

ОГЛАВЛЕНИЕ	
ПРЕДИСЛОВИЕ	4
ВВЕДЕНИЕ	6
Встреча 1. Моторы вперед!	7
<i>Отступление 1. Жесты мышью</i>	11
Задания для закрепления материала	12
Встреча 2. Робот на связи	15
Задания для закрепления материала	18
Встреча 3. Моторы вперед-2	19
<i>Отступление 2. Ещё больше возможностей</i>	20
Задания для закрепления материала	21
Встреча 4. Впереди стена	22
<i>Отступление 3. Объемный мир</i>	24
Задания для закрепления материала	25
Встреча 5. Подпрограмма – это маленькая программа	26
Задания для закрепления материала	27
Встреча 6. Лабиринт	28
<i>Отступление 4. Непостоянное время</i>	29
Задания для закрепления материала	30
Встреча 7. Бег по кругу	31
Задания для закрепления материала	32
Встреча 8. Разноцветная дорога	33
<i>Отступление 5. Раздвигаем горизонты</i>	33
<i>Отступление 6. Графики значений датчиков</i>	34
Задания для закрепления материала	36
Встреча 9. Познакомимся с неизвестными	37
<i>Отступление 7. Дело случая</i>	40
Задания для закрепления материала	41
Встреча 10. Кегельринг	42
Задания для закрепления материала	44
Встреча 11. Что, если?	45
<i>Отступление 8. Посчитаем кнопки</i>	46
<i>Отступление 9. Где логика?</i>	47
Задания для закрепления материала	48
Встреча 12. Исправление ошибок	49
<i>Отступление 10. Алгоритм «математического» сглаживания</i>	50
Задания для закрепления материала	51

Встреча 13. Движение вдоль стены. Объезд предметов	52
Задания для закрепления материала	54
Встреча 14. Движение по линии	55
Задания для закрепления материала	57
Встреча 15. Манипуляторы	58
Задания для закрепления материала.	60
Встреча 16. Таблицы и массивы	61
<i>Отступление 11. Золотая середина</i>	62
Задания для закрепления материала	64
Встреча 17. Камера контроллера ТРИК	65
<i>Отступление 12. Небольшой, но компьютер.</i>	68
<i>Отступление 13. Мультимедиа</i>	68
<i>Отступление 14. Консоль робота</i>	69
Задания для закрепления материала	70
Встреча 18. Параллельным курсом	71
Задания для закрепления материала	73
Встреча 19. Исследователь	74
Задания для закрепления материала	76
Встреча 20. Мастерская художника	77
Задания для закрепления материала	81
Встреча 21. Мы вместе	82
<i>Отступление 15. Сохраним информацию</i>	82
Задания для закрепления материала	84
Встреча 22. Гироскоп, акселерометр	85
Задания для закрепления материала	88
Встреча 23. Дистанционное управление	89
ЗАКЛЮЧЕНИЕ	93
Литература	94

М.М.Киселев
М.М.Киселев

Сборник задач по программированию механизмов и роботов

Препринт

Санкт-Петербург
2016

ПРЕДИСЛОВИЕ

Данная методическая разработка представляет собой актуальный, научно обоснованный и практически апробированный вариант серьезно спланированной работы с учащимися, как в рамках урочной деятельности, так и в системе дополнительного образования.

Ещё много лет назад знаменитый американский психолог С. Пейперт из MIT заметил, что при обучении юных программистов нужно иметь быструю и, главное, видимую обратную связь, когда обучаемый немедленно видит, именно видит в наглядной форме, а не получает ответ в виде каких-то чисел, результаты работы своей программы. Воплощая эту идею, С. Пейперт придумал «черепашку» – объект, который ползает по экрану, оставляя след, в соответствии с программой пользователя. Созданный им программный продукт Лого популярен до сих пор. Я много раз видел, когда дети, получив задание нарисовать, скажем, дом, видели, что окно или дверь оказывались вне дома, удивлялись, но быстро находили ошибку.

Сейчас уже немногие помнят, что С. Пейперт пошёл дальше и в дополнение к «черепашке» сконструировал и настоящего робота, управляемого теми же программами, что и Лого, справедливо полагая, что реальный объект воспринимается лучше любой модели. Но состояние элементной базы на тот момент не позволило сделать этот продукт дешёвым и массовым. Сегодня уже возможно выпускать контроллеры (в каком-то смысле «мозг» робота) массово и по доступной цене.

Собственно, контроллер – это обычный компьютер, часто весьма мощный. Единственное отличие от обычных ЭВМ состоит в наличии большого количества разъёмов для различных датчиков: расстояния, касания, цвета, температуры и т.п. и управляющих устройств: моторов, звуковых динамиков, электронных выключателей и т.п. Но именно это отличие диктует необходимость весьма специфического программного обеспечения, так называемого «событийно-ориентированного ПО» и соответствующего набора инструментальных средств его разработки.

Сейчас существует множество конструкторов роботов и средств программирования для них. Но одни средства имеют неподъёмную для школ цену, другие работают только на «своём» оборудовании, третьи не локализованы в России, а какие-то слишком сложны для понимания детьми. Методических материалов на русском языке очень мало, можно упомянуть разве что книги С.А. Филиппова – великого энтузиаста робототехники из Президентского лицея № 239 в Санкт-Петербурге. Но и эти книги слишком направлены на спортивную робототехнику и на кружковую работу. Мы не отрицаем важность соревнований роботов, созданных руками детей, но робот, быстро едущий по линии, имеет мало общего с задачами из реальной жизни.

Робот – это не обязательно тележка с колёсами и моторами. Устройства, включающие свет, если в комнате стало темно, открывающие форточку, если уровень углекислого газа превысил допустимый уровень, включающие обогреватель, если стало холодно – это тоже полезные роботы. Бурно развивающееся сегодня направление IoT (Internet of Things – интернет вещей), в рамках которого браслет на руке пациента сообщает врачу по интернету температуру и давление пациента, автомобиль сам общается с центром технического обслуживания, «умные дома» и многое другое – также являются хорошими примерами полезных роботов, которые надо уметь программировать.

В обязательном школьное образование включён предмет «Технология», но нет общепринятого понимания, чему учить в рамках этого предмета. Если какая-то девочка научится работать на швейной машинке, а какой-то мальчик на токарном станке – это тоже хорошо, но вряд ли это соответствует чаяниям молодёжи, да и соответствующего оборудования в школах не осталось. Нужно учить детей современным технологиям.

Более 30 лет назад академик А.П. Ершов сказал: «Программирование – вторая грамотность», в истинности этих слов мы не сомневаемся и сегодня, но учить только

программированию – мало, так как детям надо развивать микромоторику, умение пользоваться отвёрткой, дрелью, паяльником и другими быденными, но важными в жизни инструментами.

Робототехника – это практически идеальное сочетание для целей образования: дети что-то делают руками, иногда выполняя нетривиальное проектирование и проявляя чудеса изобретательности, и программируют получившиеся устройства, изучая при этом основы кибернетики. Кроме того, изучение программирования и робототехники – шаг в сторону инженерного образования, о важности которого сейчас много говорят.

Огромное значение, особенно на современном этапе, имеет нацеленность на результат. Построение и программирование робота – это не просто решение отдельной задачи, но настоящий проект, в рамках которого нужно спланировать работы, обеспечить необходимые ресурсы, выполнить проектирование, реализацию, проверку качества результата, продумать хорошую презентацию, наконец. Всему этому также нужно учиться, причем на реальных проектах, а не у доски.

Еще одна важная особенность данного учебного пособия – это хорошо проработанная связь робототехники с другими предметами, изучаемыми в школе. Важность такой связи многими декларируется, но реальных примеров очень мало.

Хотелось бы, чтобы материал данного пособия послужил основой для унификации и стандартизации методики преподавания предмета «Технология», на наш взгляд, сейчас в этом вопросе царят разброд и шатания. Основания для таких надежд есть – TRIK Studio изучают тысячи школьников и студентов на всех континентах, это бесплатный открытый продукт, мы тщательно следим за активностью пользователей, его скачивающих. Самый частый вопрос: «Когда будет опубликовано пособие с подробным разбором типовых задач и темами для самостоятельной работы?» Вот оно, читайте, изучайте, решайте задачи. Мы всегда будем рады ответить на ваши вопросы, замечания и предложения по его улучшению.

Доктор физико-математических наук, профессор А.Н.Терехов.

ВВЕДЕНИЕ

Первая часть задачника, которую Вы держите в руках, адресована широкому кругу читателей от учащихся начальной и средней школы до учителей и преподавателей как основного, так и дополнительного образования. Решение поставленной задачи, умение применять на практике накопленные знания, являются определяющими критериями глубины усвоения теории по любому предмету.

Пособие подготовлено на основе опыта преподавания авторами задачника предмета «Технология», ведения проектной деятельности и кружка «Конструирование и программирование механизмов и роботов».

Цель пособия – научить применять изученные алгоритмы и навыки программирования к задачам, которые привязаны к окружающему нас реальному миру.

Уровень сложности задач различен, и учитель может сам выбрать из них те, которые совпадают с возможностями учащихся и задачами, стоящими перед ними.

В качестве среды программирования, предложенной для изучения, выбрана разработка резидента инновационного центра «Сколково» компании «КиберТех Лабс» TRIK Studio.

Значительный вклад в развитие этого некоммерческого отечественного проекта внесли сотрудники, студенты и аспиранты кафедры системного программирования СПбГУ.

Язык программирования, который поддерживается в среде TRIK Studio, относится к визуальным языкам программирования, программы создаются при помощи пиктограмм. Интерфейс среды доступен для понимания даже учащимся начальных классов. Переход от визуального программирования к текстовому программированию на языках Pascal, RuC, C, C++, Java выполнен очень наглядно, поэтому ограничений по возрасту на использование TRIK Studio не существует.

TRIK Studio поддерживает программирование контроллера ТРИК российской разработки и контроллеров Lego Mindstorms NXT 2.0, Lego Mindstorms EV3.

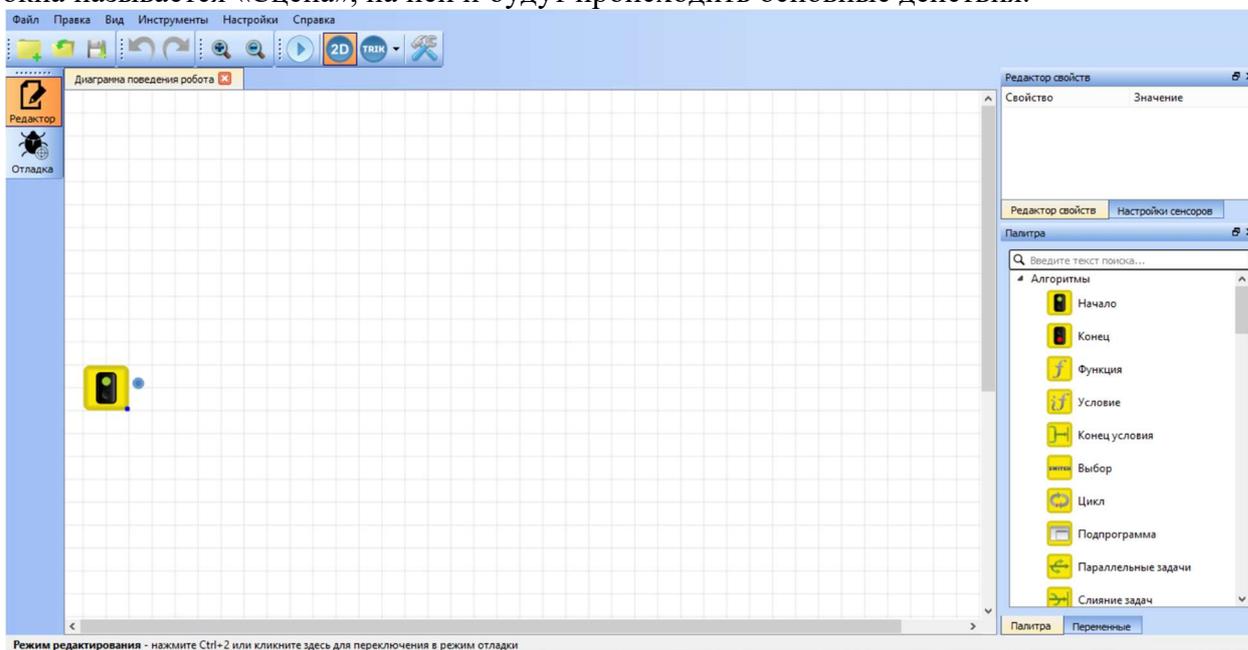
Перед каждой группой заданий для закрепления материала разбираются примеры решения задач, с использованием опыта предыдущих заданий. Подробный разбор задач сделан для контроллера ТРИК, а для Lego Mindstorms EV3 дан в сокращении.

Задачи сформулированы таким образом, что их можно решать, используя другие среды программирования и контроллеры. Некоторые задачи сформулированы сюжетно, без конкретики в постановке задания, что дает возможность преподавателю проявить креативность.

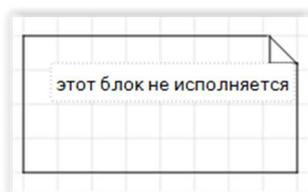
Итак, оболочка TRIK Studio установлена. Можно начинать. Удачи.

Встреча 1. Моторы вперед!

Для запуска программы ярлык, указывающий на исполняемый файл, лучше всего вынести на рабочий стол. Программы, с которыми работает TRIK Studio, называются проектами. Перед нами возникает окно с выбором проекта, над которым мы будем работать. Для начала нового проекта выбираем пункт меню «Создать проект». Центральная часть окна называется «Сцена», на ней и будут происходить основные действия.



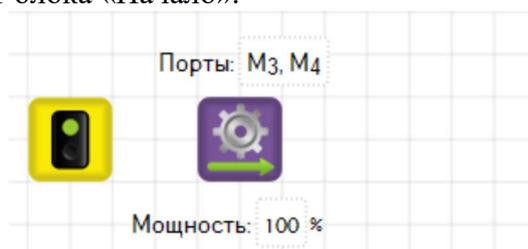
Справа находится окно «Палитра», в котором располагаются блоки, реализующие управление роботом, единственный блок, который ничего не делает – это блок «Комментарий», предназначенный для пояснения каких-нибудь действий либо для напоминания о чем-нибудь.



Любой проект начинается с блока «Начало». Удерживая левую кнопку мыши, захватываем блок «Начало» и перетаскиваем его в окно «Диаграмма поведения робота».

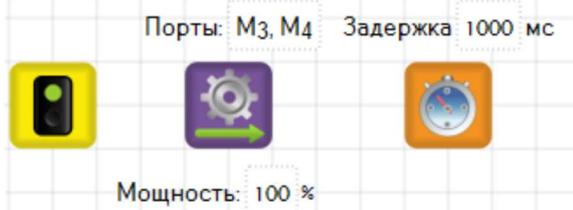


Начало положено. Попробуем заставить робота двигаться вперед. Движение осуществляется при помощи моторов. Ищем в палитре блок «Моторы вперед» и располагаем его справа от блока «Начало».



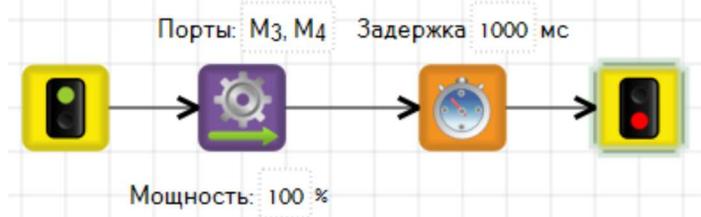
Отменить неправильное действие всегда можно при помощи стрелки «Отменить» или в меню «Правка» пункт «Отменить».

Чтобы наши моторы заработали, необходимо задать условие их работы, например, время движения. Перетаскиваем из палитры блок «Таймер» и дополняем им наш проект.



Заканчивается программа блоком «Конец».

Укажем порядок выполнения блоков в проекте, связывая их линиями соединения. Чтобы соединить два блока, нужно выделить первый блок, и зацепив синий кружок левой кнопкой мыши протянуть стрелку ко второму блоку. Если включены «Жесты», то, зажав правую кнопку мыши, можно провести линию от одного блока к другому, и тогда должна появиться стрелка, указывающая направление выполнения команд. Повторим это действие для всех пар блоков (если стрелка красная, это означает, что блоки не соединились). Программа «Моторы вперед» приобрела законченный вид.



Посмотрим на результаты ее работы. Для этого необязательно загружать программу в робота, можно воспользоваться двумерной моделью мира здесь же, в среде TRIK Studio. В меню «Настройки» выбираем «Роботы» – «Двумерная модель» и нажимаем «ОК». Запускаем программу кнопкой «Пуск», расположенной на верхней панели.



В окне «Двумерная модель» по полю 1 секунду двигается робот, назовем его ЭР-3К. В этом же окне расположены кнопки «Пуск» и «Стоп» для повторного запуска программы и остановки ее выполнения соответственно.



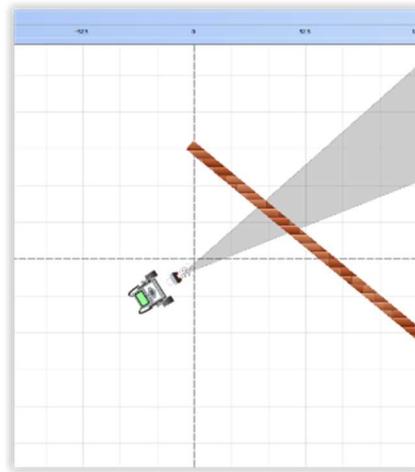
Начальное положение робота обозначено красным крестиком,



чтобы вернуть его в начальное положение, нужно нажать на кнопку



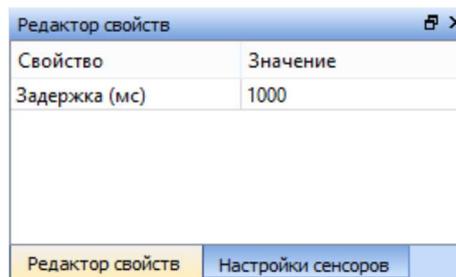
Для удобства позиционирования робота на сцену 2D модели нанесена координатная сетка с осями координат.



Сейчас мы находимся в режиме отладки, переключение в окно программы и обратно осуществляется нажатием на закладки:



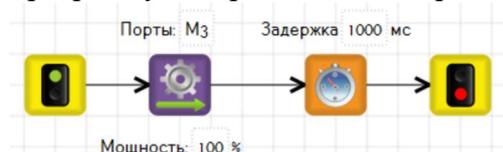
Рассмотрим нашу программу более подробно. Выделим блок «Таймер», нажав на него левой клавишей мыши. Слева в окне «Редактор свойств» появится запись «Задержка (мс) 1000». Это время, на которое установлен таймер. Поменять значение можно, щелкнув на нем и прописав нужное значение в миллисекундах (1000 миллисекунд равняются одной секунде).



У блока «Моторы вперед» в «Редакторе свойств» две записи.

Первая запись – порт. У контроллера ТРИК имеются четыре разъема, к которым можно присоединить четыре мотора. Разъемы обозначены на корпусе: М1, М2, М3 и М4. Это и есть порты. Наша программа использует М3, М4, то есть моторы подсоединены к разъемам М3 и М4. Порты можно поменять, щелкнув на записи «М3, М4» мышкой и вручную прописать нужные нам. Последняя запись – это мощность, с которой работают моторы.

Попробуем изменить программу. Уберем один из портов, например, М4.

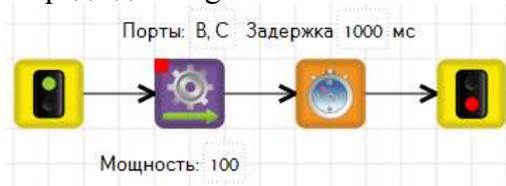


После запуска программы мы увидим, что ЭР-3К совершает поворот вокруг одного из колес, так как работает только один мотор, подключенный к разъему М3.

Сохраним результат нашей работы. Для этого выберем в меню «Файл» вкладку «Сохранить файл» или «Сохранить файл, как». Определим папку, где будет храниться наш проект, в поле «имя файла» вводим название для программы. Нажимаем «Сохранить». Программу TRIK Studio можно закрыть.

К контроллеру Lego EV3 тоже можно подключить 4 мотора, обозначаются они А, В, С и D.

Программа «моторы вперед» для Lego EV3:



Программа для поворота робота:

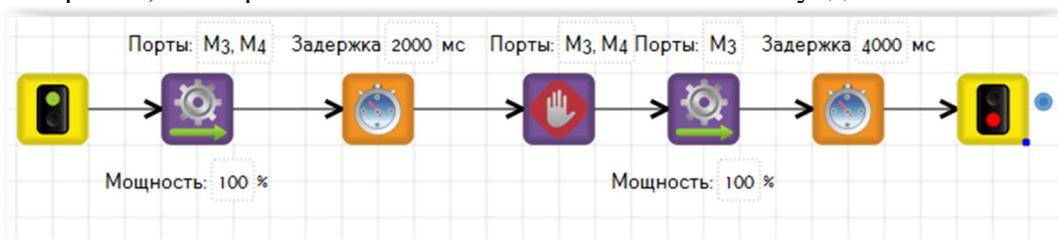


Алгоритмы сохранения проекта для ТРИК и Lego EV3 идентичны.

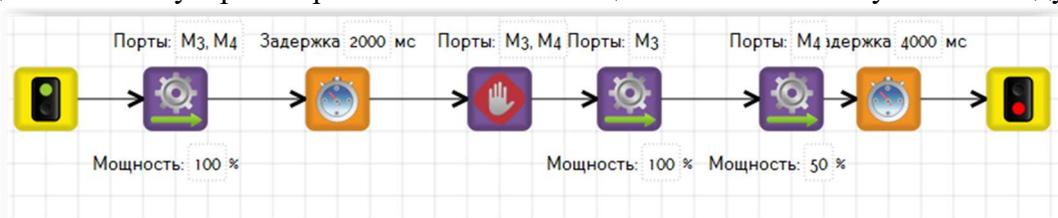
Задача: повторите маршрут ЭР-3К, который 2 секунды двигался на Север и в 2 раза дольше на Запад. (Север находится вверху сцены, Запад – слева. А где находятся Юг и Восток?)

Чтобы внести изменения в уже написанную программу, откроем сохраненный проект. В меню «Файл» выберем «Открыть». Находим папку, в которую была сохранена программа, выбираем нужный файл и нажимаем «Открыть».

На сцену добавляем блоки «Моторы вперед» и «Таймер». В первом блоке «Моторы вперед» в значении «Порты» возвращаем порт М3, а у второго блока «Моторы вперед» убираем порт М4, таймеры выставляем соответственно 2 и 4 секунды.



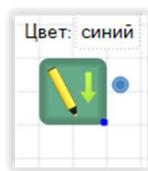
Попробуем совершить плавный поворот. Потребуется еще один блок «Моторы вперед». В нем мы уберем порт М3 и изменим мощность на 50%. Получилось следующее:



Эта же программа для Lego EV3:

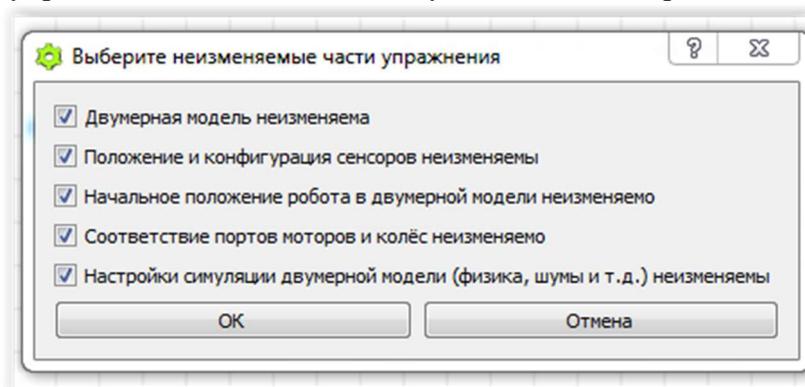


Траекторию, по которой ЭР-3К двигается по сцене, можно увидеть, используя блок «Опустить маркер» (только для двумерной модели).



Цвет траектории указывается в свойствах блока.

Любое выполненное задание можно сохранить в виде упражнения для дальнейшего его использования в качестве учебной задачи. В меню «Инструменты» выбираем «Сохранить как упражнение», появляется следующее окно настроек:



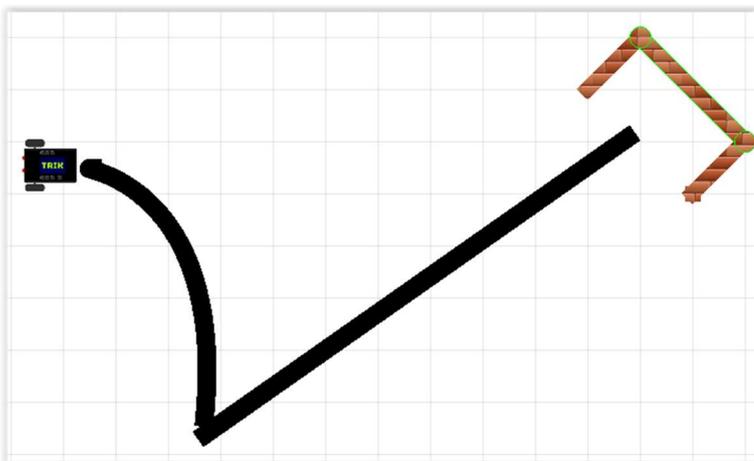
После нажатия «ОК» упражнение сохраняется с выбранными настройками.

Отступление 1. Жесты мышью

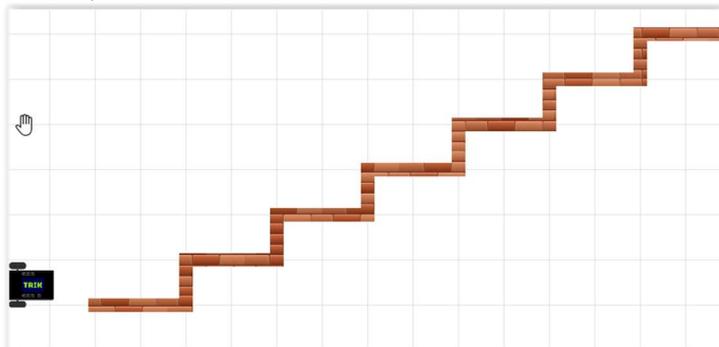
Разработчики программы TRIK Studio большое внимание уделяют usability – удобству использования. Для того чтобы каждый раз не искать нужный блок в палитре, достаточно изобразить в окне «Диаграмма поведения робота» определенный жест мышью, зажав правую кнопку. Ознакомиться со списком поддерживаемых жестов можно открыв меню «Инструменты» – «Жесты мышью». На устройствах, использующих сенсорный экран, данная опция просто незаменима.

Задания для закрепления материала

1. Определите, за сколько секунд выполняется поворот на 360 градусов обычный и «танковый».
2. Как будет выглядеть алгоритм «танкового» разворота на 180 градусов? «Танковый» разворот – это когда одно колесо крутится вперед, а другое – назад.
3. Сколько времени необходимо для того, чтобы описать полную окружность при следующих мощностях моторов: М3 – 100 %, М4 – 50%?
4. Составьте алгоритм движения по восьмерке.
5. Что изменится, если разность мощностей моторов при таком движении увеличится (уменьшится)?
6. Первое задание для нашего робота – «Патрулирование», ЭР-3К должен двигаться вдоль стены вперед-назад. Если вплотную к нашему роботу поставить еще двух точно таких же роботов и удвоить расстояние, которое составляет эта «цепочка», то получится протяженность стены.
7. Задача в патруле усложнилась, достигнув конца стены, необходимо развернуться на месте и двигаться «лицом» вперед
8. После удачного патрулирования стены ЭР-3К доверили «секретный объект», вокруг которого нужно очертить границу красного цвета. Объект представляет собой квадратное кирпичное здание со стороной вдвое меньшей, чем стена из задачи 6.
9. «Секретных объекта» стало два, они удалены друг от друга на пять корпусов робота, требуется провести границу вокруг них. Рассмотрите все варианты патрулирования.
10. ЭР-3К отправляется на «Базу» по заранее прочерченному маршруту. Траекторию, по которой будет двигаться робот, нарисуйте в окне «Двумерная модель», используя кривую Безье и линию.



11. Чтобы попасть на «Базу», необходимо подняться по лестнице с 10-ю ступеньками (лестницу рисуем при помощи стены, длина ступеньки 2 корпуса робота, высота в два раза меньше).



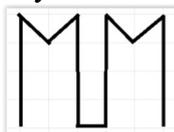
12. Задача «Стадион». Наш робот тренируется перед соревнованиями. Беговая дорожка ограничена двумя прямоугольниками, нарисованными синим цветом, один из прямоугольников находится внутри другого. Длина маленького прямоугольника равна ширине большого, а длина большого прямоугольника в два раза больше ширины маленького. Протяженность беговой дорожки вдоль короткой стороны стадиона 6 корпусов робота.

13. Задача «Челночный бег». На полу нарисованы 4 цветные линии, старт производится из-за первой линии. Добегая 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).

14. Новое увлечение ЭР-3К – мадонари (рисунки на асфальте, произведения городской живописи), первый его рисунок – луна.



15. Рисуем «Кремлевскую стену».



16. Еще одним произведением ЭР-3К стал узор в виде зигзага, напоминающий зубчики пилы.



17. Напишите на сцене при помощи инструмента «Маркер» свое имя.

18. На сцене начерчена черная линия длиной 4 клетки. Сотрите ее.

19. ЭР-3К, нанося дорожную разметку, совершил ошибку, теперь ему из сплошной линии нужно сделать прерывистую.

20. Нарисуйте два прямоугольника так, чтобы один из них был внутри другого. Размеры большого прямоугольника 7x5 клеток, заливка синяя, вложенного – 5x3, заливка зеленая. Напишите программу деления прямоугольников черной линией на 4 равные части (рассмотрите все варианты).

21. Нарисуйте красным маркером квадрат, вписанный в желтый круг радиуса 4 клетки.

22. С помощью рисунков можно передавать сообщения, а если изучить азбуку Морзе, то и целый текст. Азбука Морзе переводит буквенные символы в точки и тире

Азбука Морзе

А	• —	П	• — — •	Ь	— • • —
Б	• • • •	Р	• — •	Ы	— • • —
В	• — —	С	• • •	Й	• — — —
Г	• — •	Т	—		
Д	• • •	У	• • —	1	• — — — —
Е	•	Ф	• • — •	2	• • — — —
Ж	• • • —	Х	• • • •	3	• • — — —
З	• — — •	Ц	• — — •	4	• • • —
И	• •	Ч	• — — —	5	• • • •
К	• • — •	Ш	— — — —	6	• • • • •
Л	• — • •	Щ	• — — —	7	• — — • •
М	• — —	Э	• • • •	8	• — — — •
Н	• •	Ю	• • — —	9	• — — — •
О	— — —	Я	• • • —	0	• — — — —

Отправьте сообщение: «Привет ЭР 3К».

23. Передайте при помощи азбуки Морзе результат вычисления выражения $21-12+5-10+1$.
24. Расшифруйте и передайте азбукой Морзе послание: 00100 010 10010 0011. Ключ к шифру: 0 – точка, 1 – тире, а первые и последние знаки в букве поменяли местами между собой.
25. Объезжая поочередно небольшие препятствия, ЭР-3К вычерчивает красно-желтую кривую, при левом повороте кривая красная. Поле для «Слалома»: количество препятствий – 7, расстояние между ними 3 корпуса робота. Препятствия –

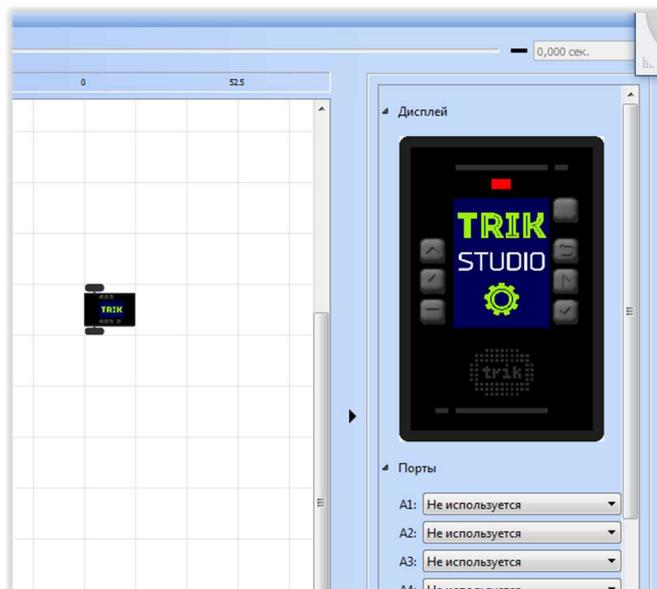
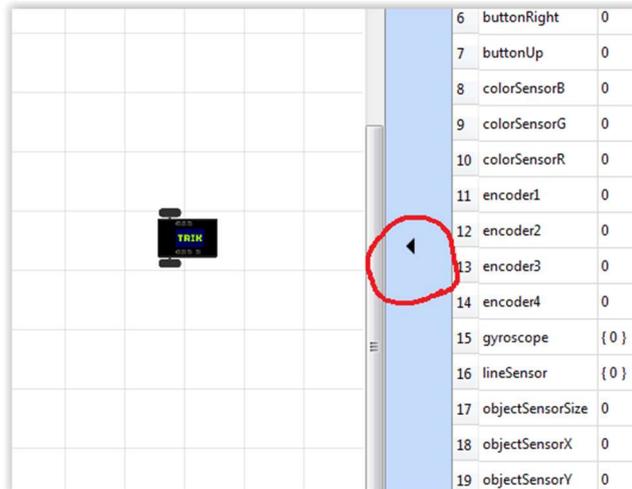


26. ЭР-3К вычерчивает цветные окружности разных радиусов, начинающиеся в одной точке, диаметры окружностей больше длины корпуса робота.
27. Нарисуйте на сцене равносторонний треугольник. Внутри него высоту к одной из сторон. Основание треугольника – 4 клетки.
28. ЭР-3К движется по квадрату, отмечая углы точками при помощи маркера.
29. Робот сначала рисует, а потом стирает круг.

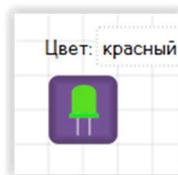
Встреча 2. Робот на связи

Часто для проверки и отладки программ требуется, чтобы робот сам сообщал нам о выполнении команд. Такая связь с исполнителем называется обратной. Рассмотрим, какие сигналы может подавать нам робот ЭР-3К.

В окне «Двумерная модель» за маленькой стрелочкой прячется изображение контроллера ТРИК:

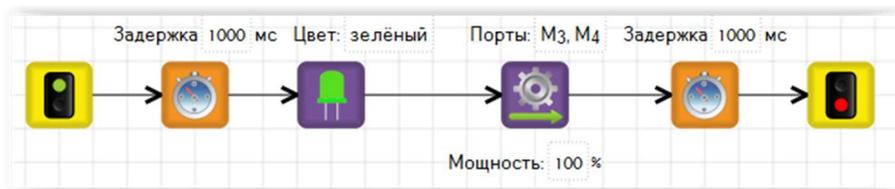


Выше дисплея на контроллере расположен светодиод, способный изменять свой цвет. В палитре для него есть иконка «Светодиод»:

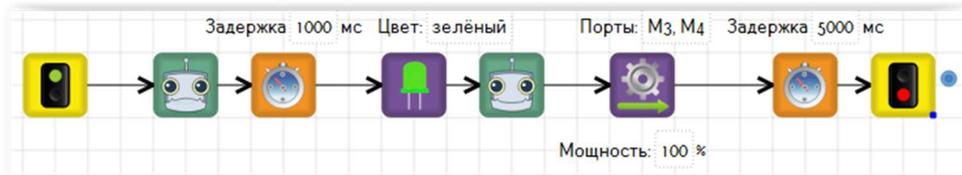


В редакторе свойств цвет светодиода меняется на зеленый, оранжевый, красный или отключается.

Получим первую обратную связь от нашего робота. Программа «Светофор»: после того, как сигнал светодиода изменит цвет с красного на зеленый, ЭР-3К начинает движение.

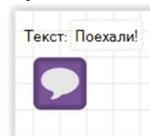


Дисплей контроллера также может быть элементом обратной связи. Поместим перед началом движения иконку «Грустный смайлик» из раздела «Рисование» палитры и иконку «Смайлик» после того, как загорится зеленый светодиод (для контроллера Lego NXT иконки со смайликами недоступны).

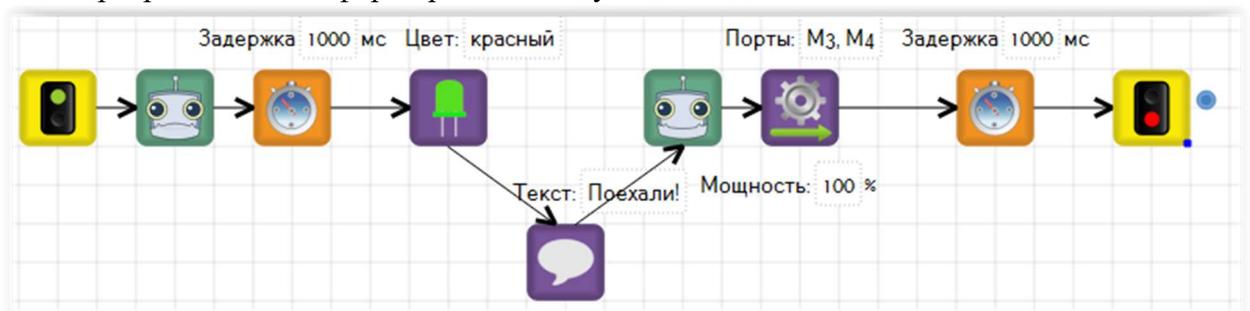


И вот ЭР-ЗК радостно катится вперед по своим делам.

Визуальная связь с нашим роботом установлена, добавим звуковых эффектов. Контроллер TRIK умеет генерировать речь, как на английском, так и на русском языке. В свойствах иконки «Сказать» указываем нужный текст.

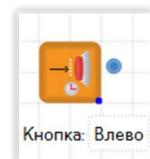


Программа «Светофор» приняла следующий вид.

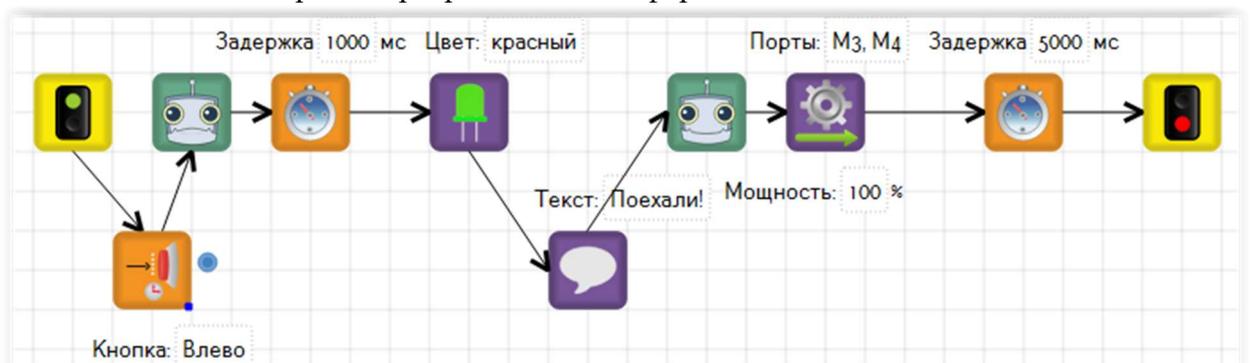


В 2D модели появляется окно в правом верхнем углу со словом «Поехали».

Добавим в программу команду «Старт»: действия начинаются только после нажатия кнопки «Влево» на контроллере.

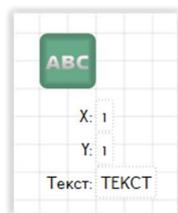


Законченный вариант программы «Светофор»:



Не забудьте нажать кнопку «Влево».

При необходимости на дисплей можно вывести любой текст. Воспользуемся блоком «Напечатать текст»:



X, Y – это координаты начала текста на дисплее.

Для контроллера Lego EV3 в качестве обратной связи можно использовать блоки «Светодиод», «Гудок» и «Играть звук», управление нажатием кнопок – как у контроллера ТРИК.

Задания для закрепления материала

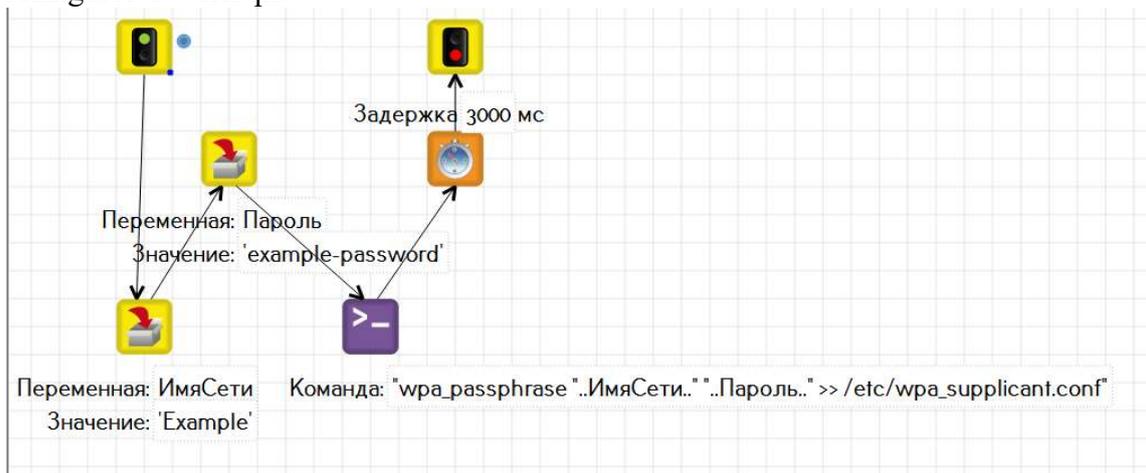
1. Озвучим кнопки контроллера: при нажатии на кнопку контроллер сообщает ее название.
2. Нажимая несколько раз кнопку «Вверх», измените цвет светодиода сначала на оранжевый, затем на зеленый.
3. Кнопка «Влево» – грустный смайлик, «Вправо» – веселый.
4. ЭР-3К едет вперед только после нажатия кнопки «Влево». Пока кнопка не нажата, на экране написано «Нажмите кнопку «Влево»».
5. Теперь мы можем управлять роботом. Пусть он начинает движение после нажатия кнопки «Влево» и останавливается после нажатия кнопки «Вправо».
6. Рисуем окружность: при нажатии кнопки «Влево» – движение прямо, если нажата кнопка «Вправо» – движение по окружности.
7. Робот двигается всего две секунды, после чего стоит и улыбается. Но его можно заставить двигаться в три раза дольше нажатием кнопки «Вниз», на дисплее появится грустный смайлик.
8. ЭР-3К иногда разговаривает сам с собой, например, когда он начинает двигаться вперед, он говорит «Еду вперед», при остановке – «Остановился, стою на месте», а при движении назад – «Еду назад». Запрограммируйте такое поведение робота.
9. Напишите программу для светофора, использующего светодиод контроллера. Красный цвет горит в два раза дольше зеленого, а оранжевый моргает 4 раза в течение 8 секунд. Время одного цикла светофора – 26 секунд.
10. Если у ЭР-3К хорошее настроение, то он светит зеленой лампочкой, если плохое, то оранжевой. Настроение показывает смайлик на дисплее.
11. У нашего робота нет указателей поворота, поэтому, делая обход вокруг гаража, он сообщает, в какую сторону происходит поворот.
12. ЭР-3К очень любит рисовать, и когда он рисует маркером круг, на экране веселый смайлик, а по окончании работы он радостно сообщает «Рисунок готов».
13. Разметка дороги: необходимо нарисовать двойную сплошную линию, при этом, рисуя 2-ю линию и двигаясь назад, необходимо подавать звуковой сигнал «Я двигаюсь назад».
14. Поставить 5 меток разных цветов, сообщая о цвете метки.
15. Пешеходный переход для людей со слабым зрением: когда загорается зеленый свет, раздается звуковой сигнал «Переход разрешен», за 5 секунд до красного – «Заканчивайте переход», перед красным сигналом – «Переход запрещен».
16. Лифт: составьте программу для движения лифта с 5 до 1 этажа с остановкой на 4 и 2 этаже. Скорость кабины 5 м/с, высота этажа 2.5 м. Двери открываются и закрываются в течение 12 с, пауза для входа-выхода – 10 с. Автоматический информатор сообщает номер этажа, на котором произошла остановка, и выводит его на дисплей.
17. Правда ли то, что расстояние, пройденное роботом за 5 секунд при мощности моторов 100% равно расстоянию, пройденному за 10 секунд при мощности – 50%?
18. В кафе размером 10x10 клеток в два ряда расположены 4 столика 4x4 клетки. Робот-официант находится возле двери на кухню в левом нижнем углу. Составьте оптимальную программу для обслуживания посетителей. При доставке заказа робот должен сообщить об этом.
19. ЭР-3К двигался 5 секунд с мощностью моторов 40% и 1 секунду с мощностью 100%. С какой мощностью моторов он должен ехать, чтобы за то же время пройти такое же расстояние, не меняя мощности?
20. К контроллеру подключен насос, качающий воду. При нажатии кнопки «Вниз» начинает литься вода, кнопкой «Вверх» останавливает воду.

Встреча 3. Моторы вперед-2

Использовать 2D модель очень удобно, но гораздо интереснее увидеть, как же будет двигаться наш робот по-настоящему.

Для подключения к роботу необходимо настроить Wi-Fi сеть. Контроллер TRIK работает как роутер, если в меню контроллера выбрать «Сеть» - «Точка доступа». На дисплее высветится IP-адрес контроллера, имя сети и пароль.

Если сеть раздается другим устройством, можно поступить следующим образом: в среде TRIK Studio в папке с примерами имеется программа для настройки сети: `configureNetwork.qrs`.



Загружаем ее и выполняем следующий алгоритм:

- включить робота в режиме точки доступа «Сеть» - «Точка доступа»;
- перевести TRIK Studio в режим работы с реальным роботом (интерпретация Wi-Fi) кнопкой на панели инструментов:



- подключиться к роботу по Wi-Fi (проверить корректность можно кнопкой «Подключиться» на панели инструментов):



- поменять значение переменной "ИмяСети", введя имя сети (SSID) в кавычках;
- поменять значение переменной "Пароль", введя пароль сети;
- запустить программу кнопкой «Выполнить»:



- после того, как программа отработает (появится меню на роботе), выбрать режим «WiFi-клиент» в меню «Сеть»;
- выбрать вновь добавленную сеть и нажать кнопку «Ввод» на роботе;
- робот через некоторое время должен подключиться к сети и показать выданный ему IP-адрес в верхней части окна настройки сети.

Теперь программы можно загружать непосредственно на робота с помощью кнопки



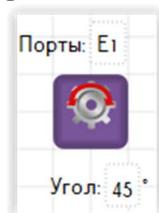
в режиме «Генерация» или работать без загрузки программ в режиме «Интерпретация».

Настройки контроллера Lego EV3 описаны в руководстве пользователя или на официальном сайте <http://www.lego.com/ru-ru/>.

Отступление 2. Ещё больше возможностей

На момент написания пособия в 2D модели отображались только два силовых мотора, шесть аналоговых датчиков и два цифровых. У контроллера ТРИК еще два силовых мотора и шесть сервомоторов в двумерном мире не существуют (у контроллера Lego EV3 – два силовых мотора), поэтому в задачах, которые используют более 2 моторов или сервомоторы, 2D модель может передать только движение робота.

Для программирования сервомоторов используется блок «Угловой сервомотор»:



В свойствах блока можно настроить порт мотора и угол поворота в градусах.

И ещё одно замечание: к портам силовых моторов мы можем подключать не только моторы, но и любую нагрузку 12 V, например, LED-ленту или лампочку.

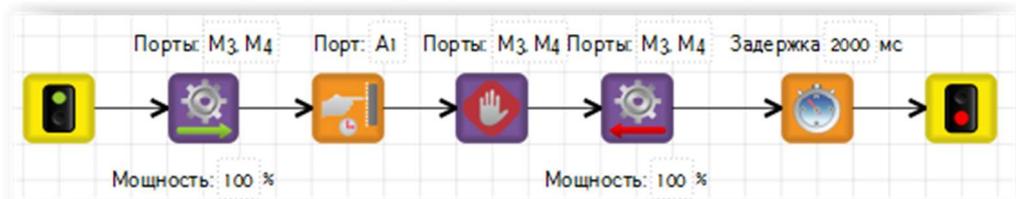
Задания для закрепления материала

Встреча 4. Впереди стена

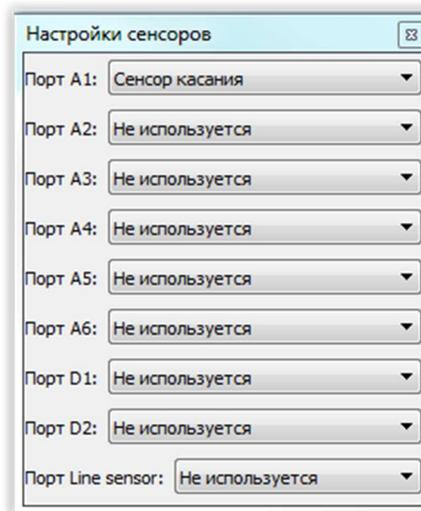
Мы уже знаем, что на контроллерах находятся разъемы для подключения моторов. На контроллере ТРИК также присутствуют порты А1-А6, D1, D2. Это порты для подсоединения датчиков. У контроллера Lego EV3 это порты 1, 2, 3 и 4.

Попробуем решить следующую задачу: ЭР-3К как мячик катится вперед, и, ударившись об стену, отскакивает назад.

Для решения задачи используем датчик касания. В палитре для него есть блок «Ждать датчик касания». Блок «Ждать датчик касания» является условием для осуществления движения, поэтому таймер нам не нужен, моторы будут работать, пока не сработает датчик касания. В свойствах блока необходимо указать порт, к которому он подсоединен. Итак, ЭР-3К двигается вперед, пока датчик не сработал, затем останавливает моторы, запускает их в обратную сторону и едет назад две секунды.



Мы написали программу, но контроллеру нашего робота надо сообщить, что к порту А1 присоединен сенсор касания. Для этого в настройках сенсоров выставляем для «Порта А1» сенсор касания. Датчик касания – аналоговый. Аналоговые датчики подключаются к портам А1-А6.



В окне «Отладка» на роботе появится изображение датчика, который с помощью мыши надо перетащить к передней (как в нашем примере) или задней части робота.

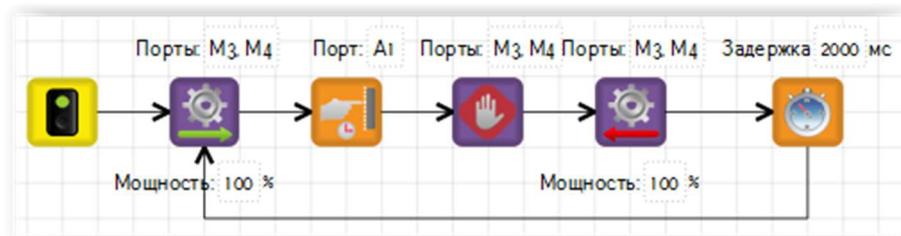


Строим стену при помощи панели «Инструменты»

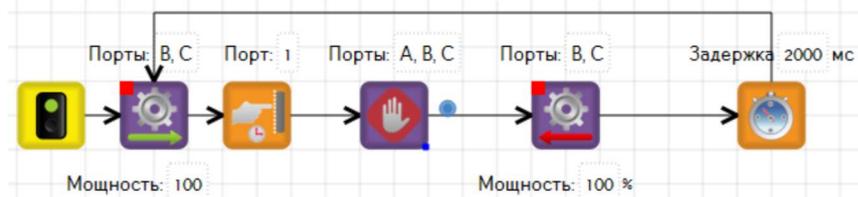


и запускаем программу.

«Штурм замка»: если несколько раз запустить нашу программу, покажется, что ЭР-ЗК пытается как таран проломить стену. Поможем ему в этом. Уберем блок «Конец» и соединим блоки «Таймер» и «Моторы вперед». Программа будет выполняться, пока мы не нажмем «Стоп».



Для решения этой задачи на контроллере Lego EV3 используем датчик касания. Не забудем выставить в настройках сенсоров, что к «Порту 1» присоединен сенсор касания.



Точно так же, как и на контроллере ТРИК, датчик нужно установить спереди:



Что делать, если мы не хотим касаться стены (например, если она окрашена)? Изучим еще один датчик – датчик расстояния (сонар). Задача прежняя: доехав до стены, отъехать назад, но на этот раз без касания. В свойствах блока «Ждать сонар» видим следующие свойства: «Дистанция», «Порт», «Считанное значение».

«Дистанция»: выставляем значение до объекта в сантиметрах, в нашем случае не доедем до стены 15 сантиметров.

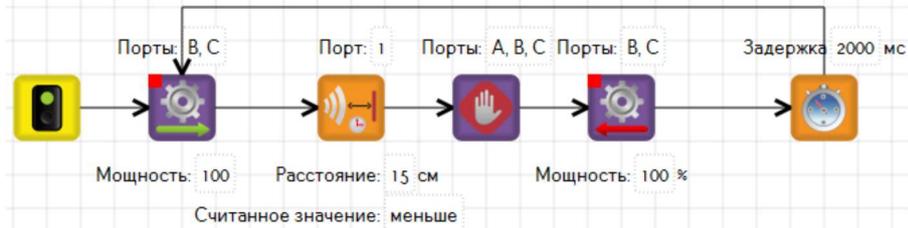
«Порт»: номер порта, к которому подключен датчик. Поскольку ультразвуковой датчик расстояния цифровой, подключение осуществляется к порту D1 или D2.

«Считанное значение» зависит от условия задачи, у нас оно должно стать меньше пятнадцати сантиметров.

Алгоритм решения задачи для ТРИК:



Для Lego EV3:



Не забудем выставить в настройках сенсоров, что к порту D1 (для Lego EV3 «1») присоединен сенсор расстояния.

Отступление 3. Объемный мир

2D модель работает только на плоскости, поэтому задачи, в которых датчики направлены вверх или вниз, в ней отображаются некорректно. Для отладки таких программ лучше использовать «живого» робота.

Задания для закрепления материала

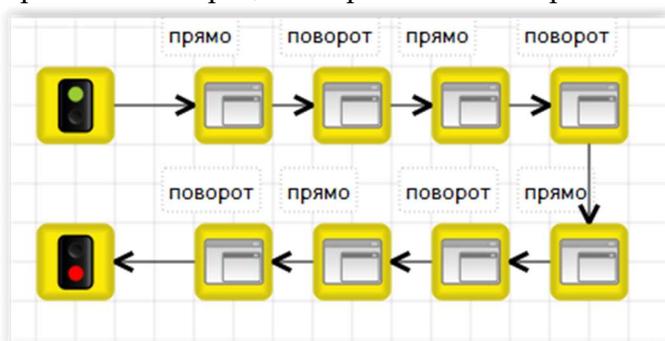
Встреча 5. Подпрограмма – это маленькая программа

Часто бывает, что какие-то операции, выполняемые роботом, повторяются в различных частях программы или в разных программах. Их можно выделить и описать отдельно в блоке «Подпрограмма». Для того чтобы в дальнейшем не выполнять одну и ту же работу, программисты составляют свою собственную библиотеку подпрограмм и используют их при необходимости. Перетащим блок «Подпрограмма» в окно «Диаграмма поведения робота». У каждой подпрограммы свое название. Щелкнув мышкой на блоке, откроем окно подпрограммы.

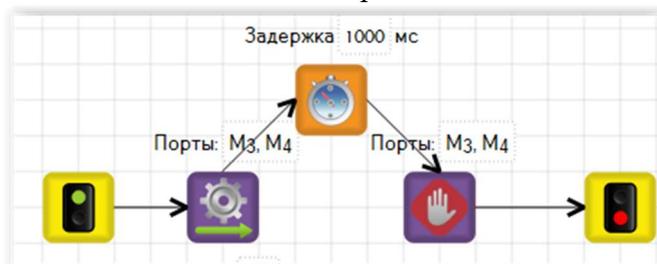
Начало любой подпрограммы – блок «Начало». Заканчивается она блоком «Конец». В отличие от программы, подпрограмма не может быть бесконечным циклом.

Задача: объехать прямоугольное здание по периметру.

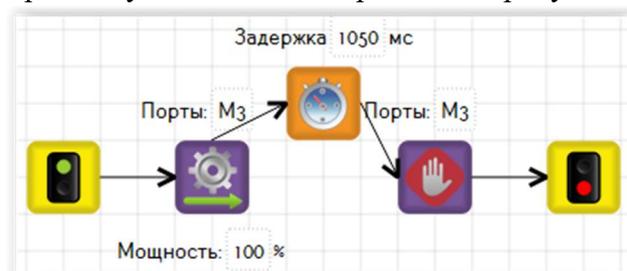
Разделим движение робота на операции: «Прямо» и «Поворот».



Подпрограмма «Прямо» – это движение по прямой.



Подпрограмма «Поворот» осуществляет поворот на 90 градусов.



Задания для закрепления материала

Встреча 6. Лабиринт

Поиск выхода из лабиринта является классической задачей, которую решают не только робототехники, но и программисты.

Для начала допустим, что лабиринт нам известен. Напишем алгоритм прохождения по заданному маршруту. Выделим подпрограммы:

«Шаг вперед»:



«Поворот налево»:



«Поворот направо»:



Подпрограмму «Шаг назад» напишите самостоятельно.

Таймер для выполняемых действий подбирается в зависимости от размеров лабиринта.

Для Lego EV3:



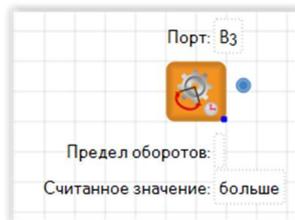
Программа будет представлять собой набор действий, описанных выше, выполненных в необходимом нам порядке.



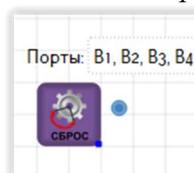
В случае если лабиринт неизвестен, необходимо использовать датчик расстояния для обнаружения стены, преграждающей дорогу. Основные алгоритмы прохождения лабиринта будут рассмотрены после изучения конечных и бесконечных циклов.

Отступление 4. Непостоянное время

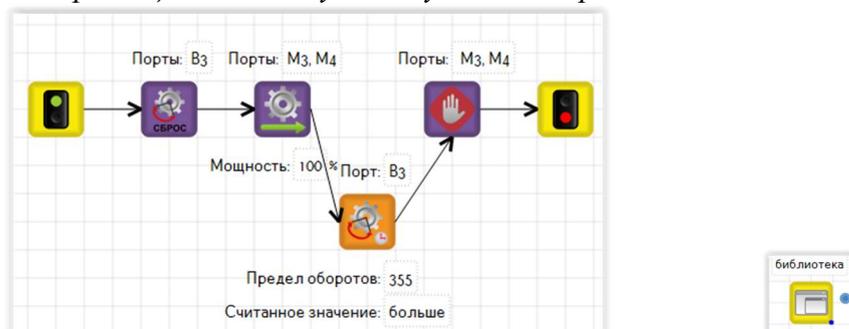
Моторы, подключаемые к контроллерам, часто оснащаются энкодерами – датчиками количества оборотов. Если нам необходимо переместить реального робота на заданное расстояние, то использование таймера не дает достаточной точности передвижения, поскольку сильно зависит от физических факторов, например, таких, как заряд аккумулятора или состояние поверхности, по которой происходит передвижение. Использование энкодеров частично решают эту проблему. В TRIK Studio для ожидания количества оборотов используется блок «Ждать энкодер»:



При подсчете оборотов мотора не забывайте обнулять показание энкодера в нужное время, в том числе и в начале программы, применяя блок «Сбросить показания энкодера», так как подсчет оборотов ведется непрерывно.



Программа перемещения на одну клетку по энкодерам выглядит так:

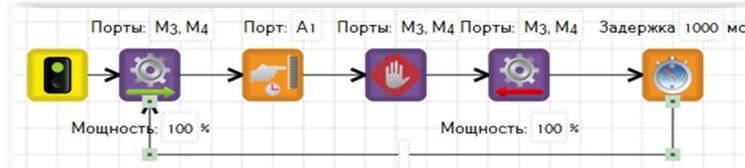


Задания для закрепления материала

Встреча 7. Бег по кругу

Роботы часто выполняют одну и ту же задачу неоднократно. Мы уже видели, как можно заставить программу выполняться бесконечное число раз. Но иногда достаточно, чтобы операция была исполнена 5, 10 или 100 раз. В программе TRIK Studio это возможно благодаря оператору «Цикл» из палитры.

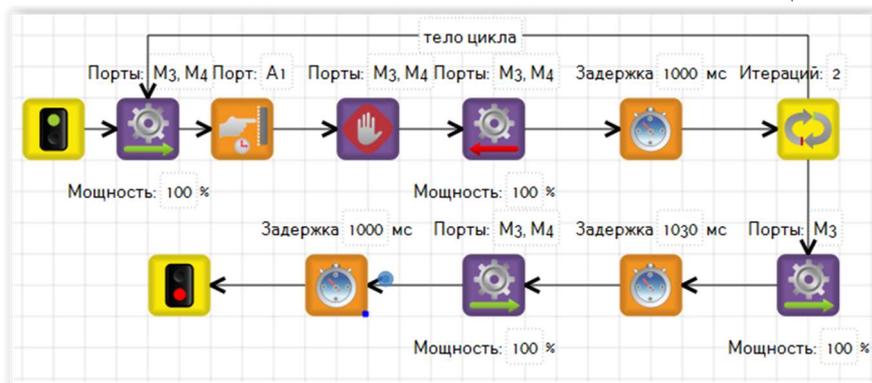
Вернемся к задаче «Штурм замка». Бесконечный цикл, который мы в ней использовали, выглядел следующим образом:



Пусть в замке проживает наш хороший знакомый, но его нет дома. Постучимся три раза, развернемся и уедем.

Используем блок «Цикл». Из оператора «Цикл» исходят две линии соединения: одна направлена на начало цикла, вторая отправляет нас дальше: на действие после цикла.

Выделим блок «Цикл». В свойствах указаны итерации, то есть нужно определить, сколько раз мы будем выполнять наше действие. Щелкаем на линию соединения, смотрящую на начало цикла, и в ее свойствах указываем условие – «Тело цикла». Последний блок исполняемого действия цикла соединяем с блоком «Цикл».



Для Lego EV3:



Итак, мы рассмотрели два типа циклов – бесконечный (безусловный) и с итерациями. Циклы с условием будут показаны после изучения операторов ветвления.

Задания для закрепления материала

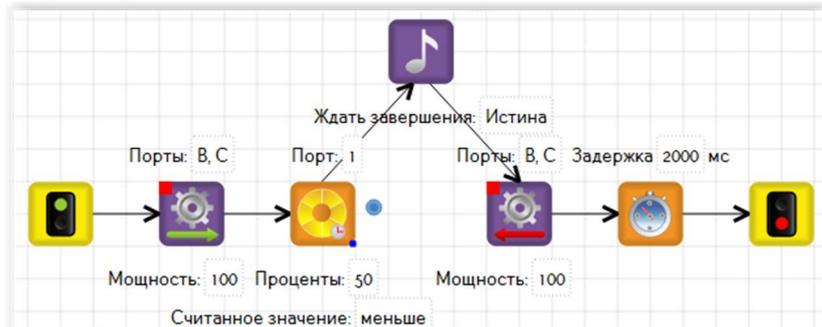
Встреча 8. Разноцветная дорога

Мы выяснили, что ЭР-ЗК может не только передвигаться, но и видеть предметы. Еще одна способность робота – это определение цвета и света. Появляется она благодаря соответствующим датчикам. В одной из задач мы успешно избежали касания окрашенной стены. Но что делать, если окрашен пол? Попробуем использовать датчик света. Любая поверхность отражает свет, но количество отраженного света зависит от свойств поверхности, чем темнее поверхность, тем меньше она отражает. Допустим, что неокрашенный пол – белый, а покрашенный – черный.

Датчик света возвращает значения от 0 до 100, где 100 соответствует белому свету и 0 – черному. Для датчика света в палитре имеется блок «Ждать свет». В свойствах блока видим три записи: «Проценты» – количество света (от 0 до 100), «Порт к которому подключаемся» и «Считанное значение», в нашем случае меньше 50 процентов. В настройках сенсоров контроллера выставляем сенсор света, подключенный к порту А1.

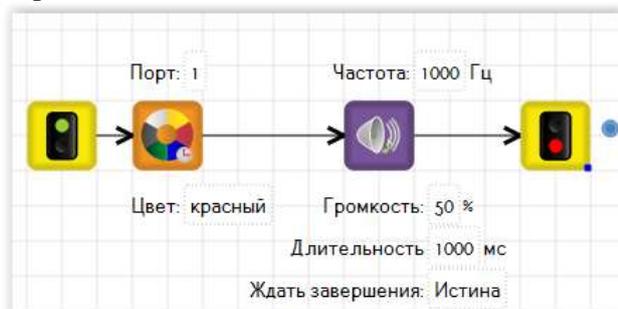


Для контроллера Lego EV3:



Теперь пятно на полу нам не страшно. В качестве обратной связи в задачах используются: для ТРИК – блок «Сказать», для Lego EV3 – блок «Гудок».

Контроллер ТРИК определяет цвета по видеокамере, а не по датчикам, поэтому следующая задача только для контроллеров Lego (работа с видеокамерой будет описана позже, в том числе и определение интенсивности света). Программа «Спелый фрукт»: определяем, что предмет красный.



В свойстве блока «Ждать цвет» указываем цвет, а в свойстве «Порт» – порт подключения датчика. Обратная связь – блок «Играть звук».

Отступление 5. Раздвигаем горизонты

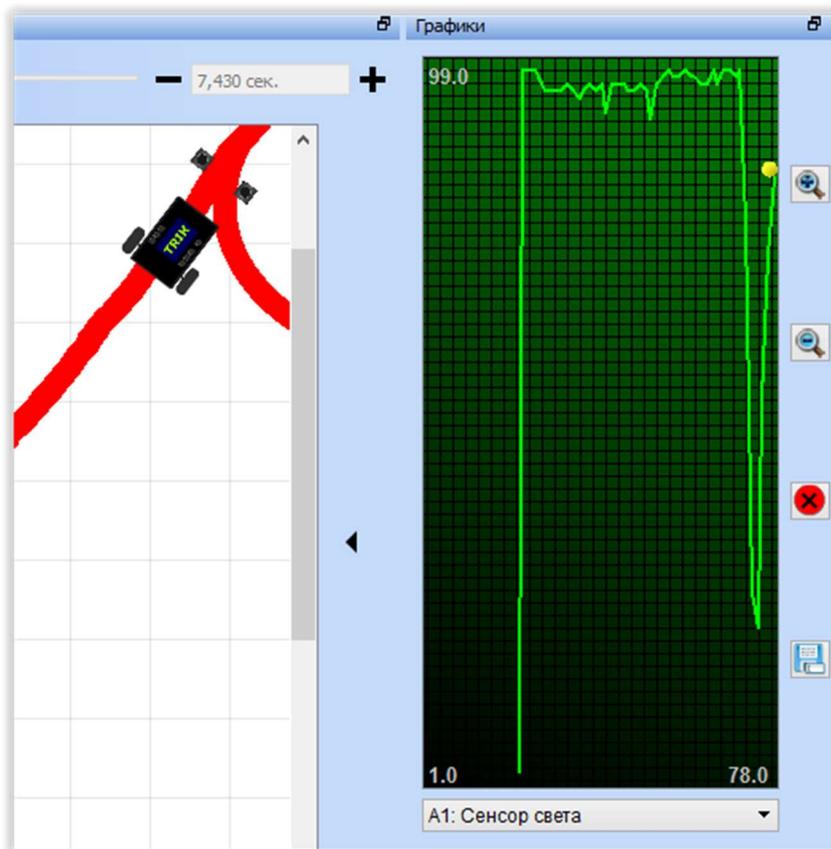
Набор датчиков, входящий в комплектацию образовательных наборов, можно существенно расширить, не прибегая к специальной доработке контроллера. Ниже приведен список датчиков, которые уже были опробованы с контроллером TRIK:

- датчик температуры -LM35DT/NOPB;
- датчик влажности - HR202, модуль датчика влажности воздуха [Гигрометр];
- датчик углеводородных газов - модуль датчика углеводородных газов MQ-2;
- датчик освещенности - модуль датчика освещенности KY-018;
- датчик звука - микрофон AVR PIC KY-037;
- датчик вибрации - модуль датчика вибрации;
- датчик Холла - MA3144 цифровой модуль датчика Холла;
- датчик движения- HC-SR501;
- потенциометр - модуль потенциометра;
- датчик тока - ACS712 модуль, датчик тока [5A];
- датчик дождя;
- датчик пламени - датчик огня;
- датчик дыма - модуль датчика газа MQ-135;
- кнопка - кнопка концевая (положения) 3-pin;
- датчик угарного газа - модуль датчика угарного газа MQ-7;
- датчик уровня жидкости;
- расхода воды - YF-S201, расходомер воды [1-30 л/мин];

В TRIK Studio для датчиков, не входящих в комплект поставки, в «Настройка сенсоров» их названия отсутствуют. Для работы с такими датчиками достаточно выбрать любой из имеющихся в настройках сенсоров, соблюдая правило – аналоговые датчики на портах A1-A6, цифровые – D1-D2.

Отступление 6. Графики значений датчиков

В среде TRIK Studio можно увидеть графическое представление значений датчиков. В режиме «Отладка», закладка «Графики». Датчик, значения которого нужно посмотреть, выбирается в выпадающем меню.



Числовые данные экспортируются в файл формата <Имя_Файла>.csv, который можно открыть, например, в MS Excel.

The screenshot shows an MS Excel spreadsheet with the following data:

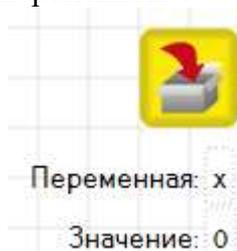
	A	B	C
1	time	value	
2	0	1	
3	1	99	
4	2	99	
5	3	99	
6	4	99	
7	5	99	
8	6	98	
9	7	98	
10	8	96	
11	9	96	
12	10	96	
13	11	96	
14	12	96	

Задания для закрепления материала

Встреча 9. Познакомимся с неизвестными

При изучении явлений природы и общества мы на каждом шагу сталкиваемся с различными величинами. Одни из них являются постоянными, неменяющимися, например, количество окон в классе или скорость света, другие же меняются в зависимости от условий. Первые величины называются постоянными или константами, вторые – переменными. В программировании переменная – это объект, которому дано имя и который может принимать различные значения.

Для объявления переменных и задания им определенных значений в TRIK Studio используется блок «Инициализация переменной».



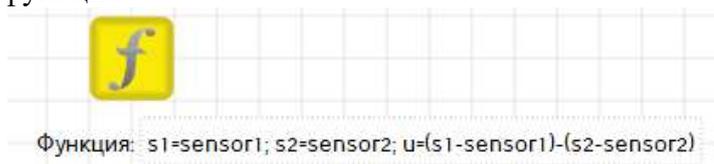
В свойствах блока имя переменной и ее значение.

Программа «Челночный бег»: бег от одной линии к другой, меняя направление на противоположное.



В этой задаче переменную Y мы объявляем 11 раз, и на каждом шаге ее значение разное, либо 1, либо (-1). Заметим, что мощность моторов также является величиной переменной: 100 или (-100). И еще одно наблюдение: при мощности (-100) действие «Моторы вперед» становится действием «Моторы назад».

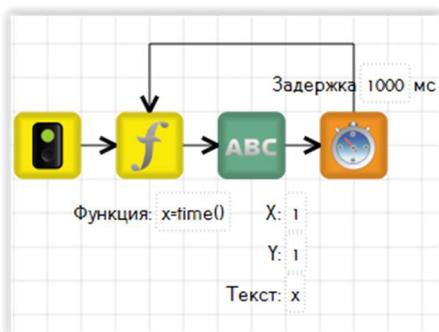
Для объявления сразу нескольких переменных или больших вычислений используется блок функция:



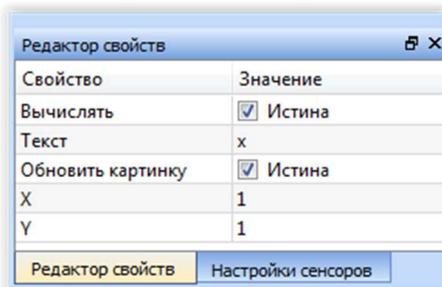
В правой части строки «Значение» разрешено использовать круглые скобки, числа, базовые арифметические операции, ранее определенные переменные, а также зарезервированные переменные, список которых можно посмотреть в режиме «Отладка» в правой части экрана.

Более подробно узнать о доступных в среде математических выражениях и операциях с ними можно прочитав раздел «Синтаксис выражений» в файле помощи, который вызывается нажатием клавиши F1 или из меню «Справка». Функции, поддерживаемые средой TRIK Studio, также подробно указаны в файле помощи.

Приведем пример применения встроенной функции time:



Программа выводит на дисплей время, прошедшее с начала исполнения. Обратим внимание, что на дисплей выводится числовое значение функции, а не строка, поэтому не забывайте ставить галочку в «Редакторе свойств», раздел «Вычислять».



Для строковых переменных работают те же правила:



Раздел «Вычислять» – Истина.

Уточним, что же такое зарезервированные переменные.

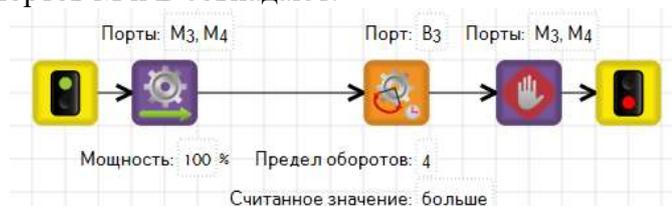
Имя	Значение
4 buttonDown	0
5 buttonEnter	0
6 buttonEsc	0
7 buttonLeft	0
8 buttonRight	0
9 buttonUp	0
10 colorSensorB	0
11 colorSensorG	0
12 colorSensorR	0
13 encoder1	0
14 encoder2	0
15 encoder3	0
16 encoder4	0
17 gamepadButton1	0
18 gamepadButton2	0
19 gamepadButton3	0
20 gamepadButton4	0
21 gamepadButton5	0

Как мы уже знаем, к контроллеру ТРИК можно присоединять датчики. В каждый момент времени датчики возвращают какое-то числовое значение на контроллер. Это и есть значение переменной sensorN.

Например, у нас к порту A1 подсоединен датчик расстояния. Робот находится вплотную около стены. Значение переменной sensorA1 равно 0. После перемещения робота на некоторое расстояние от стены значение изменится в большую сторону. То есть значение переменной sensorA1 – это то, что показывает датчик расстояния в данный момент времени.

Рассмотрим переменные encoder. В конструкторах используются моторы со встроенным датчиком оборотов. Поэтому мы всегда можем узнать, сколько оборотов совершил мотор, либо сами задать нужное количество оборотов.

Задача: проехать роботом 4 оборота колеса. Для решения этой задачи на контроллере ТРИК необходимо подключить разъемы моторов не только к портам M, но и к портам B – энкодерам. Номера портов M и B совпадают:

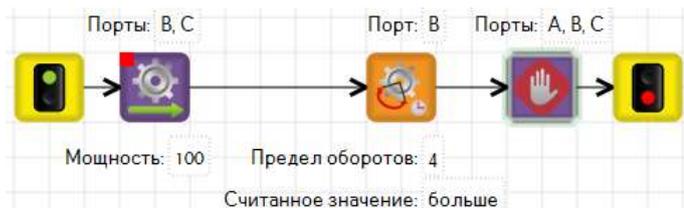


Датчику оборотов в палитре соответствует блок «Ждать энкодер».

Для Lego EV3 также есть зарезервированные переменные, но их значительно меньше:

Переменные		
	Имя	Значение
1	buttonEnter	0
2	buttonEscape	0
3	buttonLeft	0
4	buttonRight	0
5	encoderA	0
6	encoderB	0
7	encoderC	0
8	pi	3.1415926535897931
9	sensor1	0
10	sensor2	0
11	sensor3	0
12	sensor4	0

Для решения задачи с энкодерами на Lego EV3, в отличие от ТРИК, достаточно подключить моторы к соответствующим портам, у Lego EV3 нет выделенных портов для энкодеров.

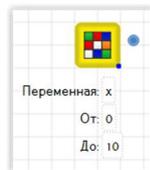


Зарезервированные переменные buttonUp, buttonDown, buttonLeft, buttonRight, buttonEsc и buttonEnter – это функции нажатия соответствующих кнопок на корпусе контроллера «Вверх», «Вниз», «Влево», «Вправо», «Отмена», «Ввод», принимающие значение 1, если кнопка нажата, и 0, если не нажата.

Оставшиеся переменные будут рассмотрены позднее.

Отступление 7. Дело случая

Рассмотрим еще один блок - «Случайное число».



В этом блоке заданной переменной присваивается случайное целое число из диапазона, который мы определим в его свойствах. Применение случайных чисел разберем в задачах для закрепления материала.

Задания для закрепления материала

Встреча 10. Кегельринг

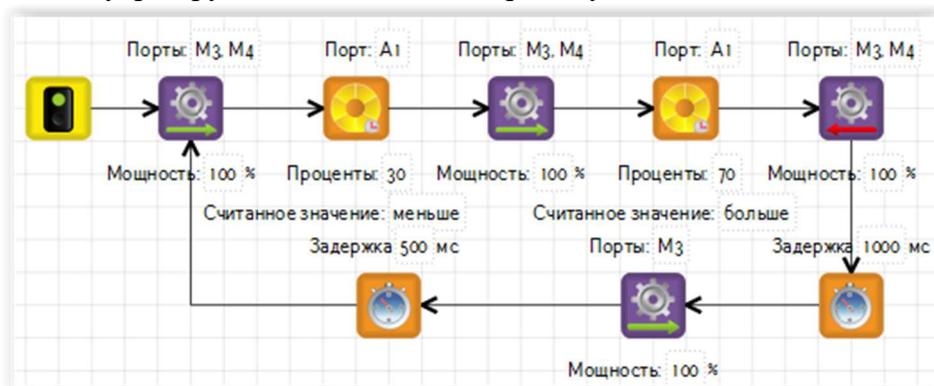
Кегельринг – это одно из состязаний, устраиваемое среди робототехников. Цель состязания – вытолкнуть за пределы очерченного круга диаметром 1 метр все расположенные внутри круга кегли (регламент соревнований можно посмотреть на сайте <http://myrobot.ru/sport/>). Конструкция робота ограничена только линейными размерами и отсутствием специальных приспособлений.

Решим несколько задач для построения нужного алгоритма.

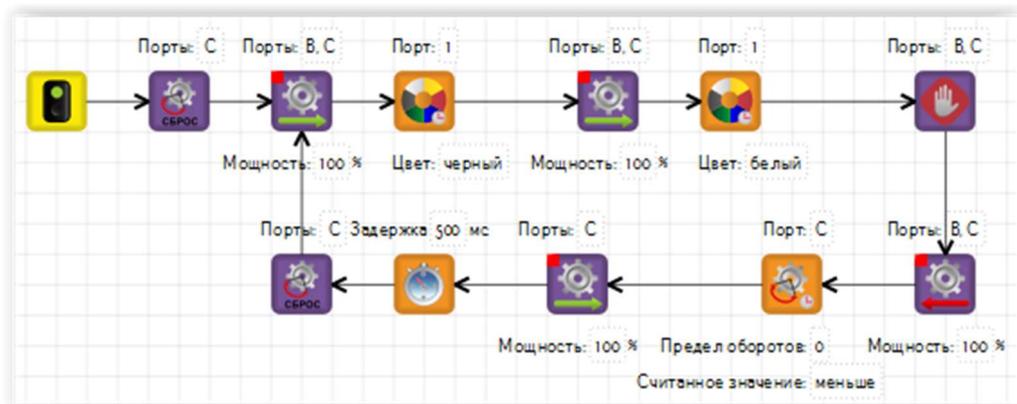
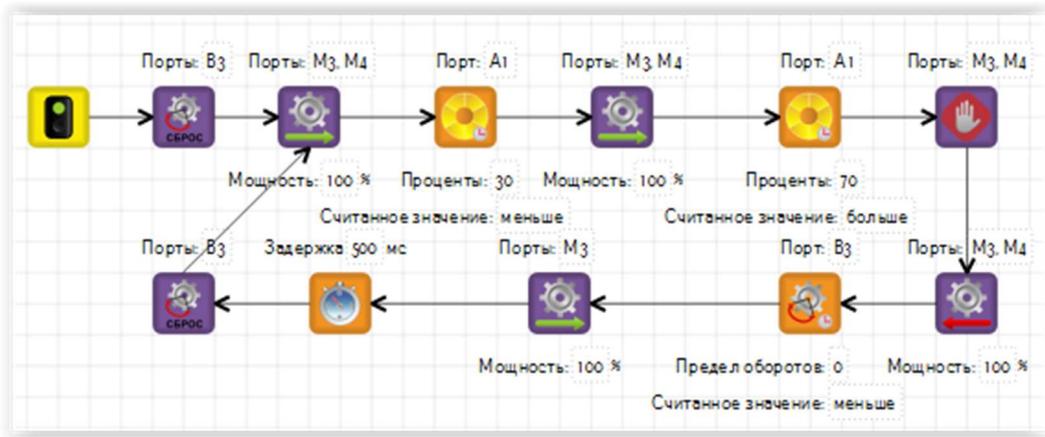
Движение внутри круга:



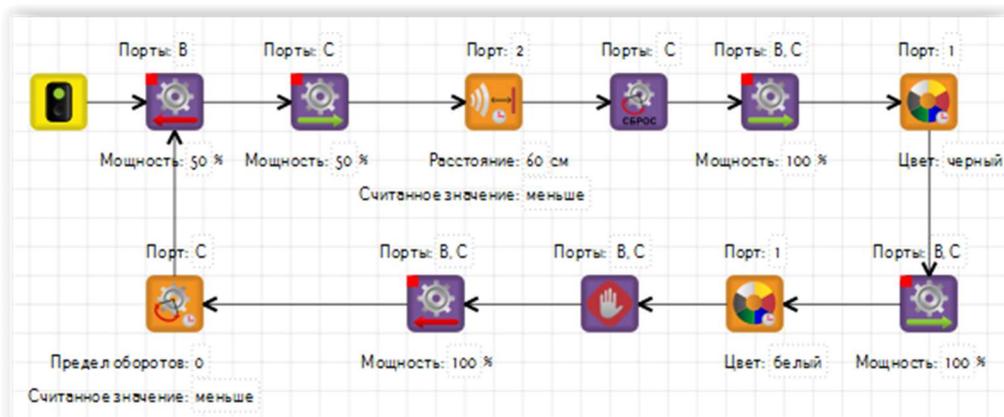
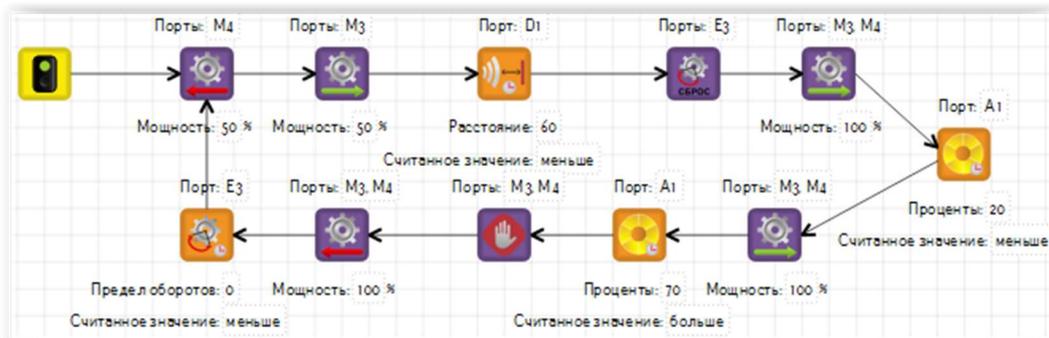
Движение внутри круга с выездом за его границу:



Движение внутри круга с возвратом по энкодерам:



Добавим сонар для более успешного поиска кеглей:

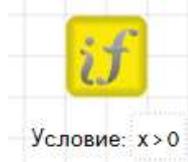


На соревнованиях часто используется алгоритм движения по спирали, перекрывающей всю площадь круга, не использующий датчики.

Задания для закрепления материала

Встреча 11. Что, если?

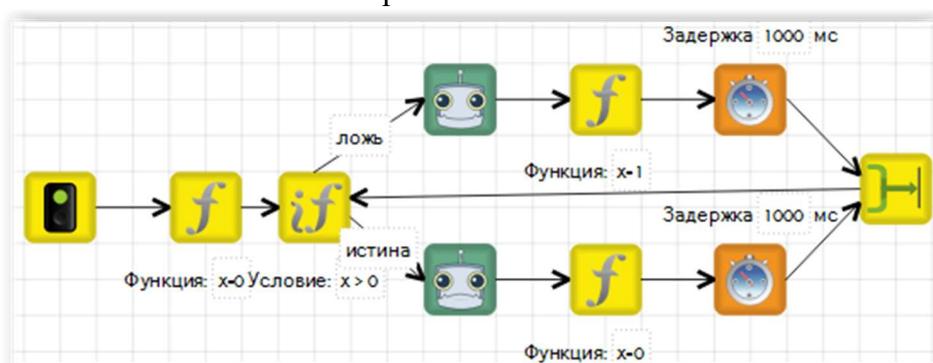
Что, если нам необходимо задать какое-нибудь условие выполнения задачи? Здесь нам придет на помощь оператор условия IF, которому в палитре соответствует блок



Заканчивается оператор IF блоком «Конец условия»:



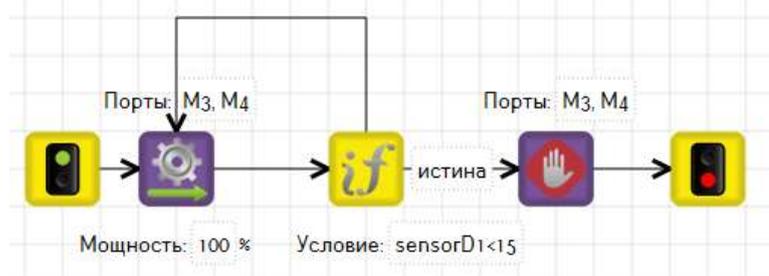
В следующем примере будем выводить на дисплей веселый и грустный смайлики по очереди, в зависимости от значения переменной X:



Задача: Продолжать движение, пока расстояние до стены не станет меньше 15, не используя блок «Ждать датчик расстояния».

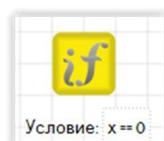


Из оператора IF обязательно должны выходить две связи «истина» и «ложь», причем не обязательно указывать обе. Программа все равно будет выполняться

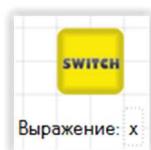


В данном случае ветви условного оператора вместе не сходятся, поэтому блок «Конец условия» в решении задачи отсутствует.

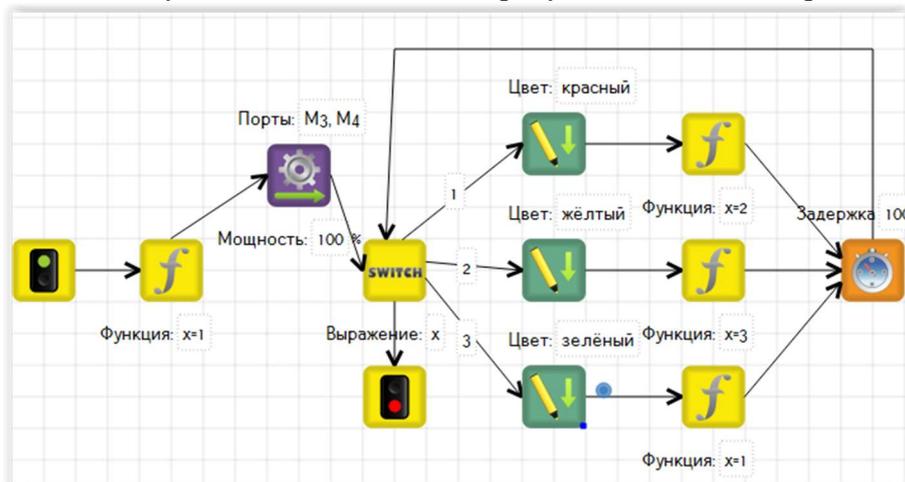
Если нам необходимо в операторе IF использовать равенство, то нужно поставить знак «==»:



В примере реализован выбор из двух вариантов, случилось событие или нет. Для осуществления выбора из нескольких вариантов в TRIK Studio используется блок «Switch»:



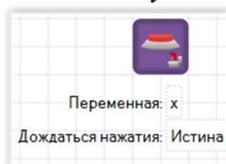
Разберем следующую задачу: необходимо нарисовать трехцветную линию, красно-желто-зеленую, цвета бесконечно чередуются в данном порядке.



Связи, исходящие из оператора «Switch», имеют значения переменной X. Исходящая связь «По умолчанию» не помечена и указывает на конец программы (наличие связи «По умолчанию» обязательно).

Отступление 8. Посчитаем кнопки

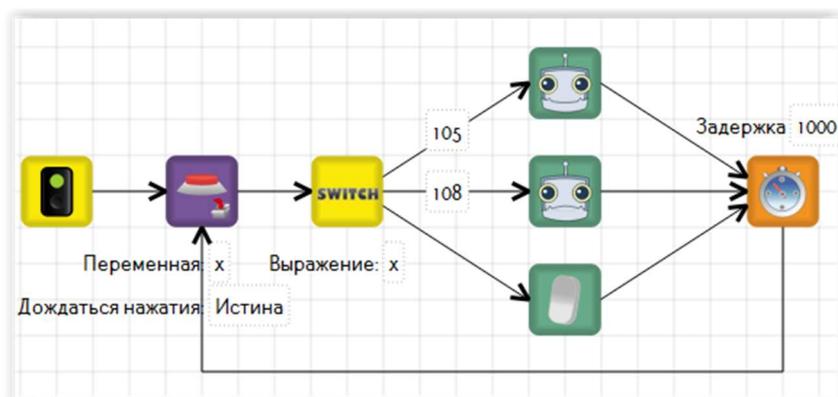
Каждая кнопка на контроллере TRIK имеет свой собственный числовой код, который можно узнать, используя блок «Получить код кнопки»:



Примеры применения:



Вывод на дисплей кода кнопки. Свойство «Вычислять» блока «Напечатать текст» – Истина.



Использование кодов для создания своего меню.

Отступление 9. Где логика?

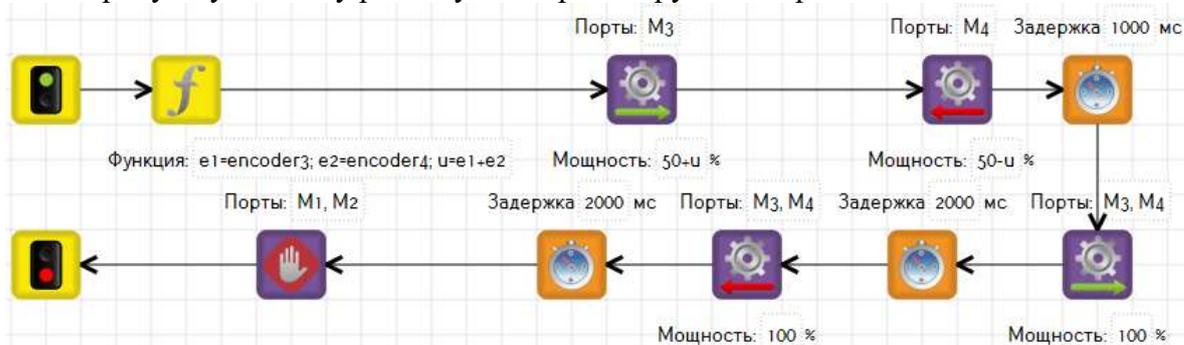
Булевый или логический тип переменной относится к базовым типам языка TRIK Studio. Логические переменные могут принимать значения истина (true) и ложь (false). К логическим операторам относятся: логическое И - &&, логическое ИЛИ - ||, логическое НЕ - !. Результатом выполнения логической операции будет логическое значение.

Задания для закрепления материала

Встреча 12. Исправление ошибок

На практике мы неоднократно сталкивались с тем, что показания одинаковых приборов могут различаться. Это связано с рядом факторов, таких как условия окружающей среды, несовершенство изделий и т.п. В наборе ТРИК датчики, используемые для конструирования, также не являются идеальными. Например, значения, возвращаемые двумя датчиками оборотов, могут не совпадать.

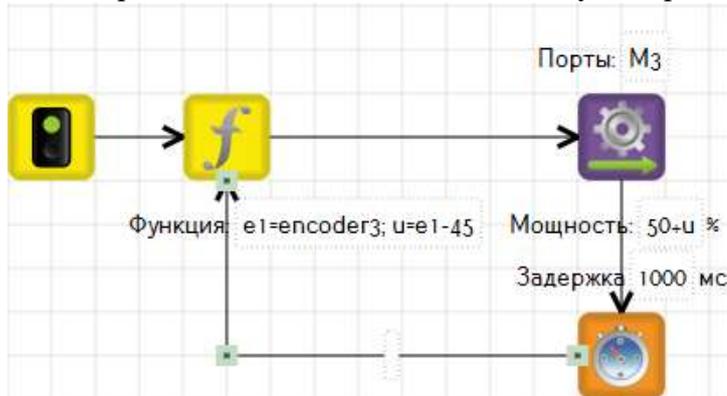
Попробуем учесть эту разницу. Синхронизируем моторы.



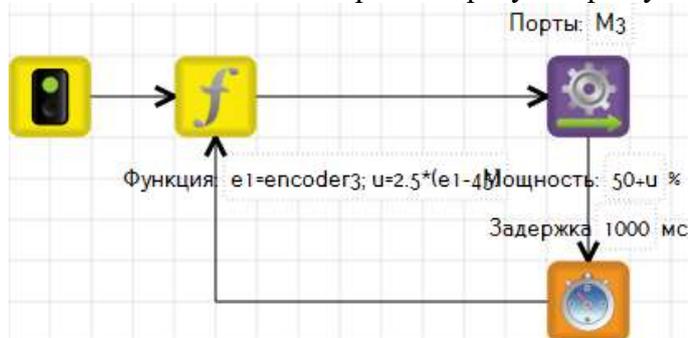
В программе мы использовали метод релейного регулирования для исправления ошибки. В нашем тексте мы не будем вдаваться в подробности теории управления, но воспользуемся некоторыми ее результатами.

Задача: Стабилизировать мотор в положении 45 градусов.

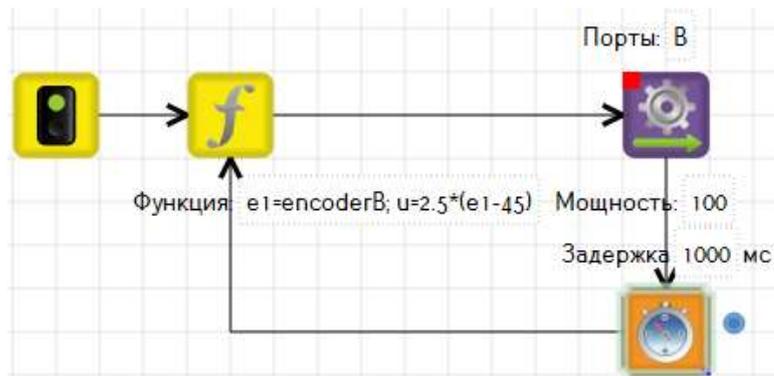
В этой задаче для исправления ошибки также воспользуемся релейным регулятором.



Мотор будет колебаться около положения 45 градусов. Более точное исправление ошибки достигается использованием пропорционального регулятора. Подправим наш алгоритм, используя коэффициент усиления регулятора 2,5. Поправка $u=2.5*(45-\text{encoder1})$. Соответственно стабилизация мотора на П-регуляторе будет выглядеть так:



Для Lego EV3 программа выглядит следующим образом:



Отступление 10. Алгоритм «математического» сглаживания

Невозможно создать измерительный прибор, который бы не зависел от внешних и внутренних условий. Наши датчики тоже не исключение. Как же тогда получить достоверную или приближенную к достоверной информацию об измерениях?

Алгоритм обработки данных, убирающий лишнюю информацию, называется фильтр.

Фильтров существует великое множество в зависимости от принципов работы и области применения. Например, самый простой из них, высчитывающий среднее арифметическое по нескольким последним измерениям:

$$X_t = (X_t + X_{t-1} + X_{t-2}) / 3$$

Часто используется экспоненциальный фильтр:

$$X_t = a * x_t + (1-a) * X_{t-1}$$

Самым, пожалуй, распространенным, является фильтр Калмана, речь о нем будет позднее.

Задания для закрепления материала

Встреча13. Движение вдоль стены. Объезд предметов

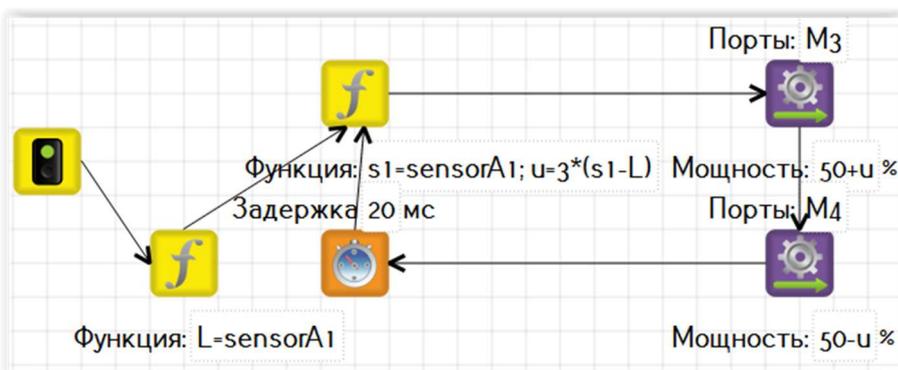
Умения видеть и объезжать препятствия, двигаться вдоль препятствия являются основными для робота, находящегося в движении.

Решим такую задачу: робот должен двигаться вдоль стены на определенном расстоянии L .

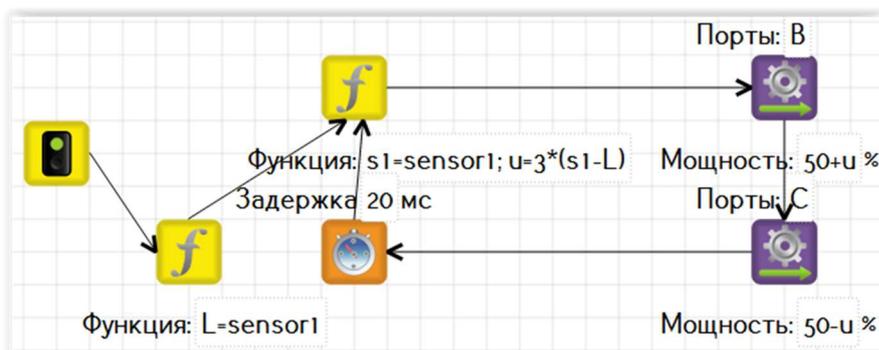
Из датчиков присутствует сонар, подключенный к порту A1.

Используем для исправления ошибки пропорциональный регулятор. Поправка $U=3*(SensorA1-L)$.

ТРИК:



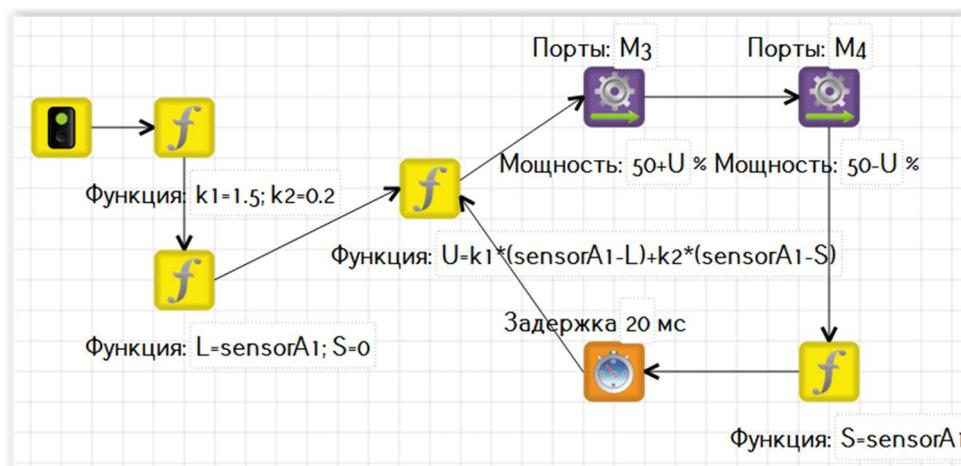
Lego EV3:



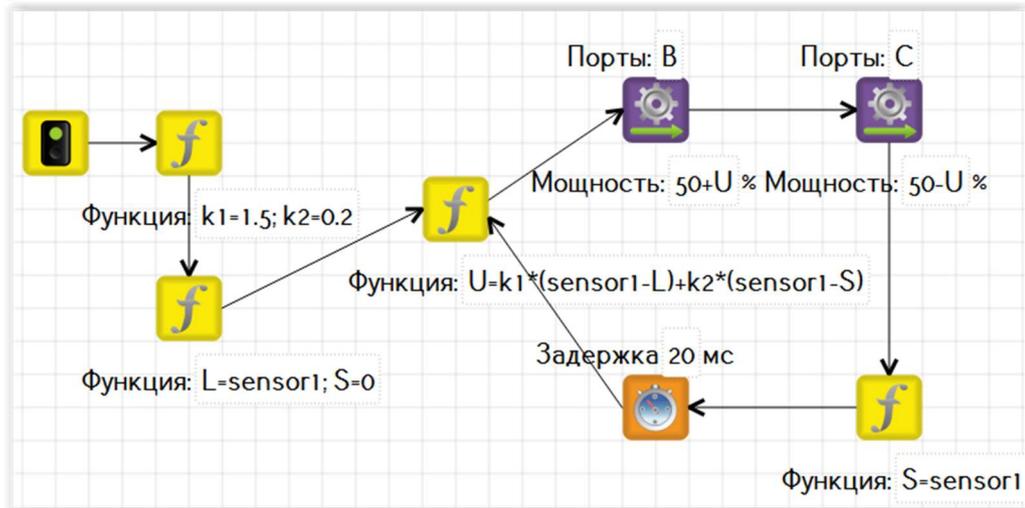
В некоторых ситуациях П-регулятор может вывести систему из устойчивого состояния. Например, если робот направлен от стены, но находится по отношению к ней ближе заданного расстояния, на моторы поступит команда еще больше повернуть от стены. Чтобы не попасть в такое положение, исправим ошибку при помощи пропорционально-дифференциального регулятора.

Поправка $U=k1*(sensorA1-L) + k2*(sensorA1-S)$, где S – расстояние до стены на предыдущем шаге измерения.

ТРИК:



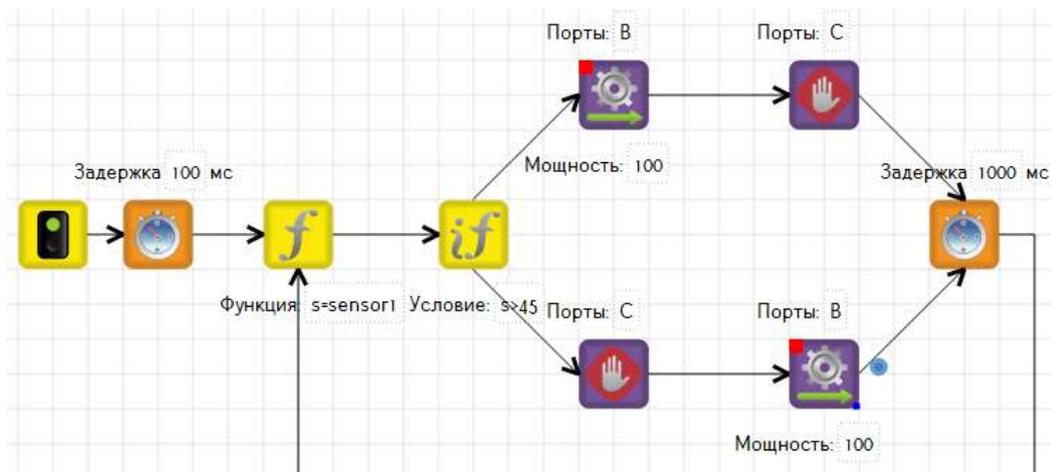
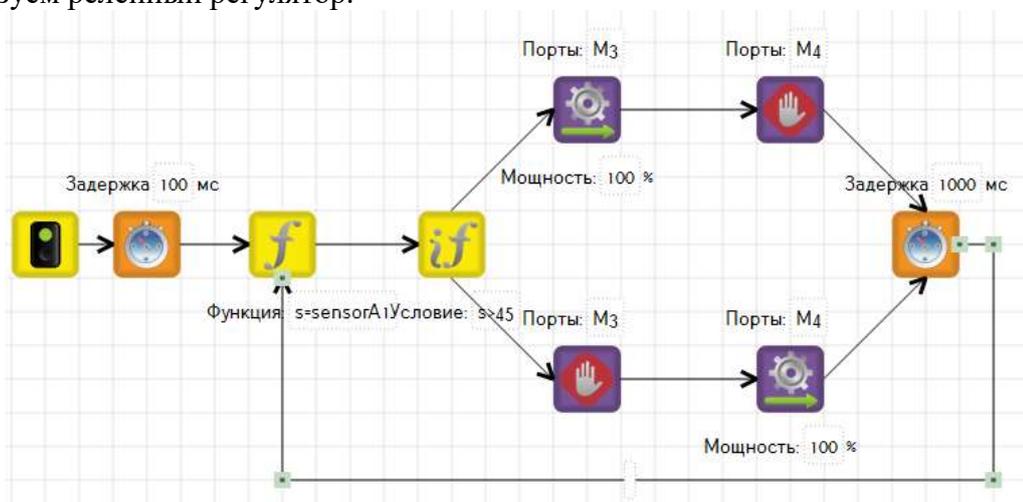
Lego NXT:



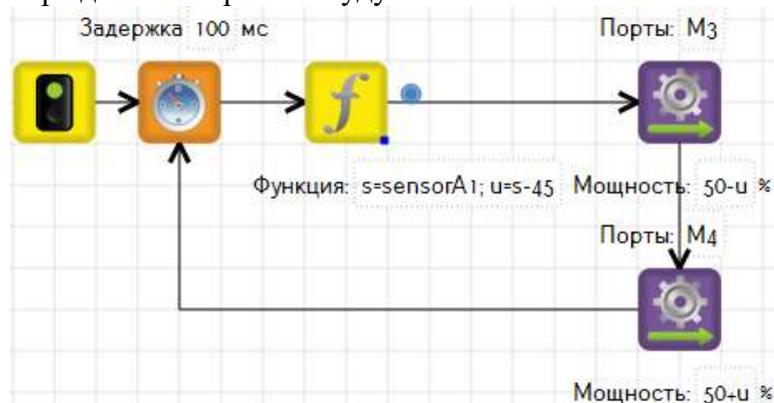
Задания для закрепления материала

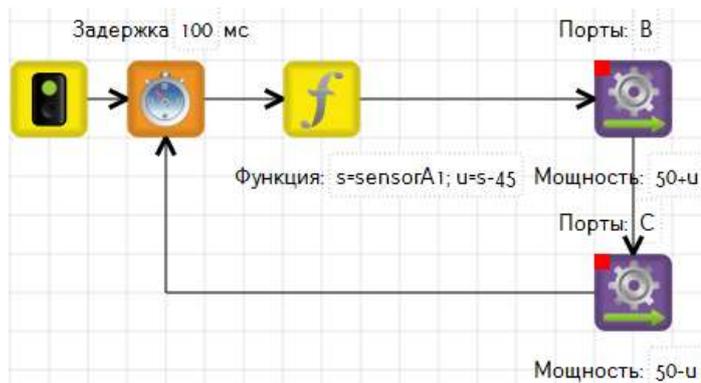
Встреча 14. Движение по линии

Достаточно большое количество задач в робототехнике связано с движением вдоль линий. Рассмотрим пример движения по черной линии с одним датчиком освещенности. Используем релейный регулятор.



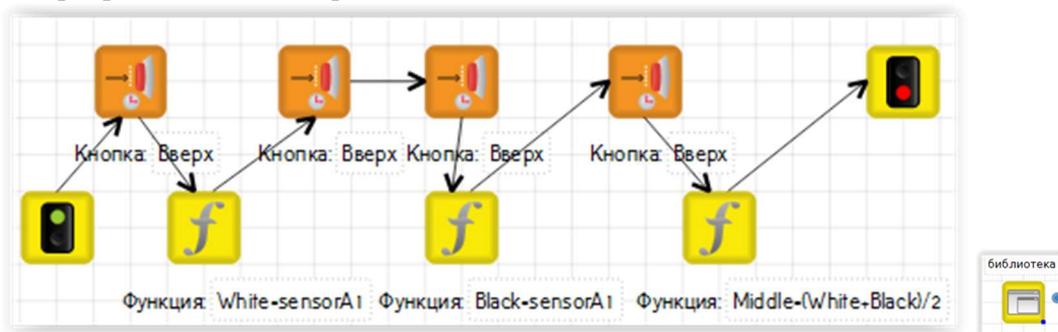
Робот движется по ломаной кривой, периодически наезжая на линию. На П-регуляторе движения робота будут более осмысленными.





Число 45 – показание датчика на границе между белым и черным цветом, для более точного определения границы необходима калибровка датчиков.

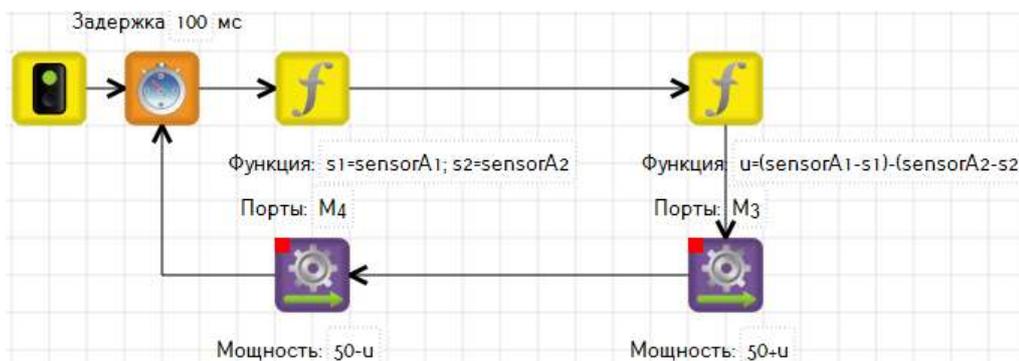
Подпрограмма для калибровки датчика света:



Ставим робота на черное, нажимаем кнопку «Вверх», еще раз нажимаем кнопку «Вверх», переставляем робота на белое, нажимаем кнопку «Вверх». Среднее значение цвета будет храниться в переменной Middle.

Модернизируем конструкцию робота, добавив еще один датчик. На такую модель практически не влияют колебания освещенности в помещении.

Сравнивая начальные показания датчиков с текущими значениями на П-регуляторе, имеем

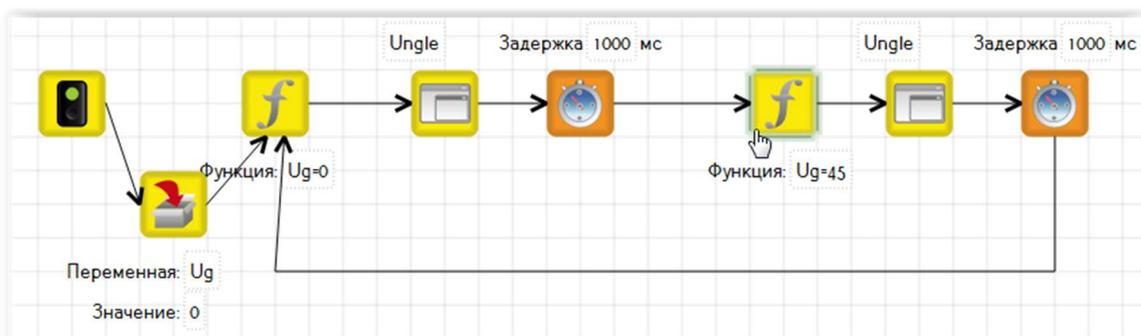


Задания для закрепления материала

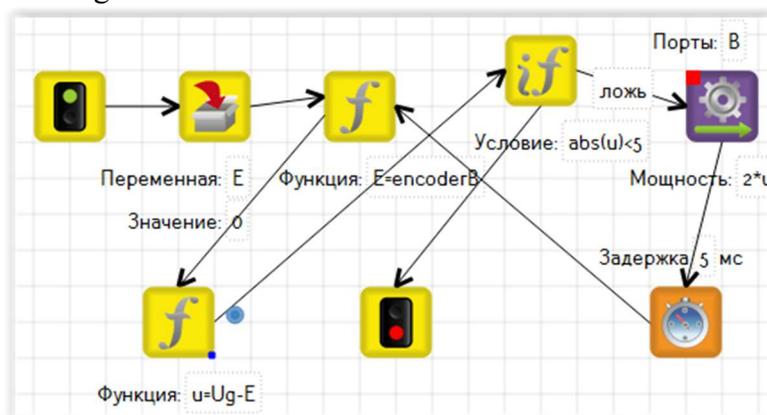
Встреча 15. Манипуляторы

На сегодняшний день ни одно из промышленных предприятий не обходится без манипуляторов. Для расширения возможностей на роботах также устанавливаются манипуляторы. Интересная модель манипулятора с тремя степенями передвижений приведена в книге С.А.Филиппова «Робототехника для детей и родителей». Напишем программу для этого манипулятора.

Первый мотор отвечает за горизонтальное перемещение захвата и фиксируется в двух положениях: 0 градусов и 45 градусов. Используем П-регулятор для стабилизации положений.



Подпрограмма Ungle:

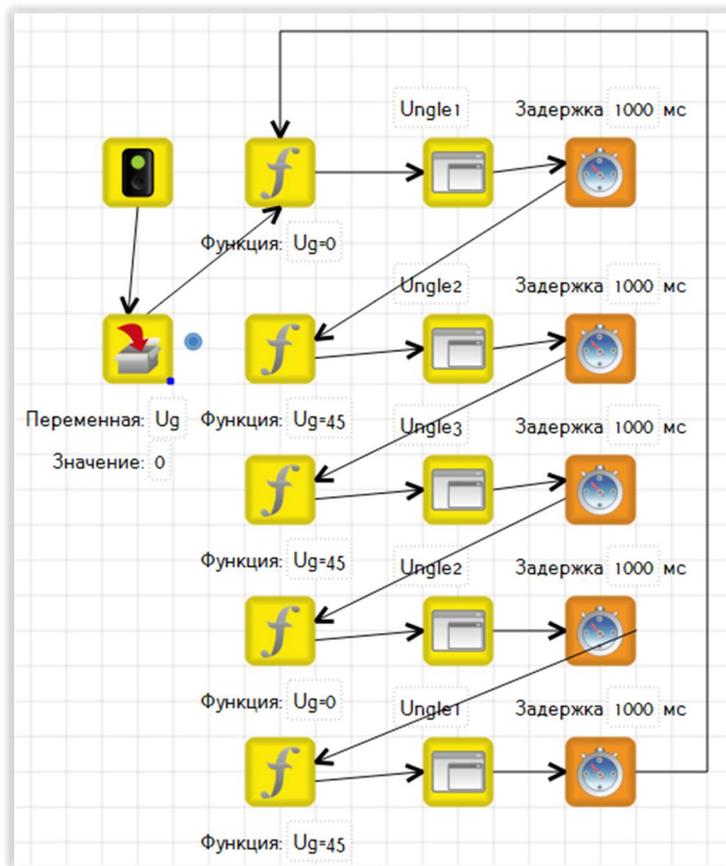


Как мы помним, подпрограмма не может быть цикличной, поэтому условие $(Ug - EncoderB) < 5$ и будет означать выход из цикла. На малых мощностях моторов их работа не стабильна, поэтому используем коэффициент 2.

Второй мотор отвечает за вертикальные перемещения и также стабилизируется в двух положениях. Т.е. задача аналогична первой.

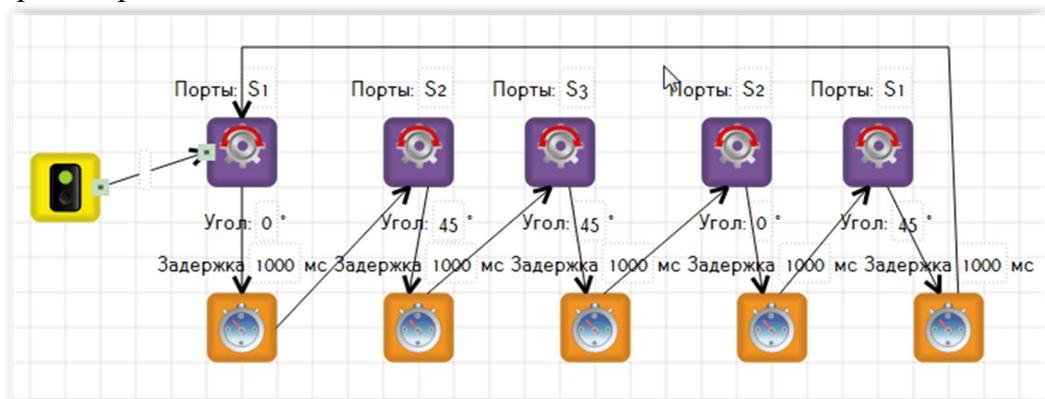
Третий мотор отвечает собственно за захват.

В результате имеем:



Подпрограммы работают с разными энкодерами.

В конструкторе ТРИК используются сервомоторы, поэтому программа для него будет гораздо проще:

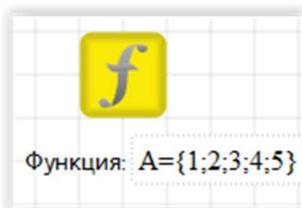


Задания для закрепления материала.

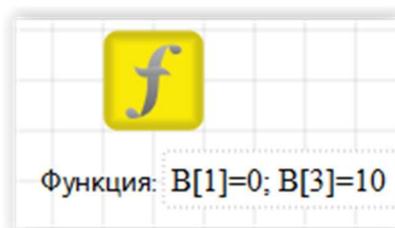
Встреча 16.Таблицы и массивы

Использование сложных структур данных в робототехнике связано с разнообразием решаемых задач. Организация данных в таблицы в TRIK Studio осуществляется в виде динамических массивов, размер которых может при необходимости меняться во время исполнения программы. Все элементы массива должны быть однотипны, и каждому из них присваивается номер элемента – индекс. Размерность массива не указывается.

Массивы в TRIK Studio можно создавать в явном виде таблицей

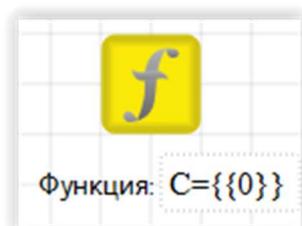


Или поэлементно

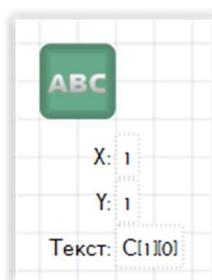


Неописанные элементы примут значение 0, пустая строка или false в зависимости от типа элементов.

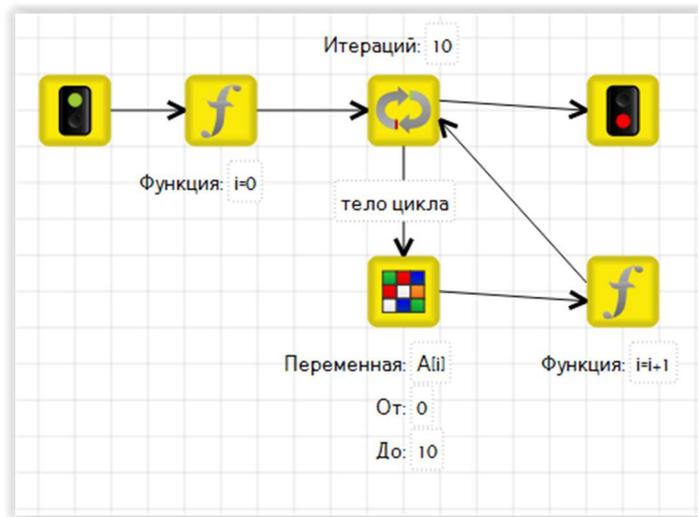
Массивы сами могут быть элементами массива, поэтому запись



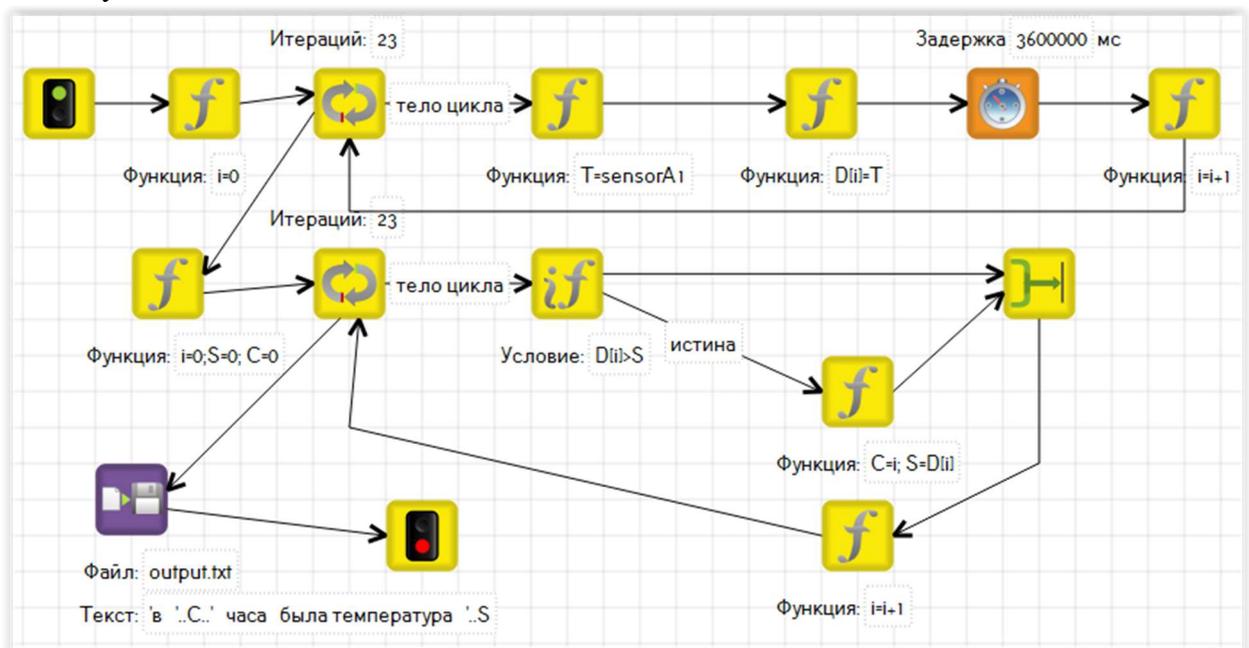
означает, что объявлен двумерный массив. Обращение к его элементам выглядит следующим образом



Заполнение массива случайными числами:



Программа для определения, в какое время был достигнут температурный максимум:



Отступление 11. Золотая середина

Как уже было сказано ранее, математических фильтров для обработки сигналов существует великое множество. Познакомимся еще с одним из них – «медианный» фильтр. Основная особенность этого фильтра – способность устранения аномальных значений в массивах данных. Медианный фильтр пропускает только среднее значение из множества значений. Для того чтобы понять, как он работает и чем отличается от среднего арифметического, разберем пример: датчик выдает следующие значения $\{4,4,5,34,8,4,2\}$. Среднее значение без учета крайних значений 4 и 2 получается – 11. Будем выбирать по три показания датчика, и упорядочивать их по возрастанию. $\{4,4,5\}$ – $\{4,4,5\}$. Значение «медианного» фильтра – 4. $\{4,5,34\}$ – $\{4,5,34\}$. Значение «медианного» фильтра – 5. $\{5,34,8\}$ – $\{5,8,34\}$. Значение «медианного» фильтра – 8. $\{34,8,4\}$ – $\{4,8,34\}$. Значение «медианного» фильтра – 8. $\{8,4,2\}$ – $\{2,4,8\}$. Значение «медианного» фильтра – 4. Отфильтрованные значения – $\{4,5,8,8,4\}$. Крайние значения 4 и 2 фильтруются,

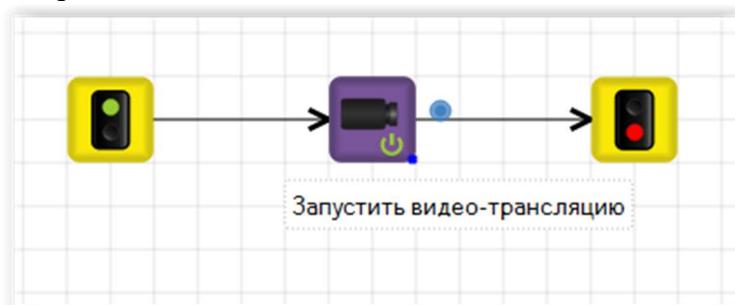
используя предыдущие и последующие значения. Среднее значение отфильтрованных данных получилось – 5,8. То есть мы всегда выбирали значение среднего элемента в подмассиве данных. Обычно используется выборка из нечетного количества элементов не менее трех.

Задания для закрепления материала

Встреча 17. Камера контроллера ТРИК

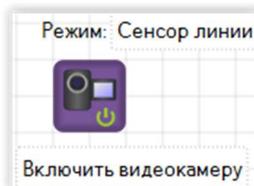
Видеомодуль или usb-камера, которые можно подключать к контроллеру ТРИК, выполняют несколько различных функций. Их можно использовать как датчик линии, датчик цвета, объектный датчик и непосредственно как камеру.

Запустим видеотрансляцию:



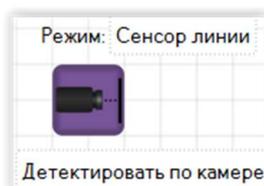
Данная программа открывает трансляцию видеосигнала. Видео можно просматривать в браузере по адресу http://<IP_адрес_контроллера>:8080/?action=stream или на пульте дистанционного управления (будет рассмотрено позднее).

Камера – датчик линии. Для использования камеры как датчика необходимо ее проинициализировать, используя блок «Включить видеокамеру»:

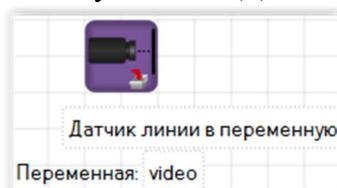


В свойствах выбирается режим камеры «Сенсор линии».

Изображение фиксируется в центре кадра и инициализируется датчик линии блоком «Детектировать по камере»:



Чтобы получить данные, задействуем блок «Датчик линии в переменную»:



Переменная video показывает отклонение объекта, в нашем случае черной линии, от центра экрана по оси X в диапазоне от (-100) до 100.

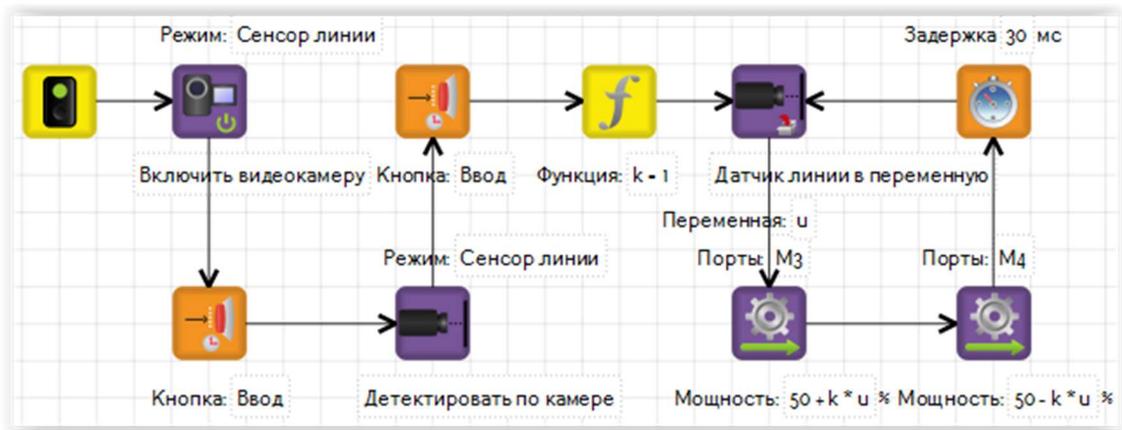
Для датчика линии имеется своя зарезервированная переменная lineSensor, которая является массивом, состоящим из трех элементов.

lineSensor [0] – это и есть переменная video.

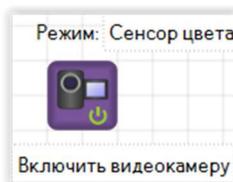
lineSensor [1] – вероятность перекрестка в процентах от 0 до 100.

lineSensor [2] – относительный размер линии от 0 до 100.

Напишем программу движения по черной линии, используя П-регулятор.



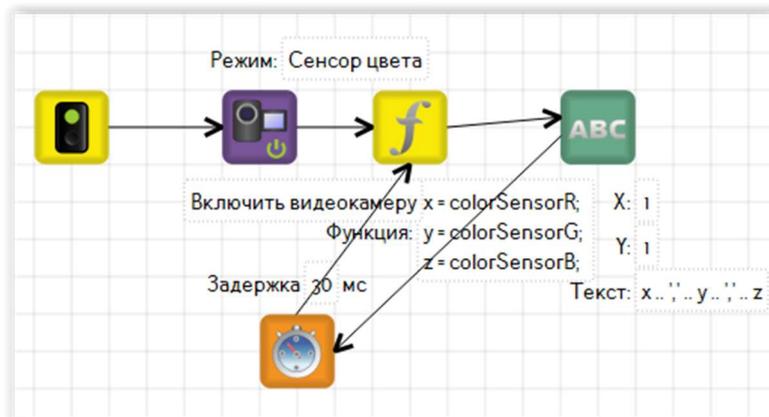
Камера – датчик цвета.



Цветовая модель – это представление цвета при помощи смешения нескольких красок, выбранных за основу. Существуют разнообразные виды цветowych моделей, отличающиеся набором входящих в них базовых цветов и изменяемых значений, придающих необходимый оттенок.

Для TRIK Studio была выбрана модель RGB. В режиме сенсора цвета камера разбивает изображение на 9 квадратов. В каждом квадрате берется среднее значение цвета – три числа от 0 до 255 в формате RGB.

Зарезервированные переменные colorSensorR, colorSensorG, colorSensorB соответствуют значениям цвета в среднем квадрате.



Формулы пересчета для различных цветowych пространств.

Система HSV.

Считаем, что:

$H \in [0,360]$;

$S, V, R, G, B \in [0,1]$;

Пусть MAX — максимальное значение из R, G и B, а MIN — минимальное из них.

$$H = \begin{cases} 0, & \text{если } MAX = MIN; \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{если } MAX = R \text{ и } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{если } MAX = R \text{ и } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{если } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{если } MAX = B \end{cases}$$

$$S = \begin{cases} 0, & \text{если } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{иначе} \end{cases}$$

$$V = MAX.$$

Система CMY(K).

Считаем, что цвета RGB находятся в интервале [0;1]

$$K = 1 - MAX(R, G, B);$$

$$C = \frac{1-R-K}{1-K};$$

$$M = \frac{1-G-K}{1-K};$$

$$Y = \frac{1-B-K}{1-K}.$$

Система YIQ:

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B;$$

$$I = 0.596 \times R - 0.274 \times G - 0.321 \times B;$$

$$Q = 0.211 \times R - 0.526 \times G + 0.311 \times B.$$

Применяя формулы преобразования цветного изображения в черно-белое, камеру можно использовать как датчик освещенности.

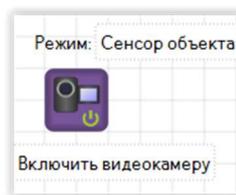
В цветовых пространствах YUV и YIQ, используемых в PAL и NTSC, яркость Y' вычисляется следующим образом:

$$Y' = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

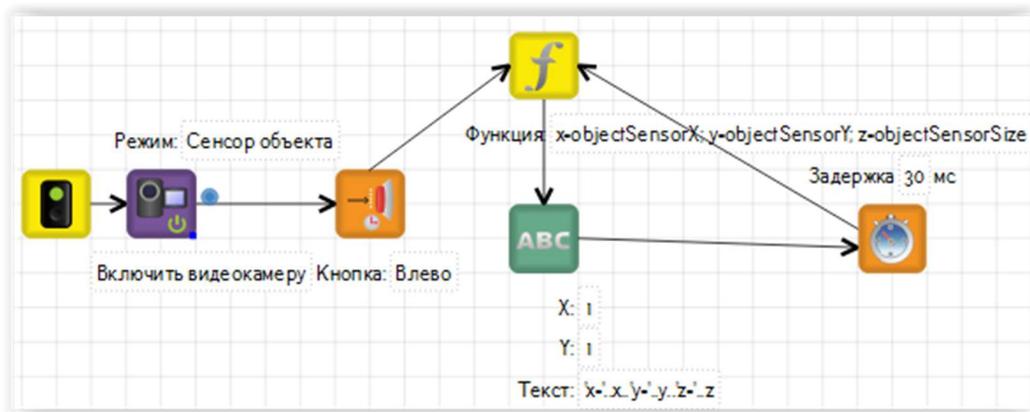
Для учёта особенностей восприятия изображения человеческим глазом (чувствительность к зелёному и синему цвету) в модели HDTV используют другие коэффициенты:

$$Y' = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$$

Камера – датчик объекта.



Зарезервированных переменных для датчика объекта три: objectSensorX и objectSensorY – координаты распознанного объекта в диапазоне от (-100) до 100, objectSensorSize – размер объекта от 0 до 100.

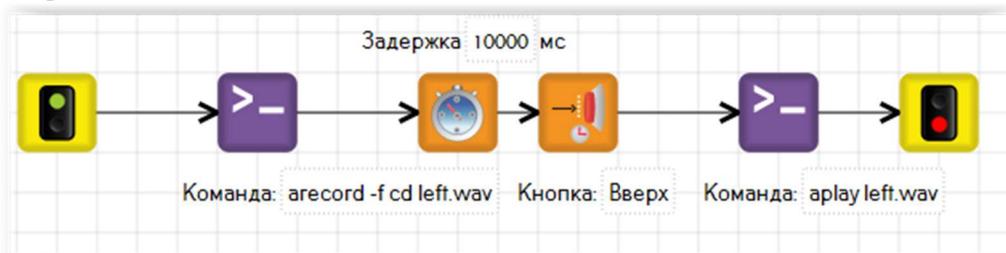


Отступление 12. Небольшой, но компьютер.

На контроллере ТРИК установлена операционная система Linux. TRIK Studio позволяет воспользоваться любыми системными командами при помощи блока



Примеры использования



Эта программа позволяет сначала записать звук с микрофона в файл left.wav, а затем его воспроизвести.

Настройка сети Wi-Fi также производилась при помощи этого блока.

Отступление 13. Мультимедиа

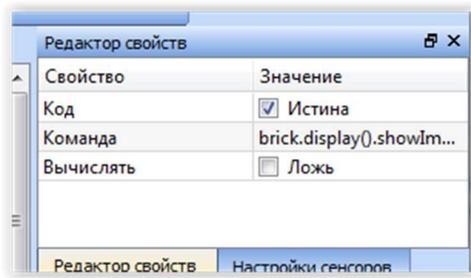
Звуковые файлы и файлы изображений на контроллере ТРИК располагаются в директории /home/root/trik/media/<Имя_Файла>.

Для воспроизведения звукового файла можно воспользоваться блоком «Играть звук»:



Указав путь к файлу на контроллере. Второй способ воспроизведения – использование «Системного вызова» для вызова проигрывателя aplay (Отступление 11).

Файлы изображения открываются на дисплее после вызова системной команды brick.display().showImage('media/trik_smile_sad.png'), с указанием пути к файлу изображения. Не забудьте указать, что это исполняемый код:



Для того чтобы сделать фотографии видеокамерой, используется системная команда `v4l2grab -W 320 -H 240 -d /dev/video2 -o doc.jpg`. Снимки сохраняются в файл `doc.jpg`. выполнение этой команды требует предварительной настройки контроллера:

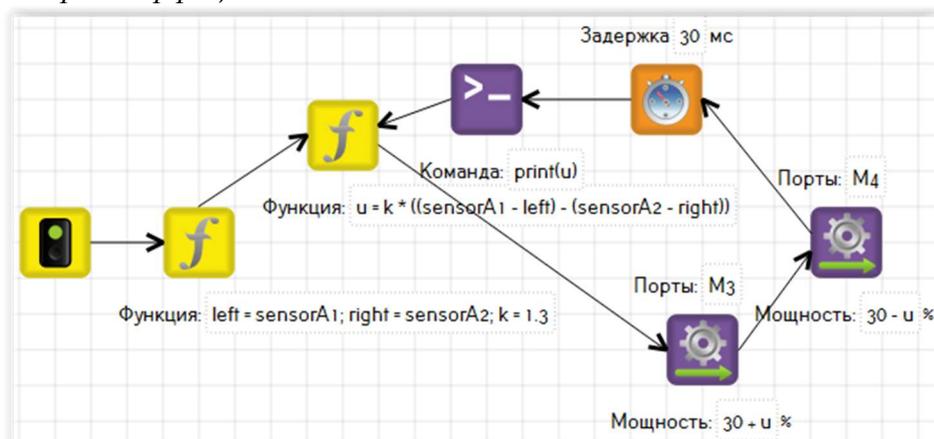
```
opkg update
opkg upgrade
opkg --force-reinstall install v4l2grab.
```

Отступление 14. Консоль робота

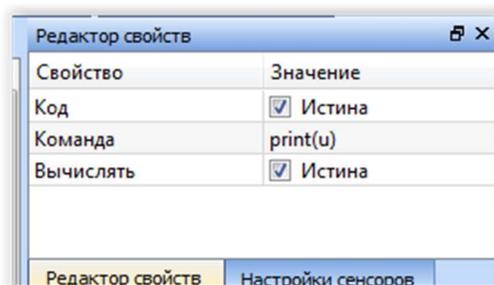
Умение выводить в консоль робота значения датчиков, флаги, значение переменных очень помогает в отладке программ.

Ниже показан пример из папки `Example` – движение по линии, в текст которого добавлен оператор `print` для вывода в консоль робота значения управляющего воздействия.

Для подбора коэффициента отладочный вывод – очень полезное знание.



Свойства блока «Системный вызов»:



Задания для закрепления материала

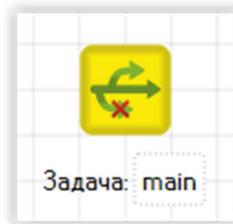
Встреча 18. Параллельным курсом

Многие практические задачи требуется решать в реальном времени, для этого может потребоваться большой объем вычислений. Современные компьютеры могут выполнять несколько операций одновременно. Такой процесс называется параллельным вычислением. Основная цель параллельных вычислений – уменьшение времени решения задачи. Контроллеры – не исключение, для организации параллельных процессов в TRIK Studio используются блоки «Параллельные задачи» и «Слияние задач»:



Параллельным задачам необходимо дать имена, чтобы можно было на них ссылаться. Имя главной задачи – «main».

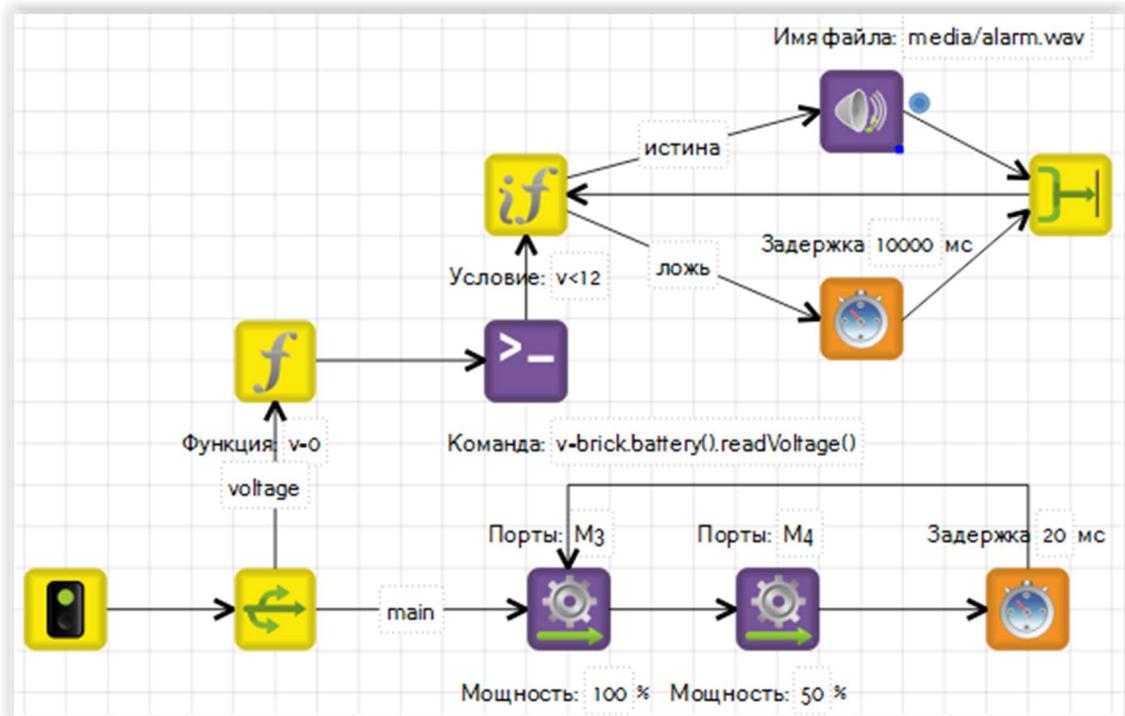
Одну из выполняемых задач можно завершить блоком:



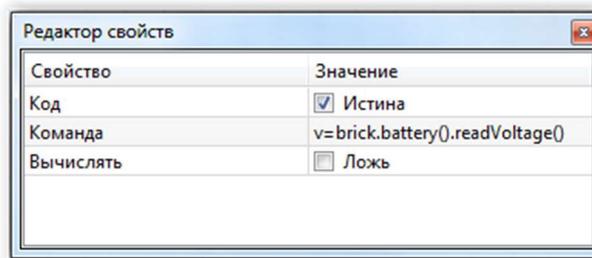
Использовать параллельные задачи в цикле нельзя, процессы будут множиться до переполнения памяти контроллера.

Параллельные процессы должны быть независимы друг от друга, то есть решаемые задачи имеют различные цели, например, воспроизведение звука при движении по траектории, или контроль заряда батареи во время выполнения задачи.

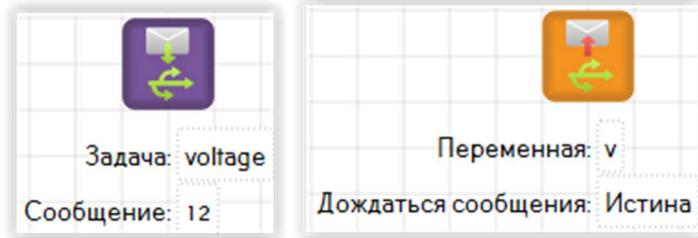
В качестве примера рассмотрим движение по кругу с контролем заряда батареи:



Контроль заряда батареи осуществляется каждые 10 секунд, и, если напряжение падает ниже 12 Вольт, раздается звуковой сигнал. Свойства системного вызова:



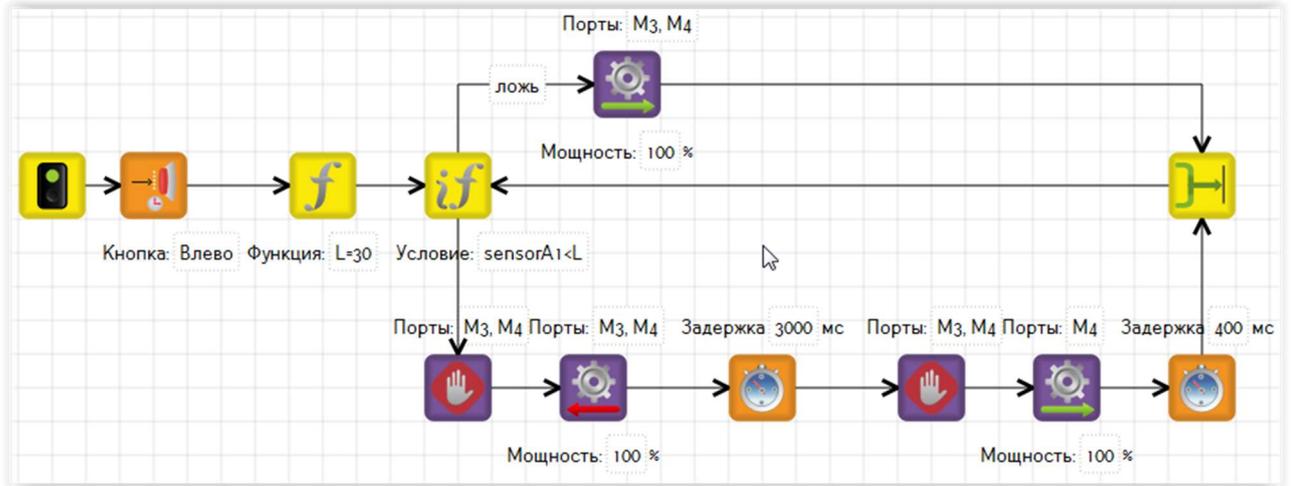
Параллельные задачи могут обмениваться сообщениями, что очень полезно для синхронного выполнения действий. Для приема и отправки сообщений используются блоки «Отправить сообщение в задачу» и «Получить сообщение из другой задачи»:



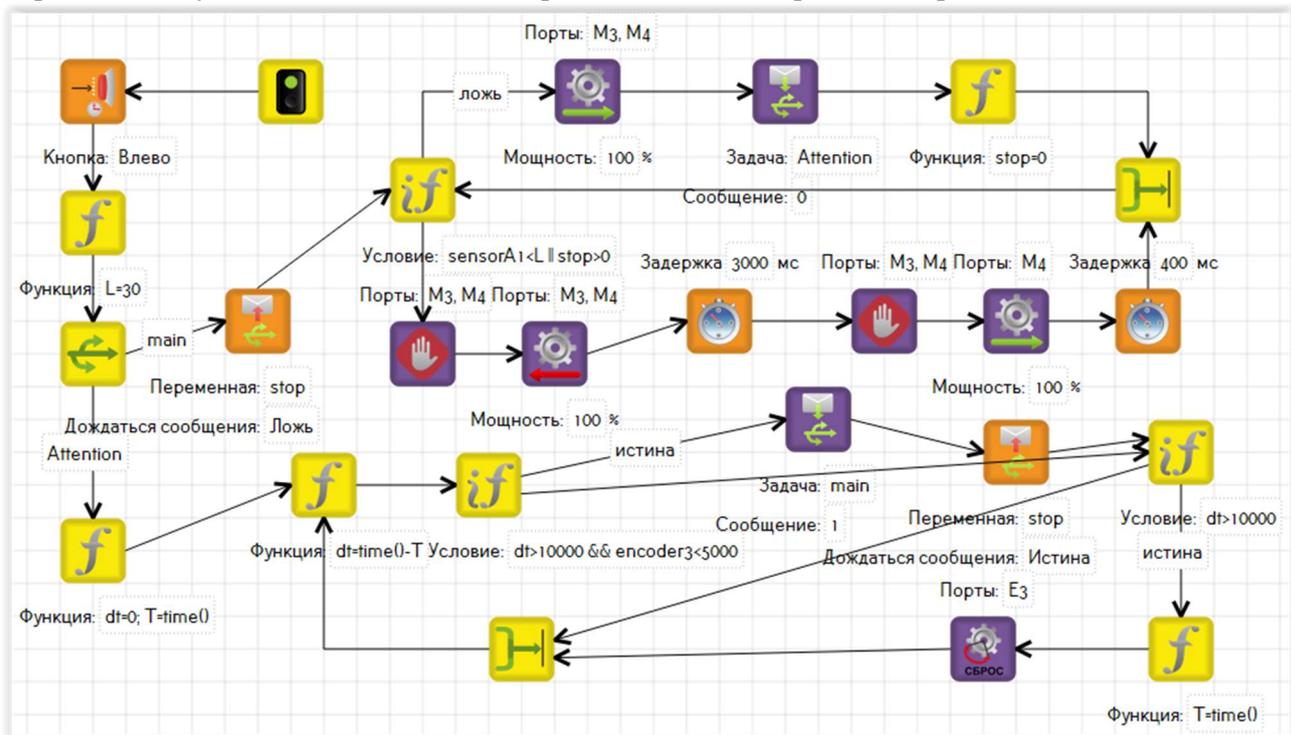
Задания для закрепления материала

Встреча 19. Исследователь

Сконструируем робота для передвижения по комнате так, чтобы он не наткнулся на предметы и не застревал. Для начала оснастим его датчиком расстояния. Алгоритм путешествия следующий: вперед до препятствия – отъезд – разворот – вперед до препятствия.



Во время движения робот может попасть в ситуацию, когда датчик не заметит препятствие и уткнется в него, при этом моторы не смогут проворачивать колеса в полной мере. Используем для защиты от блокировки колес таймер и энкодер.



Чтобы избежать небольших препятствий, добавим датчик касания, расположив его на бампере робота. В программе добавится проверка датчика касания.



В случае, когда робот уперся в препятствие, а моторы прокручивают колеса, можно добавить еще одну параллельную проверку по гироскопу или акселерометру, но о них будет рассказано позже.

Задания для закрепления материала

Встреча 20. Мастерская художника

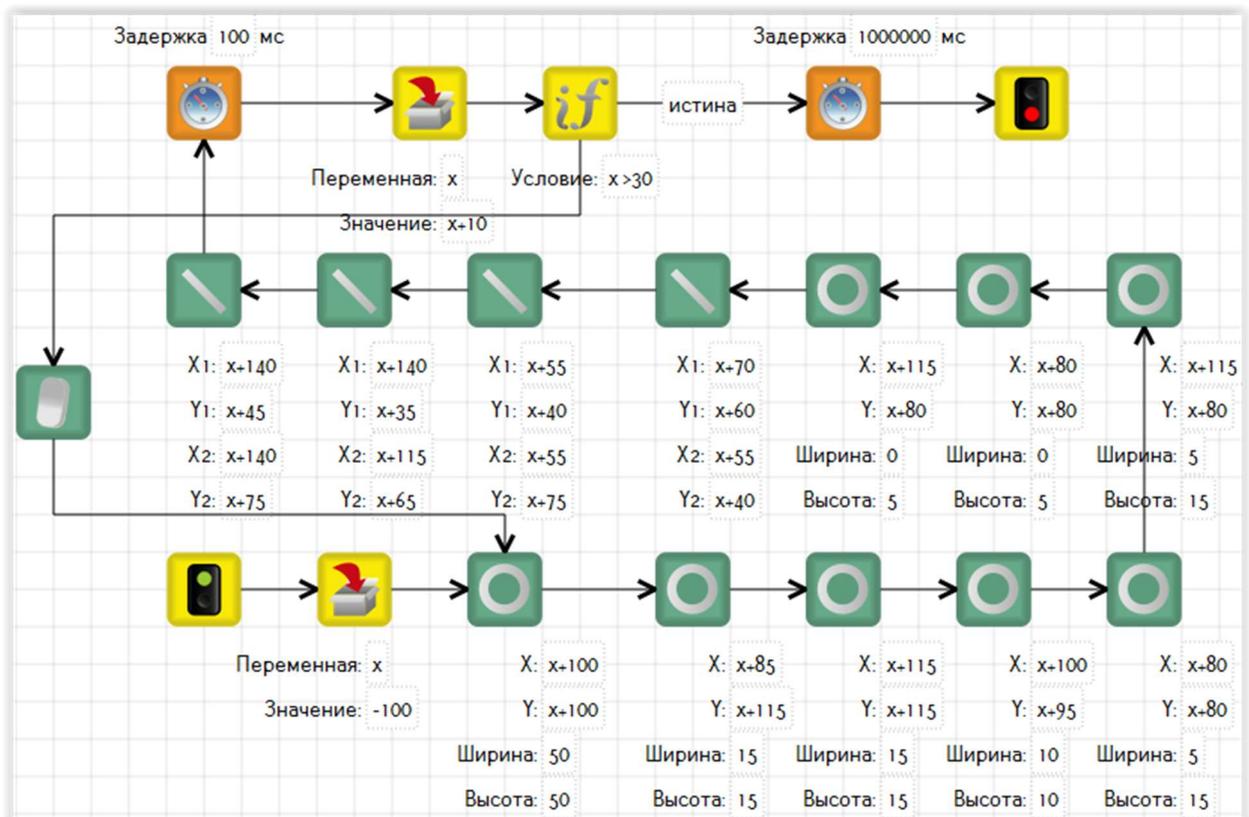
Для вывода на дисплей текста, рисунков и значков в TRIK Studio используется раздел «Рисование» в «Палитре». Приведем целиком таблицу назначений блоков из «Помощи» программы.

Название элемента	Пиктограмма	Описание
Цвет кисти	<p>Цвет: черный</p> 	Указать цвет, которым будут рисоваться простые графические фигуры на экране робота.
Ширина кисти	<p>Ширина: 5</p> 	Указать ширину линии, которой будут рисоваться простые графические фигуры на экране робота.
Нарисовать точку	 <p>X: 0 Y: 0</p>	Нарисовать на экране точку в указанных координатах.
Нарисовать линию	 <p>X1: 0 Y1: 0 X2: 10 Y2: 10</p>	Нарисовать на экране отрезок. В качестве параметров блоку указываются концы отрезка.
Нарисовать прямоугольник	 <p>X: 0 Y: 0 Ширина: 5 Высота: 5</p>	Нарисовать на экране прямоугольник. В качестве параметров указываются координаты левого верхнего угла, ширина и высота прямоугольника.

<p>Нарисовать эллипс</p>	 <p>X: 0 Y: 0 Ширина: 5 Высота: 5</p>	<p>Нарисовать на экране эллипс, вписанный в заданный прямоугольник.</p>
<p>Нарисовать дугу</p>	 <p>X: 0 Y: 0 Ширина: 5 Высота: 5 Начальный угол: 20 Конечный угол: 20</p>	<p>Нарисовать на экране дугу, задающуюся координатами прямоугольника, в который она будет вписана, и углами (в градусах) её начала и конца на окружности. Если начало и конец совпадают, будет нарисована полная окружность.</p>
<p>Смайлик</p>		<p>Нарисовать на экране смайлик.</p>
<p>Грустный смайлик</p>		<p>Нарисовать на экране грустный смайлик.</p>
<p>Цвет фона</p>	<p>Цвет: белый</p> 	<p>Задаёт цвет фона экрана.</p>
<p>Напечатать текст</p>	 <p>X: 1 Y: 1 Текст: 'текст'</p>	<p>Печатает заданную строку в заданном месте на экране робота. Значение свойства "Текст" по умолчанию трактуется как строка в чистом виде, оно так и будет выведено на экран. Чтобы система считала, что это выражение на текстовом языке (это может быть полезно, например, при отладке значения переменных), поставьте галочку "Вычислять" в редакторе свойств.</p>

Очистить экран		Стереть всё, что нарисовано на экране.
----------------	---	--

В качестве примера приведем следующую программу с движением по экрану:



В результате ее выполнения получится двигающаяся картинка



При расчетах не забывайте о существовании полярных координат.

Формулы перевода из полярных координат в прямоугольные и обратно выглядят следующим образом:

$$X = R \times \cos \alpha; Y = R \times \sin \alpha$$

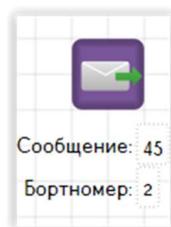
$$R = \sqrt{X^2 + Y^2}; \tan \alpha = Y/X$$

Задания для закрепления материала

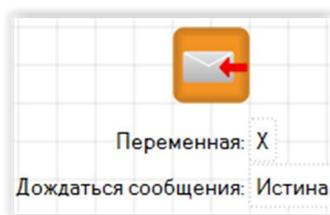
Встреча 21. Мы вместе

Контроллеры ТРИК могут взаимодействовать друг с другом по протоколу WiFi, если они находятся в одной сети. Настройка взаимодействия производится непосредственно на контроллерах в меню «Настройка» – «Взаимодействие». Каждому контроллеру присваивается свой бортовой номер и выбирается один контроллер, который будет «Ведущим». Его IP-адрес прописывается в настройках всех взаимодействующих контроллеров.

Для обмена данными используются блоки «Послать сообщение»



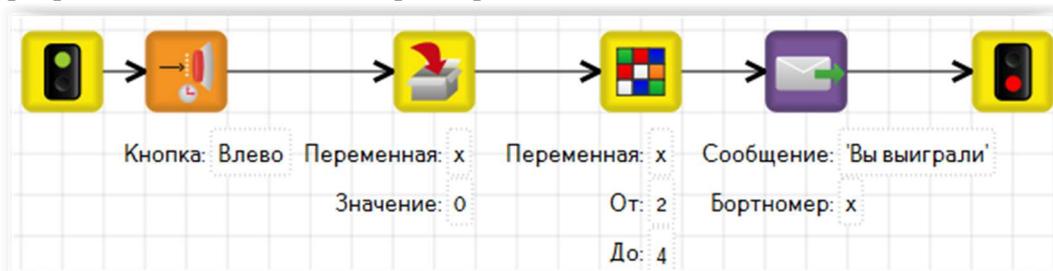
и «Получить сообщение»



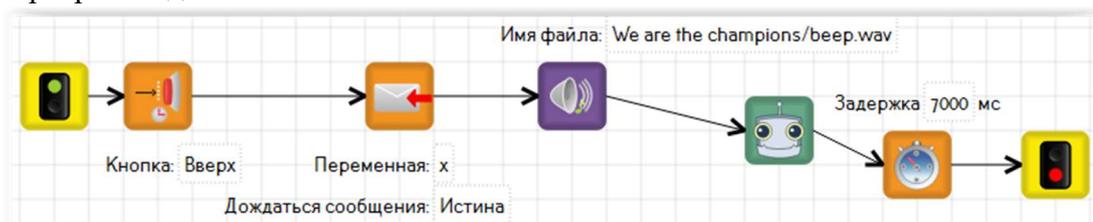
Свойство «Дождаться сообщения» останавливает работу программы в случае, если очередь сообщений пустая, либо позволяет продолжить ее выполнение, записав в переменную пустую строку или 0.

Пример взаимодействия роботов: «Лотерея». Главный контроллер случайным образом выбирает, кому из трех роботов отправить сообщение о выигрыше, счастливиц, получивший сообщение, естественно, бурно радуется.

Программа для главного контроллера

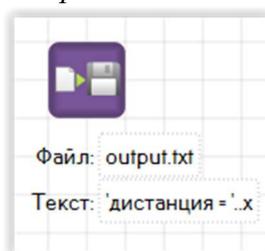


Программа для «Счастливицы»:

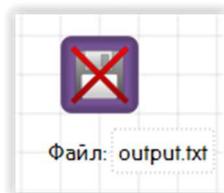


Отступление 15. Сохраним информацию

Для того чтобы сохранить нужную нам информацию, например, показание датчиков, можно записать ее в файл, который будет находиться на контроллере, воспользовавшись блоком «Записать в файл»:



Файл можно получить с робота, например, воспользовавшись программой WinSCP под Windows или scp под Linux. Для удаления файла – блок «Удалить файл»:



Не забывайте, что для работы с файлами существует возможность вызова системных команд. При настройке сети мы добавляли запись в конец файла `wpa_supplicant.conf`.

Задания для закрепления материала

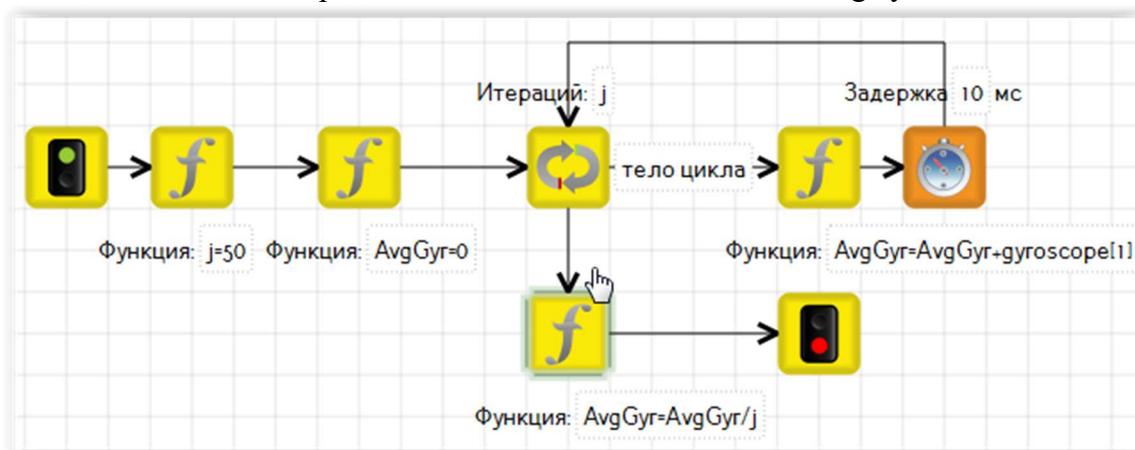
Встреча 22. Гироскоп, акселерометр

Не все датчики контроллера ТРИК требуют проводного подключения, на плате контроллера расположены гироскоп и акселерометр. Эти два датчика применяются в основном для ориентации в пространстве и стабилизации положения робота или его частей.

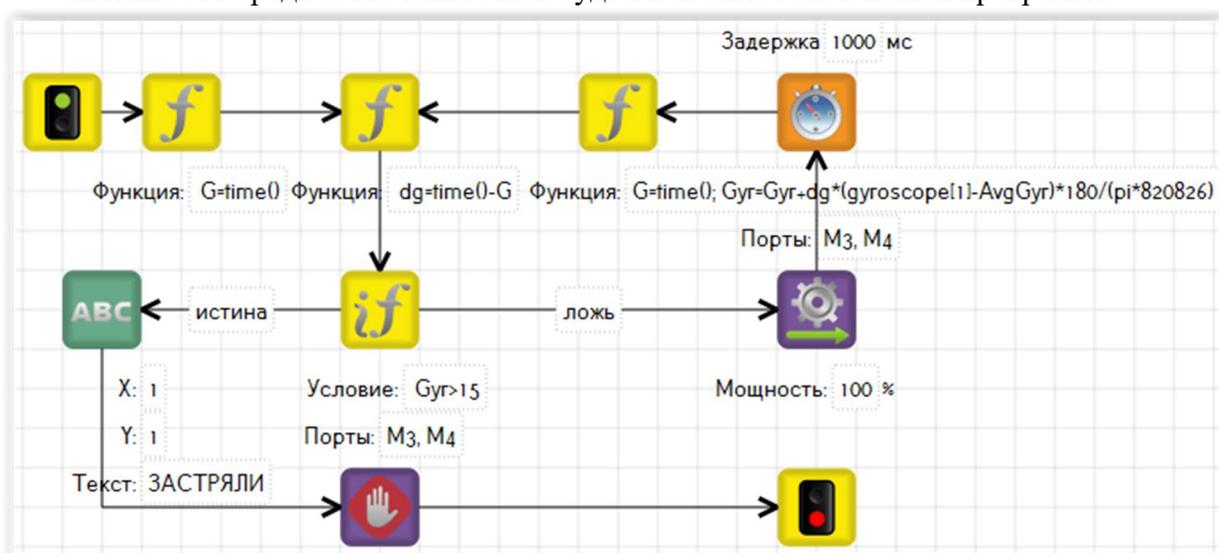
Гироскоп показывает изменение углов ориентации тела, то есть угловые скорости в трех плоскостях. Акселерометр измеряет разность между истинным ускорением объекта и гравитационным ускорением. Комбинация гироскопа и акселерометра позволяет отследить и зафиксировать движение в трехмерном пространстве.

Для этих двух датчиков зарезервированы переменные gyroscope и accelerometer, представляющие собой одномерные массивы из трех элементов, которые показывают отклонения по осям X,Y,Z.

Гироскоп – датчик наклона. Следующая программа сигнализирует о том, что угол наклона изменился, например, робот наехал на препятствие. Сначала откалибруем гироскоп, высчитаем его среднее значение в состоянии покоя AvgGyr:



Именно это среднее значение мы и будем использовать в нашей программе.



Если добавить такой контроль в программу «Исследователь», то защита от попадания в безвыходное положение нам обеспечена.

У акселерометра существует второе название – датчик виброускорения. По виброускорению можно судить о величине колебательных сил, действующих на объект, поэтому акселерометры используются помимо известного всем применения в телефонах,

планшетах и схожих гаджетах, в контрольно-измерительных приборах в различных отраслях: начиная с медицинской и пищевой и заканчивая отраслями промышленности с взрывоопасными условиями эксплуатации оборудования.

Оси координат контроллера расположены следующим образом:



Углы поворотов вокруг осей высчитываются так:

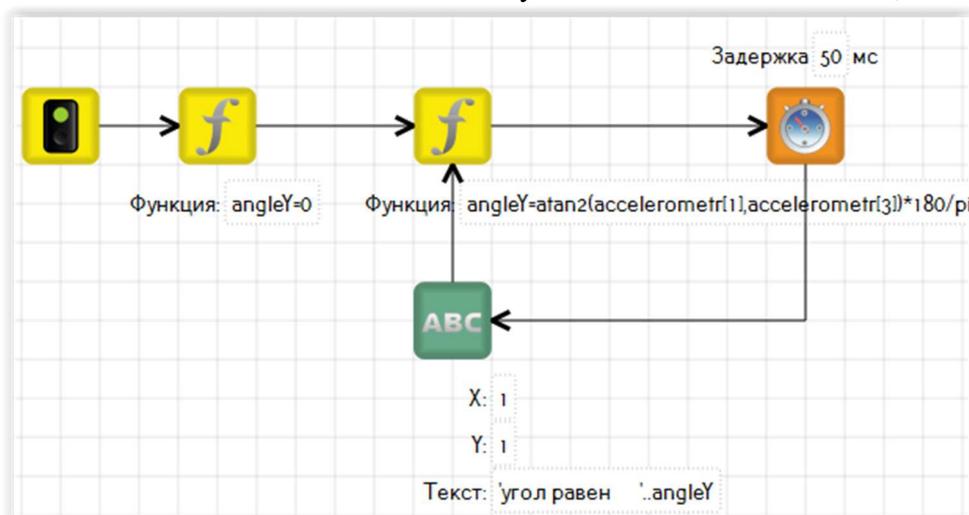
Вокруг оси Y: $\arctg(\text{accelerometer}[1]/\text{accelerometer}[3])$,

Вокруг оси X: $\arctg(\text{accelerometer}[3]/\text{accelerometer}[2])$,

Вокруг оси Z: $\arctg(\text{accelerometer}[2]/\text{accelerometer}[1])$.

Измерение углов ведется в радианах.

Выведем на дисплей значение угла отклонения по оси Y, то есть по горизонтали.



Высчитывая angleX , angleY и angleZ и сравнивая их с критическими значениями, мы получили программу контроля вибровоздействия.

Для точного позиционирования робота в пространстве использование только гироскопа или акселерометра недостаточно. Оба этих датчика наравне со своими преимуществами имеют и свои недостатки, такие как дрейф нуля и высокочастотный шум. Применяя фильтры, можно добиться приемлемых результатов.

Комплементарный фильтр для вычисления угла отклонения:

$$A = (1-K) \times \text{gyr} + K \times \text{acc}$$

Здесь

- A – отфильтрованный, результирующий угол наклона;
- gyr и acc – значения угла наклона, полученные при помощи гироскопа и акселерометра;

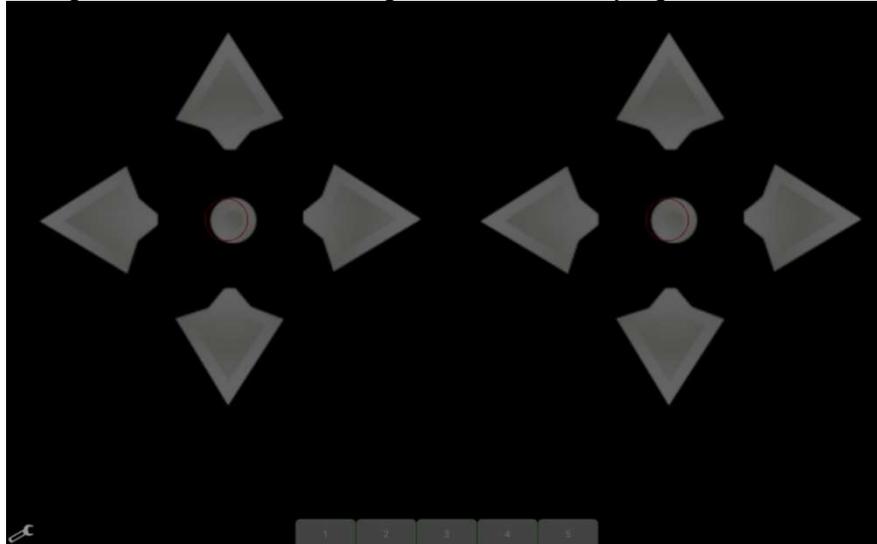
- K – коэффициент комплементарного фильтра, как правило, подбирается вручную для каждого отдельного случая.

Задания для закрепления материала

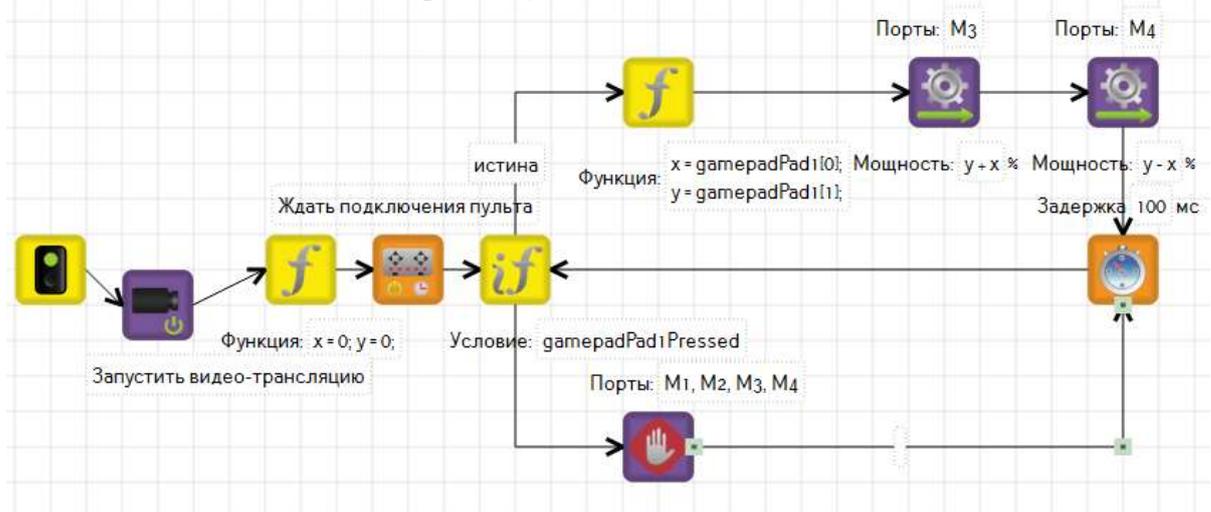
Встреча 23. Дистанционное управление

Робот может двигаться не только автономно, им можно управлять при помощи android-приложения TRIK Gamepad (ссылка для скачивания <https://play.google.com/store/apps/details?id=com.trik.gamepad&hl=ru>)

В приложении имеется два джойстика, для управления моторами и 5 программируемых кнопок, посылающих сигнал контроллеру. С помощью приложения можно выводить трансляцию видео на экран мобильного устройства.



Рассмотрим программу движения робота, находящуюся в папке «Examples» (путь по умолчанию C:\TRIKStudio\examples\trik): remoteControl.



Сначала запускается трансляция и идет ожидание подключения пульта. В блоке “if” проверяется, нажат ли левый джойстик. Если значение блока “if” истинно, то на моторы подается мощность, зависящая от того, где находится палец на координатной оси, осями являются горизонтальные и вертикальные клавиши джойстика.

Можно запрограммировать правый джойстик при помощи переменной `gamepad2Pad` и иконки «Ждать “руля” на пульте». Клавиши, расположенные внизу экрана, программируются при помощи блока «Ждать кнопки на пульте».

Список зарезервированных переменных:

`buttonDown`, `buttonEnter`,
`buttonEsc`, `buttonLeft`,

Состояние кнопок на корпусе робота: кнопки "Вниз",
"Ввод", "Отмена", "Влево", "Питание", "Вправо",

buttonRight, buttonUp	"Вверх" соответственно. Кнопка нажата – значение 1, не нажата – 0.
gamepadButton1, gamepadButton2, gamepadButton3, gamepadButton4, gamepadButton5	Состояние кнопок на Android-пульте, кнопки с 1 по 5 соответственно. Кнопка нажата – значение 1, не нажата – 0.
gamepadConnected	Состояние подключения Android-пульта, Пульт подключен – значение 1, не подключен – 0.
gamepadPad1, gamepadPad2	Координаты нажатия активных областей на Android-пульте. Каждая переменная хранит в себе массив из двух элементов, соответствующих x и y-координате точки нажатия. Координаты изменяются в границах от (-100, -100) до (100, 100), где (-100, -100) соответствует левому верхнему углу активной области. В случае если область не нажата, в переменной находится значение (-101, -101).
gamepadPad1Pressed, gamepadPad2Pressed	Содержат 1, если соответствующая активная область на Android-пульте нажата и 0, если нет.
gamepadWheel	Содержит текущий наклон Android-пульта, если он подключён и на нём включён "руль". Наклон кодируется числами от -100 (крайнее левое положение) до 100 (крайнее правое положение). Если данных о положении "руля" нет, в переменной находится число -101.

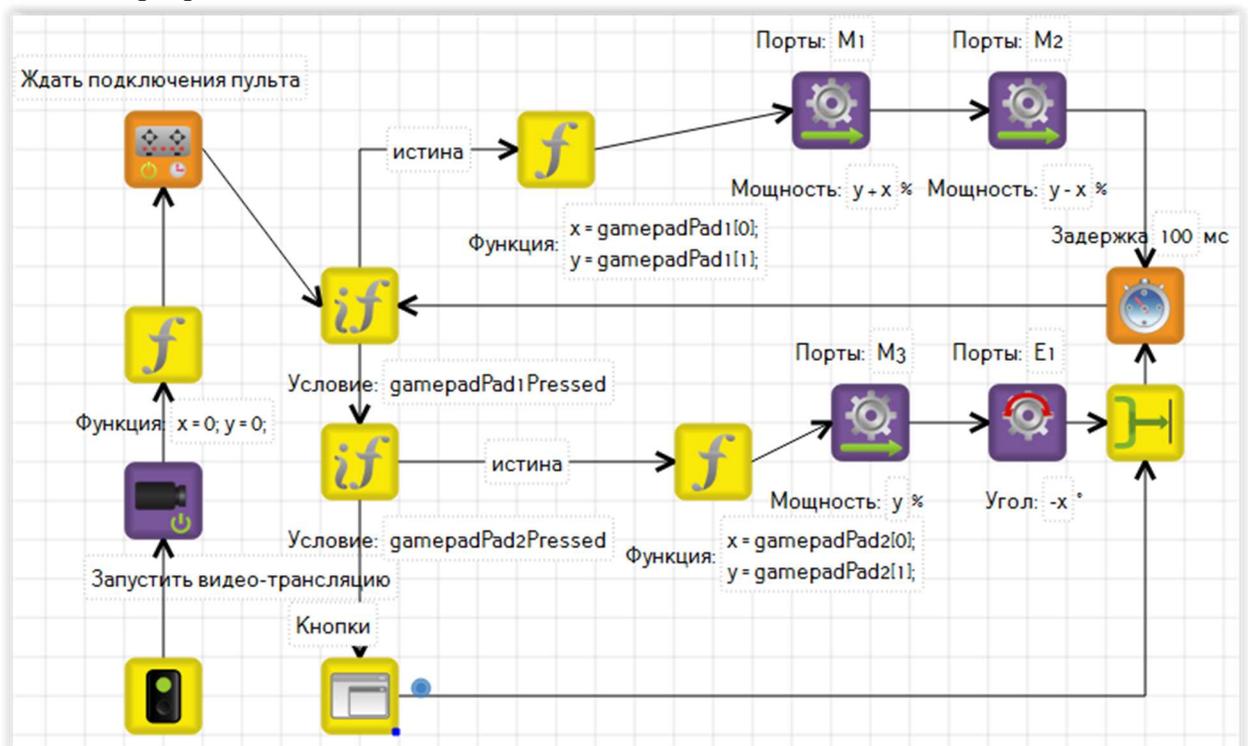
Программа для пульта управления роботом на гусеничном ходу с хватом, расположенным на стреле и видеокамерой. За подъем стрелы отвечает силовой мотор (использована червячная передача) М3. Тяговые моторы – М1 и М2. Хват открывает и закрывает сервомотор Е1.

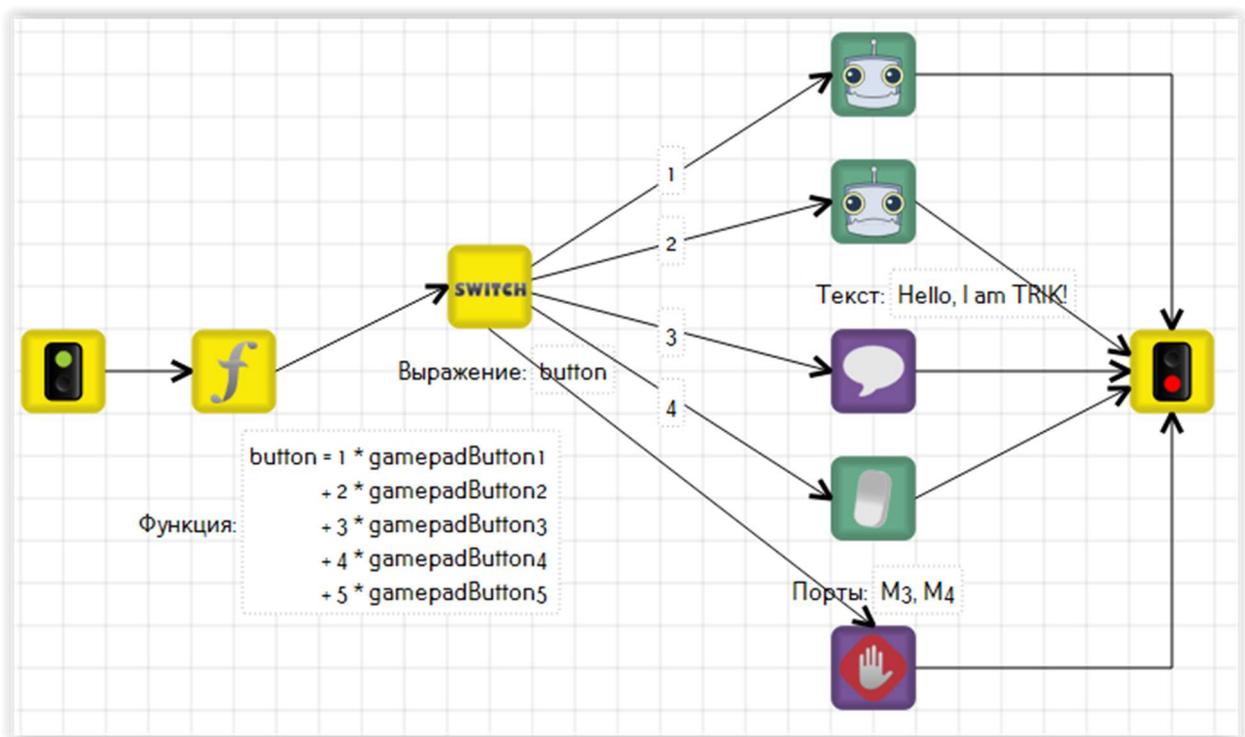


При нажатии кнопок:

- 1 – веселый «Смайлик»
- 2 – «Грустный смайлик»
- 3 – робот произносит: «Hello, I am TRIK»
- 4 – очистка дисплея.

Видеотрансляция выводит изображение с камеры на Android-пульт сразу же с началом программы.





ЗАКЛЮЧЕНИЕ

Среда TRIK Studio – действительно уникальный продукт, рассчитанный как на начинающих, так и на «продвинутых» пользователей. В первой части задачника не была затронута важная особенность TRIK Studio – возможность генерации кода программы. Создавая программу в виде диаграммы, мы всегда можем увидеть, как она выглядит на текстовых языках программирования RuC, Pascal, F#, JavaScript. Для этого достаточно нажать на иконку «Генерировать код». Обратное действие также существует. Написав программу на текстовом языке, через TRIK Studio возможно загрузить ее на контроллер для исполнения. Способы общения с контроллерами, написание программ на текстовых языках, использование специфических особенностей контроллера ТРИК – все это будет рассмотрено в продолжении этого пособия.

Надеемся, что предложенное издание было Вам чем-нибудь полезным и пригодилось в учебной деятельности. Авторы рассчитывают на получение обратной связи от читателей для того, чтобы сделать задачник лучше.

Контактная информация: m.kisselev@yandex.ru
mm.kiselev@yandex.ru.

Литература

1. The LEGO Technic Idea Book: Simple Machines Paperback– October 25, 2010 by YoshihitoIsogawa
2. Филиппов С.А. Робототехника для детей и их родителей. 3-е издание. С-Пб: Наука, 2013
3. Д.Г. Копосов. Первый шаг в робототехнику (практикум). Бином, 2012
4. Введение в программирование Lego-роботов на языке NXT-G. Учебное пособие для студентов и школьников: Учебное пособие / Джендер В.О., Денисова Л.В. - М.: Национальный открытый университет "ИНТУИТ", 2014.
5. ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА. СБОРНИК МЕТОДИЧЕСКИХ РЕКОМЕНДАЦИЙ И ПРАКТИКУМОВ. Автор: Корягин А.В., Смольянинова Н.М.
6. Пёрышкин А.В. Сборник задач по физике: 7 -9 кл.: к учебникам А.В. Пёрышкина и др. «Физика. 7 класс», «Физика. 8 класс», «Физика. 9 класс»/А.В.Пёрышкин; сост. Г.А.Лонцова. – 7-е изд., перераб. и доп. – М.: Издательство «Экзамен», 2012.
7. Поливанова К.Н. Проектная деятельность школьников: пособие для учителя / К.Н. Поливанова. – 2-е изд. – М.: Просвещение, 2011
8. Полякова Т.Н. Педагогический метод проектов: учебно-методическое пособие. – СПб: Спб АППО, 2012
9. Босова Л.Л., Босова А.Ю. Информатика и ИКТ. Методическое пособие для учителя.
10. Босова Л.Л., Босова А.Ю. Программа для углубленного изучения курса «Информатика и ИКТ» в основной школе (5-7 классы), 2012
11. С. Макконнелл - Совершенный код
12. Харольд Абельсон, Джеральд Джей Сассман - Структура и Интерпретация Компьютерных Программ
13. Д. Кнут «Искусство программирования»
14. Никлаус Вирт Алгоритмы и структуры данных. Новая версия для ОберонаИздательство: ДМК Пресс Год: 2010
15. Яншин В. В. Анализ и обработка изображений: принципы и алгоритмы. Издательство: Машиностроение Год: 1994
16. Стивен Скиена. Алгоритмы. Руководство по разработке, 2-е издание Издательство: БХВ-Петербург Год: 2011
17. Ананий Левитин Алгоритмы: введение в разработку и анализ Издательство: Вильямс Год: 2006
18. С. Окулов Программирование в алгоритмах Издательство: Бином. Лаборатория знаний Год: 2004
19. Томас Кормен, Чарльз Лейзерсон, Рональд Ривест, Клиффорд ШтайнАлгоритмы. Построение и анализ, 2-ое издание Издательство: Вильямс Год: 2005

Интернет-ресурсы

1. www.239.ru/robot -Центр робототехники Президентского ФМЛ №239.
2. www.raor.ru - Российская ассоциация образовательной робототехники.
3. www.wroboto.ru - Российский сайт, посвященный подготовке к состязаниям WRO.
4. www.myrobot.ru - Информационный сайт, посвященный робототехнике.
5. www.railab.ru - Лаборатория робототехники и искусственного интеллекта Политехнического музея.
6. www.robotchallenge.org -Ежегодный международный чемпионат по робототехнике в Австрии.

7. www.edurobots.ru - Информационный сайт "Занимательная робототехника".
8. www.robogeek.ru - Информационный сайт Robogeek.
9. www.robotics.ru- каталог сайтов по робототехнике.
10. www.trikset.com – сайт проекта ТРИК.
11. www.robolymp.ru – сайт Всероссийской робототехнической олимпиады.
12. www.lego.com/ru-ru/ - Российский сайт Lego.
13. www.russianrobofest.ru – сайт робототехнического фестиваля Робофест.
14. www.rus-robots.ru - сайт Ассоциации Спортивной Робототехники.