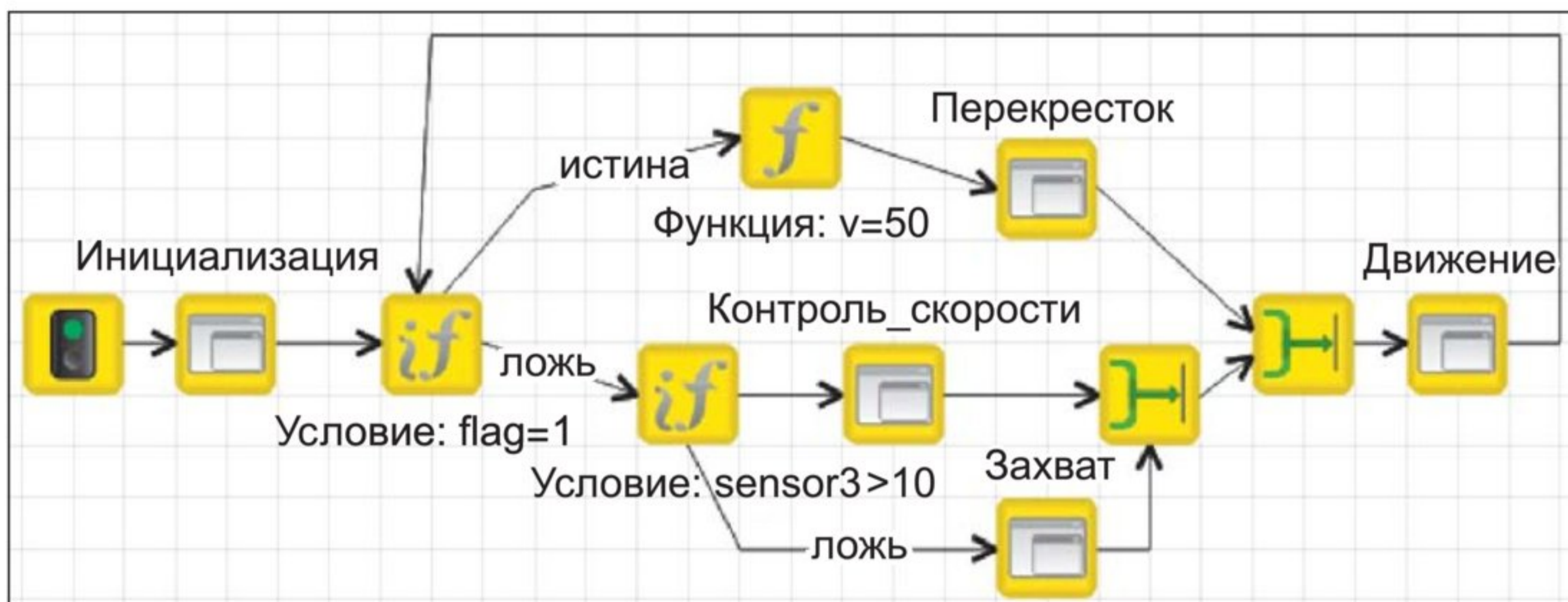
Составьте программу, по которой робот плавно остановится возле объекта на линии и произведёт захват (рис. 5.90, 5.91 и 5.92). Ожидаемое расстояние до объекта нужно уменьшить, иначе робот захватит воздух. По-



**Рис. 5.93.** Добавление подпрограммы «Инициализация»

сколько на работе используется три датчика, их следует установить на следующие порты: на 1 и 2 — датчики освещённости, на 3 — датчик расстояния.

После нескольких запусков станет понятно, что прежде чем закрыть клешни захвата, робот должен раскрыть их. Добавьте подпрограмму «Инициализация», в которой будут заданы значения переменных, произведена калибровка и раскроются клешни (рис. 5.93). Всё это робот должен сделать до начала движения.



**Рис. 5.94.** Программа с переходом этапов: транспортировка и выгрузка

*Инициализация* — это приведение программы или устройства в состояние готовности к использованию.

Следующая задача — продолжить движение с объектом. Для её решения потребуются переменная *flag*, которая определяет наличие объекта в захвате робота. В зависимости от флага меняется поведение робота. Если  $flag = 0$ , робот ищет объект. Если  $flag = 1$ , робот просто едет по линии. Начальное значение  $flag = 0$  следует указать в блоке инициализации. Надо учитывать, что при захваченном объекте обзор датчика расстояния будет перекрыт и контроль скорости по расстоянию до объекта будет невозможен. В этом случае скорость задаётся числом, например,  $v = 50$ . Значит, придётся разделить контроль скорости и регулятор движения по линии (рис. 5.94).

Теперь не составит труда при захвате добавить разворот на  $180^\circ$  с движением в обратном направлении. Объект найден и захвачен, можно возвращаться обратно.

Местом выгрузки объекта будет ближайший перекрёсток на линии. Создайте подпрограмму «Разгрузка», которая будет вызвана на перекрёстке. В этой подпрограмме необходимо задать значение переменной  $flag = 0$ . В режиме, когда объект захвачен, робот двигается по линии в ожидании перекрёстка. На перекрёстке объект будет выгружен, робот развернётся и отправится на поиски следующего объекта.



Теперь робот ведёт себя подобно Валли из известного мультфильма: всё полезное везёт к себе в домик. Такого робота можно оставить работающим на замкнутой линии с одним перекрёстком, и он будет занят полезным делом, пока не сядут батарейки.

## Проверьте себя



1. Сколько шестерёнок может быть использовано в захвате?
2. Как червячный редуктор изменяет поведение захвата?
3. Как робот может определить, что объект захвачен, если датчик расстояния перекрыт?
4. Какие действия необходимо произвести при инициализации робота?
5. Добавьте роботу датчик цвета и создайте алгоритм, по которому объекты разных цветов будут выгружены по разные стороны главного перекрёстка.
6. Постройте алгоритм для захвата и перемещения объектов, находящихся на краях линий перекрёстков.
7. Постройте алгоритм к заданию 5, по которому робот будет брать объекты только определённого цвета и транспортировать их на базу.

**Запомните** ♦ **Захват**



# Заключение

Вы освоили первую часть пособия «Уроки робототехники» и узнали, как устроены простейшие автоматы, роботы и их механизмы. Надеемся, вам удалось попробовать запустить алгоритмы управления на настоящих роботах, собранных из конструктора, или на виртуальных роботах в 2D-модели, встроенной в TRIK Studio. И, конечно, вам часто хотелось произнести главное слово робототехника: «Заработало!» Ведь основная цель технического творчества — создание устройств, исполняющих волю человека.

Если вы станете инженерами, то полученный опыт пригодится вам в работе. Если выберете иную профессию, то сможете использовать понимание принципов управления в технических системах в самых разных областях жизни. В недалеком будущем роботы станут главными помощниками человека в исследовании и улучшении окружающего мира.

Технический прогресс набирает обороты, и скоро то, что изучается на уроках как новый предмет «Робототехника», станет таким же обыденным явлением, как компьютеры или смартфоны.

# Оглавление

<b>Обращение к читателям</b> .....	3
<b>Предисловие автора</b> .....	5
Глава 1	
<b>Простейшие механизмы</b> .....	6
§ 1.1. Механизм, автомат, робот .....	7
§ 1.2. Знакомство с конструктором .....	12
§ 1.3. Механическая передача .....	20
§ 1.4. Ремённая и фрикционная передачи .....	28
§ 1.5. Соосный редуктор .....	34
§ 1.6. Построение трёхмерной модели .....	37
Глава 2	
<b>Моторные механизмы</b> .....	44
§ 2.1. Источники питания .....	45
§ 2.2. Электродвигатель .....	50
§ 2.3. Тягловые машины .....	59
Глава 3	
<b>Основы управления роботом</b> .....	66
§ 3.1. Контроллеры .....	67
§ 3.2. Среда программирования роботов .....	74
§ 3.3. Управление мобильным роботом .....	79
§ 3.4. Знакомство с датчиками .....	84



Глава 4	
<b>Возвратно-поступательное движение</b>	92
§ 4.1. От мультипликатора до маятника Капицы	92
§ 4.2. Возвратно-поступательное движение, маятник Капицы	96
§ 4.3. Шагающие роботы	100
Глава 5	
<b>Управление движением робота</b>	106
§ 5.1. Точные перемещения	106
§ 5.2. Путешествие в лабиринте	111
§ 5.3. Правило правой руки	120
§ 5.4. Защита от застреваний	125
§ 5.5. Простейшие регуляторы	130
§ 5.6. Следование по линии. Релейный и пропорциональный регуляторы	137
§ 5.7. Следование по линии с двумя датчиками. Калибровка	144
§ 5.8. Определение перекрёстков и действия на них	150
§ 5.9. Безаварийное движение	156
§ 5.10. Объекты на линии	161
§ 5.11. Транспортировка предметов	168
<b>Заключение</b>	174

*Минимальные системные требования определяются соответствующими требованиями программ Adobe Reader версии не ниже 11-й либо Adobe Digital Editions версии не ниже 4.5 для платформ Windows, Mac OS, Android и iOS; экран 10"*

*Учебное электронное издание*

**Филиппов Сергей Александрович**

**УРОКИ РОБОТОТЕХНИКИ.  
КОНСТРУКЦИЯ. ДВИЖЕНИЕ. УПРАВЛЕНИЕ**

**Составитель А. Я. Щелкунова**

*Художник В. Е. Шкерин*

*Корректоры Л. И. Трифонова, М. И. Васильева*

*Компьютерная верстка: О. Г. Лапко*

Подписано к использованию 10.05.17.

Формат 160×210 мм

Издательство «Лаборатория знаний»  
125167, Москва, проезд Аэропорта, д. 3  
Телефон: (499) 157-5272  
e-mail: [info@pilotLZ.ru](mailto:info@pilotLZ.ru), <http://www.pilotLZ.ru>



**Сергей Александрович Филиппов** — учитель и руководитель центра робототехники Президентского физико-математического лицея №239 в Санкт-Петербурге, тренер золотых медалистов WRO 2012 и победителей многих других соревнований, разработчик курсов для детей и педагогов, руководитель Всероссийского робототехнического лагеря, председатель попечительского совета фонда «Финист», организующего соревнования Робофинист, и поддерживающего детей, увлекающихся созданием роботов.

Учебное пособие знакомит с основами робототехники. Оно поможет научиться моделировать автоматические устройства с использованием робототехнических конструкторов LEGO® и TRIK, создавать алгоритмы управления роботами в среде TRIK Studio. Рассмотрены физические основы робототехники. Приведены интересные, расширяющие кругозор факты, касающиеся истории робототехники и её современных достижений.

Пособие может быть использовано на уроках технологии, занятиях в робототехнических кружках, при выполнении проектов и подготовке к участию в соревнованиях и олимпиадах.

Предназначено для школьников 5–6 классов и старше, а также всех, интересующихся робототехникой.

Не пропустите интересные проекты на платформе  
**LEGO® MINDSTORMS® EDUCATION EV3**



info@pilotLZ.ru  
www.pilotLZ.ru

