

DOI: 10.32517/2221-1993-2024-23-6-43-63

А. С. Левушкина

Средняя общеобразовательная школа № 31, г. Владимир, Россия

РАЗВИТИЕ ОСНОВ ИНЖЕНЕРНОГО МЫШЛЕНИЯ ШКОЛЬНИКОВ С ПОМОЩЬЮ СРЕДЫ TRIK STUDIO И/ИЛИ НА БАЗЕ ПЛАТФОРМЫ ARDUINO

Аннотация

Сегодня одна из важнейших задач российского образования — популяризация инженерно-технологических знаний среди школьников, профессиональная ориентация молодежи на получение инженерных специальностей. Такие ориентиры системы образования обусловлены особой важностью для российской экономики инженерной подготовки кадров, усилением внимания государства к научно-технологической грамотности молодежи.

С 1 сентября 2022 года вступил в силу новый Федеральный государственный образовательный стандарт основного общего образования, и для учащихся V—IX классов изучение робототехники стало обязательным в рамках дисциплины «Технология» (ранее ребята занимались робототехникой только в кружках и инженерных лицеях).

Для более эффективного освоения робототехники и овладения основами инженерных наук мы предлагаем включить интегрированный модуль робототехники в изучение раздела «Алгоритмы и программирование» курса информатики. Обучение робототехнике на уроках информатики может быть реализовано с помощью среды TRIK Studio и/или на базе платформы Arduino для развития основ инженерного мышления.

В статье представлены материалы для проведения уроков в среде TRIK Studio, а также описываются проекты, которые можно реализовать на платформе Arduino.

Ключевые слова: робототехника, обучение робототехнике, программирование, исполнитель Робот, робототехнический конструктор ТРИК, среда TRIK Studio, Arduino Uno.

1. Введение. Актуальность внедрения робототехники в обучение в школе для развития инженерного образования

Сегодня в России существует проблема недостаточной обеспеченности инженерными кадрами и низкого статуса инженерного образования. Для решения этой проблемы

в последнее время уделяется большое внимание концепции STEM-образования. Аббревиатура STEM расшифровывается как S — наука (*англ.* science), T — технология (technology), E — инженерное искусство (engineering), M — математика (mathematics). STEM-подход позволяет комплексно подходить к решению различных задач: как с научной, так и с практической точек зрения [1, 5].

Контактная информация

Левушкина Арина Сергеевна, учитель математики и информатики, средняя общеобразовательная школа № 31, г. Владимир, Россия; адрес: 600022, Россия, г. Владимир, ул. Завадского, д. 7; e-mail: arina.lan@yandex.ru

A. S. Levushkina

School 31, Vladimir, Russia

DEVELOPING THE BASICS OF ENGINEERING THINKING IN SCHOOLCHILDREN USING THE TRIK STUDIO ENVIRONMENT AND/OR BASED ON THE ARDUINO PLATFORM

Abstract

Today, one of the most important tasks of Russian education is the popularization of engineering and technological knowledge among schoolchildren, the professional orientation of young people to obtain engineering specialties. Such guidelines for the education system are due to the particular importance of engineering training for the Russian economy, and the increased attention of the state to the scientific and technological literacy of young people.

On September 1, 2022, a new Federal State Educational Standard for Basic General Education came into force, and for students in 5–9th grades the study of robotics became mandatory in the discipline "Technology" (previously, children studied robotics only in clubs and engineering lyceums).

For more effective development of robotics and mastering the basics of engineering sciences, we propose to include an integrated robotics module in the study of the "Algorithms and programming" section of the informatics course. Teaching robotics at informatics lessons can be implemented using the TRIK Studio environment and/or based on the Arduino platform to develop the foundations of engineering thinking.

The article presents materials for conducting lessons in the TRIK Studio environment, and also describes projects that can be implemented on the Arduino platform.

Keywords: robotics, teaching robotics, programming, executor Robot, TRIK robotic constructor, TRIK Studio environment, Arduino Uno.

Для того чтобы процесс обучения школьников в области инженерии был эффективен, необходимо найти такие области знаний, которые были бы понятны учащимся и вызывали у них мотивацию к наращиванию инженерных знаний и умений. Здесь важна роль школы, так как именно в школе у ребенка формируется осознанный выбор своей будущей профессии, связанной с наукоемкими технологиями и инженерными специальностями, закладываются первоначальные знания и кругозор в области инженерии.

Подготовка детей к изучению технических наук — это одновременно и обучение, и техническое творчество, она способствует воспитанию активных, увлеченных своим делом людей, обладающих инженерно-конструкторским мышлением. Очень важно как можно раньше выявить технические наклонности учащихся и развивать школьников в этом направлении, создав благоприятную соответствующую творческую среду.

Именно **робототехника** является мощнейшим средством развития уникальных навыков и способностей ребенка в различных областях технического творчества и, соответственно, может служить инструментом для профессиональной ориентации молодежи в области инженерно-технического образования [8].

С начала XXI века робототехника становится одним из приоритетных направлений в сфере экономики, машиностроения, здравоохранения, военного дела и других направлений деятельности человека. Сегодня на улицах можно видеть шагающих роботов, роботы-помощники широко используются на транспорте, в исследованиях Земли и космоса, в медицине, на конвейерном промышленном производстве, в массовом производстве промышленных товаров и товаров народного потребления, в быту применяются сенсорная автоматика и робототехника. Поэтому специалисты, обладающие знаниями в этой области, востребованы сегодня как никогда [10].

Образовательная робототехника представляет собой новую, актуальную педагогическую технологию, которая находится на стыке перспективных областей знания: механики, электроники, автоматизации, конструирования, программирования, схемотехники. Однако внедрение образовательной робототехники на уровне общего образования — сложная задача. Один из вариантов — введение в школьную программу интегрированного модуля образовательной робототехники. Интегрированный модуль представляет собой включение в курсы информатики блоков образовательной робототехники с использованием межпредметных связей.

Учителя должны быть готовы к введению робототехники на том уровне, который можно предложить учащимся в рамках изучения содержательной линии «Алгоритмы и программирование» в школьном курсе информатики. Но здесь возникают две проблемы: 1) недостаточная подготовка учителей информатики в области образовательной робототехники и 2) отсутствие материальной базы. Недостаточная подготовка учителей во многом связана с недостаточной разработанностью методики использования робототехники в учебном процессе, отсутствием учебных пособий для учащихся и методических рекомендаций для учителей [4]. Да, есть сборники с огромным количеством

задач, но у педагога зачастую нет времени на анализ и отбор этих задач. И не все предлагаемые в этих сборниках задания можно реализовать на уроках. Поэтому *есть необходимость в отборе и группировке по блокам заданий, подходящих для выполнения в школе и позволяющих изучить как темы курса информатики, так и основы робототехники.*

2. Опыт включения робототехники в изучение содержательной линии «Алгоритмы и программирование» школьного курса информатики

Робототехнические образовательные конструкторы позволяют эффективно формировать ключевые компетенции обучающихся в области информатики, в частности, по разделу «Алгоритмы и программирование».

Изучение алгоритмизации невозможно без знакомства с понятиями «исполнитель», «формальный исполнитель», «система команд исполнителя». Овладение школьниками этими понятиями происходит, как правило, с помощью «КуМира» — системы программирования, предназначенной для поддержки начальных курсов информатики и программирования в средней школе. Исполнители в этой системе (Робот, Чертежник, Черепаша) выполняют программу, которая вводится в текстовом редакторе, все действия исполнителя отображаются на экране. В системе «КуМир» представлены также формальные исполнители Кузнечик, Вертун, Рисователь и др. Некоторые педагоги для первоначального знакомства школьников с исполнителями предпочитают использовать Робот-Blockly К. Ю. Полякова [7] или Робота образовательной платформы «Яндекс Учебник» [11].

Что объединяет привычные подходы к изучению перечисленных выше исполнителей? То, что учащимся сразу предлагаются команды СКИ и ученики принимают их как данность — они не задаются вопросом, как, например, получилось, что исполнитель умеет выполнять именно эту команду. Зачастую у них нет представления о том, что стоит за той или иной командой. Например, используя команду «движение вперед», они с трудом вспоминают про датчик расстояния, который необходимо подключать на определенный порт. Как работать с таким датчиком, знают единицы, а многие датчик расстояния даже и не видели, не говоря уже о его использовании на практике.

Нет сомнения в том, что использование робототехнических наборов позволит изменить преподавание, вывести его на новый уровень.

2.1. Исполнитель «робот-тележка» в среде TRIK Studio

Использование российского робототехнического конструктора ТРИК и среды программирования TRIK Studio позволило сделать новый шаг в образовательной робототехнике. Среда TRIK Studio [12] относится к визуальным средам программирования — программы в ней создаются с помощью пиктограмм. Но в этой среде программы можно писать и с использованием сложного текстового языка.

Среда TRIK Studio — продукт, рассчитанный как на начинающих, так и на «продвинутых» пользователей. Особенностью этой среды является режим отладки, или **интерактивный режим имитационного моделирования**, в котором можно наглядно видеть работу программы. В данном режиме команды выполняет «робот-тележка», который перемещается на экране и выполняет различные действия. Можно создавать линии и препятствия для реализации таких алгоритмов, как, например, «движение до препятствия», «движение вдоль линии». Возможность одновременного использования виртуального и сконструированного робота позволяет более гибко организовать процесс обучения как основам робототехники, так и основам алгоритмизации [9].

Предлагаемый ниже набор задач позволяет организовать изучение учащимися основных алгоритмических структур (следование (линейный оператор), ветвление (условный оператор, оператор выбора), цикл (цикл с параметром, цикл с предусловием, цикл с постусловием)) с помощью исполнителя «робот-тележка» в среде TRIK Studio.

2.1.1. Элементарные действия

Вспомним систему команд исполнителя Робот: направо, налево, кругом (разворот на 180°), вперед на n шагов, назад на n шагов — учащиеся сразу используют команды для организации движения Робота. При этом принимают их как данность, не задаваясь вопросами, как именно это происходит, благодаря чему. Например, Робот выполняет команду поворота направо — какие механизмы позволяют ему это сделать?

В случае включения в раздел «Алгоритмы и программирование» изучения элементов робототехники

учащиеся будут знать, *что стоит за каждой командой*. Для этого учитель может вывести ребят на мозговой штурм:

Учитель. Перед вами робот. Он умеет выполнять те же команды, что и виртуальный робот. Давайте подумаем, что позволяет роботу, например, двигаться вперед. Что нужно сделать для того, чтобы он начал движение? (*Возможный ответ: завести моторы.*)

В TRIK Studio для подачи мощности на мотор существует отдельный блок «Моторы вперед» (рис. 1). У этого блока два свойства: порты и скорость (мощность). (При использовании данного блока на экране отображается название «Скорость», хотя подразумевается, что вводится мощность, установленная в процентах. Диапазон подаваемой мощности: от –100 до 100 %.)

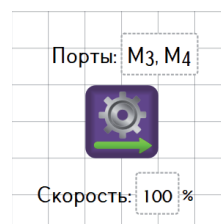


Рис. 1. Блок «Моторы вперед»

У контроллера ТРИК четыре порта для подключения силовых моторов: M1, M2, M3 и M4. Подключение моторов в 2D-модели по умолчанию: левый — к порту M3, правый — к порту M4. Если вас не устраивает подключение моторов по умолчанию, настройку подключения можно изменить в режиме отладки на центральной панели двумерной модели в разделе *Моторы* (рис. 2).

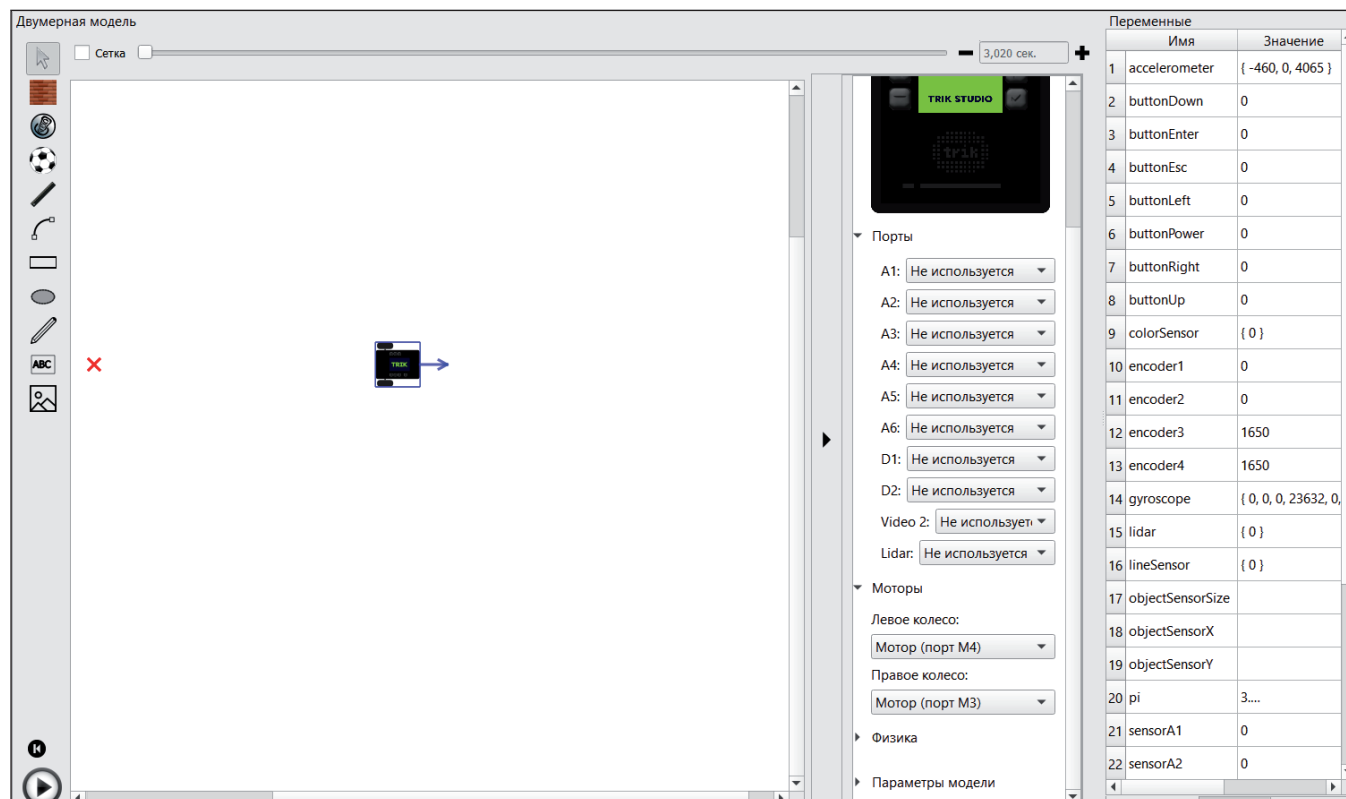


Рис. 2. Настройка подключения моторов

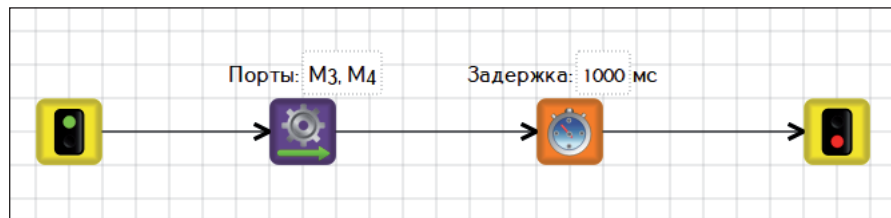


Рис. 3. Программа движения по таймеру

Действия для организации движения робота:

- вперед на n шагов;
- назад на n шагов;
- повороты направо, налево, кругом (разворот на 180°)

будем называть **элементарными действиями**.

Первое элементарное действие, которое мы рассмотрим, — **движение вперед**. Движение вперед базовой тележки задается подачей на левый и правый моторы одинаковой скорости (мощности).

При реализации команды «движение вперед» возможно **несколько вариантов для остановки этого движения**.

Вариант 1. Тайм-модель — движение осуществляется по таймеру.

В программе при организации движения вперед используется таймер для отсчета заданного количества времени в миллисекундах (рис. 3, 4).

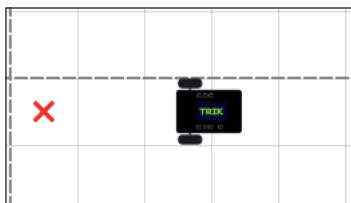


Рис. 4. Движение по таймеру (положение до/после выполнения)

Возникает вопрос: во всех ли задачах уместно организовывать движение робота, основываясь на показаниях таймера? (*Возможный ответ: нет, не всегда — например, если нам надо переместиться на определенное расстояние.*)

Вариант 2. Точные перемещения — использование блока «Ожидание значения энкодера».

В этом варианте нужно ждать, пока показание счетчика количества оборотов на заданном моторе не достигнет указанного значения параметра «Предел оборотов»

(см. реализацию задачи 1). Если мы установим значение оборота равным 360, то движение осуществляется ровно на одну клетку (в случае двумерной сетчатой модели) или на расстояние, равное 17,5 см (1 клетка = 17,5 см).

Вариант 3. Точные перемещения — проезд определенного расстояния.

В рамках первоначального знакомства учащихся с элементарными действиями подбор значений для подсчета количества оборотов, которые необходимо сделать роботу для проезда определенного расстояния, не является первостепенной задачей. Этим вопросом могут заняться те учащиеся, кто захочет детальнее изучить данный способ организации движения. В дальнейшем этот вариант движения не рассматривается.

При реализации решения задач будем использовать первые два варианта.

Задача 1. Движение вперед.

Движение вперед на точное количество клеток (задача типа «добраться до базы, стоянки»).

Итак, первая программа — **движение на одну клетку вперед** — будет выглядеть так, как показано на рисунке 5. Ее реализация представлена на рисунке 6.

Блоки, представленные на рисунке 5 (слева направо):

- «Начало» — начальная точка выполнения программы;
- «Сбросить показания энкодера» — сбросить показания количества оборотов моторов по заданным портам; порты устанавливаются по умолчанию;
- «Моторы вперед» — включить моторы по заданным портам с заданной мощностью;
- «Ждать энкодер» — ждать, пока показание счетчика количества оборотов на заданном моторе не достигнет указанного значения параметра «Предел оборотов»;
- «Моторы стоп» — выключить моторы по заданным портам;
- «Конец» — конец программы.

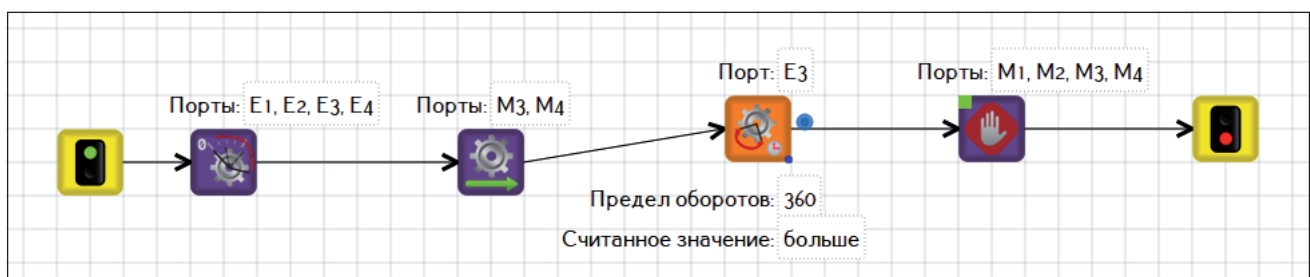


Рис. 5. Программа «Задача 1»

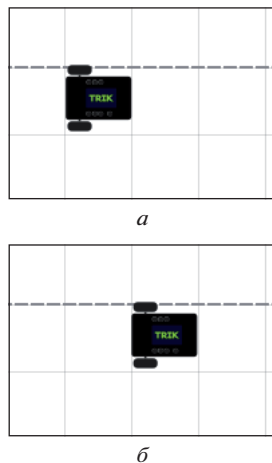


Рис. 6. Выполнение программы «Задача 1». Начальное (а) и конечное (б) положения робота: реализация движения на одну клетку вперед в двумерной модели с сеткой или же движения на определенное расстояние — в данном случае на 17,5 см

Задача 2. Движение назад.

Дойти до базы и вернуться в исходное положение.

Возможно несколько вариантов реализации данной задачи.

Вариант 1. Использование блока «Моторы назад».

Учащиеся наверняка предложат использовать этот блок (рис. 7) — по аналогии с использованием блока «Моторы вперед» в задаче 1.

Тогда учитель может задать вопрос: можно ли поменять в уже созданной программе движения вперед какие-то значения, чтобы робот ехал назад? То есть можно ли использовать для организации движения назад блок «Моторы вперед»?

Вариант 2. Для движения назад можно использовать блок «Моторы вперед», подав мощность -100% (рис. 8).

Вариант 3. Организация разворота (рис. 9).

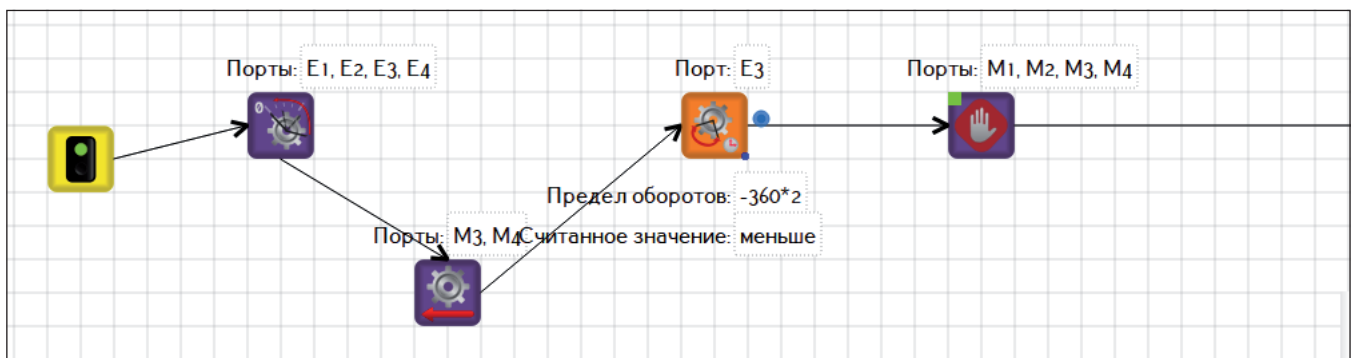


Рис. 7. Программа «Задача 2 (1)» с использованием команды «Моторы назад»

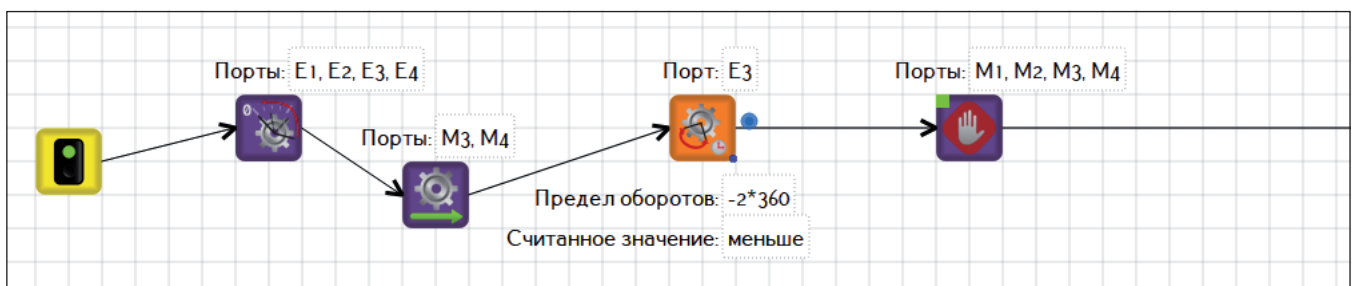


Рис. 8. Программа «Задача 2 (2)» с использованием команды «Моторы вперед» с мощностью, равной -100%

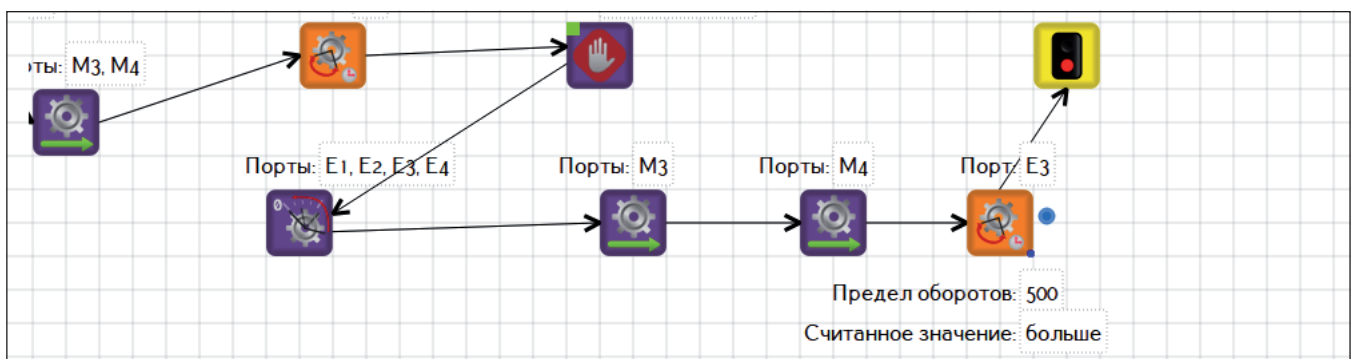


Рис. 9. Программа «Задача 2 (3)» с использованием команды разворота

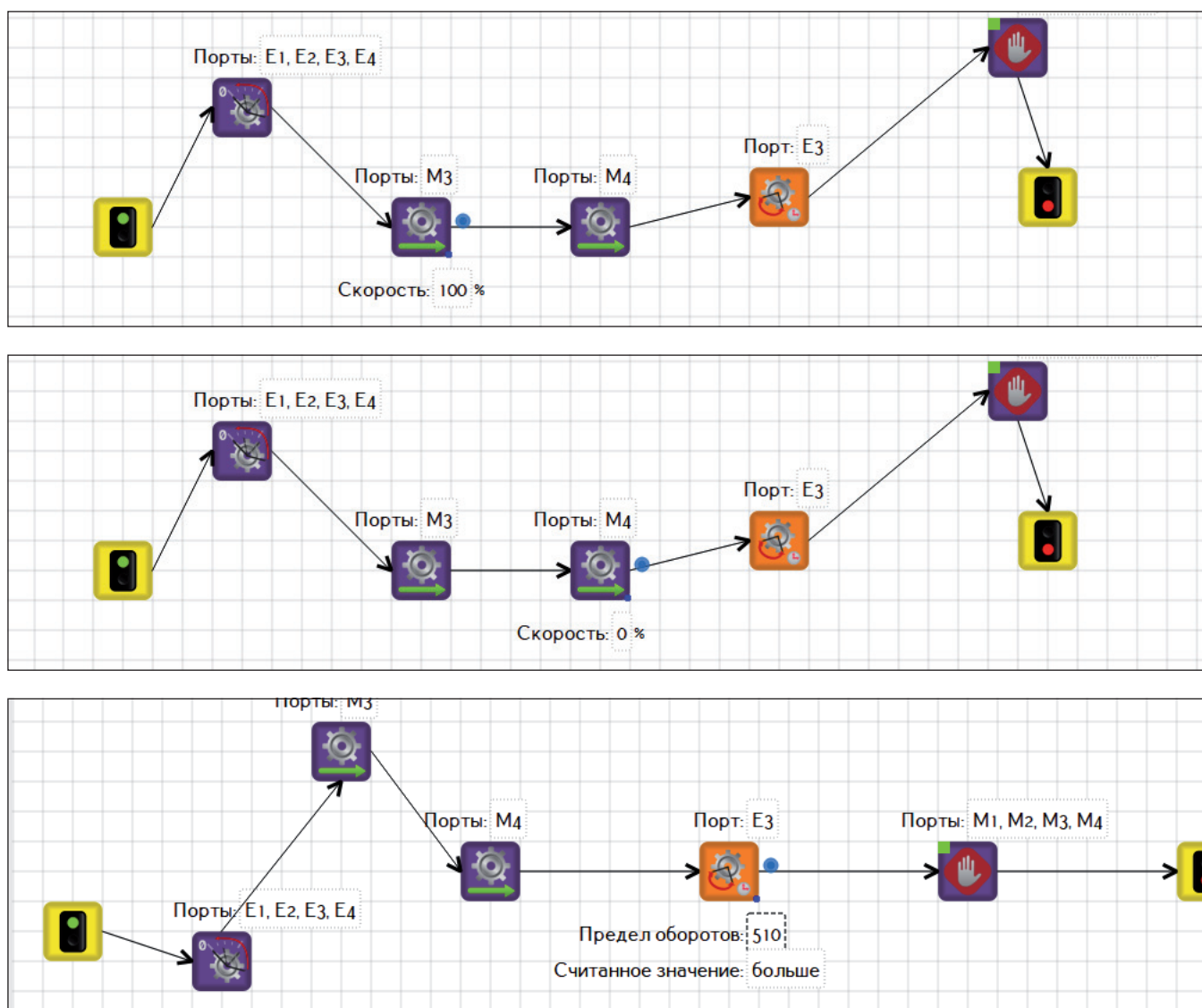


Рис. 10. Программа «Поворот налево»
(представлены три скрина, чтобы отобразить параметры в трех разных блоках, так как в программе отображаются значения только того блока, который нажат)

В представленном на рисунке 9 фрагменте разворот организован с помощью поворота на месте. Подается одинаковая мощность с разными знаками на два колеса: $M3 = 100$, $M4 = -100$. Предел оборотов считаем по показанию счетчика, который отсчитывает значения, поступающие от порта M3 на порт E3.

Следующие три варианта связаны с **организацией поворота**.

Повороты можно разделить на три типа:

- **резкий поворот** — мощность подается только на одно колесо;
- **плавный поворот** — мощность подается на два колеса, но на одно из них больше;
- **поворот на месте** — одинаковая мощность с разными знаками подается на два колеса.

Вариант 4. Резкий поворот.

Мощность подается *только на одно колесо*, т. е. одно колесо стоит, а второе крутится; например: $M3$ (правый мотор) = 100, $M4 = 0$ — крутится только правое колесо, происходит поворот налево (на 90°) (рис. 10).

Видим, что робот уехал от границы клетки и осуществил поворот (рис. 11).

Для **самостоятельного выполнения** учащимся предлагается составить программу поворота *направо*.

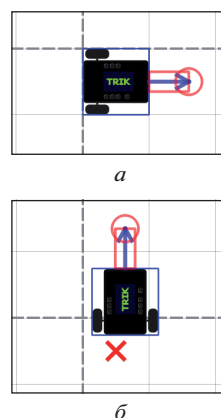


Рис. 11. Выполнение программы «Поворот налево». Начальное (а) и конечное (б) положения робота

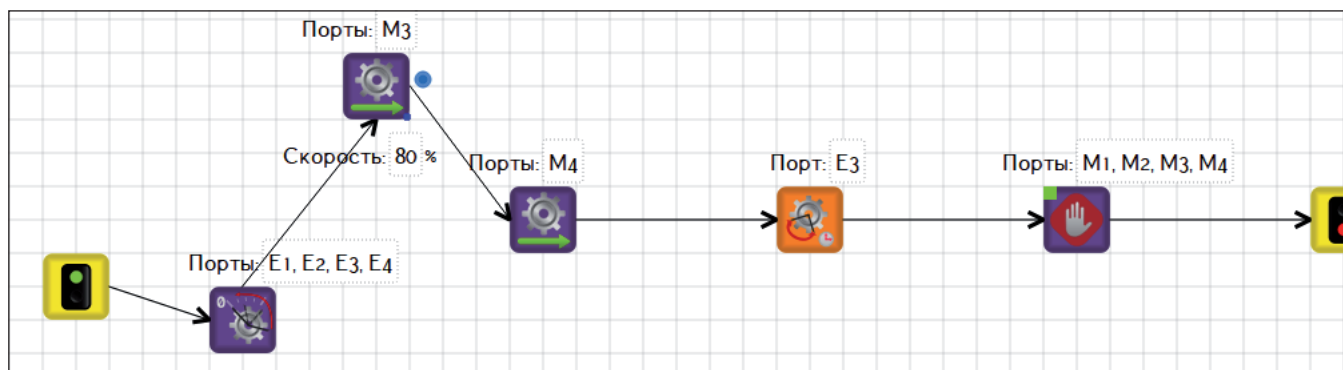


Рис. 12. Программа «Плавный поворот»

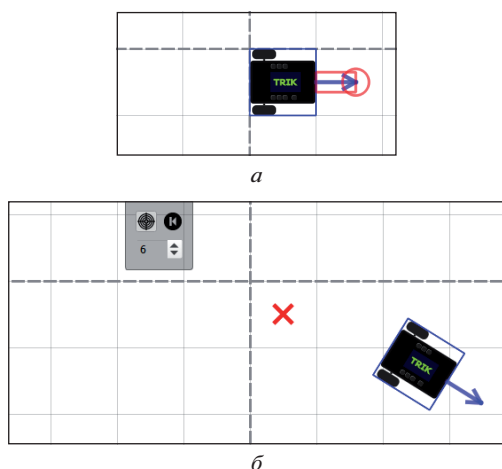
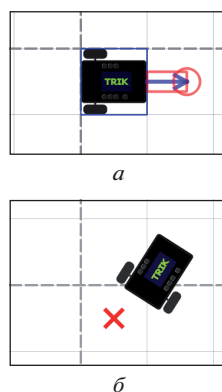
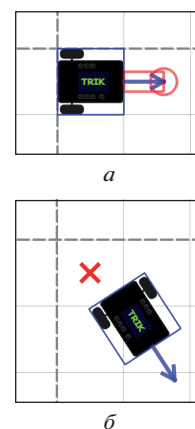


Рис. 13. Выполнение программы «Плавный поворот». Начальное (а) и конечное (б) положения робота

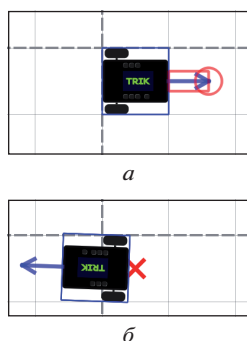
Рис. 14. Выполнение программы «Плавный поворот»: $M3 = 80, M4 = 30$. Начальное (а) и конечное (б) положения роботаРис. 15. Выполнение программы «Плавный поворот»: $M3 = 30, M4 = 80$. Начальное (а) и конечное (б) положения робота**Вариант 5.** Плавный поворот по кривой.

Мощность подается на два колеса, но на одно из них подается большая мощность (рис. 12–15).

Вариант 6. Поворот на месте.

Подается одинаковая мощность с разными знаками на два колеса (эта программа уже реализована — см. рис. 12, меняем только мощность) (рис. 16).

Когда учащиеся познакомились с тем, как реализовать команды, входящие в состав команд исполнителя «робот-тележка», можно предложить им такую задачу:

Рис. 16. Выполнение программы «Плавный поворот»: $M3 = 100, M4 = -100$. Начальное (а) и конечное (б) положения робота**Задача 3. Обогнуть угол.**

Организовать движение из стартовой позиции в конечную, выполнив следующие условия. Робот должен проехать вперед со скоростью 60, обогнуть угол, затем проехать вперед с максимальной скоростью и остановиться в зеленом круге (рис. 17). Использовать энкодерную модель [12].

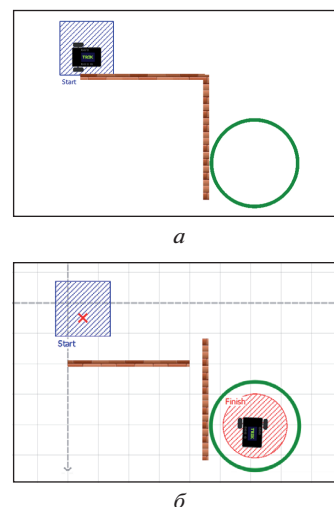


Рис. 17. Стартовая позиция робота (а) и результат выполнения программы «Задача 3» (б)

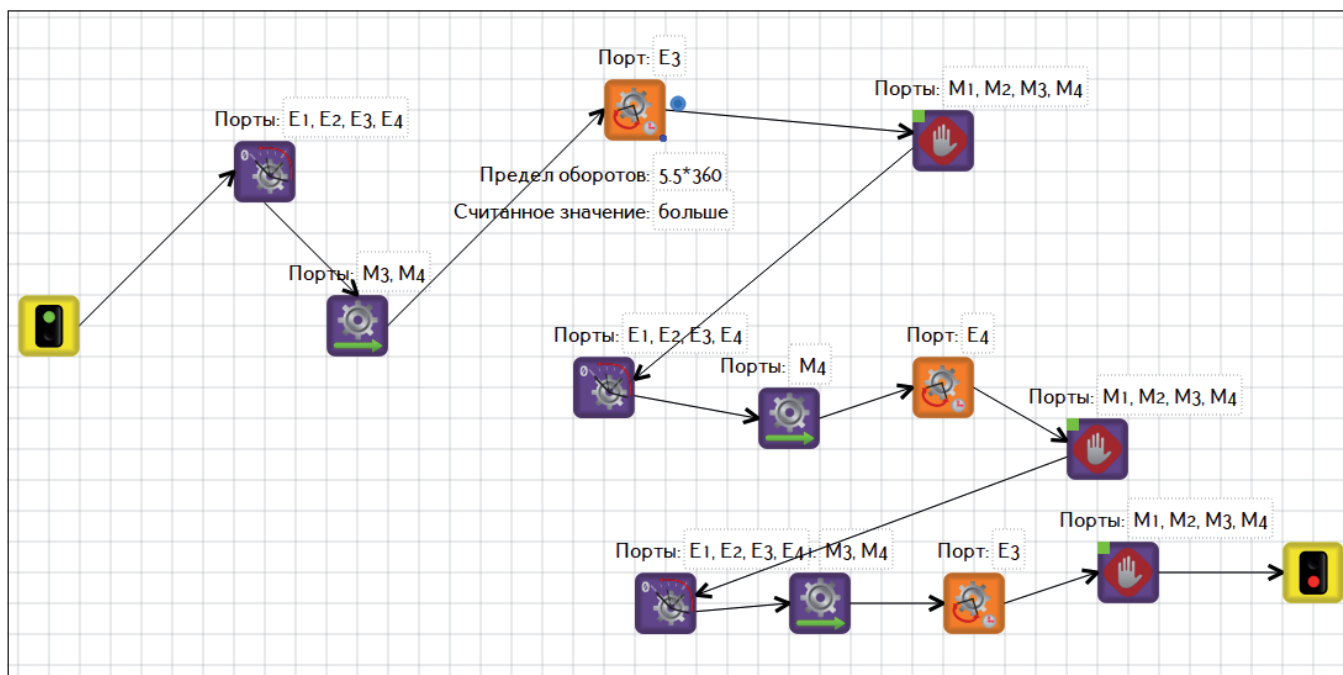


Рис. 18. Программа «Задача 3»

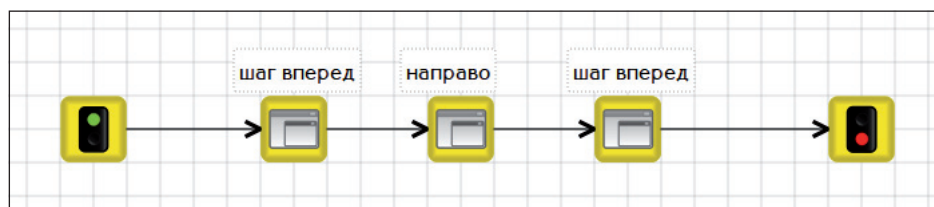


Рис. 19. Программа с подпрограммами

Стартовая позиция и результат выполнения программы представлены на рисунках 17, а и 17, б соответственно. Программа решения задачи представлена на рисунке 18.

Для сравнения: если мы будем решать эту задачу с помощью учебного исполнителя Робот в среде «Ку-Мир», то увидим, что команд будет гораздо меньше.

После знакомства с командами движения целесообразно вводить вспомогательные алгоритмы, подпрограммы для улучшения читаемости программы. Для этого необходимо действия для организации движения

Робота — вперед на n шагов, назад на n шагов, повороты направо, налево, кругом (разворот на 180°), — которые рассматривались выше, сохранить в библиотеке с соответствующими названиями команд (рис. 19).

Робот-тележка имеет блок «Рисование», т. е. может оставлять след при движении. Можно предложить следующую задачу (подобные задачи широко представлены на соревнованиях по спортивной робототехнике).

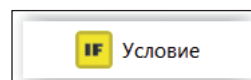
Задача 4. Соединение точек.

Соединить точки согласно шаблону (рис. 20).

Фрагмент программы и результат его выполнения представлены на рисунках 21 и 22 соответственно.

2.1.2. Оператор ветвления

Условному оператору **if** соответствует блок «Условие»:



Заканчивается оператор **if** блоком «Конец условия»:

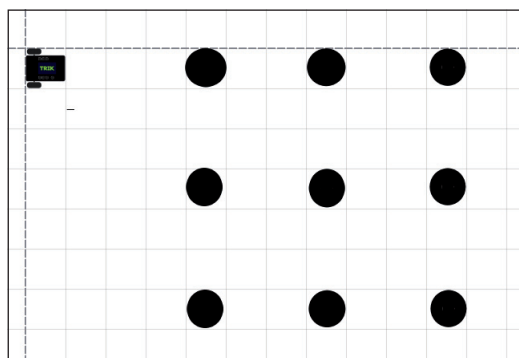
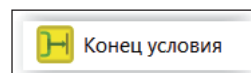


Рис. 20. Шаблон для задачи 4

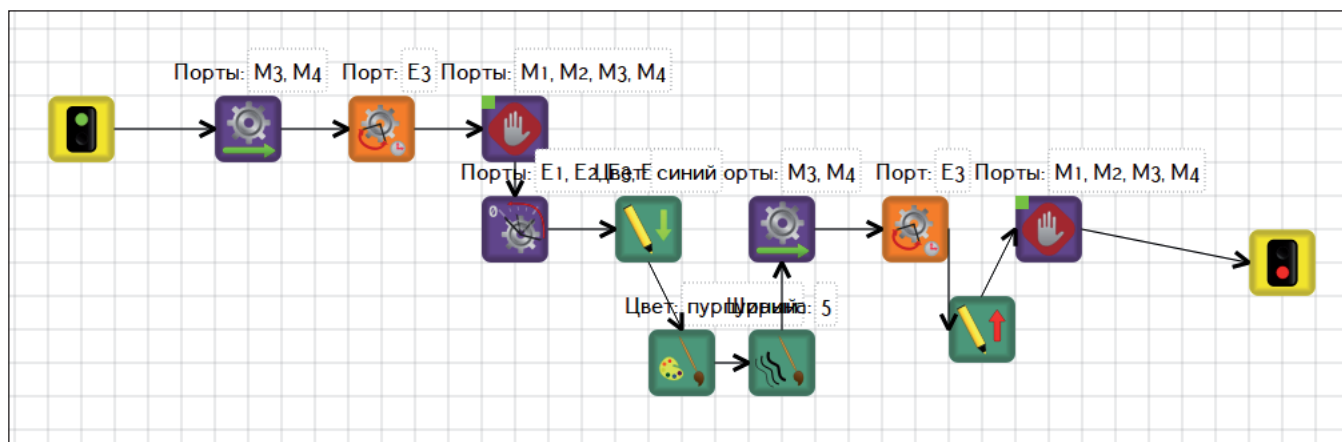


Рис. 21. Фрагмент программы «Задача 4»

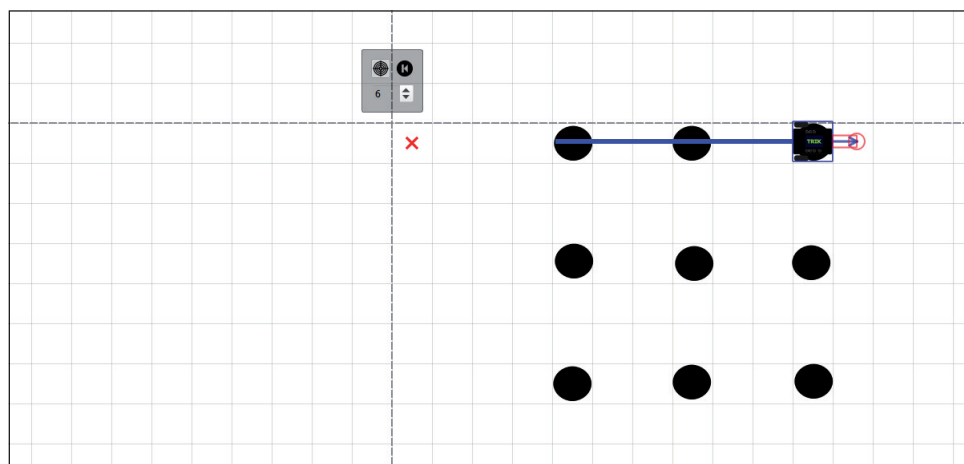


Рис. 22. Фрагмент выполнения программы «Задача 4»

Рассмотрим задачи, которые можно предложить учащимся при изучении условного оператора. Выделим основные задачи (базовые) — для совместного решения и индивидуальные задачи.

Базовые задачи.

Задача 5. Движение до стены.

Двигаться до стены, пока расстояние до нее не станет меньше 20 см.

Замечание: не использовать блок «Ждать датчик расстояния».

Программа для решения задачи и результат ее выполнения представлены на рисунках 23 и 24 соответственно. В данной задаче использован датчик расстояния, подключенный на порт D1.

Задача 6. Смайлики [12].

Вывести на экран на 3 секунды или дольше грустный смайлик, если робот далеко от стены, и веселый, если близко. За границу близко-далеко считать значение 50 инфракрасного датчика.

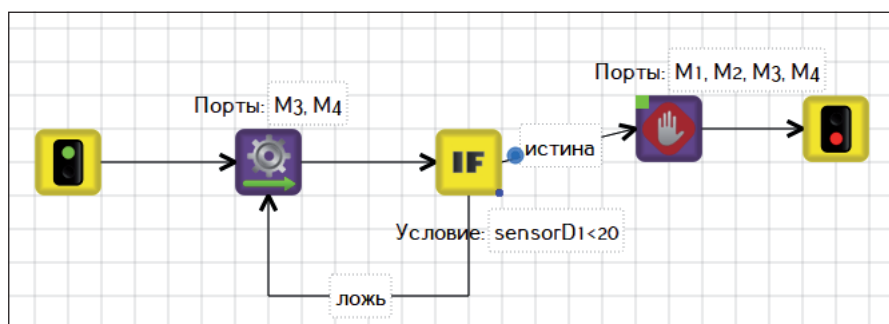


Рис. 23. Программа «Задача 5»

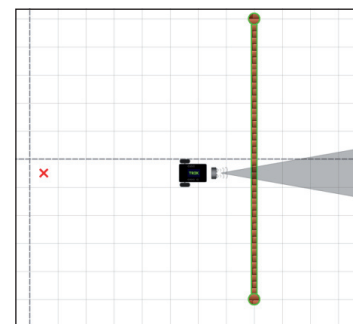


Рис. 24. Результат выполнения программы «Задача 5»

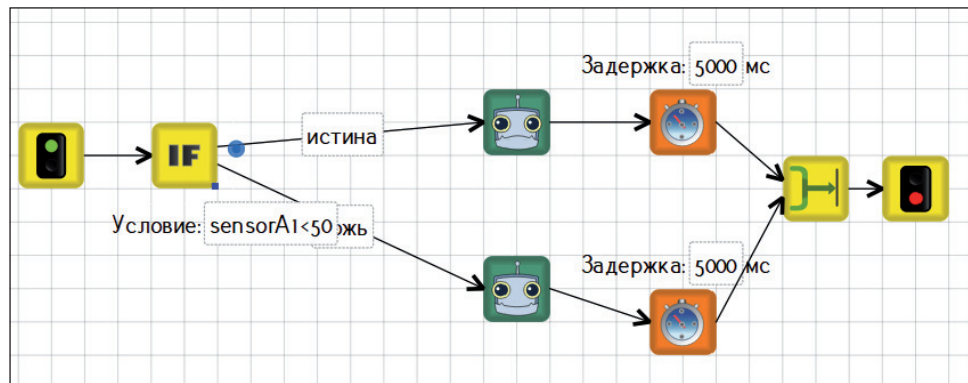


Рис. 25. Программа «Задача 6»

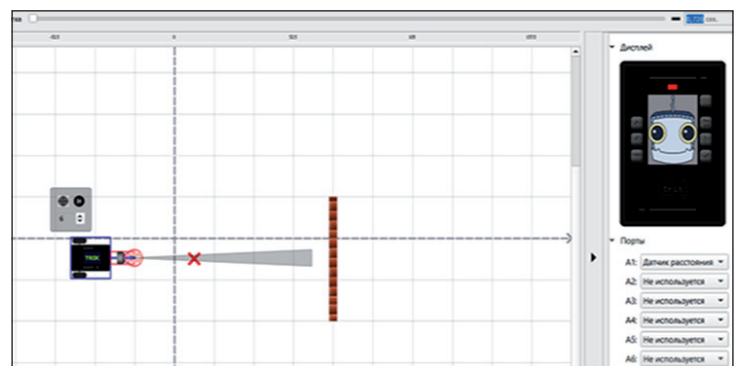
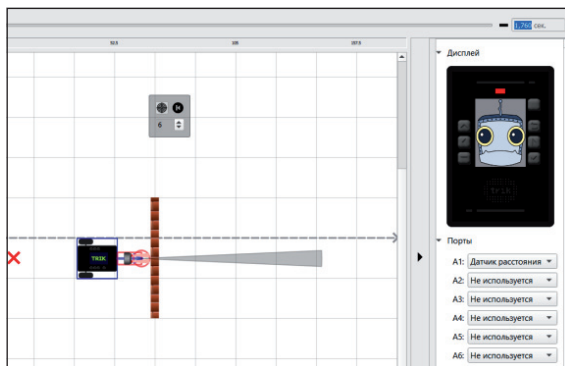


Рис. 26. Результат выполнения программы «Задача 6»

Программа решения задачи и результат выполнения программы представлены на рисунках 25 и 26 соответственно.

Задача 7. Трехцветная линия [3].

Нарисовать трехцветную линию — красный, желтый, синий цвета бесконечно чередуются в заданном порядке.

В решении этой задачи (рис. 27) используется **оператор switch** с тремя значениями переменной.

Switch представляет собой структуру, построенную по принципу выбора, и содержит все возможные варианты условий и инструкции, которые следует выполнить в каждом конкретном варианте.

В блоке проверяется выражение, от блока отходят связи, на которых указываются возможные значения этого выражения. Одна связь *обязательно* должна быть

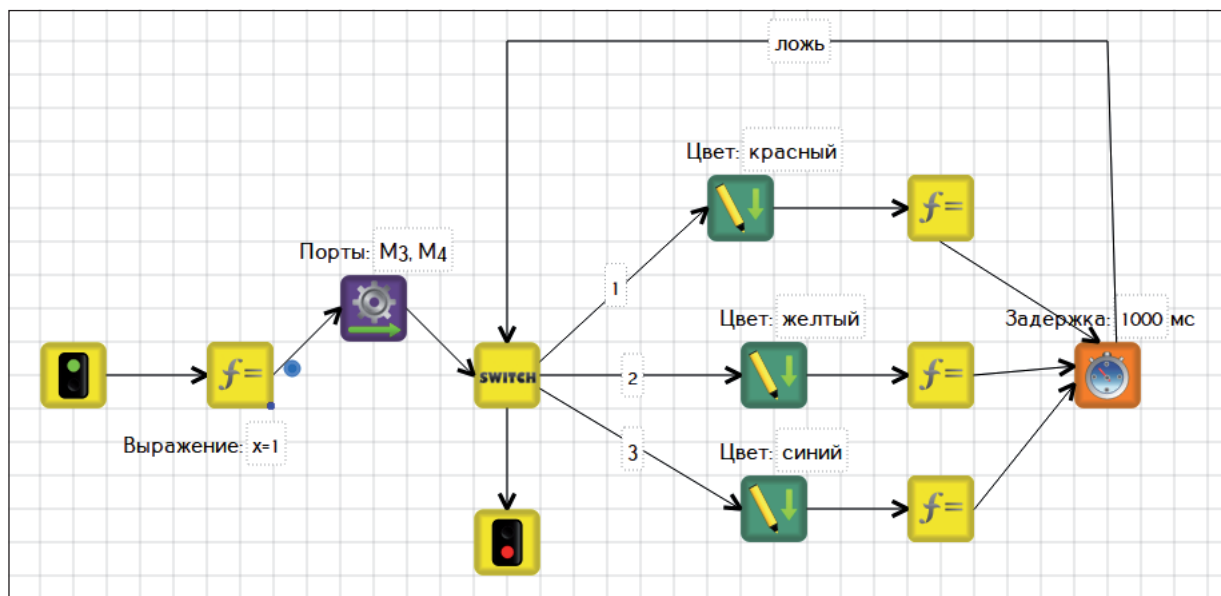


Рис. 27. Программа «Задача 7 (1)»

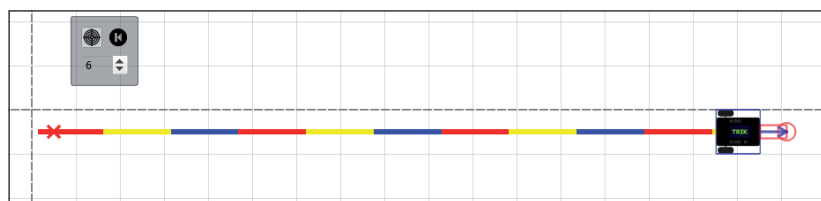


Рис. 28. Результат выполнения программы «Задача 7 (1)»

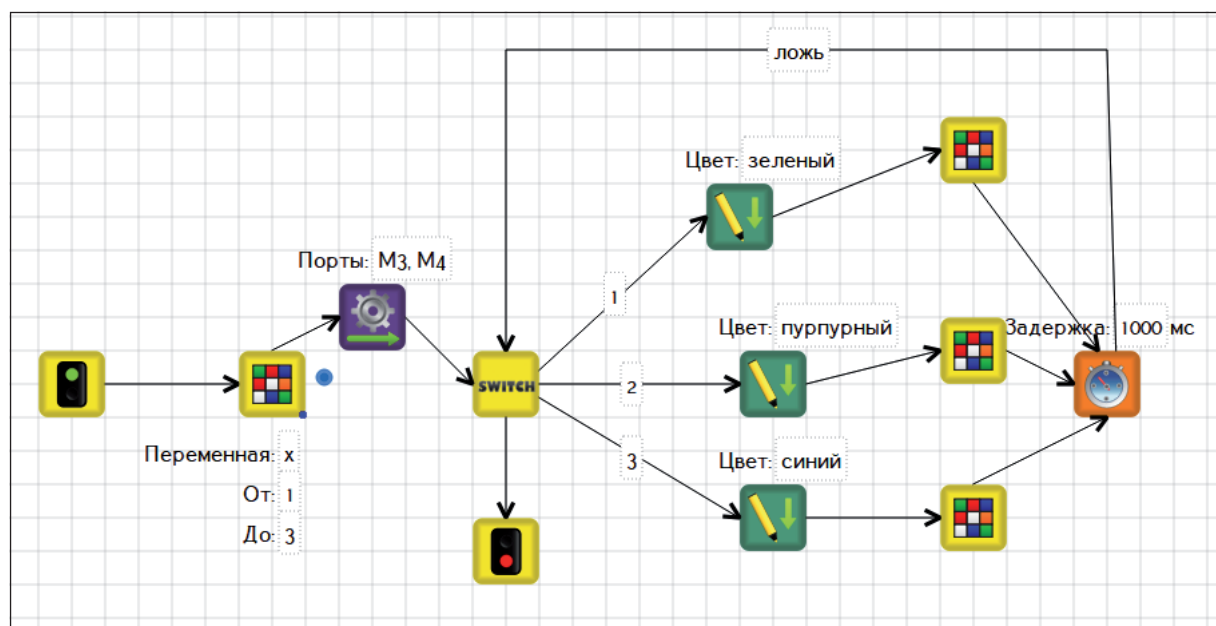


Рис. 29. Программа «Задача 7 (2)»

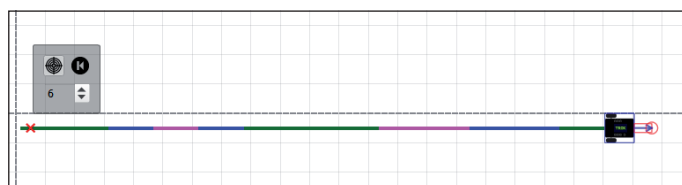


Рис. 30. Результат выполнения программы «Задача 7 (2)»

пустая (default) — по ней алгоритм будет двигаться, если не выполнено ни одно из условий.

Программа решения задачи 7 и результат выполнения программы представлены на рисунках 27 и 28 соответственно. Блок «Выражение» позволяет посчитать значение заданного выражения, также в данном блоке допускается определение переменных.

Дополнительно рассмотрим, как решить задачу 7 с помощью датчика случайных чисел. Используя

блок «Случайное число»  Случайное число

указываем переменную и значения, которые она может принимать, — от какого до какого (рис. 29).

Результат выполнения программы представлен на рисунке 30.

Задача 8. Настройка робота [12].

Робот движется прямо через черные и белые поля. Требуется непрерывно выводить на экран веселый смайлик, если робот на черном поле, и грустный, если

на белом. За границу считать значение 50 датчика света.

Программа решения задачи 8 и результат выполнения программы представлены на рисунках 31 и 32 соответственно. В данной задаче используется датчик освещенности, подключенный к порту A1.

Задача 9. Туннель.

Дан туннель, нужно выбраться из него (рис. 33). Используются два датчика расстояния. Выход расположен либо правее, либо левее движения робота. Вывести веселый смайлик, если выход найден.

Примечание. Тоннель учащиеся создают сами, или им выдается заранее созданная обстановка.

Фрагмент программы решения задачи 9 и результат выполнения этого фрагмента представлены на рисунках 34 и 35 соответственно. Здесь описаны лишь определение туннеля и вывод смайлика, необходимо добавить еще разворот и движения для выхода.

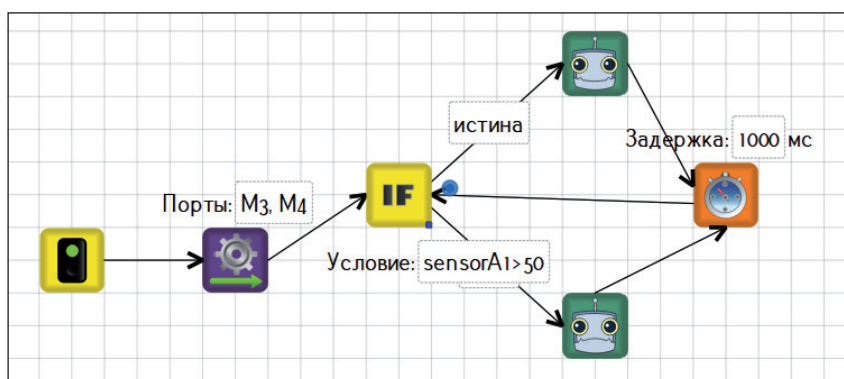


Рис. 31. Программа «Задача 8»

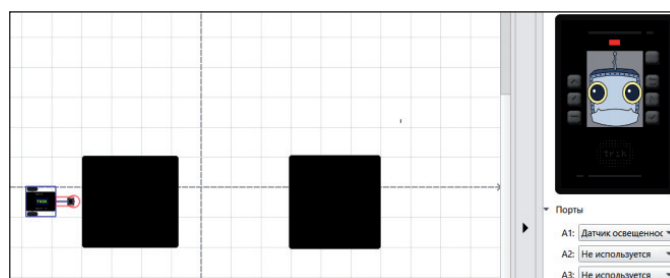
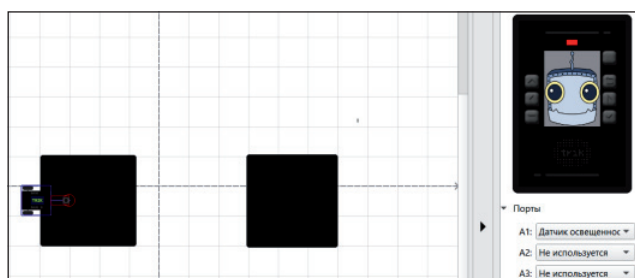


Рис. 32. Результат выполнения программы «Задача 8»

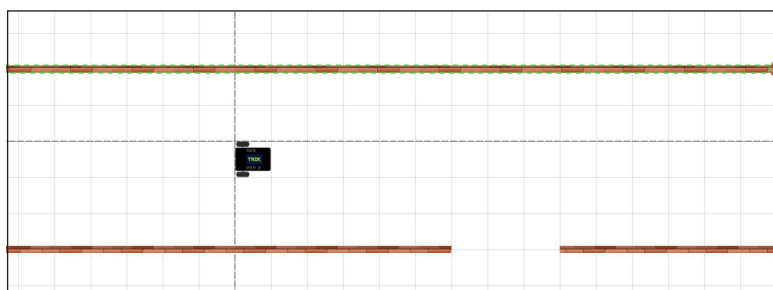


Рис. 33. Вариант возможного туннеля в задаче 9

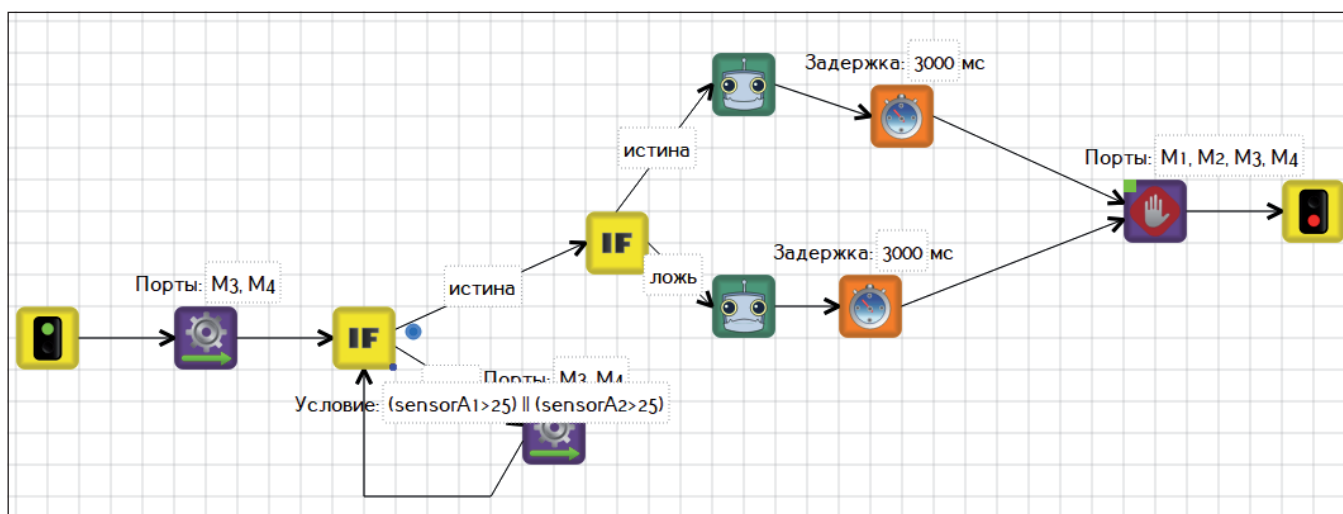


Рис. 34. Фрагмент программы «Задача 9»

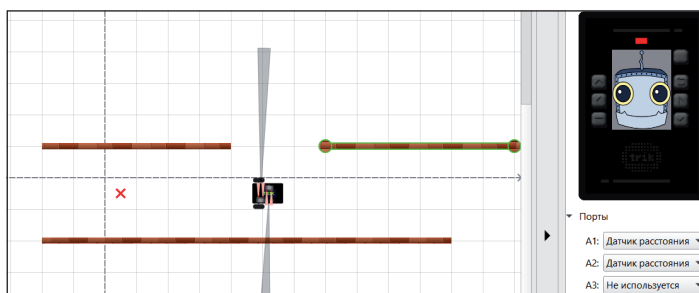


Рис. 35. Фрагмент выполнения программы «Задача 9»

Индивидуальные задачи.**Задача 10. Туннель.**

Дан туннель (рис. 36). Рисовать линию до тех пор, пока стена есть и слева, и справа. Как только стена обрывается, прекратить рисовать линию. Используются три датчика: два — для определения протяженности стены, третий — для прекращения движения.

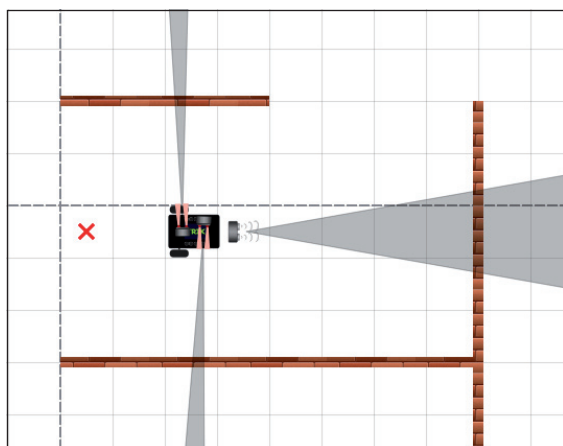


Рис. 36. Стартовая позиция задачи 10

Задача 11. Стены в туннеле.

Дан туннель (рис. 37), используются два датчика расстояния. Определить, до какой стены (левой/правой) расстояние больше.

Примечание. Стены учащиеся создают сами, или им выдается заранее созданная обстановка.

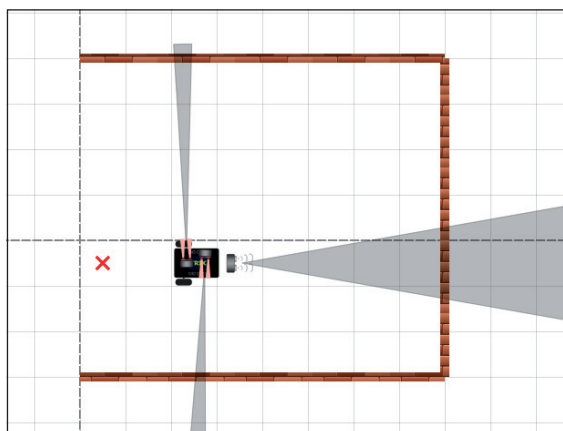


Рис. 37. Стартовая позиция задачи 11

Задача 12. Разноцветные квадраты.

Нарисовать три квадрата со сторонами четырех разных цветов:

- верхняя сторона — зеленая;
- нижняя сторона — желтая;
- левая сторона — красная;
- правая сторона — пурпурная.

Задача 13. Вывод на экран дисплея робота данных, связанных с ИК-датчиком.

Вывести на экран:

- веселый смайлик, если ИК-датчик показывает значение до 40;
- слово «неопределенность» (можно условно вывести на экран, например, сделать белый цвет фона), если ИК-датчик показывает значение в диапазоне от 40 до 60;
- грустный смайлик — в противном случае.

Задача 14. Время суток.

Вывести на экран случайным образом одно из четырех времен суток: утро, день, вечер, ночь, устанавливая цвет фона активной картинки на экране робота. Использовать оператор switch, блок «Случайное число», цвет фона на вкладке «Рисование».

Программа решения задачи 14 и результат выполнения программы представлены на рисунках 38 и 39 соответственно.

Задача 15. Кентервильский робот [9].

Привидение в Кентервильском замке каждую ночь рисует лужи красной краской. Убедившись, что лужа красная, он довольный скрывается из виду. Когда красная краска заканчивается, он рисует лужи зеленой краской и расстроенный отключается. Научите робота определять цвет лужи (рис. 40) и выключаться, если лужа зеленая. В первый раз робот всегда в приподнятом настроении. Для решения используйте датчик освещенности.

Пояснение. Выводить на экран:

- веселый смайлик (на ваш выбор, но не более 4 секунд), если робот видит красную лужу (значение датчика освещенности — больше 72) или пустой пол (значение датчика освещенности — меньше 5);
- грустный смайлик (в течение 3 секунд) — в противном случае (зеленая лужа: значение датчика освещенности — от 59 до 69).

И закончить выполнение программы.

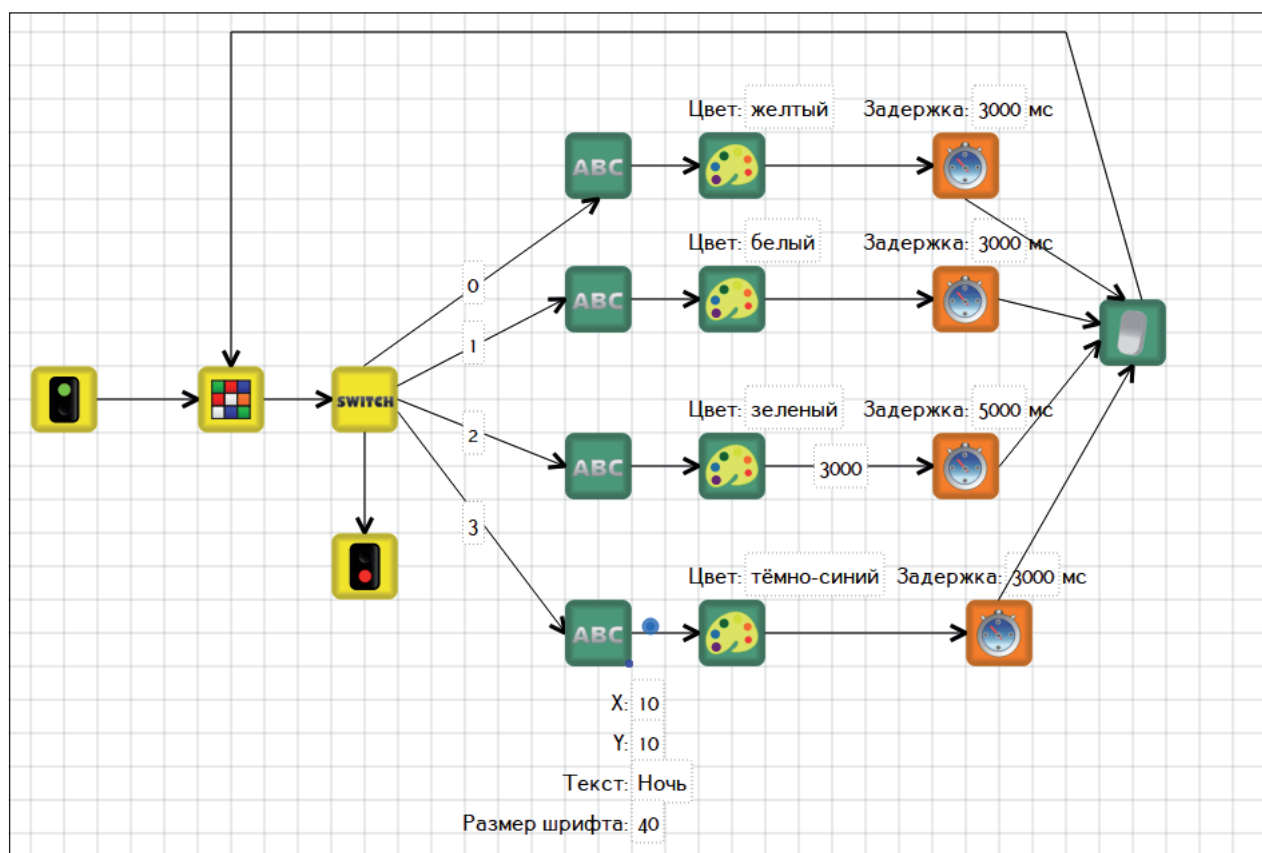


Рис. 38. Программа «Задача 14»

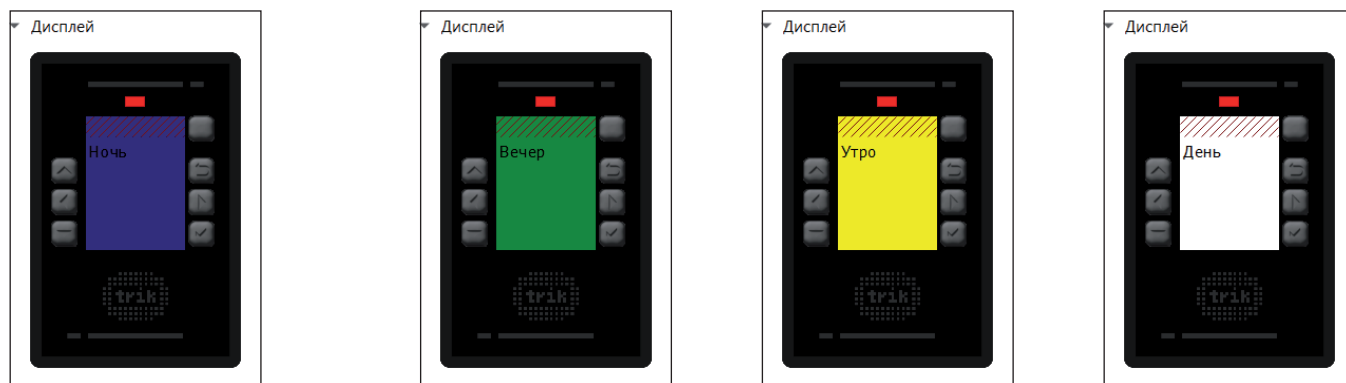


Рис. 39. Результат выполнения программы «Задача 14»

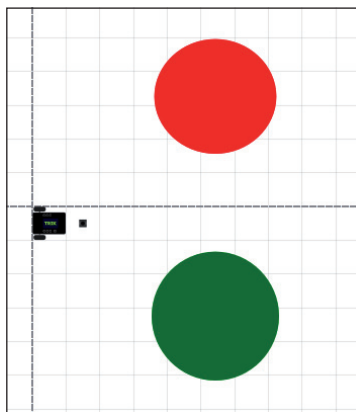


Рис. 40. Стартовая позиция задачи 15

2.1.3. Циклы

Для организации циклов используются следующие блоки:



Цикл с предусловием

— для организации

циклов с пост- и предусловием;



Цикл

— для организации циклов со счет-

чиком.

Фрагменты программ с соответствующими циклами приведены на рисунках 41–43.

Базовые задачи.

Цикл со счетчиком.

Задача 16. Движение вперед на определенное расстояние.

Проехать вперед ровно на 87,5 см.

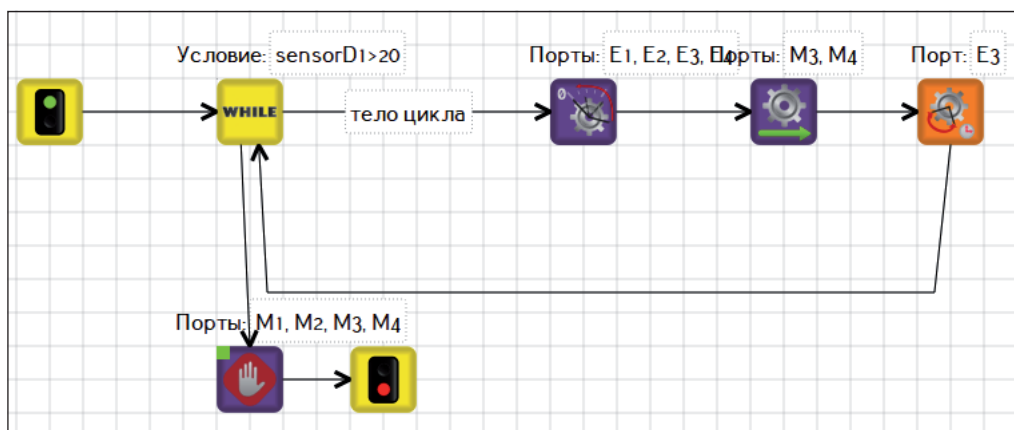


Рис. 41. Программа с использованием цикла с предусловием

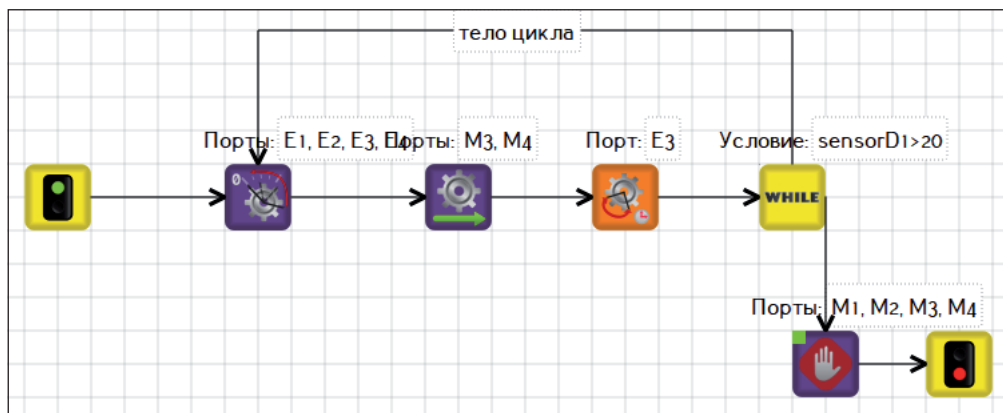


Рис. 42. Программа с использованием цикла с постусловием

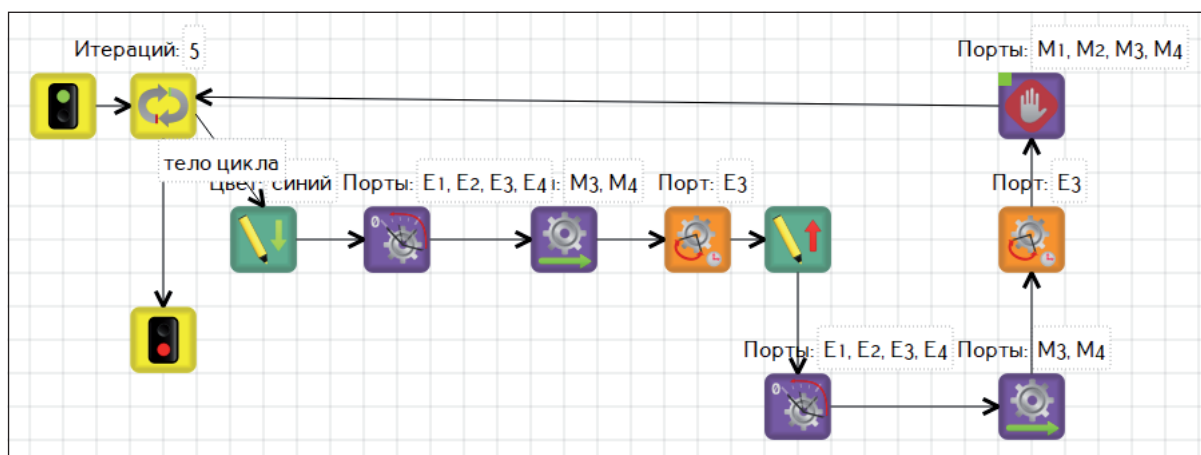


Рис. 43. Программа с использованием цикла со счетчиком

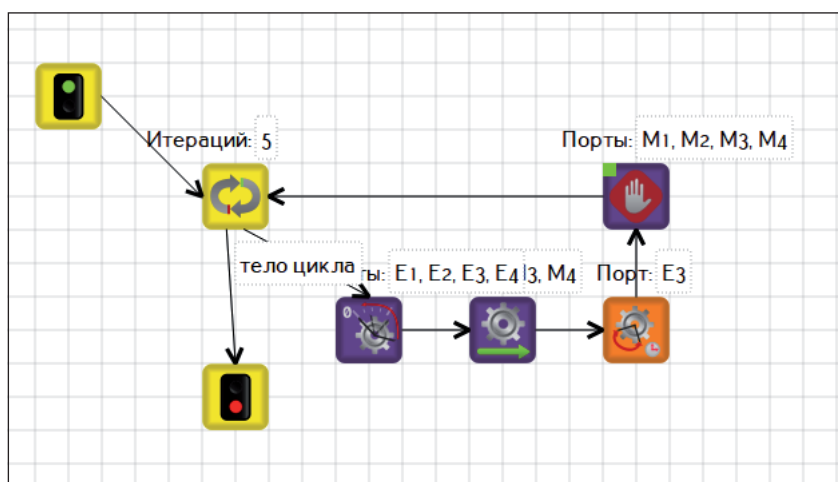


Рис. 44. Программа «Задача 16»

Так как одна клетка поля равна 17,5 см, то задача сводится к тому, чтобы проехать на пять клеток вперед (рис. 44).

Задача 17. Пунктирная линия.

Нарисовать пунктирную линию пять раз (дополнение: и вернуться назад в исходную точку).

Программа решения задачи 17 и результат выполнения программы представлены на рисунках 45 и 46 соответственно.

Задача 18. Пешеходный переход (вложенный цикл).

Дописать предыдущую задачу так, чтобы получился пешеходный переход: рисовать такие же пунктирные линии друг под другом (минимум четыре раза).

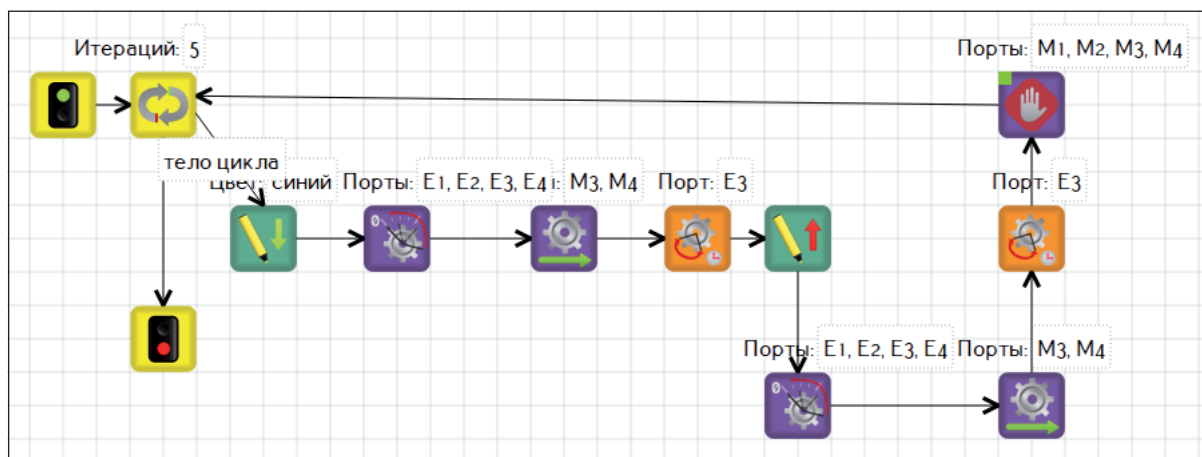


Рис. 45. Программа «Задача 17»

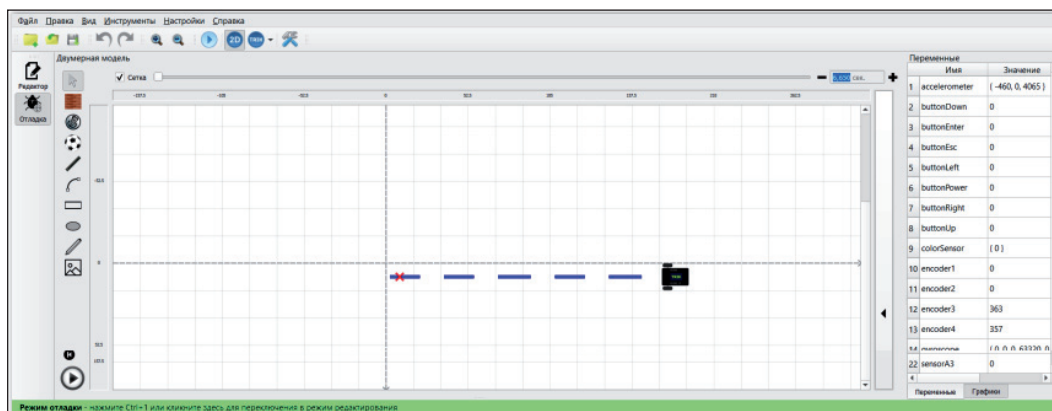


Рис. 46. Выполнение программы «Задача 17»

Задача 19. Остановиться до стены.

Робот должен ехать вперед до стены. Остановиться за 20 см до нее.

Один из возможных вариантов решения представлен на рисунке 47.

Если использовать энкодер, то робот будет двигаться быстрее (рис. 48, 49).

Результат выполнения программы представлен на рисунке 50.

Задача 20. Лабиринт.

Написать программу, которая позволит роботу проехать в центр лабиринта, используя датчик ультразвука, и остановиться на зеленом поле.

Замечание: принять, что схема лабиринта известна.

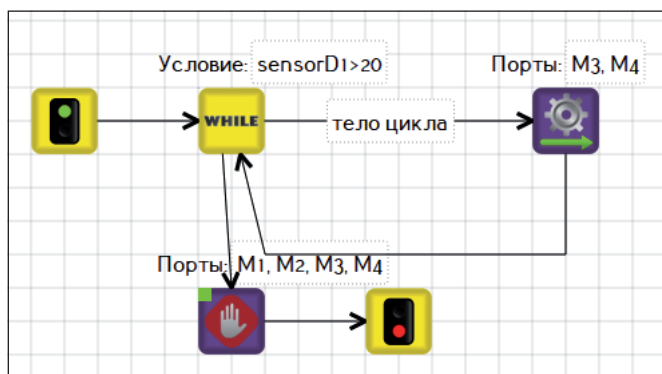


Рис. 47. Программа «Задача 19 (1)», реализованная с помощью цикла с предусловием и с использованием ультразвукового датчика расстояния

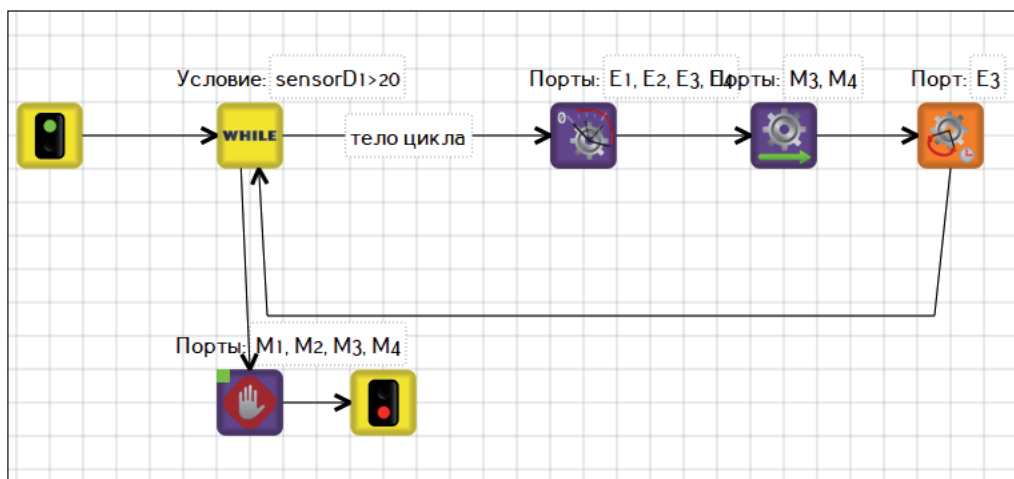


Рис. 48. Программа «Задача 19 (2)», реализованная с помощью цикла с предусловием

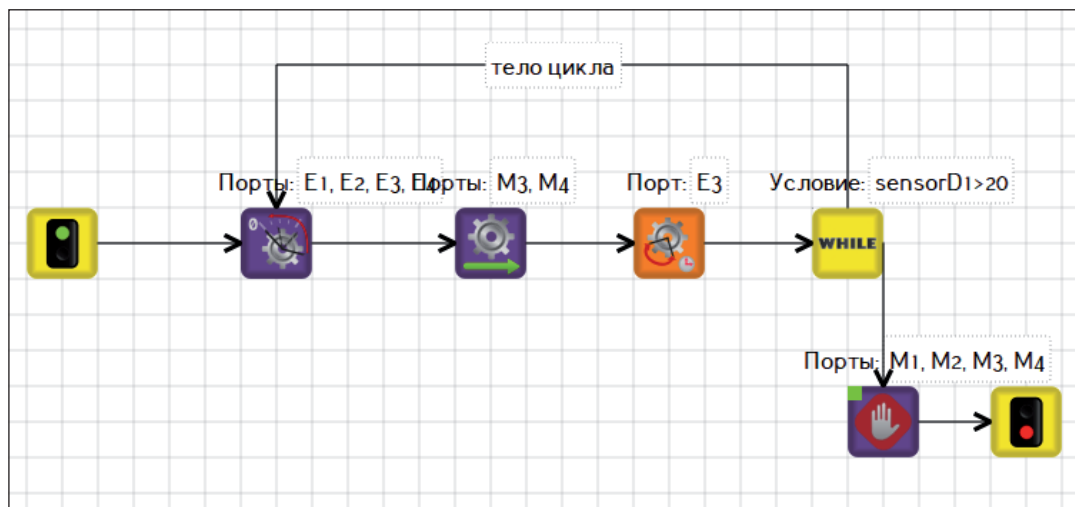


Рис. 49. Программа «Задача 19 (3)», реализованная с помощью цикла с постусловием

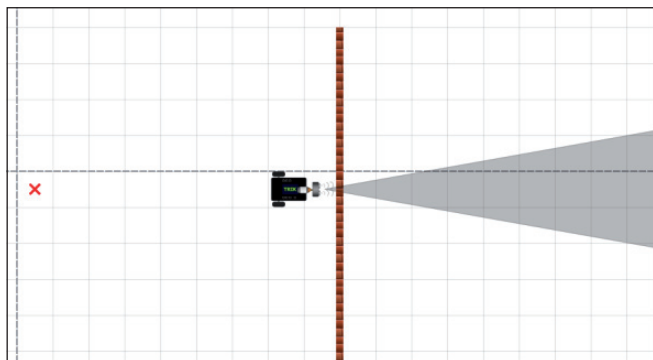


Рис. 50. Выполнение программы «Задача 19»

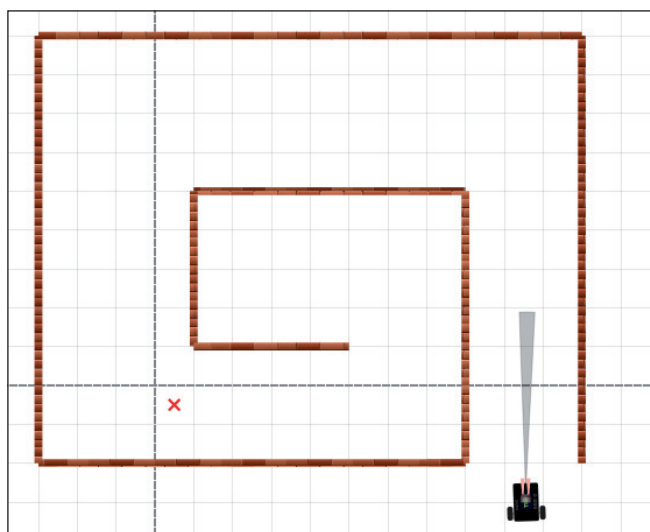


Рис. 51. Образец лабиринта для задачи 20

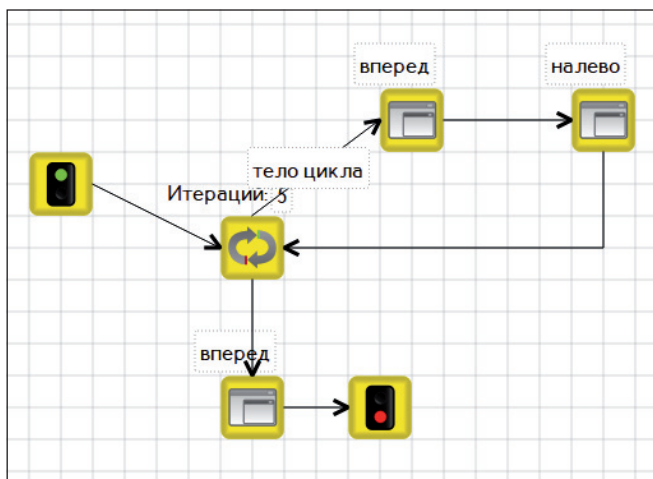


Рис. 52. Программа «Задача 20»

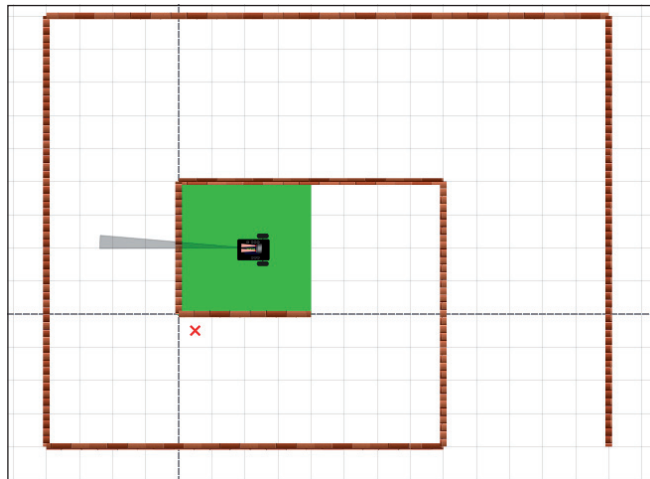


Рис. 53. Выполнение программы «Задача 20»

Примечание. Схему учащиеся создают сами, или им выдается заранее созданная обстановка (рис. 51).

Программа решения задачи 20 и результат выполнения программы представлены на рисунках 52 и 53 соответственно. В данной программе использованы команды, рассмотренные выше: движение вперед и поворот налево. Для читаемости программы (восприятия) использованы подпрограммы.

Робота можно оснастить датчиками касания (порты подключения А1—А6).

Задача 21. Ехать до стены.

Ехать вперед до стены, после касания остановиться и сообщить об этом.

Один из вариантов решения представлен на рисунке 54, результат выполнения программы — на рисунке 55.

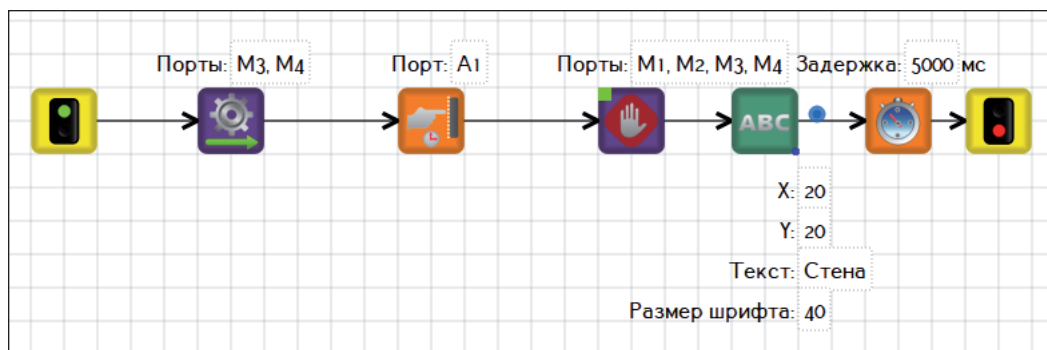


Рис. 54. Программа «Задача 21»

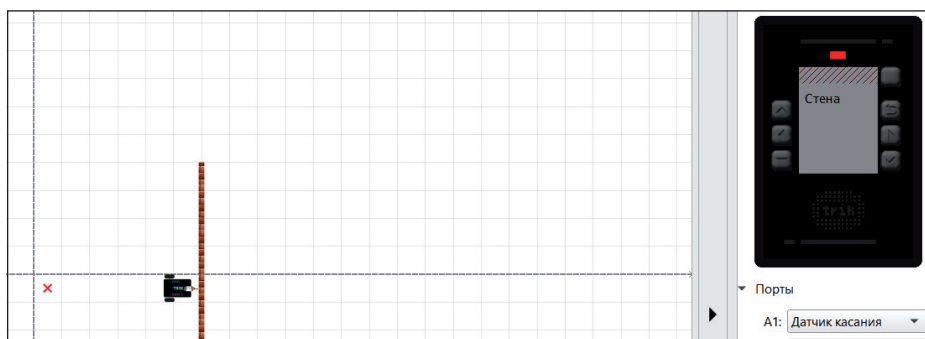


Рис. 55. Результат выполнения программы «Задача 21»

Индивидуальные задачи.**Задача 22. Не доехать до стены.**

Не доехать до стены 15 см. Использовать УЗ-датчик расстояния.

Задача 23. Выбраться из лабиринта.

Написать программу, выполнив которую робот должен выбраться из лабиринта и остановиться (рис. 56).

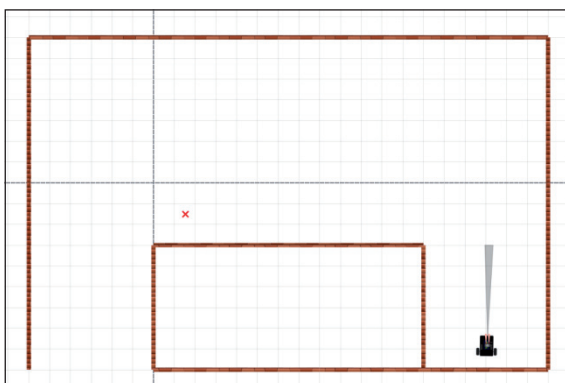


Рис. 56. Стартовая позиция задачи 23

Задача 24. Лабиринт с единственным выходом.

(Для наиболее подготовленных учащихся.) Есть лабиринт с единственным выходом. Необходимо выйти из него, используя правило правой руки.

Задача 25. Автобусный парк [3].

В маршруте движения робота пять остановок. Перед каждой остановкой он говорит: «Остановка», а перед началом движения: «Движение возобновлено». Напишите для него эту программу.

На рисунке 57 остановки обозначены зелеными прямоугольниками. Расстояние до остановок не постоянное.

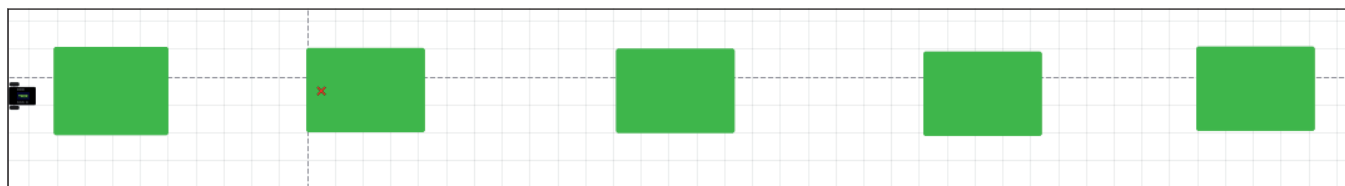


Рис. 57. Стартовая позиция задачи 25

Задача 26. За водой [3].

Робот спускается к реке за водой. Достигнув реки (синяя линия), он возвращается обратно. Требуется заполнить емкость объемом 50 литров. За один раз робот может принести одно ведро (12 литров) воды.

Замечание. В контексте данной задачи фразу «принести одно ведро» следует понимать и использовать для расчетов итераций движения.

Задача 27. Край стола.

Необходимо четыре раза проехать от края до края стола, при этом не упасть с него.

Задача 28. Стадион.

Беговая дорожка стадиона ограничена двумя прямоугольниками. Один из прямоугольников находится внутри другого (рис. 58). Напишите программу, которая позволит проехать три круга по данному стадиону. Размеры стадиона неизвестны.

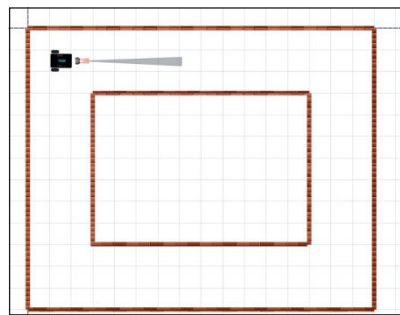


Рис. 58. Стартовая позиция задачи 28

Задача 29. Челночный бег [3].

На полу нарисованы четыре цветные линии, старт производится из-за первой линии. Доехав до каждой линии, пересекаем ее и возвращаемся к месту старта. Напишите программу, если известно, что расстояние от красной линии до синей — 6 клеток, от зеленой до

желтой — 8, от синей до зеленой — 4, а расстояние от первой до последней линии минимально (максимально).

Задача 30. Рисуем квадраты.

Нарисовать четыре одинаковых квадрата на некотором расстоянии друг от друга.

Задача 31. Узор.

Разрисовать поле размером 4×4 узором из квадрата и следующего за ним на некотором расстоянии треугольника. Данный узор повторяется, пока не заполнит все поле.

2.2. Создание электронных управляемых устройств на базе платформы Arduino

При изучении курса программирования в средней и старшей школе можно включить в изучаемый материал работу по созданию электронных управляемых устройств на базе платформы Arduino. Интересные и несложные в сборке устройства дают представление о принципах их работы. Arduino — одна из распространенных платформ для образовательной робототехники. Это удобная платформа быстрой разработки электронных устройств как для новичков, так и для профессионалов.

На базе платформы Arduino школьники могут конструировать и программировать модели электронных управляемых систем, не вдаваясь в сложные вопросы схемотехники. Эта уникальная инженерно-конструкторская среда позволит как конструировать и программировать простые проекты, так и дойти в своем творчестве до сложных проектов и даже до управления робототехническими устройствами.

В таблице приведены примеры проектов на базе платформы Arduino, которые можно использовать для изучения основных алгоритмических структур. Подробнее познакомиться с Arduino и с реализацией проектов можно в статье автора [6].

3. Выводы

Образовательная робототехника — это инновационный курс ознакомления школьников с техническим творчеством [2]. Это инструмент, закладывающий прочные основы системного мышления, он интегрирует информатику, математику, физику, черчение, технологию, естественные науки с развитием инженерного творчества. Занятия робототехникой дают хороший задел

на будущее, вызывают у ребят интерес к научно-техническому творчеству, способствуют целенаправленному выбору профессии инженерной направленности [10].

Робототехника неразрывно связана с программированием. Программирование может быть слишком сложным и утомительным для ребенка, если изучать его с помощью «традиционного» абстрактного метода. Но в случае робототехники ученик имеет возможность управлять физическим роботом и сразу видит, что идет не так, на практике узнает предел возможностей технологий.

Внедрение в практику работ по созданию электронных управляемых устройств на базе Arduino или же изучение роботов в среде TRIK Studio позволит учащимся освоить основные приемы конструирования и программирования и получить необходимые знания и навыки для дальнейшей самореализации в области инженерии, изобретательства, информационных технологий и программирования. Педагоги смогут формировать и развивать у школьников познавательный интерес, логическое и пространственное мышление, закладывать первоначальные знания и развивать кругозор в области инженерии, что позволит формировать у учащихся осознанный выбор своей будущей профессии, связанной с наукоемкими технологиями и инженерными специальностями.

Список источников

1. Ворон А. М. Использование конструкторов ТЕТРИКС в STEM образовании // VII Всероссийская конференция «Современное технологическое обучение: от компьютера к роботу». Сборник тезисов (Санкт-Петербург, 30 марта 2017 г.). СПб.: Полиграфическое предприятие № 3, 2017. С. 6–8. <https://sportrobotics.ru/files/47996/filename/sbornik-2017.pdf>
2. Данчук И. И. Робототехника в развитии технического творчества школьников во внеурочной деятельности в дополнительном образовании // Педагогический журнал. 2018. Т. 8. № 6А. С. 113–122. EDN: ZAHIGT.
3. Киселев М. М., Киселев М. М. Сборник задач по программированию механизмов и роботов. СПб., 2016. 96 с.
4. Кондрук А. В. Использование основ робототехники в учебном процессе. Костанай: ФАО «НЦПК “Ордеу” ИПК ПР по Костанайской области», 2017. 48 с.
5. Кочагина М. Н., Белушкина А. И., Звезгинцева А. Д., Кашкина Н. А., Смирнова И. В. Использование STEM-оборудования при обучении математике // CONTINUUM. Математика. Информатика. Образование. 2023. № 1 (29). С. 8–22. EDN: CJRANY. DOI: 10.24888/2500-1957-2023-1-8-22.
6. Левушкина А. С. Использование Arduino в изучении раздела «Алгоритмизация и программирование» школьного курса информатики как подготовка к освоению основ инженер-

Таблица

Примеры некоторых проектов на базе платформы Arduino для изучения алгоритмических структур

Линейный алгоритм	Алгоритм ветвления (условный алгоритм)	Циклический алгоритм
Проект 1. Зажигаем светодиод Проект 2. Мигающий светодиод Проект 3. Передача сигнала SOS Проект 4. Железнодорожный светофор Проект 5. Светофор	Проект 6. Устройство с режимом управления свечением светодиода с помощью кнопки или переключателя Проект 7. Устройство, реагирующее на изменение расстояния Проект 8. Метеостанция (наблюдение за температурой) Проект 9. «Умный» ночник (яркость свечения такого ночника будет меняться в зависимости от уровня света)	Проект 10. Сервопривод — вращение фигуры Проект 11. Ночник, светящийся всеми цветами радуги

ных наук // Информатика в школе. 2021. № 3. С. 15–31. EDN: KVTBOY. DOI: 10.32517/2221-1993-2021-20-3-15-31.

7. Поляков К. Ю. Использование языка Blockly для управления виртуальными исполнителями // Информатика в школе. 2021. № 1. С. 20–25. EDN: UGHHIL. DOI: 10.32517/2221-1993-2021-20-1-20-25.

8. Толстова Н. А., Бондаренко Д. А., Ганьшин К. Ю. Образовательная робототехника как составляющая инженерно-технического образования // Наука. Инновации. Технологии. 2013. № 3. С. 171–177. EDN: SYNELT.

9. Филимонова Е. В., Савчук А. Ю. Робот LEGO MIND-STORMS EV3 в среде TRIK Studio как новый исполнитель для

обучения алгоритмизации и основам робототехники в школьном курсе информатики // Информатика в школе. 2021. № 7. С. 48–56. EDN: WTVDDZY. DOI: 10.32517/2221-1993-2021-20-7-48-56.

10. Чупин Д. Ю., Ступин А. А., Ступина Е. Е., Класов А. Б. Образовательная робототехника: учебное пособие для студентов педагогических вузов. Новосибирск: Агентство «Сибпринт», 2019. 114 с.

11. Яндекс Учебник. Модуль 3.1. Алгоритмы и исполнители. https://education.yandex.ru/lab/library/common-course-container/informatics/collection/5info_5grade_2024-25/?grade=5&module_id=837

12. TRIK Studio. <https://trikset.com/products/trik-studio>