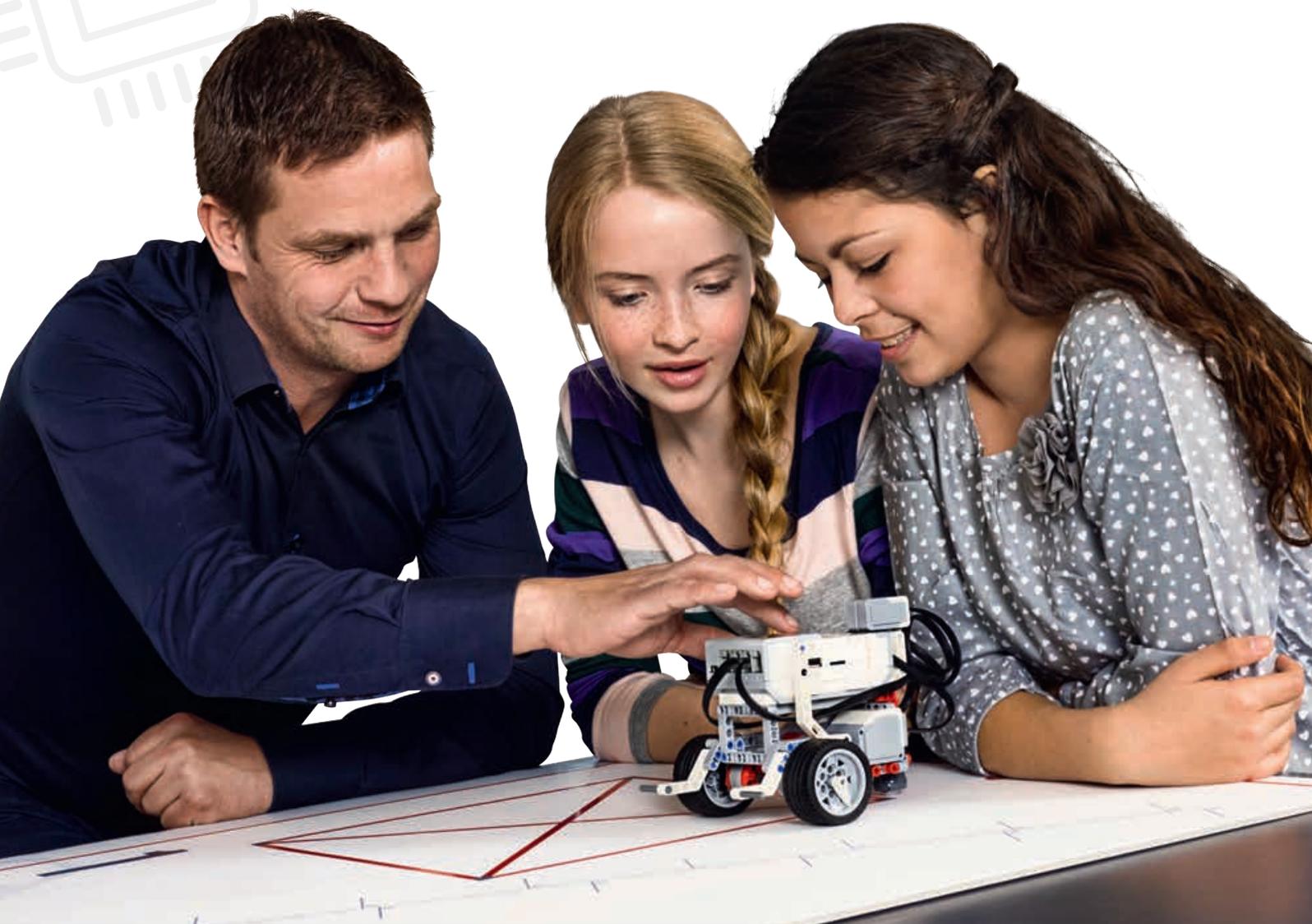


LEGO® MINDSTORMS® Education EV3

ПРОГРАММА ЗАНЯТИЙ ПО ИНФОРМАТИКЕ



Введение

Программа занятий по информатике LEGO® MINDSTORMS® Education EV3



Введение

Программа занятий по информатике LEGO® MINDSTORMS® Education EV3

Одна из основных проблем освоения школьного курса информатики и в первую очередь программирования заключается в преимущественно теоретическом характере изучения содержания, которое на самом деле теснейшим образом связано с нашей повседневной жизнью. Создание образовательных сред, в которых можно придать процессу обучения интерактивный характер, увязать изучаемый материал с решением практических задач и тем самым мотивировать обучающихся, позволяет существенно повысить эффективность образовательного процесса. Визуальная среда программирования робототехнических моделей LEGO- позволяет не только упростить и сделать понятным и доступным каждому процесс создания алгоритмов, но и соединить его с увлекательным делом конструирования разнообразных автоматизированных устройств и моделей (в том числе роботов). Интеграция робототехники в процесс изучения информатики представляет собой весьма актуальную задачу, поскольку позволяет соединить воедино информационные и материальные технологии, изучаемые чаще всего в отрыве друг от друга.

Компания LEGO® Education представляет учебно-методический комплекс по информатике для 7-9 классов основной школы (обучающихся возрастом 12-15 лет).

Эти материалы помогут учителям просто и увлекательно познакомить детей с содержанием информатики на основе изучения современных сервисных технологий, используемых в повседневной жизни. Такой подход поможет учащимся усваивать основные понятия алгоритмики и программирования, а также технического конструирования в контексте решения реальных практических задач.

Коллектив LEGO® Education разработал учебно-методический комплекс (УМК) по информатике на основе MINDSTORMS® Education EV3, чтобы с одной стороны помочь детям в усвоении такого абстрактного предмета как информатика, а с другой - решить целый ряд задач образовательного стандарта: придание личностного смысла процессу учения, формирование регулятивных, познавательных и коммуникативных универсальных учебных действий в процессе изучения информатики, информационных и материальных технологий. Учебно-методический комплекс включает в себя 12 практических занятий (проектов), объемом до 36 учебных часов, и может быть использован как на уроках информатики и технологии, так и во внеурочной деятельности в тесной привязке к материалу основных занятий.

Данный раздел содержит следующую информацию:

- **Ключевые задачи предлагаемого учебно-методического комплекса**
- **Место в учебном плане общеобразовательной школы**
- **12 проектных заданий (занятий) в привязке к примерной основной образовательной программе основного общего образования**
- **Организация занятий**
- **Учебник по робототехнике**
- **Использования Интернет ресурсов, включая видеохостинги**
- **Текстовое программирование**

Введение

Ключевые задачи предлагаемого учебно-методического комплекса

В Федеральном государственном образовательном стандарте (ФГОС) основной школы, а также в утвержденной в 2015 году Примерной основной образовательной программе основного общего образования (ПООП ООО) основное внимание уделяется формированию универсальных учебных действий (УУД). Это требует серьезной переработки содержания программ всех предметов основной школы, которая позволила бы достигать решения как предметных задач, так и личностных, и метапредметных результатов.

Согласно ПООП ООО, одним из путей формирования УУД в основной школе являются: включение обучающихся в учебно-исследовательскую и проектную деятельность, гибкое сочетание урочных, внеурочных форм, а также самостоятельной работы учащегося. Образовательный стандарт потребовал от школы массового внедрения проектных форм работы. Любое действие такого рода должно быть обеспечено качественным учебно-методическим материалом. Этому требованию в полной мере удовлетворяет учебно-методический комплекс LEGO® Education по информатике. Благодаря отработанному методическому подходу и всесторонней поддержке педагогической работы УМК дает в руки учителя богатый инструментарий для работы с детьми и формирования коммуникативных, регулятивных и познавательных универсальных учебных действий, умения работать с информацией, владения информационно-коммуникационными технологиями, смысловым чтением и т.д.

Перечислим основные метапредметные результаты, на формирование которых целенаправленно работает учебно-методический комплекс LEGO® Education по информатике, и укажем используемые для этого педагогические средства и приемы:

- умение развивать мотивы и интересы своей познавательной деятельности - формируется через использование мотивирующей образовательной среды в виде конструктора LEGO - увлекательной, интересной и познавательной игрушки, знакомой с раннего детства каждому ребенку;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией - обеспечивается интерактивностью современного автоматизированного устройства (роботехнической модели): учащиеся получают уникальную возможность видеть сразу же результат своих действий (написанной ими программы), корректировать его по мере необходимости, чтобы достичь намеченной учебной цели;
- умение устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы - при выполнении задач каждого занятия учащимся приходится решать исследовательские задачи во время отладки программы, чтобы достичь требуемого результата;

Введение

- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач - вся работа с УМК построена на освоении и использовании различных представлений (текстовых и пиктографических) алгоритмов, управляющих поведением реального объекта;
- смысловое чтение - в процессе постоянной самостоятельной работы с разнообразными информационными источниками сети Интернет и интегрированными в информационную среду УМК (учебник по робототехнике, библиотеки готовых программ и т.д.);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение - занятия спланированы с использованием групповой формы работы учащихся, совместного решения учебных задач и рефлексивной формы анализа продуктов учебной деятельности;
- умение осознанно использовать речевые средства в соответствии с задачей коммуникации для выражения своих чувств, мыслей и потребностей; планирования и регуляции своей деятельности; владение устной и письменной речью, монологической контекстной речью - в процессе групповой работы учащиеся не только взаимодействуют друг с другом, но и постоянно делятся друг с другом результатами своей работы и обосновывают выбранные ими способы решения учебных задач;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий - благодаря тому, что основным объектом и одновременно средством решения учебных задач являются ИКТ: микрокомпьютер в программируемом ими роботизированном автомобиле и компьютер с информационной оболочкой УМК, служащий для поиска информации, программирования, фиксации и представления результатов и т.д.



Введение

Место в учебном плане общеобразовательной школы

Используя задания учебно-методического комплекса, можно сформировать у детей понимание важности программирования для решения самых разнообразных задач повседневной жизни. Обучающиеся приобретут практический опыт использования нескольких языков программирования (как с текстовой, так и с графической нотацией), проведения экспериментов и исследований, а также в результате изучения рекомендаций, включенных в программное обеспечение LEGO® MINDSTORMS® Education EV3. Учебно-методический комплекс также содержит примеры решений предложенных задач по программированию.

Данный учебно-методический комплекс можно эффективно использовать при изучении информатики на самых разных этапах освоения программы. Например, в форме отдельных проектов, выполняемых в различные моменты изучения теоретического материала. Или в самом начале изучения программирования, в целях создания мотивационной основы для освоения достаточно сложного содержания.

Содержание комплекса предназначено в первую очередь для 7-9 классов. Вместе с тем его легко можно адаптировать и для 10-11 классов. Хотя данный материал предназначен в основном для изучения информатики, он также предлагает широкие возможности межпредметной интеграции с такими предметами, как технология, математика и физика.

Содержащиеся в учебно-методическом комплексе планы занятий существенно сократят затраты времени на планирование их проведения, а по мере накопления опыта работы по готовым планам учителя могут научиться разрабатывать дальнейшие занятия самостоятельно. В конце описания каждого занятия представлены полностраничные примеры иллюстраций и программ, указанных в поурочных планах. Их можно использовать как раздаточный материал.

Желаем вам приятной работы!

ОБЗОРНАЯ ТАБЛИЦА 12 ЗАНЯТИЙ – ПРИВЯЗКА К ПРИМЕРНОМУ УЧЕБНОМУ ПЛАНУ ПО ИНФОРМАТИКЕ

Занятие	Задача	Содержание примерной программы при изучении которого можно использовать УМК LEGO® MINDSTORMS® Education	Рассматриваемые программные блоки EV3 и конструкции текстового языка
1	Создание в среде визуального программирования EV3 программы разворота в три приема. Введение в программирование. Аппаратное и программное обеспечение микрокомпьютера EV3. Простые перемещения автономного движущегося робота и повороты.	Понятия: исполнитель, управление, сигнал, обратная связь, компьютер и микроконтроллер - устройства управления, программное управление, алгоритм, линейный алгоритм, программа, отладка и запуск программы, датчик. Составление алгоритмов и программ по управлению исполнителями. Словесное описание алгоритмов. Описание алгоритма с помощью блок-схем. Отличие словесного описания алгоритма, от описания на формальном алгоритмическом языке. Компьютер и управляемый им исполнитель (в том числе робот); компьютер, получающий сигналы от цифровых датчиков в ходе наблюдений и экспериментов, и управляющий реальными (в том числе движущимися) устройствами.	– Управление движением – Блок – Ультразвуковой датчик – Звук

Введение

Занятие	Задача	Содержание примерной программы при изучении которого можно использовать УМК LEGO® MINDSTORMS® Education	Рассматриваемые программные блоки EV3 и конструкции текстового языка
2	<p>Создание программы разворота в три приема на языке текстового программирования ROBOTC.</p> <p>Сравнение текстового и визуального программирования. Ученики составляют текстовые программы, опираясь на материал, усвоенный на первом занятии.</p>	<p>Понятия: исполнитель, управление, сигнал, обратная связь, компьютер и микроконтроллер - устройства управления, программное управление, алгоритм, линейный алгоритм, программа, отладка и запуск программы, датчик. Составление алгоритмов и программ по управлению исполнителями.</p> <p>Алгоритмический язык, язык программирования. Системы программирования. Запись алгоритмических конструкций в выбранном языке программирования.</p> <p>Знакомство с документированием программ. Составление описания программы по образцу.</p>	<ul style="list-style-type: none"> – Управление движением – Блок – Ультразвуковой датчик – Звук
3	<p>Составление программы управления роботом, который при столкновении с препятствием сдает назад.</p> <p>Использование программных блоков для отображения графического и светового состояния микрокомпьютера EV3. Предупреждающие знаки на автомобилях.</p>	<p>Робототехника – наука о разработке и использовании автоматизированных технических систем. Автономные роботы. Микроконтроллер. Сигнал. Обратная связь: получение сигналов от цифрового датчика касания.</p> <p>Примеры роботизированных систем (автономная система управления транспортным средством).</p> <p>Автономные движущиеся роботы. Исполнительные устройства, датчики. Система команд робота. Конструирование робота. Моделирование робота парой: исполнитель команд и устройство управления. Ручное и программное управление роботами.</p> <p>Пример учебной среды разработки программ управления движущимися роботами. Алгоритмы управления движущимися роботами. Реализация алгоритма “движение до препятствия”.</p> <p>Анализ алгоритмов действий роботов. Испытание механизма робота, отладка программы управления роботом.</p> <p>Направление мобильного автономного робота по прямой линии при помощи блока движения и рулевого управления.</p> <p>Световое и графическое отображение информации.</p>	<ul style="list-style-type: none"> – Рулевое управление – Блок – Датчик касания
4	<p>Программирование работы автоматических фар: включение “фар” при наступлении “темноты” и выключение, когда снова станет “светло”.</p> <p>Изучение работы датчика цвета.</p> <p>Настройки освещенности.</p> <p>Автоматические фары на автомобилях и автоматическое управление уличным освещением.</p>	<p>Конструкция «ветвление». Условный оператор: полная и неполная формы.</p> <p>Выполнение и невыполнения условия (истинность и ложность высказывания). Простые и составные условия. Запись составных условий.</p> <p>Конструкция «повторения»: циклы с заданным числом повторений, с условием выполнения, с переменной цикла.</p> <p>Обратная связь: получение сигналов от цифрового датчика цвета (освещенности).</p> <p>Реализация алгоритма «включение света при уменьшении освещенности».</p>	<ul style="list-style-type: none"> – Ожидание – Датчик цвета – Отображение на дисплее – Время – Цикл – Датчик касания – Прерывание цикла – Многозадачность

Введение

Занятие	Задача	Содержание примерной программы при изучении которого можно использовать УМК LEGO® MINDSTORMS® Education	Рассматриваемые программные блоки EV3 и конструкции текстового языка
5	<p>Программирование распознавания красного цвета и остановки колесного робота при красном сигнале светофора и возобновления движения при зеленом сигнале.</p> <p>Применение датчика цвета для распознавания цветов системы LEGO® и интенсивности отраженного света.</p> <p>Программирование движения по линии.</p> <p>Автомобильный автопилот.</p>	<p>Обратная связь: получение сигналов от цифрового датчика цвета.</p> <p>Примеры роботизированных систем (система управления движением в транспортной системе, автономная система управления транспортным средством).</p> <p>Реализация алгоритма «следование вдоль линии».</p> <p>Анализ алгоритмов действий роботов. Испытание механизма робота, отладка программы управления роботом. Влияние ошибок измерений и вычислений на выполнение алгоритмов управления роботом.</p> <p>Конструкция «ветвление». Условный оператор: полная и неполная формы.</p> <p>Выполнение и невыполнения условия (истинность и ложность высказывания). Простые и составные условия. Запись составных условий.</p> <p>Конструкция «повторения»: циклы с заданным числом повторений, с условием выполнения, с переменной цикла.</p>	<ul style="list-style-type: none"> – Ожидание – Движение и рулевое управление – Датчик цвета – Цикл – Переключатель – Прерывание цикла
6	<p>Программирование колесного робота на движение задним ходом, с подачей предупреждающих гудков при приближении к препятствию и затем автоматическую остановку на заданном расстоянии.</p> <p>Изучение работы ультразвукового датчика.</p> <p>Изучение принципов работы систем автомобильных парктроников.</p>	<p>Обратная связь: получение сигналов от ультразвукового датчика расстояния.</p> <p>Примеры роботизированных систем (автономная система управления транспортным средством).</p> <p>Реализация алгоритма «сигналы парктроника».</p> <p>Понимание принципа работы ультразвукового датчика за счет отражения волн и умение программировать датчик на определение расстояния.</p> <p>Освоение математических программных блоков и функций.</p> <p>Освоение возможности переноса показаний с одного блока в другой через канал передачи данных.</p>	<ul style="list-style-type: none"> – Движение и рулевое управление – Ожидание – Ультразвуковой датчик – Цикл – Математика – Звук
7	<p>Программирование запуска двигателя колесного робота при одновременном выполнении трех условий: срабатывание датчиков касания и расстояния, а также кнопки интеллектуального блока.</p> <p>Изучение принципа работы систем автоматического запуска автомобиля без ключа.</p>	<p>Высказывания. Простые и сложные высказывания. Диаграммы Эйлера-Венна. Логические значения высказываний. Логические выражения. Логические операции: «и» (конъюнкция, логическое умножение), «или» (дизъюнкция, логическое сложение), «не» (логическое отрицание). Правила записи логических выражений. Приоритеты логических операций.</p> <p>Использование блока логики в сочетании с блоком переключения.</p> <p>Применение сочетания нескольких датчиков для запуска программы микрокомпьютера EV3.</p>	<ul style="list-style-type: none"> – Ожидание – Датчик касания – Ультразвуковой датчик – Дисплей – Время – Датчик касания – Кнопки интеллектуального блока – Логика – Переключатель – Цикл – Движение и рулевое управление

Введение

Занятие	Задача	Содержание примерной программы при изучении которого можно использовать УМК LEGO® MINDSTORMS® Education	Рассматриваемые программные блоки EV3 и конструкции текстового языка
8	<p>Программирование ускорения и замедления колесного робота при нажатии на один из двух датчиков касания.</p> <p>Изучение принципа работы системы круиз-контроля автомобиля.</p>	<p>Оператор присваивания. Представление о структурах данных.</p> <p>Константы и переменные. Переменная: имя и значение. Типы переменных: целые, вещественные, символьные, строковые, логические.</p> <p>Конструкция «ветвление». Условный оператор: полная и неполная формы.</p> <p>Выполнение и невыполнения условия (истинность и ложность высказывания). Простые и составные условия. Запись составных условий.</p> <p>Алгоритмический язык, язык программирования. Системы программирования. Запись алгоритмических конструкций в выбранном языке программирования.</p> <p>Использование блока переменных для хранения информации. Разработка многоуровневых программ. Подпрограмма.</p>	<ul style="list-style-type: none"> – Ожидание – Датчик касания – Цикл – Переключатель – Переменная – Математика – Движение и рулевое управление – Мои блоки / подпрограммы
9	<p>Создание программы, заставляющей робота двигаться по заданному маршруту.</p> <p>Исследование программы сортировщика по цвету. Массивы.</p>	<p>Константы и переменные. Переменная: имя и значение. Типы переменных: целые, вещественные, символьные, строковые, логические. Табличные величины (массивы). Одномерные массивы. Двумерные массивы.</p> <p>Использование блока переменных для хранения информации. Использование блока операций над массивами.</p>	<ul style="list-style-type: none"> – Переменная – Ожидание – Кнопки интеллектуального блока – Цикл – Звук – Операции над массивами – Время – Движение и рулевое управление – Мои блоки / подпрограммы
ИТОГОВЫЕ ПРОЕКТЫ			
10	<p>Проектирование самоходного колесного робота, который может двигаться из пункта А в пункт В, обходя препятствия.</p>	<p>Программное управление самодвижущимся роботом. Понятие об этапах разработки программ и приемах отладки программ.</p> <p>Компьютер, получающий сигналы от цифровых датчиков в ходе наблюдений и экспериментов, и управляющий реальными (в том числе движущимися) устройствами.</p> <p>Составление алгоритмов и программ по управлению исполнителями.</p>	<p>Можно использовать любые программные блоки из рассмотренных за прошедшие недели.</p>

Введение

Занятие	Задача	Содержание примерной программы при изучении которого можно использовать УМК LEGO® MINDSTORMS® Education	Рассматриваемые программные блоки EV3 и конструкции текстового языка
11	Конструирование и программирование самоходного колесного робота, который может двигаться из пункта А в пункт В, обходя препятствия.	Понятие об этапах разработки программ: составление требований к программе, выбор алгоритма и его реализация в виде программы на выбранном алгоритмическом языке, отладка программы с помощью выбранной системы программирования, тестирование.	Можно использовать любые программные блоки из рассмотренных за прошедшие недели.
12	Анализ технического решения, внесение изменений и вывод об эффективности технического решения.	Анализ алгоритмов действий роботов. Испытание механизма робота, отладка программы управления роботом. Влияние ошибок измерений и вычислений на выполнение алгоритмов управления роботом.	Можно использовать любые программные блоки из рассмотренных за прошедшие недели.

Применение программы занятий

ОРГАНИЗАЦИЯ ЗАНЯТИЙ

Большинство занятий строятся на выполнении учащимися последовательности связанных друг с другом практических задач, решение которых представляет собой программу управления движущимся колесным роботом. Возможные решения к этим задачам описаны в поурочных планах, а программы для EV3 доступны для загрузки. Занятие также включает процесс конструирования робота. Обычно это модель из встроенного в информационную среду УМК учебника по робототехнике. С 1-го по 9-е занятие рекомендуется сделать упор на проектировании базовых алгоритмов управления роботом, чтобы ученики были хорошо подготовлены к трем последним занятиям, когда на основе владения базовыми алгоритмами и средой программирования EV3 учащиеся по группам проектируют, конструируют и программируют комплексный проект самодвижущегося автомобиля (робота), для выполнения которого им потребуются знания и умения всех предыдущих занятий.

Работа над каждым заданием должна включать этап обсуждения структуры / проекта программы с учениками.

УЧЕБНИК ПО РОБОТОТЕХНИКЕ

На каждом из занятий ученикам будет требоваться хорошее знание программного обеспечения EV3. Такие знания можно сформировать путем ознакомления с руководствами учебника по робототехнике, указанными в конце плана каждого занятия (под заголовком “Примечания для учителя”). Самостоятельная работа с этими руководствами, в сочетании с традиционным обучением через общение с педагогом, помогут ученикам развить навыки, необходимые для решения задач каждого занятия. Эти ресурсы помогают учащимся формировать навыки работы с информацией.

Раздел с руководствами из учебника по робототехнике делится на “Новые руководства по робототехнике” и ссылки на ранее рассмотренные руководства, которые можно использовать на данном занятии.



Введение

ИСПОЛЬЗОВАНИЕ ИНТЕРЕНЕТ РЕСУРСОВ, ВКЛЮЧАЯ ВИДЕОХОСТИНГИ

Данная программа занятий не предлагает прямых ссылок на поисковые системы или видеохостинги. Это связано с тем, что со временем ссылки могут устареть. Также в некоторых учебных заведениях действуют ограничения на пользование видеохостингами, вплоть до полного запрета на их использование.

В целях преодоления таких ограничений мы указали ряд ключевых слов, которые можно ввести в выбранную поисковую систему и найти подходящие ресурсы.

Пример.

Для начала работы по программе можно посмотреть видеосюжет с рассказом об автомобиле BMW, оснащенным автопилотом, изначально показанный в программе BBC Top Gear. Чтобы найти это видео, введите в поисковую систему или строку поиска видеохостинга следующее:

Ключевые слова: BMW driverless car top gear (BMW самоходный автомобиль Top Gear)

После ввода этих слов появятся ссылки на видеоролик, где вы сможете увидеть самоходный автомобиль в действии.

ТЕКСТОВОЕ ПРОГРАММИРОВАНИЕ

В данном издании настоящей программы в качестве текстового языка программирования используется нотация ROBOTC.

Однако данная программа занятий не предназначена для обучения педагогов и учеников работе с указанным языком программирования. Основная задача данной программы показать, как программы для EV3 могут быть написаны с использованием текстового языка программирования класса C++. Вся необходимую информацию о языках семейства ROBOTC можно найти по следующему адресу: <http://www.robotc.net/>

По возможности предложенные программы ROBOTC в точности соответствуют аналогичным программам, написанным на LabVIEW (основной язык программирования EV3). Однако могут встречаться и различия в используемой логике в зависимости от применяемой нотации. Все выявленные различия обозначаются в соответствующей программе на ROBOTC зеленым цветом шрифта.

учащимся формировать навыки работы с информацией.

Раздел с руководствами из учебника по робототехнике делится на "Новые руководства по робототехнике" и ссылки на ранее рассмотренные руководства, которые можно использовать на данном занятии.



Содержание

ЗАНЯТИЕ 1	Выполнение разворота в три приема	13-32
ЗАНЯТИЕ 2	Использование ROBOTC для программы разворота в три приема	33-55
ЗАНЯТИЕ 3	Движение робота задним ходом	56-75
ЗАНЯТИЕ 4	Освещение пути	76-96
ЗАНЯТИЕ 5	Светофоры и автоматизированные рельсовые системы	97-118
ЗАНЯТИЕ 6	Звуковой сигнал заднего хода	119-140
ЗАНЯТИЕ 7	Запуск двигателя автомобиля без ключа	141-159
ЗАНЯТИЕ 8	Круиз-контроль	160-180
ЗАНЯТИЕ 9	Мобильные роботы	181-196
ИТОГОВЫЕ ПРОЕКТЫ		197-221
ЗАНЯТИЕ 10	Проектирование собственного мобильного автономного робота	197-206
ЗАНЯТИЕ 11	Конструирование и программирование собственного мобильного автономного колесного робота	207-213
ЗАНЯТИЕ 12	Анализ, доработка и представление собственного мобильного автономного колесного робота	214-221



ЗАНЯТИЕ 1

Выполнение разворота в три приема



Занятие 1

Выполнение разворота в три приема

ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ И ПЛАТФОРМУ MINDSTORMS® EV3

На этом занятии ученики получают первые вводные знания об информатике и познакомятся с понятием программирования. На этом и последующих занятиях дети узнают о важности программирования в повседневной жизни.

Каждое занятие строится на ранее полученных знаниях и опыте и дает ученикам необходимые знания для последующего создания мобильного автономного автомобиля – колесного робота, который может проехать из пункта А в пункт В самостоятельно, т.е. без управления водителем.

Чтобы лучше погрузить учеников в изучаемый материал, можно инициировать осуждения технологий, изучаемых на этом занятии, в начале каждого урока. Примерные вопросы для обсуждения:

- Какими характеристиками такой автомобиль должен обладать по мнению учеников? Обсудите такие характеристики, как разнообразные сервисные системы для безопасности и системы навигации.
- Что еще могут предложить ученики?
- Какие автоматизированные системы есть в автомобилях их родителей? Это могут быть датчики расстояния для обнаружения препятствий (парктроники), автоматические системы остановки и системы предупреждения пешеходов и т.д.

Примечание: Можно организовать посещение автомобильного завода или ближайшего автосалона, чтобы узнать больше об автоматических функциях некоторых автомобилей.

ЗНАКОМСТВО С МИКРОКОМПЬЮТЕРОМ EV3 И ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ EV3

Необходимо выделить время на сборку базовой модели EV3 (не более 15 минут), используя Базовое ПО LME EV3 или инструкцию, входящую в Базовый набор LME EV3.

В основном урок посвящен использованию моторов и написанию алгоритма для выполнения разворота в три приема.

Объясните, что при помощи разворота в три приема автомобили разворачиваются на 180 градусов.

Для наглядности можно использовать видео из интернета с разворотом в три приема / разворотом на дороге.



Занятие 1

План урока

ПЛАН УРОКА

На уроке должны быть выполнены три основные задачи. Решения к ним указаны в плане урока / разделе с ресурсами.

РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание, что с помощью алгоритмов можно выполнять определенную последовательность команд;
- понимание принципа работы блока независимого управления моторами, чтобы управлять перемещениями колесного робота;
- сборка и программирование простого колесного робота с использованием блока независимого управления моторами, синхронизация для разворота на 180° в три приема;
- знакомство с командой "ожидание" и блоком ультразвукового датчика.

ОСНОВНЫЕ ПОНЯТИЯ

Вход, выход, алгоритм, ожидание, ультразвуковой датчик, отладка программы.

ВВЕДЕНИЕ

- Объясните ученикам, что на уроке они будут собирать и программировать мобильного колесного робота, который должен выполнять разворот в три приема. Чтобы привести собранные модели в движение, ученикам сперва придется ознакомиться с программным обеспечением EV3.
- Продемонстрируйте ученикам готовую базовую модель EV3 и объясните, что они должны собрать такую же модель, опираясь на встроенные инструкции в программном обеспечении EV3. Инструкции находятся в разделе программного обеспечения EV3 **Самоучитель/Инструкции по сборке**.
- Покажите, как осуществляется доступ к программному обеспечению EV3, и как при помощи блока независимого управления моторами можно заставить колесного робота двигаться.



Рисунок 1



Рисунок 2



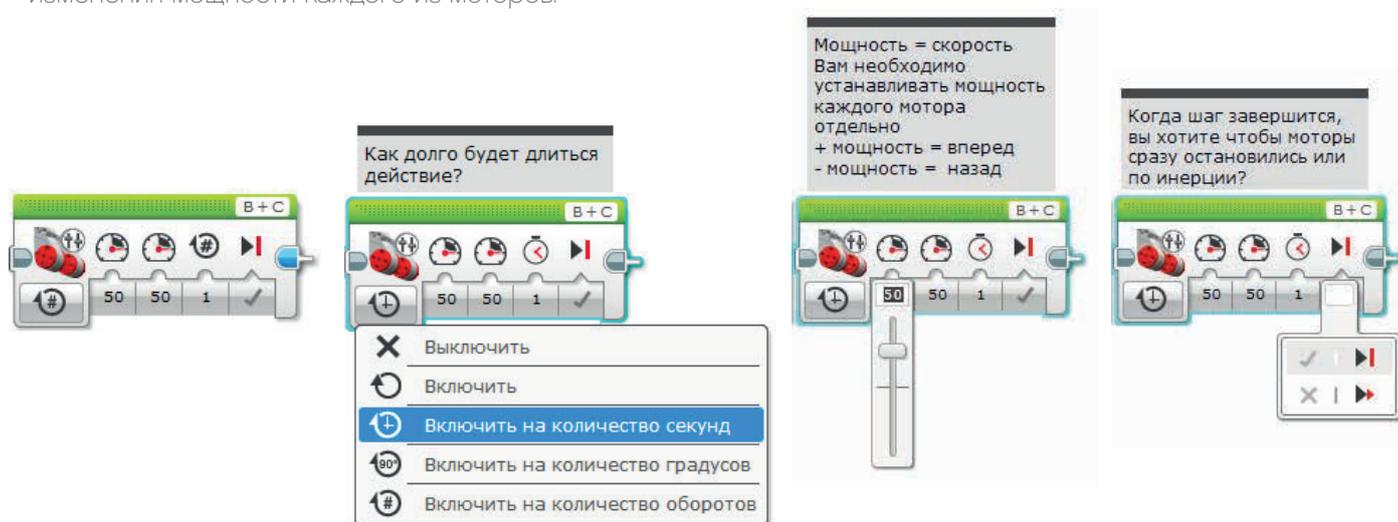
Рисунок 3

Занятие 1

План урока

ВВЕДЕНИЕ: ИССЛЕДОВАНИЕ

- Ученики в парах собирают базовую модель.
- По завершении сборки они должны начать новый проект в ПО EV3 и провести эксперименты с блоком независимого управления моторами.
- Попросите учеников исследовать различные способы разворота колесного робота и эффекты изменения мощности каждого из моторов.



ОСНОВНОЕ ЗАДАНИЕ 1

- Попросите учеников рассказать о том, какие способы они применяли для разворота робота с помощью блока независимого управления моторами. Пусть они объяснят, как пришли к такому решению.
- Поясните, что они должны будут использовать то, что они узнали из наблюдений за разворотом модели, для моделирования разворота в три приема.
- Можно показать классу видео о выполнении разворота в три приема.
- Далее обучающиеся должны создать в программном обеспечении EV3 программу разворота в три приема, задав разные значения мощности для каждого мотора, что заставит робота двигаться в нужном направлении.
- Напомните детям о необходимости постоянного анализа и отладки созданной ими программы, чтобы робот мог развернуться именно на 180 градусов.
- Примечание: Можно при помощи клейкой ленты обозначить "дорожную разметку", чтобы дети научились разворачивать робота в ограниченном пространстве.
- Примечание: В предлагаемых программных решениях перед заключительным этапом робот переходит в режим прямолинейного движения перед тем, как тронуться. Однако в реальности это не всегда возможно, соответственно, обучающиеся могут включить движение по дуге перед переходом к прямолинейному движению. Напомните детям, что у таких задач нет неправильных решений.

Занятие 1

План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 1
ВКЛАДКА: MAIN 1

Сделайте разворот в три приема используя управление и время

Поворот вправо и остановка через 1.5 секунды

Обратный разворот влево и остановка через 1 секунду

Теперь надо поехать вперед в против.направлении

ОСНОВНОЕ ЗАДАНИЕ 2

- Расскажите о простейших вариантах применения ультразвукового датчика.
- Спросите класс, каким образом можно сделать программу более автономной, и как можно распознавать все препятствия, возникающие в процессе движения колесного робота задним ходом.
- Продемонстрируйте блок ожидания и то, как его можно использовать с ультразвуковым датчиком.
- Ученики должны усовершенствовать программу таким образом, чтобы мобильный колесный робот останавливался в заданной точке в ответ на сигнал ультразвукового датчика.
- Поясните, что их программы имитируют действия водителя, нажимающего на тормоза, когда сдающий назад автомобиль приближается к препятствию.

ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 1
ВКЛАДКА: MAIN 2

Сделайте разворот в три приема используя управление и время. Добавим ультразвуковой датчик как парковочный датчик

Поворот вправо и остановка через 1.5 секунды

Обратный разворот влево

Ожидание ультразву.датчика для определения помехи

Остановка на 1 секунду. (Выключаем моторы, Ждем 1 секунду)

Теперь надо поехать вперед в против.направлении



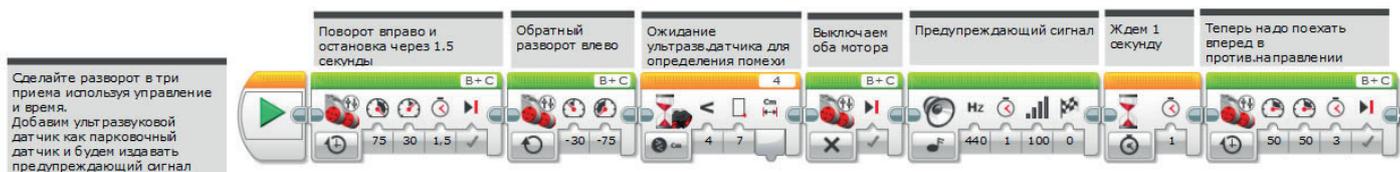
Занятие 1

План урока

ОСНОВНОЕ ЗАДАНИЕ 3

- Добавить колесному роботу функции безопасности на основе ультразвукового датчика (например, предупреждающие звуковые сигналы).
- Спросите класс, что происходит с автомобилем, когда он сдает назад и приближается к препятствию – можно слышать предупреждающий сигнал.
- Покажите звуковой блок, с помощью которого можно добавить предупреждающий сигнал после остановки колесного робота. Проявите ученикам, как работает звуковой блок и где он должен быть в программе.
- Ученики должны изменить составленные ими программы, включив в них звуковой блок таким образом, чтобы он инициировал предупреждающий сигнал, когда колесный робот оказывается на определенном расстоянии от препятствия.
- Желательно, чтобы они постоянно отлаживали программы, пока не добьются необходимой эффективности звуковой сигнализации и остановки на подходящем расстоянии.

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 1
ВКЛАДКА: MAIN 3



ПОДВЕДЕНИЕ ИТОГОВ

- Перечислите основные моменты, которые были усвоены на данном уроке.
- Убедитесь, что ученики понимают термины, использованные в ходе занятия, и могут ими оперировать, повторите значения основных слов.
- Попросите одну-две группы продемонстрировать свои программы. Обсудите, что получилось и над чем еще можно поработать.
- Объясните ученикам, что на следующей неделе они будут решать те же задачи, но уже не при помощи программного обеспечения EV3, а пользуясь текстовым языком программирования.



Занятие 1

Рабочие карточки

ПОДВЕДЕНИЕ ИТОГОВ

Цель сегодняшнего занятия - ближе познакомить вас с программным обеспечением LEGO® MINDSTORMS® Education EV3 и помочь вам начать активную работу с ним.

Вы уже смогли немного поэкспериментировать с блоком независимого управления моторами, заставляя робота перемещаться по классу. Теперь, опираясь на этот опыт, вы должны решить три задачи.

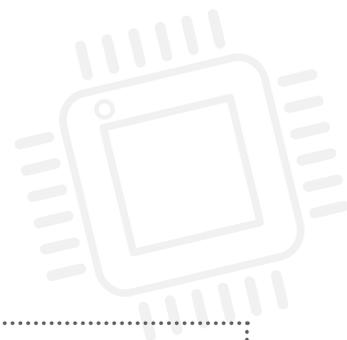
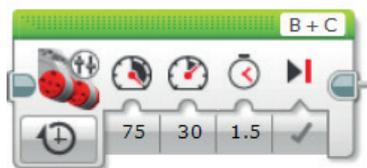
Удачи!

ЗАДАЧА 1

Запрограммируйте колесного робота на разворот в три приема.

Робот должен выполнить поворот при движении вперед, затем сдать назад и снова двигаться вперед. Посмотрите видео еще раз, чтобы вспомнить, как выглядит такой разворот. Следите за тем, чтобы не пересечь дорожную разметку!

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 1

Рабочие карточки

ЗАДАЧА 2

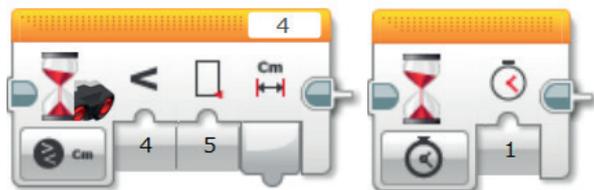
Сейчас вы должны провести эксперимент с одним из датчиков EV3 – ультразвуковым датчиком.

Запрограммируйте колесного робота на разворот в три приема, используя ультразвуковой датчик в качестве парковочного радара, чтобы робот при движении задним ходом останавливался на заданном расстоянии от препятствия.

Может ли ваш робот затормозить перед тем, как тронуться в новом направлении?

Здесь вам потребуются знания о блоке ожидания, также нужно установить ультразвуковой датчик на заднем торце робота.

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 1

Рабочие карточки

ЗАДАЧА 3

Моделирование предупреждающих звуковых сигналов.

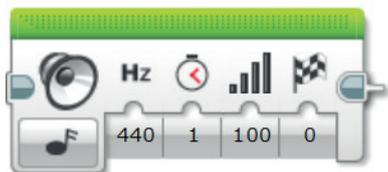
Что часто происходит, когда автомобиль сдает назад и приближается к препятствию?

Теперь ваш колесный робот останавливается по сигналу ультразвукового "датчика парковки". Можете ли вы усовершенствовать программу таким образом, чтобы робот издавал предупреждающий звуковой сигнал сразу после включения тормозов при заднем ходе?

Программу необходимо постоянно отлаживать, чтобы предупреждающий звуковой сигнал выключался одновременно с остановкой робота. Какую часть программы нужно изменить?

Используемые блоки

Используйте те же блоки, что и в задачах на программирование 1 и 2, но также не забывайте об этом :



Plan your program first. Write it in pseudo-code below:

Занятие 1

Рабочие карточки

По завершении занятия по программированию важно записать свои мысли и наблюдения. Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?



Мысли и наблюдения

Занятие 1

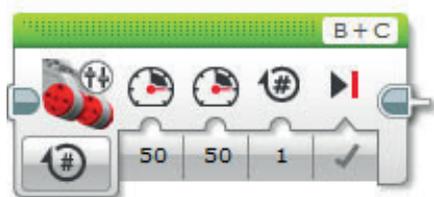
Примечания для учителя

РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

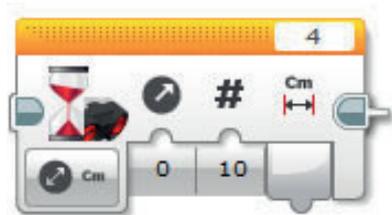
Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами.

НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Самоучитель > Основы > Независимое управление моторами



Самоучитель > Основы > Остановка у объекта



Занятие 1

Приложение

ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 1:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 1

Приложение



Занятие 1

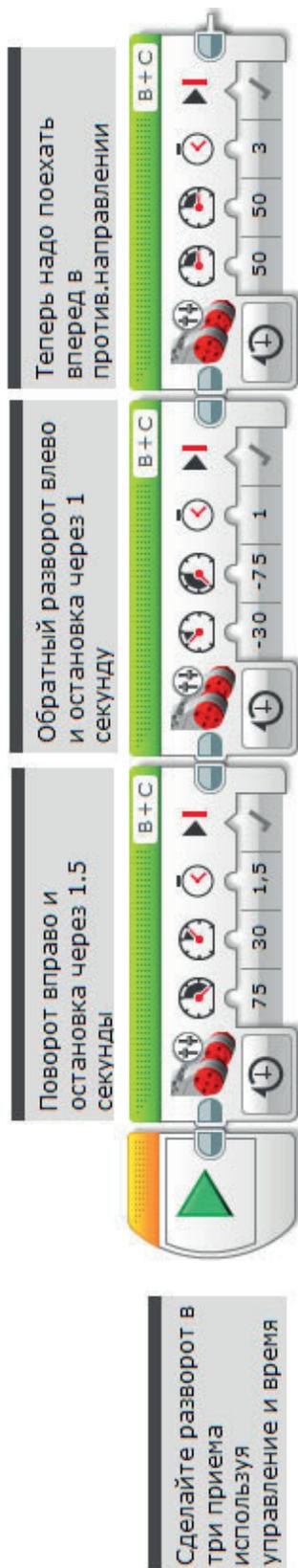
Приложение



Занятие 1

Приложение

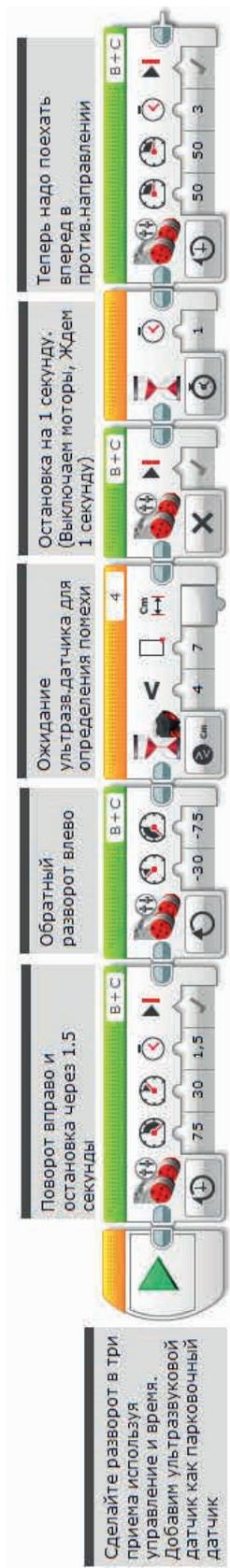
ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 1
ВКЛАДКА: MAIN 1



Занятие 1

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 1
ВКЛАДКА: MAIN 2



Занятие 1

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 1
ВКЛАДКА: MAIN 3

Сделайте разворот в три приёма используя управление и время. Добавьте ультразвуковой датчик как парковочный датчик и будем издавать предупреждающий сигнал

Поворот вправо и остановка через 1.5 секунды

Обратный разворот влево

Ожидание ультразвукового датчика для определения помехи

Выключаем оба мотора

Предупреждающий сигнал

Ждем 1 секунду

Теперь надо поехать вперед в противонаправлении

Block Label	Block Type	Value 1	Value 2	Value 3
Поворот вправо и остановка через 1.5 секунды	Motor	75	30	1.5
Обратный разворот влево	Motor	-30	-75	
Ожидание ультразвукового датчика для определения помехи	Wait	4		
Выключаем оба мотора	Motor	X		
Предупреждающий сигнал	Sound	440	1	100
Ждем 1 секунду	Wait	1		
Теперь надо поехать вперед в противонаправлении	Motor	50	50	3

Занятие 1

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session1_1.c

```
Activity1_1.c
1  #pragma config(StandardModel, "EV3_REMBO")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard!!**/
3
4  /*
5  Create a three point turn using timing and steering.
6  */
7
8  task main()
9  {
10     //Turn to the right and stop after 1.5 seconds.
11     setMotorSpeed(motorB, 75);
12     setMotorSpeed(motorC, 30);
13     sleep(1500);
14
15     //Reverse to the left and stop after 1 second.
16     setMotorSpeed(motorB, -30);
17     setMotorSpeed(motorC, -75);
18     sleep(1000);
19
20     //You should now be straight and facing the other way. Drive off.
21     setMotorSpeed(motorB, 50);
22     setMotorSpeed(motorC, 50);
23     sleep(3000);
24 }
25
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 1

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session1_2c

```
Activity1_2.c
1  #pragma config(StandardModel, "EV3_REMBOТ")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  /*
5  Create a three point turn using timing and steering.
6  Introducing the Ultrasonic Sensor to act as parking sensors.
7  */
8
9  task main()
10 {
11     //Turn to the right and stop after 1.5 seconds.
12     setMotorSpeed(motorB, 75);
13     setMotorSpeed(motorC, 30);
14     sleep(1500);
15
16     //Reverse to the left while the Ultrasonic Sensor sees a value greater than or
17     setMotorSpeed(motorB, -30);
18     setMotorSpeed(motorC, -75);
19     while(getUSDistance(sonarSensor) >= 7)
20     {
21         //Keep reversing while the Ultrasonic Sensor sees a value greater than or eq
22     }
23
24     //Once the Ultrasonic Sensor sees a value less than 7cm.
25     //Stop the robot for 1 second.
26     setMotorSpeed(motorB, 0);
27     setMotorSpeed(motorC, 0);
28     sleep(1000);
29
30     //You should now be straight and facing the other way. Drive off.
31     setMotorSpeed(motorB, 50);
32     setMotorSpeed(motorC, 50);
33     sleep(3000);
34 }
35
```

Данный код может быть изменен, в связи с регулярными обновлениями ROBOTC.

Занятие 1

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session1_3c

```
Activity1_3.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**//
3
4  /*
5  Create a three point turn using timing and steering.
6  Introducing the Ultrasonic Sensor to act as a parking sensor.
7  Play an alert before moving off.
8  */
9
10 task main()
11 {
12     //Turn to the right and stop after 1.5 seconds
13     setMotorSpeed(motorB, 75);
14     setMotorSpeed(motorC, 30);
15     sleep(1500);
16
17     //Reverse to the left while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
18     setMotorSpeed(motorB, -30);
19     setMotorSpeed(motorC, -75);
20     while(getUSDistance(sonarSensor) >= 7)
21     {
22         //Keep reversing while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
23         sleep(10);
24     }
25
26     //Once the Ultrasonic Sensor sees a value less than 7cm.
27     //Play sound alert.
28     //Wait for 1 second.
29     setMotorSpeed(motorB, 0);
30     setMotorSpeed(motorC, 0);
31     playTone(440, 100);
32
33     // Wait for the tone to be done playing
34     while(bSoundActive)
35     {
36         sleep(10);
37     }
38
39     sleep(1000);
40
41     //You should now be straight and facing the other way. Drive off.
42     setMotorSpeed(motorB, 50);
43     setMotorSpeed(motorC, 50);
44     sleep(3000);
45 }
46
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 2

Текстовые команды для разворота в три приема (текстовое программирование)



Занятие 2

Указания по занятию про разворот в три приема

На этом занятии будет повторяться материал занятия 1, но теперь обучающиеся перейдут от визуального программирования к текстовому программированию.

Обсудите с учениками различные среды программирования и причины, по которым существуют разные среды программирования.

Рассмотрите идею, что все люди разные, соответственно, им подходят разные способы обучения. Одни предпочитают текстовый язык, другие - визуальный. Обратите внимание учеников на то, что в течение последующих недель ученики должны будут писать текстовые программы, соответствующие составленным ими программам EV3.

Попросите учеников выполнить несколько простых заданий по перемещению робота, чтобы они до конца поняли принцип действия различных блоков движения и управления моторами.



ЗНАКОМСТВО УЧЕНИКОВ С ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ РОБОТС

Объясните, что программное обеспечение на языке ROBOTC имеет множество отличий.

Например, при программировании в ROBOTC необходимо указывать, какие датчики и моторы используются.

В основном на уроке повторяются задания, которые выполнялись на занятии 1. Таким образом, ученики должны написать алгоритм для выполнения разворота в три приема.



Занятие 2

План урока

РЕЗУЛЬТАТЫ

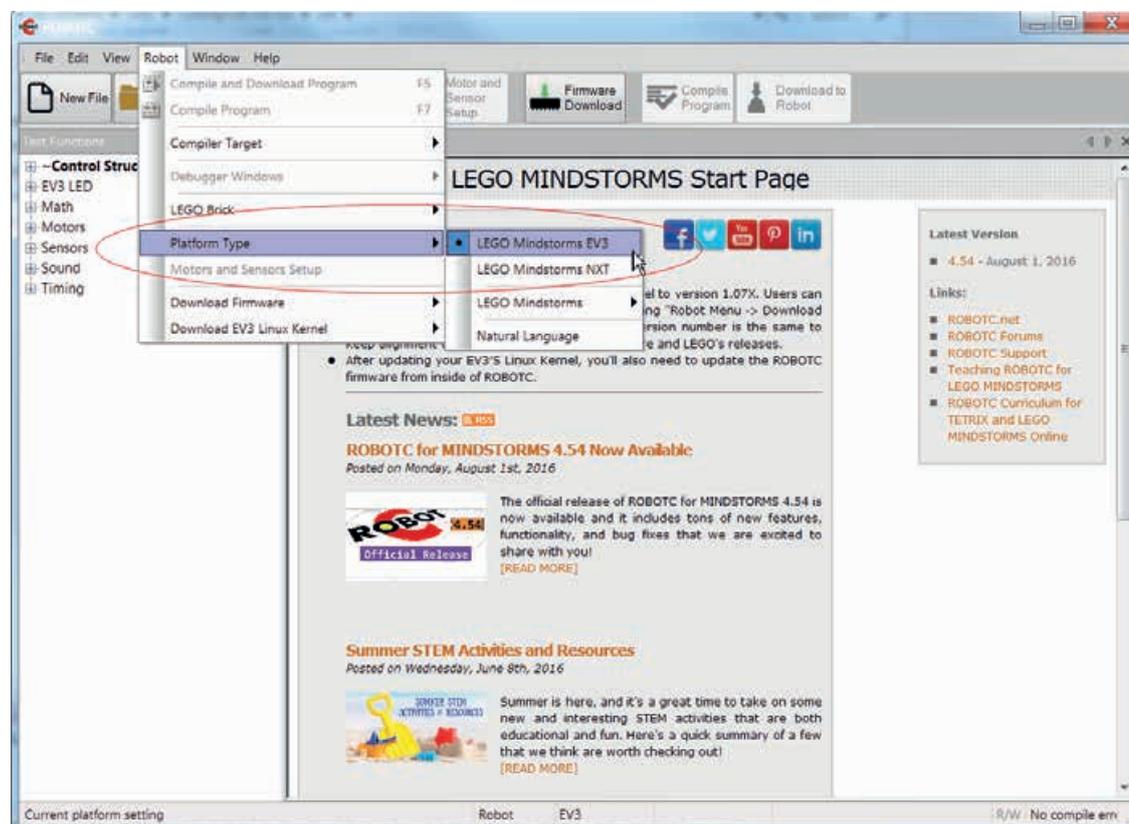
Ученики должны приобрести следующие знания и умения:

- применение программы ROBOTC для создания текстовых программ, аналогичных программам EV3, составленным на занятии 1;
- понимание важности синтаксиса, т.е. двоеточия, скобок и т.д.;
- умение организовать сессию ROBOTC;
- понимание, что с помощью алгоритмов можно выполнять определенную последовательность команд;
- сборка и программирование простого колесного робота для разворота на 180° в три приема;
- знакомство с командами "установить скорость мотора" (setMotorSpeed) и "сон" (Sleep).



ВВЕДЕНИЕ

- Напомните ученикам, что они делали на прошлом занятии. Они собирали колесного робота. Если они еще не делали этого, сконструируйте базовую модель, опираясь на бумажное пособие в базовом комплекте EV3.
- Покажите классу, как осуществлять доступ к программе ROBOTC. Покажите, где в среде ROBOTC следует выбрать платформу, которую нужно использовать. Объясните, что этой программой пользуются некоторые компании, выпускающие роботов и занимающиеся автоматизацией.



Занятие 2

План урока

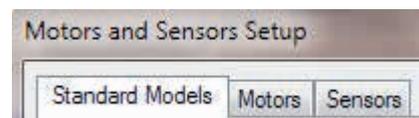
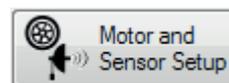
- Укажите, где она сообщает пользователю о том, какая платформа используется (как выделено выше). В случае другой платформы, для перехода на платформу EV3 следуйте указаниям на сайте ROBOTC. Необходимо выполнить следующие действия:-
 1. Измените тип платформы на LEGO® MINDSTORMS® EV3
 2. Обновите встроенное программное обеспечение EV3 (операционную систему), нажав на Download EV3 Linux Firmware (“Загрузить встроенное программное обеспечение Linux для EV3”), а затем на Standard File (“Стандартный файл”).
 3. Установите постоянное программное обеспечение ROBOTC, нажав на Download Firmware (“Обновление встроенного ПО”) и Standard File (“Стандартный файл”).

Установка ROBOTC для программирования

- Объясните, что в отличие от работы с программой EV3, в программе ROBOTC нужно указывать, какие датчики и моторы используются для данной сессии. Пользователи могут настроить программное обеспечение на соответствие аппаратному обеспечению двумя способами. Они заключаются в следующем:

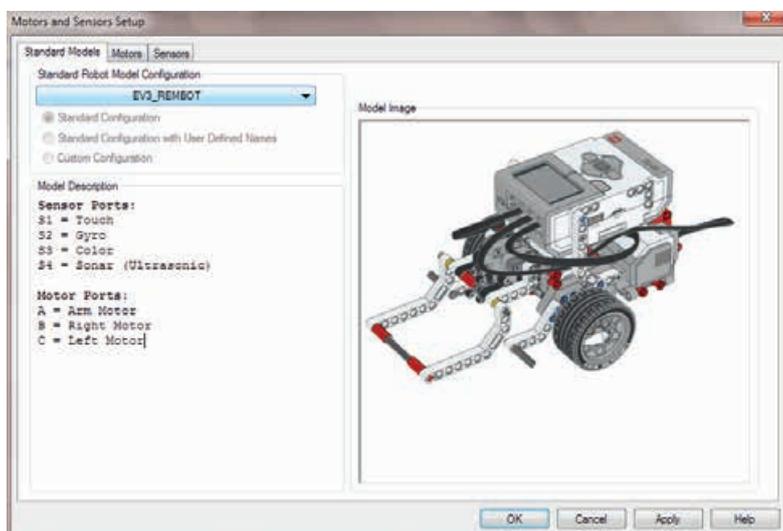
4. использование предварительно заданных моделей;
5. ручное конфигурирование программы.

- Сначала необходимо нажать на кнопку настройки мотора и датчиков Motor and Sensor Setup.
- Теперь пользователь может либо выбрать стандартную модель по вкладке Standard Models, либо настроить собственных роботов через вкладки Motors (“Моторы”) и Sensors (“Датчики”).



Использование стандартных моделей

- Нажмите на вкладку Standard Models (“Стандартные модели”).
- Затем нажмите на выпадающее меню и выберите EV3_REMBOТ. Это стандартная модель из учебника по робототехнике. После выбора появляется изображение модели из учебника по робототехнике, а также порты для датчиков и моторов.



Занятие 2

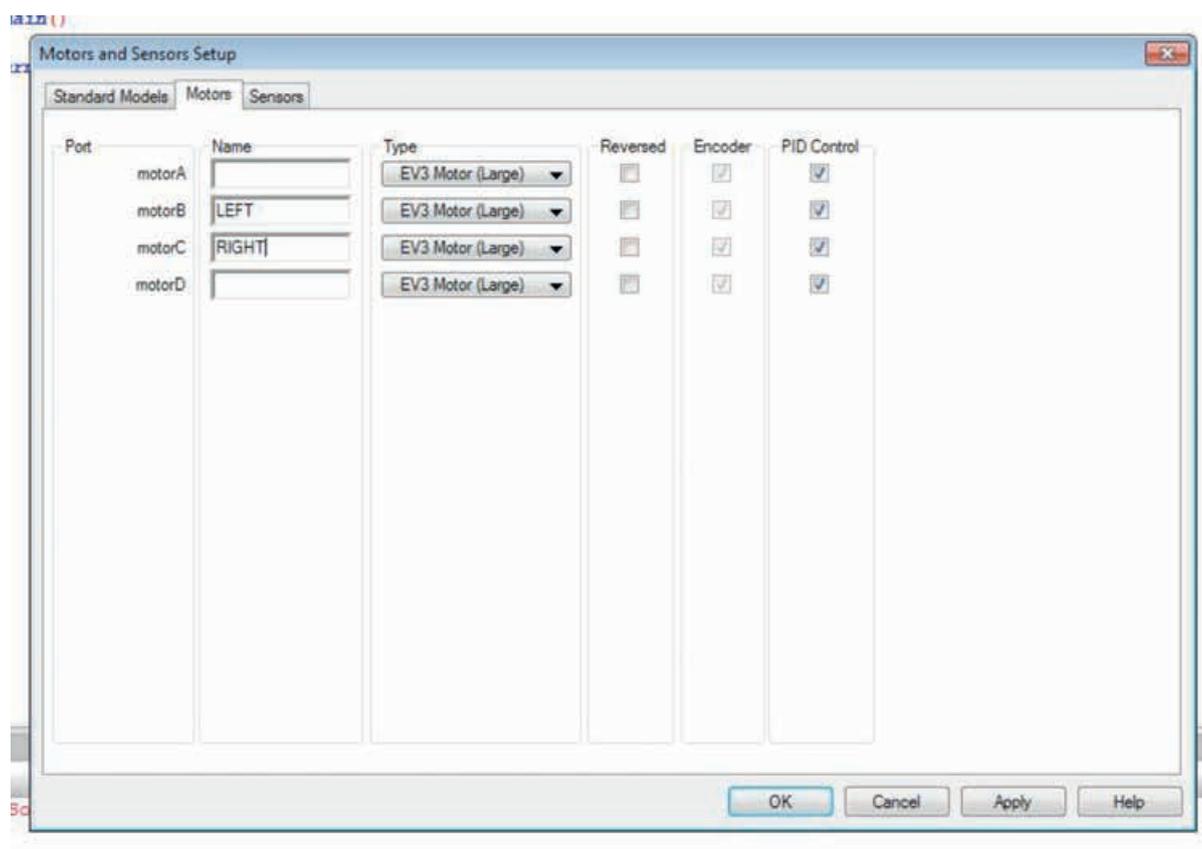
План урока

- Проверьте, отображается ли в пункте #pragma config в верхней части экрана программы значение EV3_REMBOT.

```
LEGO Start Page SourceFile002.c*
1 #pragma config(StandardModel, "EV3_REMBOT")
2 /**!!Code automatically generated by 'ROBOTC' configuration wizard      !!*/
3
4 task main()
5 {
6
7
8
9 }
```

Установка моторов и датчиков вручную

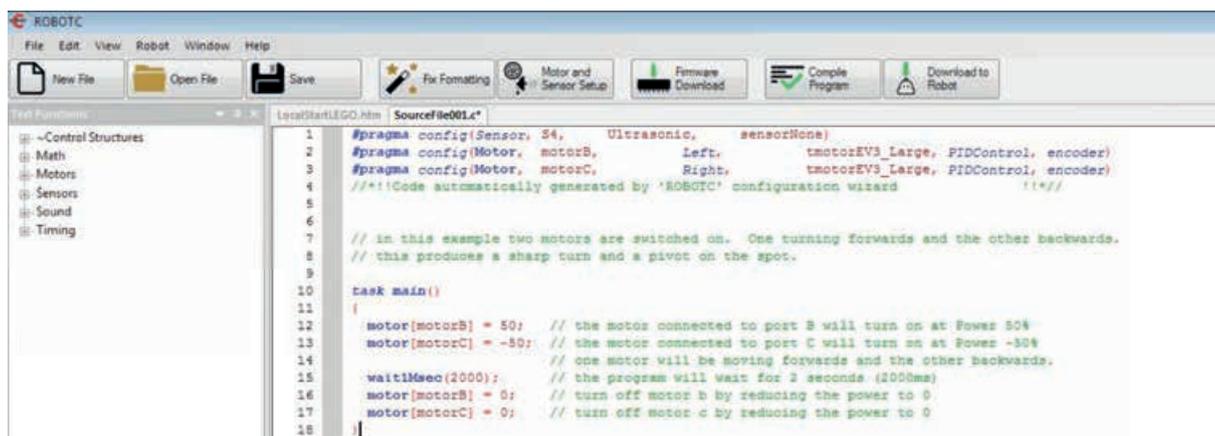
- Нажмите на вкладку Motor ("Мотор").
- Назовите мотор B LEFT ("ЛЕВЫЙ"), а мотор C RIGHT ("ПРАВЫЙ"), как показано ниже. Далее нажмите на ОК для подтверждения настройки.



Занятие 2

План урока

- Теперь можно продемонстрировать ученикам программирование простого поворота на основе текстового языка.
- В заключение поясните, что к программе можно делать примечания, добавив две косые черты, как показано ниже. Важно, чтобы ученики понимали каждую часть текстовой программы так же хорошо, как они понимают блоки в программном обеспечении EV3.



```
1 #pragma config(Sensor, S4, Ultrasonic, sensorNone)
2 #pragma config(Motor, motorB, Left, tmotorEV3_Large, PIDControl, encoder)
3 #pragma config(Motor, motorC, Right, tmotorEV3_Large, PIDControl, encoder)
4 /**Code automatically generated by 'ROBOTC' configuration wizard */
5
6
7 // in this example two motors are switched on. One turning forwards and the other backwards.
8 // this produces a sharp turn and a pivot on the spot.
9
10 task main()
11 {
12   motor[motorB] = 50; // the motor connected to port B will turn on at Power 50%
13   motor[motorC] = -50; // the motor connected to port C will turn on at Power -50%
14   // one motor will be moving forwards and the other backwards.
15   wait1Msec(2000); // the program will wait for 2 seconds (2000ms)
16   motor[motorB] = 0; // turn off motor b by reducing the power to 0
17   motor[motorC] = 0; // turn off motor c by reducing the power to 0
18 }
```



Занятие 2

План урока



ВВЕДЕНИЕ: ИССЛЕДОВАНИЕ

- Ученики в парах или в группах по трое собирают базовую модель. Этот этап уже может быть выполнен на занятии 1.
- По завершении сборки они должны начать новую программу и провести эксперименты с различными способами разворота. Ниже представлено три примера.

```
LEGO Start Page | Explore1.c | explore2.c | explore3.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  task main()
5  {
6      //motor c reverses at power 50
7      //motor b drives forward at power 50
8      //robot will pivot centrally for 1 second (1000ms).
9      setMotorSpeed(motorB, 50);
10     setMotorSpeed(motorC, -50);
11     sleep(1000);
12 }
```

Пример 1: разворот вокруг центральной оси.

```
LEGO Start Page | Explore1.c | explore2.c | explore3.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  task main()
5  {
6      //motor c drives at power 50
7      //robot will pivot on stationary wheel for 1 second (1000ms).
8      setMotorSpeed(motorC, 50);
9      sleep(1000);
10 }
```

Пример 2: разворот на одном колесе.

```
LEGO Start Page | Explore1.c | explore2.c | explore3.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  task main()
5  {
6      //motor b drives forward at power 75
7      //motor c drives forward at power 50
8      //robot will turn gently for 1 second (1000ms).
9      setMotorSpeed(motorB, 75);
10     setMotorSpeed(motorC, 50);
11     sleep(1000);
12 }
```

Пример 3: медленный разворот по дуге.

Занятие 2

План урока

ОСНОВНОЕ ЗАДАНИЕ 1

- Попросите учеников рассказать о том, какие способы они применяли для разворота робота. Пусть они объяснят, как пришли к такому решению.
- Поясните, что они должны будут использовать то, что они узнали из наблюдений за разворотом модели, для моделирования разворота в три приема.
- Можно показать классу видео о выполнении разворота в три приема.
- Далее обучающиеся должны создать программу разворота в три приема на языке ROBOTC, моделируя рулевое управление посредством установки разных значений мощности для каждого мотора.
- Напомните детям о необходимости постоянного анализа и отладки созданной ими программы, чтобы робот мог развернуться на 180 градусов.
- Примечание: Можно при помощи клейкой ленты обозначить “дорожную разметку”, чтобы дети научились разворачивать робота в ограниченном пространстве.
- Примечание: В предлагаемых программных решениях перед заключительным этапом робот переходит в режим прямолинейного движения перед тем, как тронуться. Однако в реальности это не всегда возможно, соответственно, обучающиеся могут включить движение по дуге перед переходом к прямолинейному движению. Напомните детям, что у таких задач нет неправильных решений.
- Предложите ученикам сравнить два подхода к решению одной задачи. Какой из них легче и эффективнее?



Занятие 2

План урока

ВОЗМОЖНОЕ РЕШЕНИЕ

ИМЯ ФАЙЛА: SESSION2_1.C

```
Activity2_1.c
1  #pragma config(StandardModel, "EV3_REMBO")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard!!*/
3
4  /*
5  Create a three point turn using timing and steering.
6  */
7
8  task main()
9  {
10     //Turn to the right and stop after 1.5 seconds.
11     setMotorSpeed(motorB, 75);
12     setMotorSpeed(motorC, 30);
13     sleep(1500);
14
15     //Reverse to the left and stop after 1 second.
16     setMotorSpeed(motorB, -30);
17     setMotorSpeed(motorC, -75);
18     sleep(1000);
19
20     //You should now be straight and facing the other way. Drive off.
21     setMotorSpeed(motorB, 50);
22     setMotorSpeed(motorC, 50);
23     sleep(3000);
24 }
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.



Занятие 2

План урока

ОСНОВНОЕ ЗАДАНИЕ 2

- Расскажите о применении ультразвукового датчика. Покажите классу, как добавить еще один датчик в конфигурацию. Для добавления нужно просто войти в меню Motors and Sensors Set up ("Настройка моторов и датчиков").
- Спросите класс, каким образом можно сделать программу более автономной, и как можно регистрировать все препятствия, возникающие в процессе движения колесного робота задним ходом.
- Продемонстрируйте команду проверки условия (While) и как ее можно использовать с ультразвуковым датчиком. Обратите внимание класса на настройки мотора в составе цикла с проверкой условия и настройки мотора вне цикла с проверкой условия.
- Ученики должны усовершенствовать программу таким образом, чтобы колесный робот останавливался в заданной точке в ответ на сигнал ультразвукового датчика.
- Поясните, что их программы имитируют действия водителя, нажимающего на тормоза, когда сдающий назад автомобиль приближается к препятствию.
- Предложите ученикам сравнить два подхода к решению одной задачи. Какой из них легче и эффективнее?



Занятие 2

План урока

ВОЗМОЖНОЕ РЕШЕНИЕ

ИМЯ ФАЙЛА: SESSION2_2.C

```
Activity2_2.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  /*
5  Create a three point turn using timing and steering.
6  Introducing the Ultrasonic Sensor to act as parking sensors.
7  */
8
9  task main()
10 {
11     //Turn to the right and stop after 1.5 seconds.
12     setMotorSpeed(motorB, 75);
13     setMotorSpeed(motorC, 30);
14     sleep(1500);
15
16     //Reverse to the left while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
17     setMotorSpeed(motorB, -30);
18     setMotorSpeed(motorC, -75);
19     while(getUSDistance(sonarSensor) >= 7)
20     {
21         //Keep reversing while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
22     }
23
24     //Once the Ultrasonic Sensor sees a value less than 7cm.
25     //Stop the robot for 1 second.
26     setMotorSpeed(motorB, 0);
27     setMotorSpeed(motorC, 0);
28     sleep(1000);
29
30     //You should now be straight and facing the other way. Drive off.
31     setMotorSpeed(motorB, 50);
32     setMotorSpeed(motorC, 50);
33     sleep(3000);
34 }
35
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.



Занятие 2

План урока

ОСНОВНОЕ ЗАДАНИЕ 3

- Добавьте колесному роботу функции безопасности на основе ультразвукового датчика (например, предупреждающие звуковые сигналы).
- Спросите класс, что происходит с автомобилем, когда он сдает назад и приближается к препятствию – можно слышать предупреждающий сигнал.
- Расскажите о команде звукового сигнала, с помощью которой можно добавить предупреждающий сигнал после остановки колесного робота.
Продемонстрируйте ученикам, как работает команда звукового сигнала и где она должна быть в программе.
- Ученики должны изменить составленные ими программы, включив в них команду звукового сигнала таким образом, чтобы она инициировала предупреждающий сигнал, когда колесный робот оказывается на определенном расстоянии от препятствия.
- Объясните ученикам целесообразность использования задержки от 1 до 10 мс в цикле ожидания.
- Желательно, чтобы они постоянно отлаживали программы, пока звуковая сигнализация и остановка на подходящем расстоянии не заработают достаточно эффективно.



Занятие 2

План урока

ВОЗМОЖНОЕ РЕШЕНИЕ ИМЯ ФАЙЛА: SESSION2_3.C

```
Activity2_3.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  /*
5  Create a three point turn using timing and steering.
6  Introducing the Ultrasonic Sensor to act as a parking sensor.
7  Play an alert before moving off.
8  */
9
10 task main()
11 {
12     //Turn to the right and stop after 1.5 seconds
13     setMotorSpeed(motorB, 75);
14     setMotorSpeed(motorC, 30);
15     sleep(1500);
16
17     //Reverse to the left while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
18     setMotorSpeed(motorB, -30);
19     setMotorSpeed(motorC, -75);
20     while(getUSDistance(sonarSensor) >= 7)
21     {
22         //Keep reversing while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
23         sleep(10);
24     }
25
26     //Once the Ultrasonic Sensor sees a value less than 7cm.
27     //Play sound alert.
28     //Wait for 1 second.
29     setMotorSpeed(motorB, 0);
30     setMotorSpeed(motorC, 0);
31     playTone(440, 100);
32
33     // Wait for the tone to be done playing
34     while(bSoundActive)
35     {
36         sleep(10);
37     }
38
39     sleep(1000);
40
41     //You should now be straight and facing the other way. Drive off.
42     setMotorSpeed(motorB, 50);
43     setMotorSpeed(motorC, 50);
44     sleep(3000);
45 }
46
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ПОДВЕДЕНИЕ ИТОГОВ

- Перечислите основные моменты, которые были усвоены на данном уроке.
- Убедитесь, что ученики понимают термины, использовавшиеся в ходе занятия, и могут ими оперировать, повторите значения основных слов.
- Попросите одну-две группы продемонстрировать свои программы. Обсудите, что получилось и над чем еще можно поработать.
- Предложите оценить текстовые программы. Кто предпочел бы использовать текстовый язык в качестве основного языка программирования?

Занятие 2

Рабочие карточки учеников



ЗАДАЧИ НА СЕГОДНЯ

Цель сегодняшнего занятия - познакомить вас с программным обеспечением ROBOTC и помочь вам начать работу с ним.

Вы уже смогли немного поэкспериментировать с командой “установить скорость мотора” (`setMotorSpeed`), заставляя робота перемещаться по классу. Теперь, опираясь на этот опыт, вы должны решить три задачи.

Удачи!

ЗАДАЧА 1

Запрограммируйте колесного робота на разворот в три приема.

Робот должен выполнить поворот при движении вперед, затем сдать назад и снова двигаться вперед.

Посмотрите видео еще раз, чтобы вспомнить, как выглядит такой разворот. Следите за тем, чтобы не пересечь дорожную разметку!

Используемые команды

`setMotorSpeed` (установить скорость мотора) **`sleep`** (сон)

Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 2

Рабочие карточки учеников

ЗАДАЧА 2

Сейчас вы должны провести эксперимент с одним из датчиков EV3 – ультразвуковым датчиком.

Запрограммируйте колесного робота на разворот в три приема, используя ультразвуковой датчик в качестве парковочного радара, чтобы робот при движении задним ходом останавливался на заданном расстоянии от препятствия.

Может ли ваш робот затормозить перед тем, как тронуться в новом направлении?

Здесь вам потребуются знания о команде проверки условия (While), также нужно установить ультразвуковой датчик на заднем торце робота.



Используемые команды

setMotorSpeed (установить скорость мотора) **sleep** (сон) **while**
(проверка условия) **getUSDistance** (измерить расстояние УЗ)

Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 2

Рабочие карточки учеников

ЗАДАЧА 3

Моделирование предупреждающих звуковых сигналов.

Что часто происходит, когда автомобиль сдает назад и приближается к препятствию?

Теперь ваш колесный робот останавливается по сигналу ультразвукового “датчика парковки”. Можете ли вы усовершенствовать программу таким образом, чтобы робот издавал предупреждающий сигнал непосредственно перед включением тормозов при заднем ходе?

Программу необходимо постоянно отлаживать, чтобы предупреждающий сигнал выключался одновременно с остановкой робота. Какую часть программы нужно изменить?

Используемые команды

setMotorSpeed (установить скорость мотора) **sleep** (сон) **getUSDistance**
(измерить расстояние УЗ) **playTone** (подать звуковой сигнал)

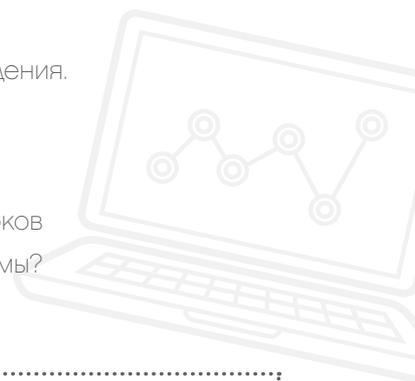
Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 2

Рабочие карточки учеников

По завершении занятия по программированию важно записать свои мысли и наблюдения. Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?



Мысли и наблюдения

Занятие 2

Приложение

ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 2:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 2

Приложение



Занятие 2

Приложение



Занятие 2

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session2_1.c

```
Activity2_1.c
1  #pragma config(StandardModel, "EV3_REMOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard!!**/
3
4  /*
5  Create a three point turn using timing and steering.
6  */
7
8  task main()
9  {
10     //Turn to the right and stop after 1.5 seconds.
11     setMotorSpeed(motorB, 75);
12     setMotorSpeed(motorC, 30);
13     sleep(1500);
14
15     //Reverse to the left and stop after 1 second.
16     setMotorSpeed(motorB, -30);
17     setMotorSpeed(motorC, -75);
18     sleep(1000);
19
20     //You should now be straight and facing the other way. Drive off.
21     setMotorSpeed(motorB, 50);
22     setMotorSpeed(motorC, 50);
23     sleep(3000);
24 }
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 2

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session2_2.c

```
Activity2_2.c
1  #pragma config(StandardModel, "EV3_REMBO")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  /*
5  Create a three point turn using timing and steering.
6  Introducing the Ultrasonic Sensor to act as parking sensors.
7  */
8
9  task main()
10 {
11     //Turn to the right and stop after 1.5 seconds.
12     setMotorSpeed(motorB, 75);
13     setMotorSpeed(motorC, 30);
14     sleep(1500);
15
16     //Reverse to the left while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
17     setMotorSpeed(motorB, -30);
18     setMotorSpeed(motorC, -75);
19     while(getUSDistance(sonarSensor) >= 7)
20     {
21         //Keep reversing while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
22     }
23
24     //Once the Ultrasonic Sensor sees a value less than 7cm.
25     //Stop the robot for 1 second.
26     setMotorSpeed(motorB, 0);
27     setMotorSpeed(motorC, 0);
28     sleep(1000);
29
30     //You should now be straight and facing the other way. Drive off.
31     setMotorSpeed(motorB, 50);
32     setMotorSpeed(motorC, 50);
33     sleep(3000);
34 }
35
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 2

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session2_3.c

```
Activity2_3.c
1  #pragma config(StandardModel, "EV3_REMOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
3
4  /*
5  Create a three point turn using timing and steering.
6  Introducing the Ultrasonic Sensor to act as a parking sensor.
7  Play an alert before moving off.
8  */
9
10 task main()
11 {
12     //Turn to the right and stop after 1.5 seconds
13     setMotorSpeed(motorB, 75);
14     setMotorSpeed(motorC, 30);
15     sleep(1500);
16
17     //Reverse to the left while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
18     setMotorSpeed(motorB, -30);
19     setMotorSpeed(motorC, -75);
20     while(getUSDistance(sonarSensor) >= 7)
21     {
22         //Keep reversing while the Ultrasonic Sensor sees a value greater than or equal to 7cm.
23         sleep(10);
24     }
25
26     //Once the Ultrasonic Sensor sees a value less than 7cm.
27     //Play sound alert.
28     //Wait for 1 second.
29     setMotorSpeed(motorB, 0);
30     setMotorSpeed(motorC, 0);
31     playTone(440, 100);
32
33     // Wait for the tone to be done playing
34     while(bSoundActive)
35     {
36         sleep(10);
37     }
38
39     sleep(1000);
40
41     //You should now be straight and facing the other way. Drive off.
42     setMotorSpeed(motorB, 50);
43     setMotorSpeed(motorC, 50);
44     sleep(3000);
45 }
46
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 3

Движение робота задним ходом



Занятие 3

Движение робота задним ходом

На этом занятии дети научатся программировать робота таким образом, чтобы он подавал визуальные сигналы пешеходам, другим водителям и пассажирам при движении задним ходом.

Используя блок светового и графического отображения состояния интеллектуального блока, можно создавать программы, которые будут визуально обозначать действия робота.

Также с помощью датчика касания можно имитировать передачи переднего и заднего хода.



Занятие 3

План урока

РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание, что с помощью алгоритмов можно выполнять определенную последовательность команд;
- направление колесного робота по прямой линии при помощи блока движения и рулевого управления;
- использование блока ожидания для программирования датчиков касания;
- использование функций светового и графического отображения состояния на интеллектуальном блоке;
- развитие алгоритмического мышления путем создания более сложных алгоритмов.



ОСНОВНЫЕ ПОНЯТИЯ

Ввод, вывод, алгоритм, ожидание, датчик касания, наладка, блок движения и рулевого управления.

ВВЕДЕНИЕ

- Объясните ученикам, что в ходе урока они будут программировать колесного робота для имитации заднего хода с задействованием фар и индикаторов приборной панели, а также познакомятся с еще одним датчиком – датчиком касания.
- Для начала они должны кратко обсудить, что происходит, когда автомобиль движется задним ходом. Обратите внимание класса на использование фар заднего хода для предупреждения других водителей и пешеходов. Помимо этого для водителя и пассажиров автомобиля предусмотрены индикаторы на приборной панели, например, показывающие, какая передача включена, то есть в каком направлении будет движение.
- При помощи программы EV3 продемонстрируйте три разных режима работы датчика касания (нажатие, освобождение, удар) и блок световой индикации состояния интеллектуального блока.



Занятие 3

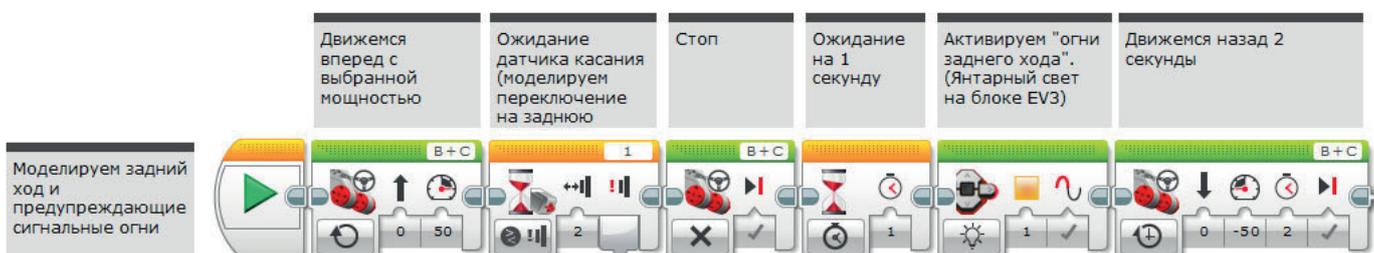
План урока

РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание, что с помощью алгоритмов можно выполнять определенную последовательность команд;
- направление колесного робота по прямой линии при помощи блока движения и рулевого управления;
- использование блока ожидания для программирования датчиков касания;
- использование функций светового и графического отображения состояния на интеллектуальном блоке;
- развитие алгоритмического мышления путем создания более сложных алгоритмов.

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 3 ВКЛАДКА: MAIN 1



ОСНОВНОЕ ЗАДАНИЕ 2

- Предложите применить в качестве "главной передачи" еще один датчик касания.
- Продемонстрируйте классу возможность использования дополнительного датчика касания, а также функцию программного обеспечения MINDSTORMS, позволяющую автоматически обнаруживать, к какому входу он подключен.
- Дети расширяют полученные ранее знания, составляя программы для имитации передачи переднего и заднего хода. Второй датчик касания ("главная передача") служит для запуска программы.

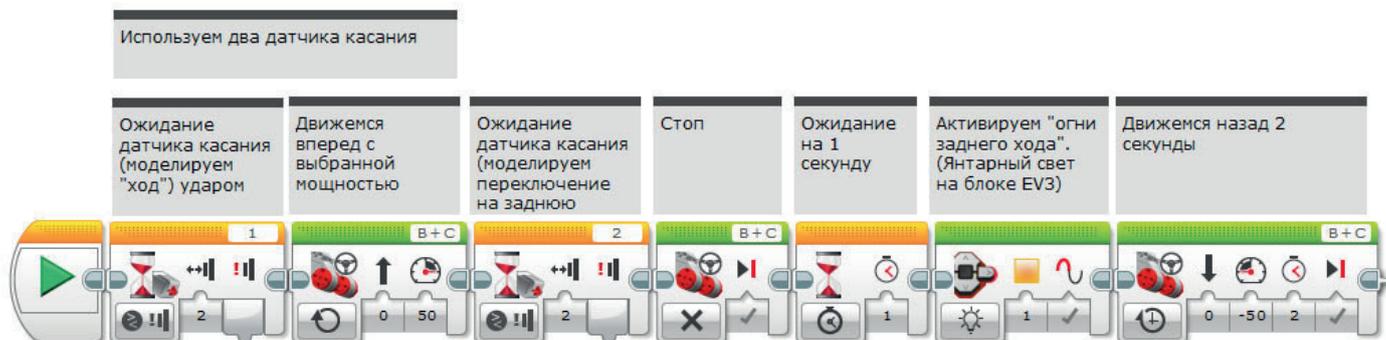


Занятие 3

План урока



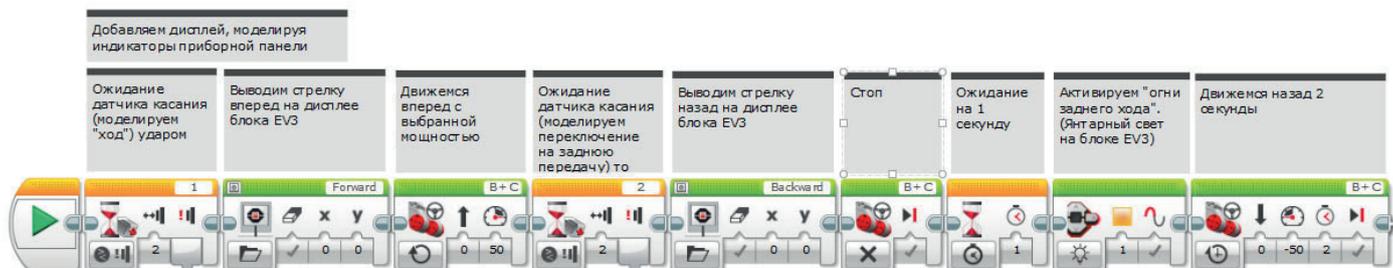
ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 3 ВКЛАДКА: MAIN 2



ОСНОВНОЕ ЗАДАНИЕ 3

- Использование дисплея интеллектуального блока EV3 как индикатора приборной панели.
- Необходимо далее усовершенствовать составленные программы таким образом, чтобы задействовать дисплей интеллектуального блока EV3. Данная функция добавляется через блок управления дисплеем. При движении колесного робота передним или задним ходом на экране должны отображаться стрелки, показывающие направление движения.
- Продемонстрируйте классу блок управления движением.
- При желании дети могут исследовать функции блока управления движением и настроить экраны в соответствии с особенностями своих программ.
- Каким образом, по мнению класса, можно использовать экран в автоматизированном автомобиле? Помогите ученикам догадаться, что экран может показывать важную информацию для пассажиров самоходного автомобиля.

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 3 ВКЛАДКА: MAIN 3



Занятие 3

План урока

ПОДВЕДЕНИЕ ИТОГОВ

- Попросите одну-две группы продемонстрировать составленные программы и движение колесных роботов.
- Сколько всего вариантов придумала группа? Это полезное упражнение, способствующее пониманию того, что задача может иметь множество решений.
- Сообщите ученикам, что в следующий раз они будут экспериментировать с другим датчиком (цвета) для совершенствования системы автоматизации колесного робота.



Занятие 3

Рабочие карточки

ЗАДАЧИ НА СЕГОДНЯ

Задачи сегодняшнего урока потребуют от вас использования и углубления полученных ранее знаний по программированию. Вы должны использовать еще один датчик (датчик касания), а также функции интеллектуального блока EV3. Интеллектуальный блок нужно запрограммировать так, чтобы активировать его световые сигналы и использовать экран для визуальной индикации.

К концу решения третьей задачи под управлением созданных программ робот будет имитировать передний и задний ход с включением фар заднего хода и индикаторов приборной панели.

ЗАДАЧА 1

Составьте программу, которая заставит колесного робота двигаться вперед и сдавать назад при нажатии на датчик касания.

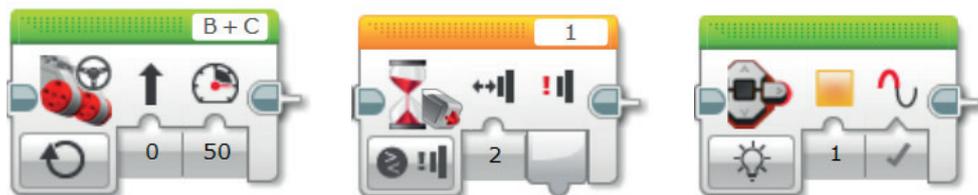
Испытайте составленную программу, затем усложните ее:

Какие сигналы вы видите снаружи машин, когда они сдают назад, помогающие пешеходам и другим участникам дорожного движения понимать, что происходит?

Ваш робот должен включать предупреждающие фары заднего хода.

Сымитируйте работу фар заднего хода при помощи интеллектуального блока EV3 и светового индикатора статуса.

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 3

Рабочие карточки

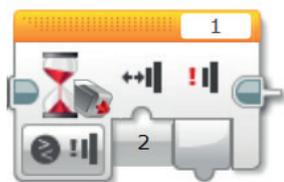
ЗАДАЧА 2

Можете ли вы усовершенствовать программу таким образом, чтобы у колесного робота было две передачи (переднего и заднего хода)? Робот должен “трогаться” (начинать движение вперед) при активации “главной передачи”.

Полезная информация: вам потребуется второй датчик касания.

Используемые блоки

Используйте те же блоки, что и в задаче на программирование 1, но также не забывайте об этом блоке:



Начните с плана программы. Запишите ее ниже в виде

Занятие 3

Рабочие карточки

ЗАДАЧА 3

Что происходит внутри автомобиля при переключении передач?

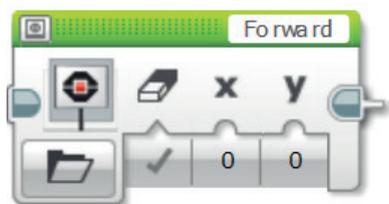
Часто на приборной панели присутствуют индикаторы/изображения, сообщающие водителю, какая передача действует в данный момент. Можете ли вы симитировать такие сигналы в вашей программе, используя блок управления дисплеем?

Внимательно изучите блок управления дисплеем и найдите подходящие изображения для обозначения движения вперед и назад.

Новая программа должна быть расширенной версией предыдущей программы и по-прежнему включать работу фар заднего хода.

Используемые блоки

Используйте те же блоки, что и в задачах на программирование 1 и 2, но также не забывайте о следующем блоке:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 3

Рабочие карточки

По завершении занятия по программированию важно записать свои мысли и наблюдения. Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?



Мысли и наблюдения

Занятие 3

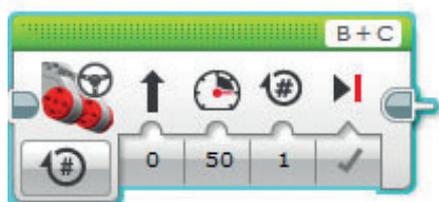
Примечания для учителя

РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

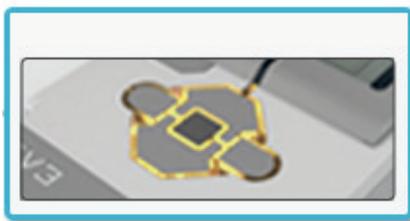
Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами.

НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Основы > Движение по прямой



Аппаратное обеспечение > Световой индикатор состояния интеллектуального блока



Аппаратное обеспечение > Датчик касания



Аппаратное обеспечение > Дисплей интеллектуального блока



РАНЕЕ РАССМОТРЕННЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Основы > Остановка у объекта



Занятие 3

Приложение

ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 3:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 3

Приложение



Занятие 3

Приложение



Занятие 3

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 3
ВКЛАДКА: MAIN 1

Моделируем задний ход и предупреждающие сигнальные огни	
Двигаемся вперед с выбранной мощностью	
Ожидание датчика касания (моделируем переключение на заднюю)	
Стоп	
Ожидание на 1 секунду	
Активируем "огни заднего хода" (Яртарный свет на блоке EV3)	
Двигаемся назад 2 секунды	

Занятие 3

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 3
ВКЛАДКА: MAIN 2

Используем два датчика касания	
Ожидание датчика касания ("ход") ударом	1
Движемся вперед с выбранной мощностью	50
Ожидание датчика касания (моделируем переключение на заднюю)	2
Стоп	X
Ожидание на 1 секунду	1
Активируем "огни заднего хода" (Янтарный свет на блоке EV3)	1
Движемся назад 2 секунды	0 -50 2



Занятие 3

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 3
ВКЛАДКА: MAIN 3

The image displays a sequence of programming blocks in the LEGO Mindstorms EV3 software interface. The blocks are arranged vertically and include the following instructions:

- Добавляем датчик, моделируя индикаторы приборной панели.** (Add sensor, simulating dashboard indicators.)
- Ожидание датчика касания ("ход") ударом** (Waiting for touch sensor ("step") hit)
- Выводим стрелку вперед на дисплее блока EV3** (Output arrow forward on EV3 block display)
- Движется вперед с выбранной мощностью** (Moves forward with selected power)
- Ожидание датчика касания (моделируем переключение на заднюю передачу) TO** (Waiting for touch sensor (simulating gear shift to reverse) TO)
- Выводим стрелку назад на дисплее блока EV3** (Output arrow backward on EV3 block display)
- Стоп** (Stop)
- Ожидание на 1 секунду** (Waiting for 1 second)
- Активируем "огни заднего хода" (Яркий свет на блоке EV3)** (Activate "reverse lights" (Bright light on EV3 block))
- Движемся назад 2 секунды** (Move backward 2 seconds)

Each block is accompanied by a graphical representation of the EV3 brick with the corresponding function block highlighted. The interface also shows a toolbar with various icons for navigation and execution.

Занятие 3

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session3_1.c

```
Activity3_1.c
1  #pragma config(StandardModel, "EV3_REMBO")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard      !!**/
3
4  /*
5  Create a program where the robot drives forward until the Touch Sensor is pressed.
6  The robot then reverses flashing an orange LED.
7  */
8
9  task main()
10 {
11     //Set the MotorSpeed to 50%.
12     setMotorSpeed(motorB, 50);
13     setMotorSpeed(motorC, 50);
14
15     //Wait for touch sensor to be pressed.
16     while(getTouchValue(touchSensor) ==0)
17     {
18         sleep(10);
19     }
20
21     //Set the MotorSpeed to 0% or off. Wait 1 second.
22     setMotorSpeed(motorB, 0);
23     setMotorSpeed(motorC, 0);
24     sleep(1000);
25
26     //Flash the EV3 LED Orange.
27     setLEDColor(ledOrangeFlash);
28
29     //Set the MotorSpeed to -50%(reverse). Wait 2 seconds.
30     setMotorSpeed(motorB, -50);
31     setMotorSpeed(motorC, -50);
32     sleep(2000);
33 }
34
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 3

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session3_2.c

```
Activity3_2.c
1  #pragma config(Sensor, S1, touchSensor1, sensorEV3_Touch)
2  #pragma config(Sensor, S2, touchSensor2, sensorEV3_Touch)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorA, armMotor, tmotorEV3_Large, PIDControl, encoder)
6  #pragma config(Motor, motorB, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  #pragma config(Motor, motorC, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
8  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
9
10 /*
11 Create a program that starts the robot on the press of the Touch Sensor. The robot
12 then drives forward until a second Touch Sensor is pressed touched. The robot then reverses
13 flashing an orange LED.
14 */
15
16 task main()
17 {
18 //Wait for Touch Sensor 1 to be pressed.
19 while(getTouchValue(touchSensor1) ==0)
20 {
21 sleep(10);
22 }
23
24 //Set the MotorSpeed to 50%.
25 setMotorSpeed(motorB, 50);
26 setMotorSpeed(motorC, 50);
27
28 //Wait for Touch Sensor to be pressed.
29 while(getTouchValue(touchSensor2) ==0)
30 {
31 sleep(10);
32 }
33
34 //Set the MotorSpeed to 0% or off. Wait 1 second.
35 setMotorSpeed(motorB, 0);
36 setMotorSpeed(motorC, 0);
37 sleep(1000);
38
39 //Flash the EV3 LED Orange.
40 setLEDColor(ledOrangeFlash);
41
42 //Set the MotorSpeed to -50%(reverse). Wait 2 seconds.
43 setMotorSpeed(motorB, -50);
44 setMotorSpeed(motorC, -50);
45 sleep(2000);
46 }
47
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 3

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session3_3.c

```
Activity3_3.c
1  #pragma config(Sensor, S1, touchSensor1, sensorEV3_Touch)
2  #pragma config(Sensor, S2, touchSensor2, sensorEV3_Touch)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorA, armMotor, tmotorEV3_Large, PIDControl, encoder)
6  #pragma config(Motor, motorB, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  #pragma config(Motor, motorC, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
8  /*!!Code automatically generated by 'ROBOTC' configuration wizard !!*/
9
10 /*
11 Create a program that communicates messages via the EV3 screen.
12 Users develop the program from the previous activity.
13 NOTE: ***THIS PROGRAM USES TEXT INSTEAD OF IMAGES***
14 */
15
16 task main()
17 {
18
19 //Wait for Touch Sensor 1 to be pressed.
20 while(getTouchValue(touchSensor1) ==0)
21 {
22 sleep(10);
23 }
24
25 //Display the word "forward" on the EV3 screen.
26 displayCenteredBigTextLine(4, "forward");
27
28 //Set the MotorSpeed to 50%
29 setMotorSpeed(motorB, 50);
30 setMotorSpeed(motorC, 50);
31
32 //Wait for Touch Sensor 2 to be pressed.
33 while(getTouchValue(touchSensor2) ==0)
34 {
35 sleep(10);
36 }
37
38 //Set the MotorSpeed to 0% or off. Wait 1 second.
39 //Display the word "reverse" on the EV3 screen.
40 displayCenteredBigTextLine(4, "reverse");
41 setMotorSpeed(motorB, 0);
42 setMotorSpeed(motorC, 0);
43
44 sleep(1000);
45
46 //Flash the EV3 LED Orange.
47 setLEDColor(ledOrangeFlash);
48
49 //Set the MotorSpeed to -50%(reverse). Wait 2 seconds.
50 setMotorSpeed(motorB, -50);
51 setMotorSpeed(motorC, -50);
52 sleep(2000);
53 }
54
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 4

Освещение пути



Занятие 4

Освещение пути

Обучающиеся ознакомятся с работой датчика цвета и научатся имитировать автоматические функции автомобиля с помощью различных режимов (распознавание цвета и освещенности).

Их программы станут более сложными за счет включения датчиков, реагирующих на изменения окружающей среды.

Дети должны составить программы, имитирующие автоматическую работу фар, а затем включить в них функцию ручной блокировки фар.



Занятие 4

План урока

РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание, что с помощью алгоритмов можно выполнять определенную последовательность команд;
- знание простой булевой логики и некоторых вариантов ее применения;
- использование блока ожидания для программирования датчика цвета;
- понимание того, что датчик цвета имеет несколько функций и может измерять и реагировать на ряд параметров;
- расширение применения датчика и понимание принципов работы блока управления дисплеем;
- понимание принципов работы блока цикла и многозадачность / параллельное программирование.



ОСНОВНЫЕ ПОНЯТИЯ

Вход, выход, алгоритм, ожидание, датчик цвета, наладка, освещенность, цикл, булева логика, параллельное программирование.

ВВЕДЕНИЕ

- Объясните ученикам, что в ходе двух последующих занятий они будут изучать различные функции датчика цвета, а также программировать колесного робота на имитацию работы автоматических фар. Помимо этого, датчик цвета поможет им с делать робота более автономным.
- Поясните, что во время занятий класс будет наблюдать за тем, как изменения освещенности вызывают определенные реакции программы, и что это пример булевой логики (управление программой через принятие решений).
- Попросите привести примеры ситуаций из реальной жизни, когда изменения освещенности приводят к определенным действиям (например, автоматическое включение фар автомобиля или уличных фонарей).
- Сообщите, что далее обучающиеся будут знакомиться с работой цикла и искать способы его реализации в составленных ими программах.



Занятие 4

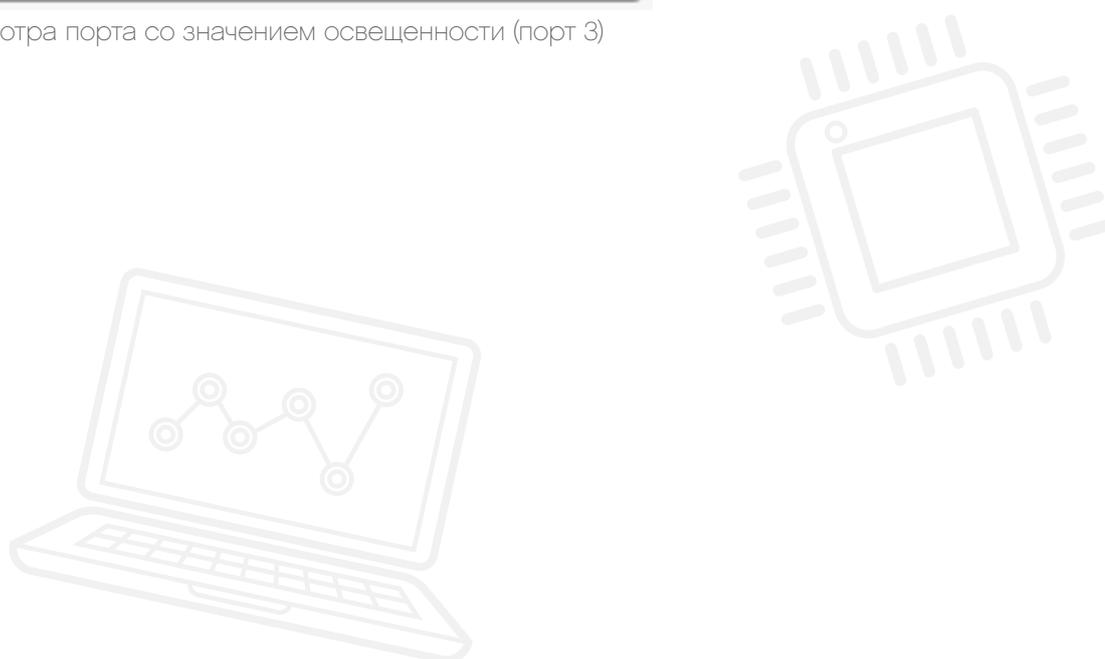
План урока

РЕЗУЛЬТАТЫ

- Расскажите о датчике цвета и продемонстрируйте классу, что он может работать в нескольких режимах:
 - режим цвета – распознавание цветов системы LEGO;
 - освещенность – распознавание интенсивности окружающего света;
 - интенсивность отраженного света, позволяющая роботу следовать по заданному маршруту.
- Спросите, какие функции автомобиля могли бы реагировать на изменение освещенности. Автоматические фары. Когда света на улице становится очень мало, включаются фары.
- Ученики составляют программы, имитирующие этот процесс, используя датчик цвета в режиме сравнения освещенности и дисплей интеллектуального блока EV3, который будет обозначать включение фар (для этого можно выбрать, например, изображение лампочки).
- Покажите классу функцию просмотра порта в программе MINDSTORMS® EV3, с помощью которой можно считывать показатели освещенности.
- Обучающиеся должны измерить освещенность в комнате и решить, при каком уровне освещенности будет включаться дисплей.
- Если ученики испытывают трудности при работе с датчиком цвета, можно напомнить им об инструкциях по сборке из учебника по робототехнике: мобильная база с датчиком цвета и функцией движения вперед. Также они могут придумать собственное решение.



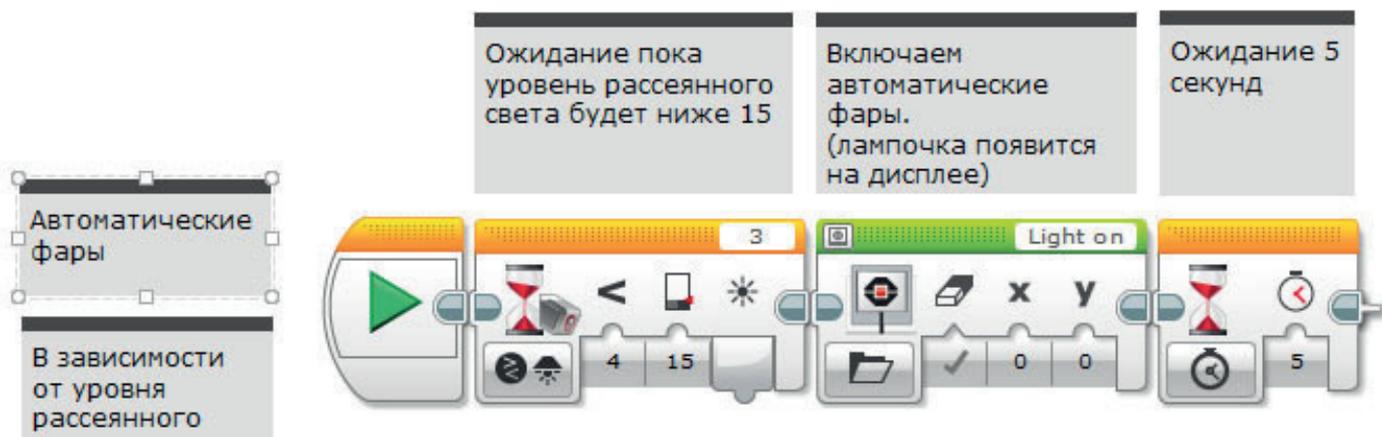
Функция просмотра порта со значением освещенности (порт 3)



Занятие 4

План урока

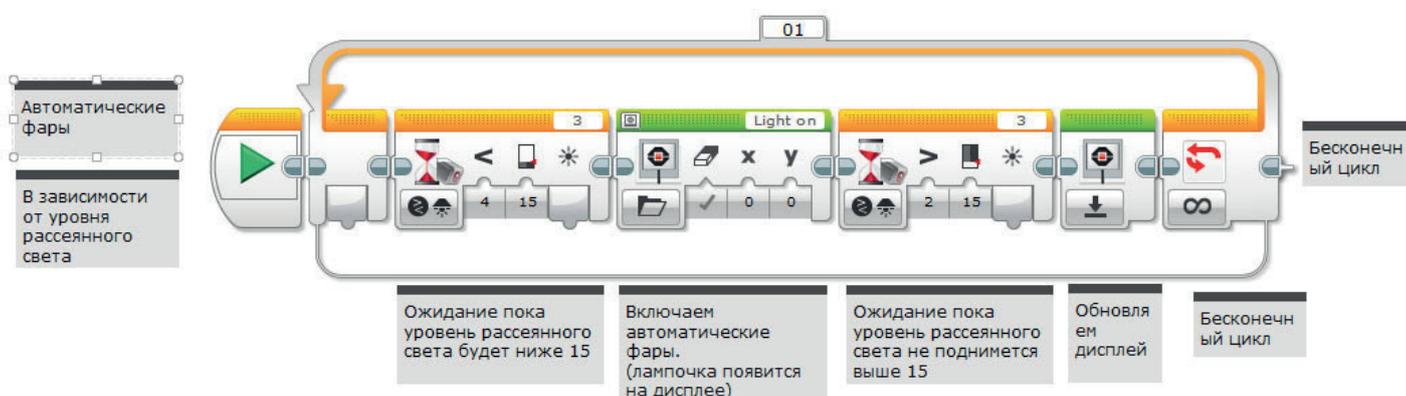
ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 4 ВКЛАДКА: MAIN 1



ОСНОВНОЕ ЗАДАНИЕ 2

- Необходимо составить программу, которая будет реагировать на изменения освещенности и более точно имитировать работу автоматических фар. Программа должна будет включать "фары" при наступлении "темноты" и выключать их, когда снова станет "светло".
- Расскажите классу о блоке цикла и продемонстрируйте его работу с помощью программного обеспечения EV3.
- Объясните понятие цикла и каким образом цикл обеспечивает бесконечное выполнение программ до тех пор, пока они не будут остановлены вручную.
- Обратите внимание обучающихся на то, что настройки на освещенность в программе необходимо постоянно контролировать и отлаживать, чтобы программа работала правильно.

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 4 ВКЛАДКА: MAIN 2



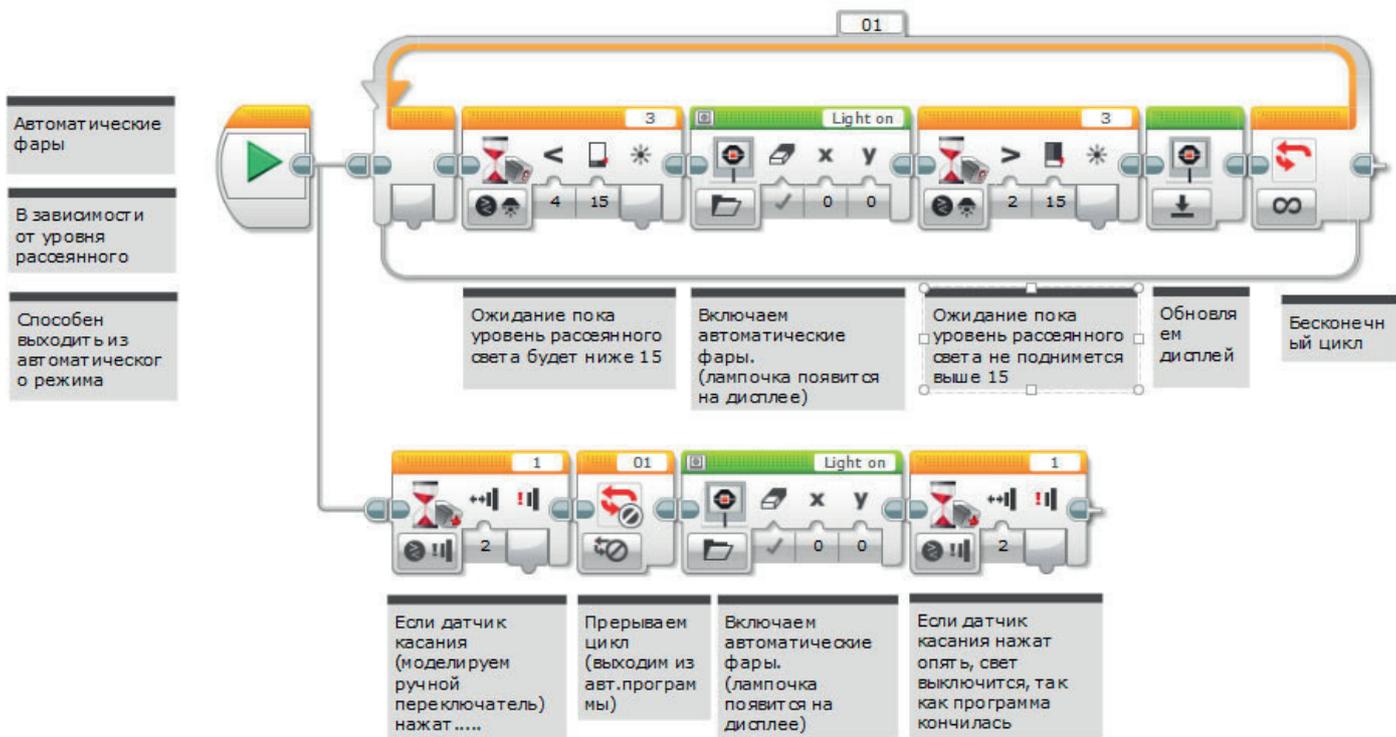
Занятие 4

План урока

ОСНОВНОЕ ЗАДАНИЕ 3

- Применение многозадачности / параллельного программирования для имитации автоматического и ручного включения и выключения фар.
- Необходимо усовершенствовать программы, включив в них параллельное программирование. Обучающиеся должны составить такие дополнения к программам автоматической работы фар, чтобы у "водителя" появилась возможность отключать автоматическую программу и включать фары вручную.
- Для имитации ручного выключателя потребуется датчик касания.
- Данная задача представляет собой дальнейшие возможности для применения булевой логики.
- Покажите классу, как перетащить "провод" от блока пуска для создания параллельной программы.
- Продемонстрируйте ученикам блок прерывания цикла и поясните, что его можно использовать для остановки любых действий в цикле параллельной программы.

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 4
ВКЛАДКА: MAIN 3



Занятие 4

План урока

ПОДВЕДЕНИЕ ИТОГОВ

- Примечание: ученики могут попробовать заменить блок управления дисплеем на блок световой индикации состояния интеллектуального блока, чтобы сделать действия колесного робота более реалистичными. Также они могут сочетать оба блока!
- В ходе этого занятия было рассмотрено много новых понятий, а также изучено несколько новых блоков программного обеспечения EV3. Воспользуйтесь ими для повторения понятий и проверки понимания учениками принципов их работы.
- Попросите одну-две группы продемонстрировать свои программы.
- Сообщите ученикам, что на следующем занятии они будут осваивать еще один вариант применения датчика цвета, который позволит добиться большей автономности машины.



Занятие 4

План урока

ЗАДАЧИ НА СЕГОДНЯ

Сегодня мы рассмотрим одну из функций датчика цвета – способность измерять освещенность и реагировать на ее изменение.

Автоматические фары автомобиля могут измерять освещенность и реагировать соответствующим образом (автоматически включаться и выключаться).

Помимо этого вы научитесь задавать роботу два алгоритма одновременно при помощи параллельного программирования (многозадачности).

ЗАДАЧИ НА СЕГОДНЯ

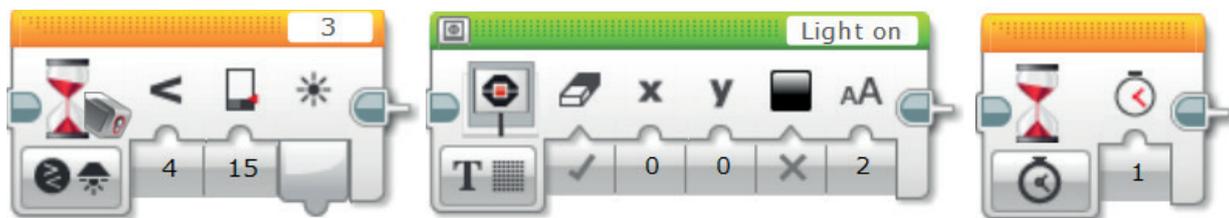
Что происходит с некоторыми автомобилями при наступлении темноты? Включаются фары! Можете ли вы составить программу, имитирующую работу автоматических фар автомобиля?

Найдите изображение лампочки для дисплея EV3 и включите его в свою программу. Чтобы включить эту лампочку, вам потребуется датчик цвета.

Для надлежащей работы программы необходимы показания освещенности из функции просмотра портов.

Примечание: блок управления дисплеем можно заменить блоком световой индикации состояния интеллектуального блока, либо использовать оба.

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 4

Рабочие карточки учеников

ЗАДАЧА 2

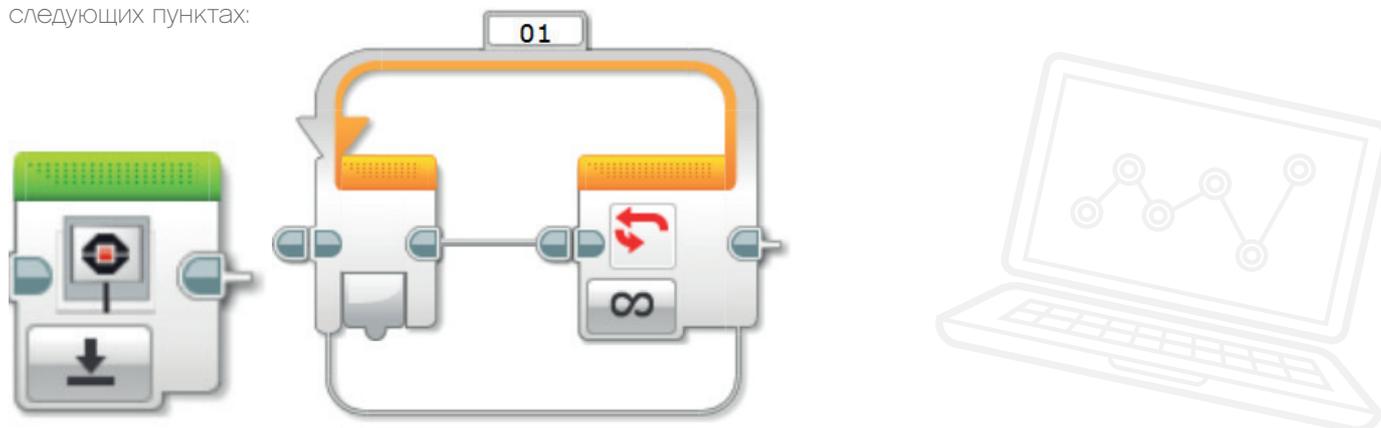
Теперь, когда автоматические фары вашего автомобиля успешно включаются, вам необходимо расширить программу, чтобы они выключались, когда снова становится светло.

Для этого нужно создать повторяющуюся программу, которая может повторять себя, чтобы пользователю не приходилось ее запускать заново.

Примечание: блок управления дисплеем можно также заменить блоком световой индикации состояния интеллектуального блока, либо использовать оба.

Используемые блоки

Используйте те же блоки, что и в задаче на программирование 1, но также не забывайте о следующих пунктах:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 4

Рабочие карточки учеников

ЗАДАЧА 3

Что если водителю захочется больше управлять автоматическими фарами, например, включать и выключать их вручную?

Многие современные автомобили имеют такую функцию, которая позволяет водителям отменять автоматическое выполнение программ.

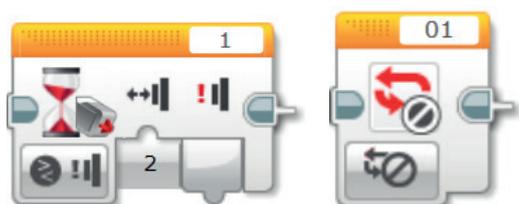
Попробуйте симитировать такую функцию в вашей программе за счет параллельного программирования или многозадачности. Для имитации ручного выключателя можно использовать датчик касания.

Полезная информация: для отключения автоматического управления также может потребоваться блок прерывания цикла.

Примечание: блок управления дисплеем можно также заменить блоком световой индикации состояния интеллектуального блока, либо использовать оба

Используемые блоки

Используйте те же блоки, что и в задачах на программирование 1 и 2, но также не забывайте о следующих двух:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 4

Рабочие карточки учеников

По завершении занятия по программированию важно записать свои мысли и наблюдения.

Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?



Мысли и наблюдения

Занятие 4

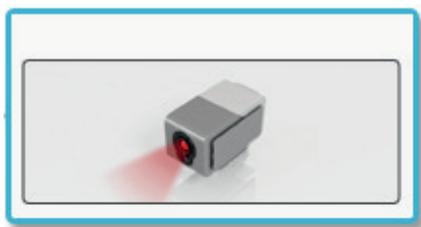
Примечания для учителя

РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

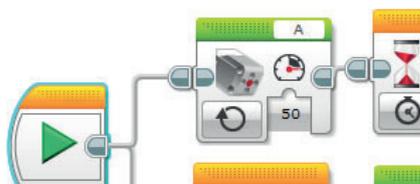
Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами.

НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

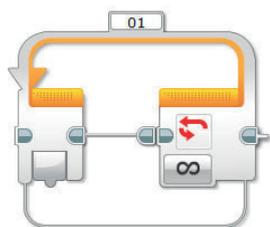
Аппаратное обеспечение > Датчик цвета – Свет



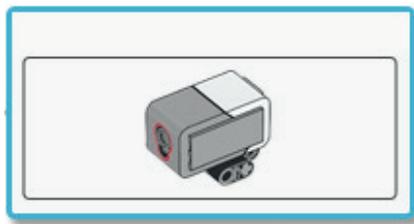
Расширенные знания > Многозадачность



Расширенные знания > Цикл



Руководство по сборке > Мобильная база с датчиком цвета и функцией движения вперед



РАНЕЕ РАССМОТРЕННЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

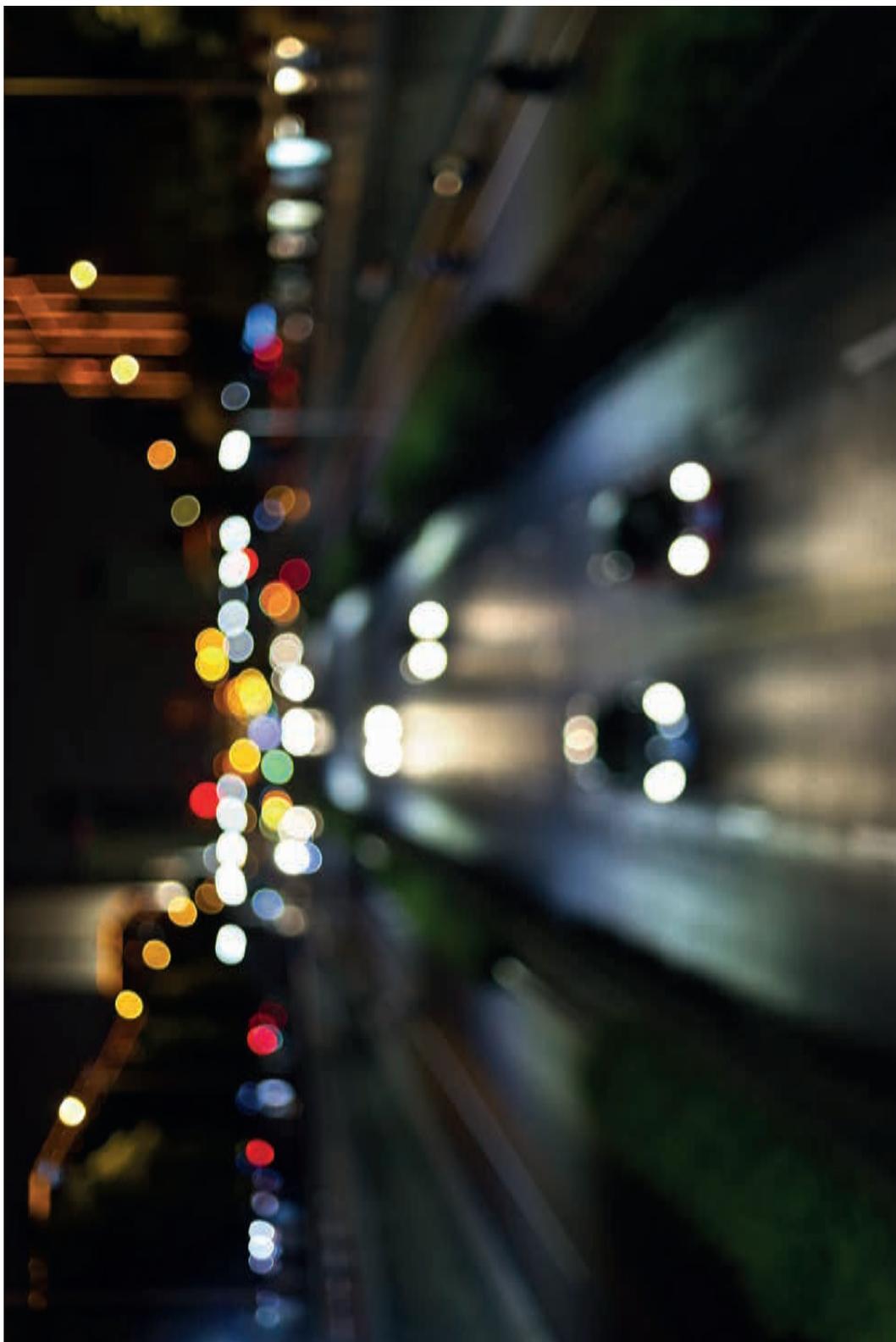
Аппаратное обеспечение > Дисплей интеллектуального блока



Занятие 4

Приложение

ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 4:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 4

Приложение



Занятие 4

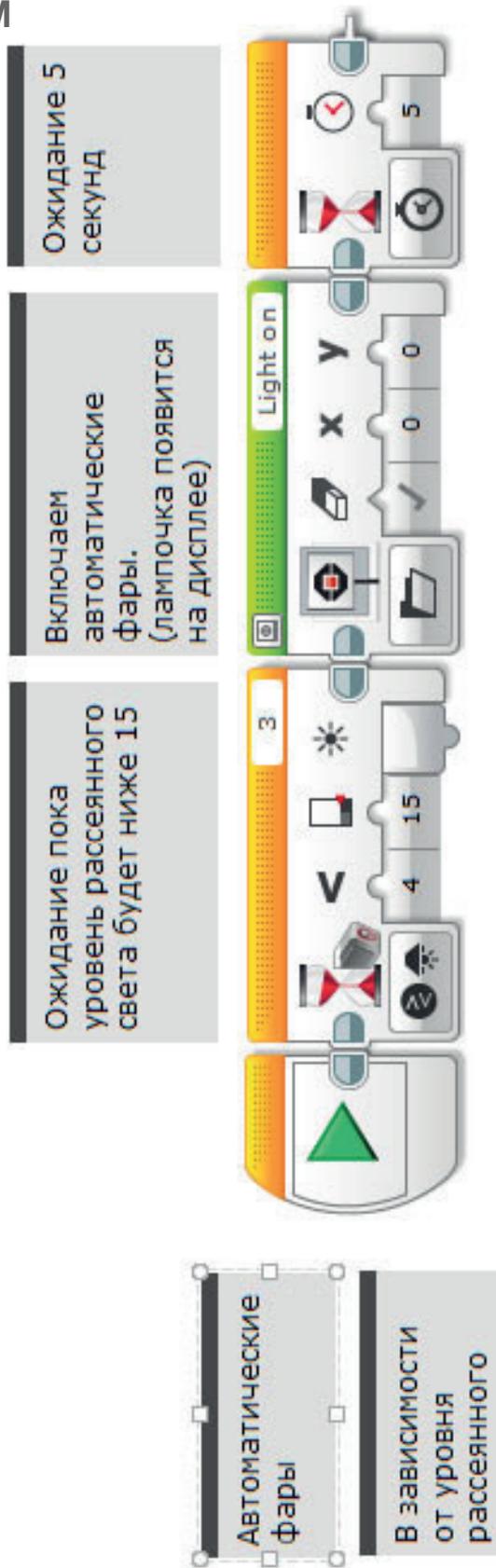
Приложение



Занятие 4

Приложение

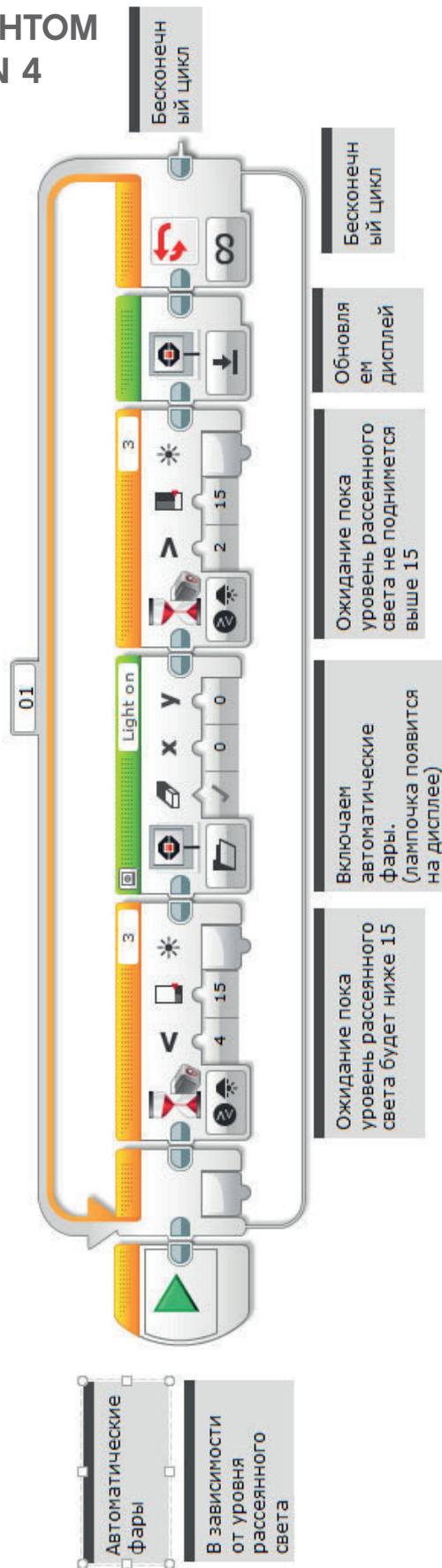
ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 4
ВКЛАДКА: MAIN 1



Занятие 4

Приложение

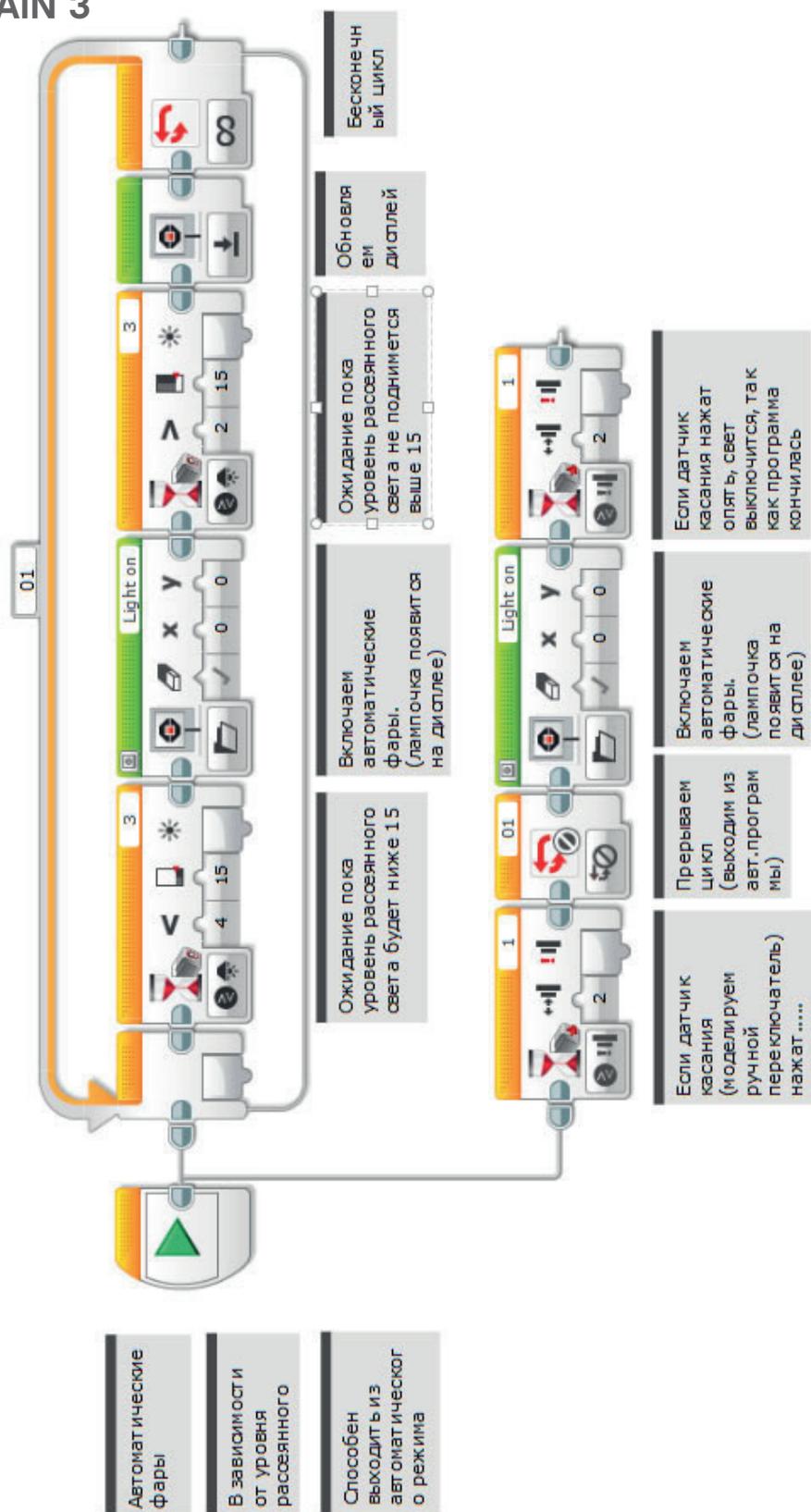
ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 4
ВКЛАДКА: MAIN 2



Занятие 4

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 4
ВКЛАДКА: MAIN 3



Занятие 4

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session4_1.c

```
Activity4_1.c
1  #pragma config(Sensor, S1, touchSensor, sensorEV3_Touch)
2  #pragma config(Sensor, S2, gyroSensor, sensorEV3_Gyro)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color, modeEV3Color_Ambient)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorB, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
6  #pragma config(Motor, motorC, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
8
9  /*
10 Create a program that turns on the lights automatically using the Color Sensor
11 when the ambient light level drops below a certain level.
12 */
13
14 task main()
15 {
16 //Wait for the level of ambient light intensity to drop below 15.
17 while(getColorAmbient(colorSensor) >= 15)
18 {
19 //Do nothing while we're waiting.
20 sleep(1);
21 }
22
23 //Display Image "Light on" on the LCD Screen.
24
25 drawBmpfile(0, 127, "Light on");
26
27 //Stay on for 5 seconds.
28 sleep(5000);
29 }
30
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 4

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session4_2.c

```
Activity4_2.c
1  #pragma config(Sensor, S1, touchSensor, sensorEV3_Touch)
2  #pragma config(Sensor, S2, gyroSensor, sensorEV3_Gyro)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color, modeEV3Color_Ambient)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorB, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
6  #pragma config(Motor, motorC, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  /**!Code automatically generated by 'ROBOTC' configuration wizard !*/
8
9  /*
10 Create an autonomous program that uses the Color Sensor to switch the light on and off
11 depending on the ambient light levels.
12 */
13
14 task main()
15 {
16 //Loop forever.
17 while(true)
18 {
19
20 //Wait for the level of ambient light intensity to drop below 15.
21 while(getColorAmbient(colorSensor) >= 15)
22 {
23 //Do nothing while we're waiting.
24 sleep(1);
25 }
26
27 //Display Image "Light on" on the LCD Screen.
28 drawBmpfile(0, 127, "Light on");
29
30 //Wait for the level of ambient light intensity to rise above 15.
31 while(getColorAmbient(colorSensor) <= 15)
32 {
33 //Do nothing while we're waiting.
34 sleep(1);
35 }
36
37 //Erase the display to clear the LCD screen.
38 eraseDisplay();
39 }
40 }
41
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 4

Приложение

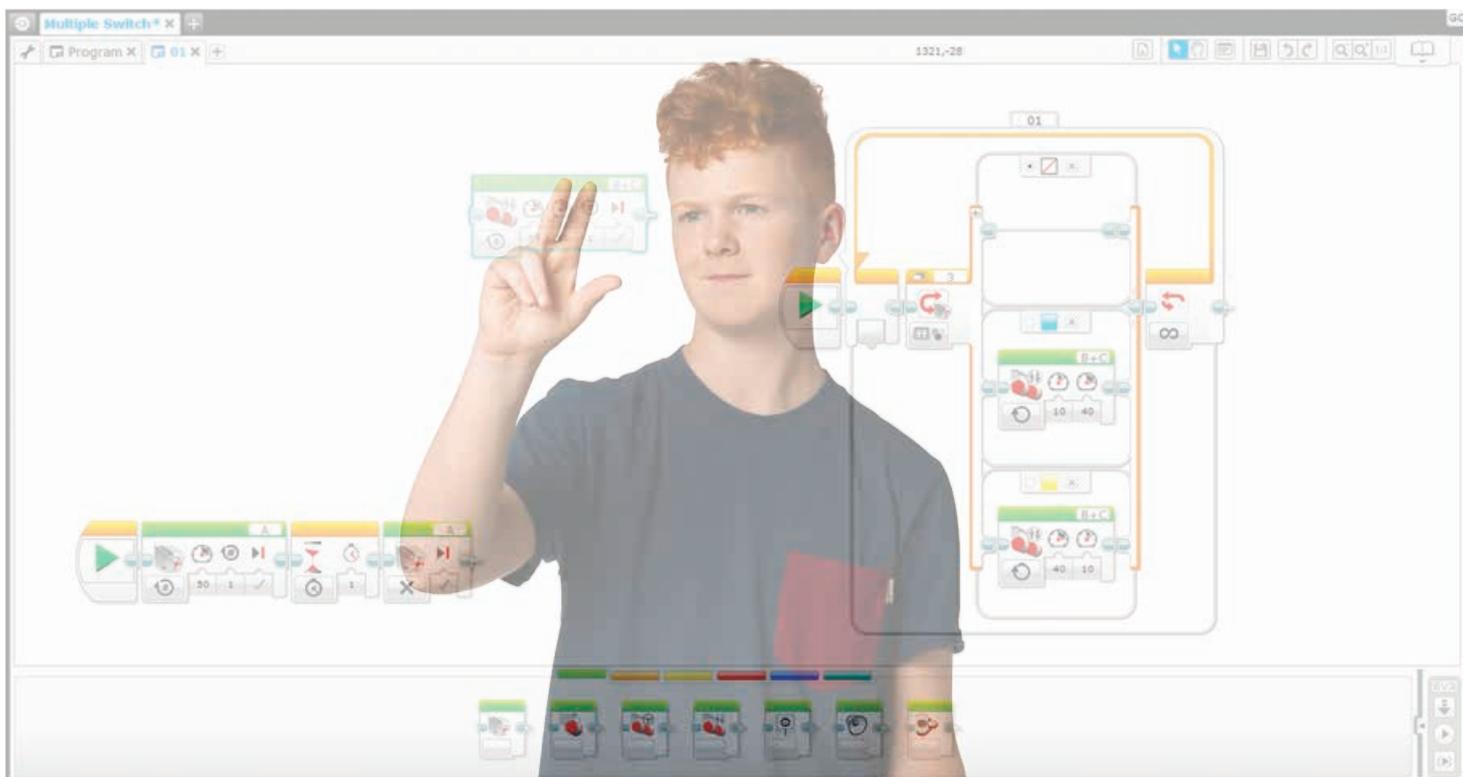
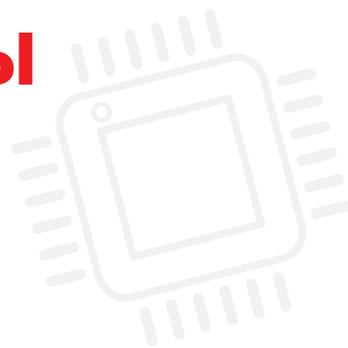
ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session4_3.c

```
Activity4_3.c
1  #pragma config(StandardModel, "EV3_REMOT")
2  /**Code automatically generated by 'ROBOTC' configuration wizard      !***
3
4  /*
5  Create an autonomous program that uses the Color Sensor to switch the light on and off
6  depending on the ambient light levels. Pressing the Touch Sensor overrides the loop
7  turning on the light regardless of the ambient light levels until the Touch Sensor is
8  pressed again.
9  NOTE: A sleep in the monitorTouchSensor task is used to replace the bump function in
10 the EV3 software.
11 */
12
13 task monitorTouchSensor()
14 {
15
16     //Wait for Touch Sensor to be touched.
17     while(getTouchValue(touchSensor) ==0)
18     {
19         //Do nothing until touched.
20         sleep(10);
21     }
22
23     //Stop task "main" (which is task #0)
24     stopTask(0);
25
26     //Display Image "Light on" on the LCD Screen.
27     drawBmpfile(0, 127, "Light on");
28
29     sleep (1000);
30
31     //Wait for Touch Sensor to be touched.
32     while(getTouchValue(touchSensor) == 0)
33     {
34         //Do nothing until touched.
35         sleep(10);
36     }
37 }
38
39 task main()
40 {
41     startTask(monitorTouchSensor);
42
43     //Loop forever
44     while(true)
45     {
46         //Wait for the level of ambient light intensity to drop below 15.
47         while(getColorAmbient(colorSensor) >= 15)
48         {
49             //Do nothing while we're waiting.
50             sleep(1);
51         }
52
53         //Display Image "Light on" on the LCD Screen.
54         drawBmpfile(0, 127, "Light on");
55
56         //Wait for the level of ambient light intensity to rise above 15.
57         while(getColorAmbient(colorSensor) <= 15)
58         {
59             //Do nothing while we're waiting.
60             sleep(10);
61         }
62
63         //Erase the display to clear the LCD screen.
64         eraseDisplay();
65     }
66 }
67
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 5

Светофоры и автоматизированные рельсовые системы



Занятие 5

Светофоры и автоматизированные рельсовые системы

На этом занятии ученики углубят понимание принципов работы датчика цвета и расширят опыт по использованию блока цикла, составив программу, которая имитирует работу светофора с сигналами "стоять – двигаться".

Кроме того, они запрограммируют колесного робота на движение по заданному пути (или линии) и узнают о блоке ветвлений.



Занятие 5

План урока



РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание, что с помощью алгоритмов можно выполнять определенную последовательность команд;
- расширение знаний о булевой логике и ее применении;
- использование блока ожидания для программирования датчика цвета;
- понимание того, что датчик цвета имеет несколько функций и может измерять ряд параметров;
- расширенное применение датчика цвета для распознавания цветов системы LEGO® и интенсивности отраженного света;
- расширенное понимание блока цикла;
- изучение понятия ветвлений и его использования для команд "истинно" и "ложно".

ОСНОВНЫЕ ПОНЯТИЯ

Вход, выход, алгоритм, ожидание, датчик цвета, отладка, освещенность, отраженный свет, цикл, булева логика, ветвление.

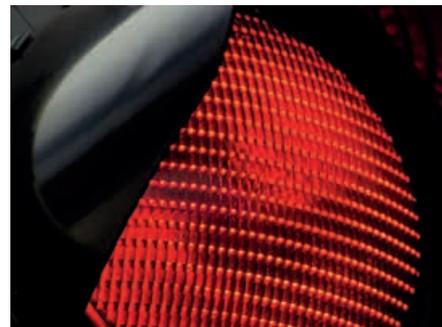
ВВЕДЕНИЕ

- Сообщите ученикам, что они снова будут работать с датчиком цвета. Теперь они должны исследовать его возможность распознавать цвета системы LEGO® и реагировать на них. Также они лучше поймут, как датчик реагирует на свет. Они должны создать программу, которая заставит колесного робота двигаться по заданному пути за счет распознавания интенсивности отраженного света. Работа разбита на три задачи.
- Поясните, что датчик цвета необходим для того, чтобы сделать робота более автономным, имитировать реакцию машины-робота на светофоры. Также объясните обучающимся, что составленная ими программа должна обеспечивать движение колесного робота по заданному пути или "дороге".
- Попросите привести примеры реальных ситуаций, где могут использоваться автоматические транспортные средства, например, в составе системы монорельсового транспорта московского метрополитена.
- Сообщите, что далее обучающиеся будут знакомиться с работой переключателей и искать способы их применения в составленных ими программах.
- Также они будут знакомиться с работой цикла и искать способы его реализации в составленных ими программах.



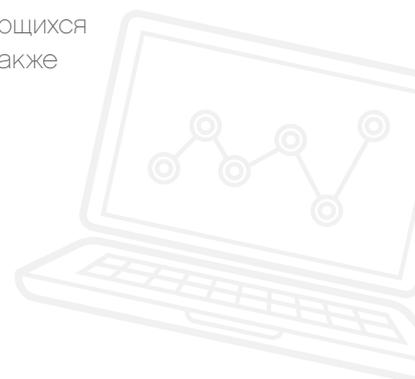
Занятие 5

План урока



ОСНОВНОЕ ЗАДАНИЕ 1

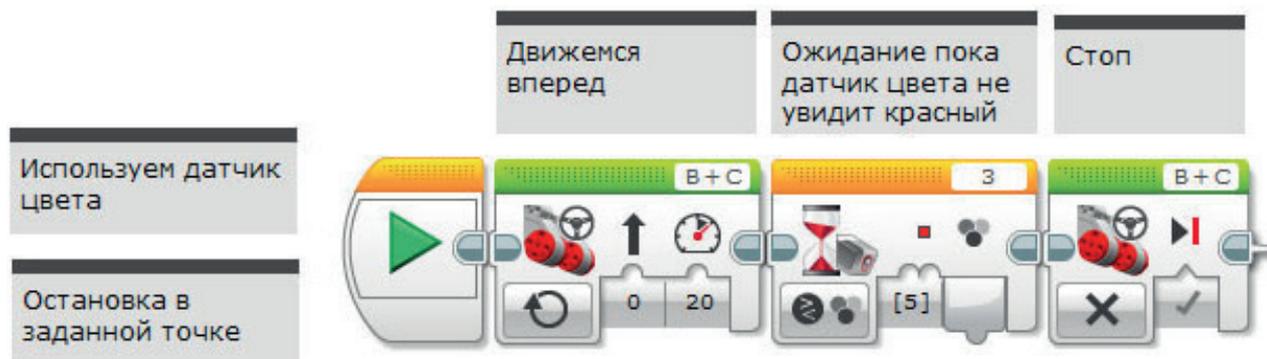
- Обучающиеся приступают к изучению функций датчика цвета, который распознает цвета системы LEGO®. Для этого они программируют колесного робота на движение по столу и остановку на "красном сигнале светофора".
- Для этого потребуется блок ожидания. Обратите внимание учеников на то, что робот может исключать цвета и игнорировать их при обнаружении.
- Класс составляет программу, останавливающую мотор по сигналу датчика цвета, когда датчик распознает красный цвет.
- Ученики должны настроить датчик цвета на распознавание красного цвета.
- Можно организовать дополнительную работу в этом направлении, попросив обучающихся поэкспериментировать с другими цветами и убедиться, что с ними их программы также работают корректно.



Занятие 5

План урока

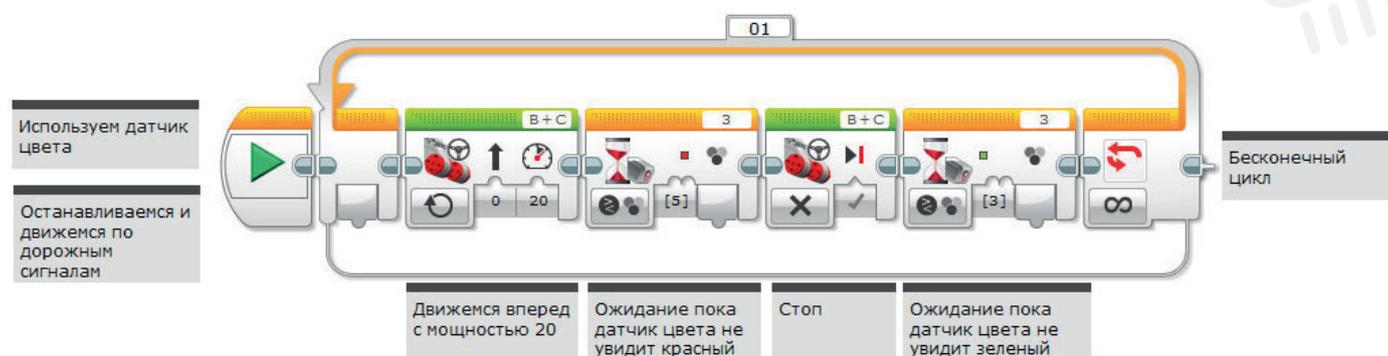
ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 5
ВКЛАДКА: MAIN 1



ОСНОВНОЕ ЗАДАНИЕ 2

- При выполнении этой задачи ученики должны более точно имитировать работу светофора, научив робота реагировать на последовательность зеленых и красных сигналов.
- Напомните классу о блоке цикла, повторите понятие цикла, вспомнив, что он обеспечивает бесконечное выполнение программы до ее остановки вручную.
- Внедрение цикла даст возможность установить множество "светофоров" на маршруте робота.
- Напомните ученикам о необходимости исключения всех других цветов, чтобы датчик цвета эффективно реагировал на выбранные цвета (красный и зеленый).

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 5
ВКЛАДКА: MAIN 2



Занятие 5

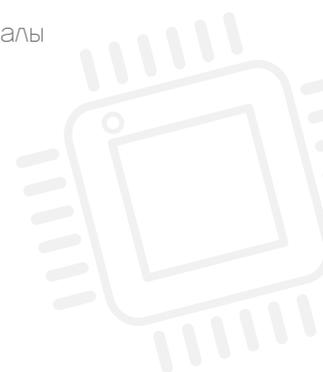
План урока



ОСНОВНОЕ ЗАДАНИЕ 3

Движение по линии – Создание автоматизированного самоходного автомобиля.

- При выполнении задачи ученики должны найти решение, как направить автоматизированный автомобиль по дороге или определенному маршруту. Как можно научить колесного робота следовать по линии за счет отраженного света?
- Для этого ученики должны ознакомиться с блоком ветвлений, который будет действовать внутри контура. Поясните, что блок ветвлений обеспечит автоматическое выполнение программы и, следовательно, автономное действие колесного робота.
- Расскажите о том, что блок ветвлений используется для контроля последовательности выполнения программы и настройки блока переключения по умолчанию на основе датчика касания служат классическим примером булевой логики. Покажите, как перенастроить блок на датчик цвета и расскажите про точку срабатывания. Пороговая точка используется для создания оператора “истинно / ложно” (до пороговой точки выполняется одно действие, после точки срабатывания – другое).
- Подскажите ученикам, что для создания программы, способной вести колесного робота по линии, потребуется предусмотреть движения робота из стороны в сторону в процессе перемещения вдоль линии. Другими словами, робот будет попеременно поворачиваться то вправо, то влево, в зависимости от того, пройдена ли пороговая точка. Предупредите, что используемый блок движения и рулевого управления необходимо установить на “Включено” (On).
- После того, как ученики добьются движения робота вдоль линии, его работу можно усовершенствовать, чтобы она больше напоминала движение автомобиля, то есть чтобы робот больше двигался по прямой, а не поворачивался из стороны в сторону.
- Необходимо выделить время на объяснение понятия ветвлений и того, каким же образом он служит примером булевой логики.
- Обучающиеся вновь будут работать с датчиком цвета, но на этот раз они должны запрограммировать его таким образом, чтобы он реагировал на интенсивность отраженного света.
- Полезная информация: им потребуется значение интенсивности цвета из функции просмотра порта, чтобы определить, какое значение ввести в блок ожидания.
- Полезная информация: наиболее эффективно использовать черную ленту на очень светлой (или белой) поверхности.
- Такую программу можно усовершенствовать, например, добавив второй датчик цвета. При этом робот сможет сочетать две программы – движения по линии и реагирование на сигналы светофора, имитируя автоматизированную пассажирскую систему, например, московский монорельсовый транспорт.



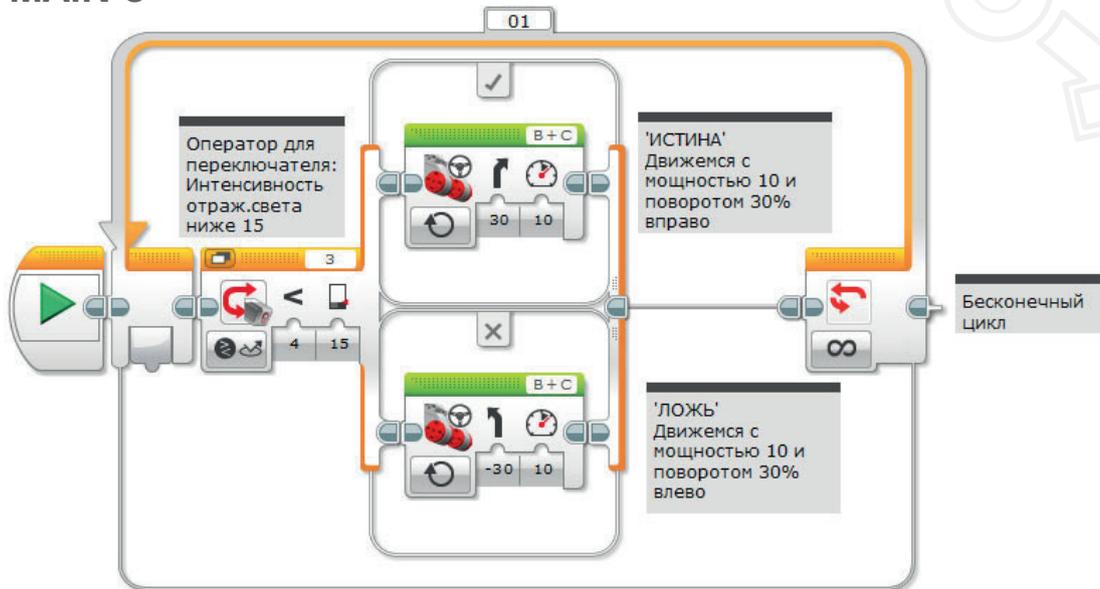
Занятие 5

План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 5
ВКЛАДКА: MAIN 3

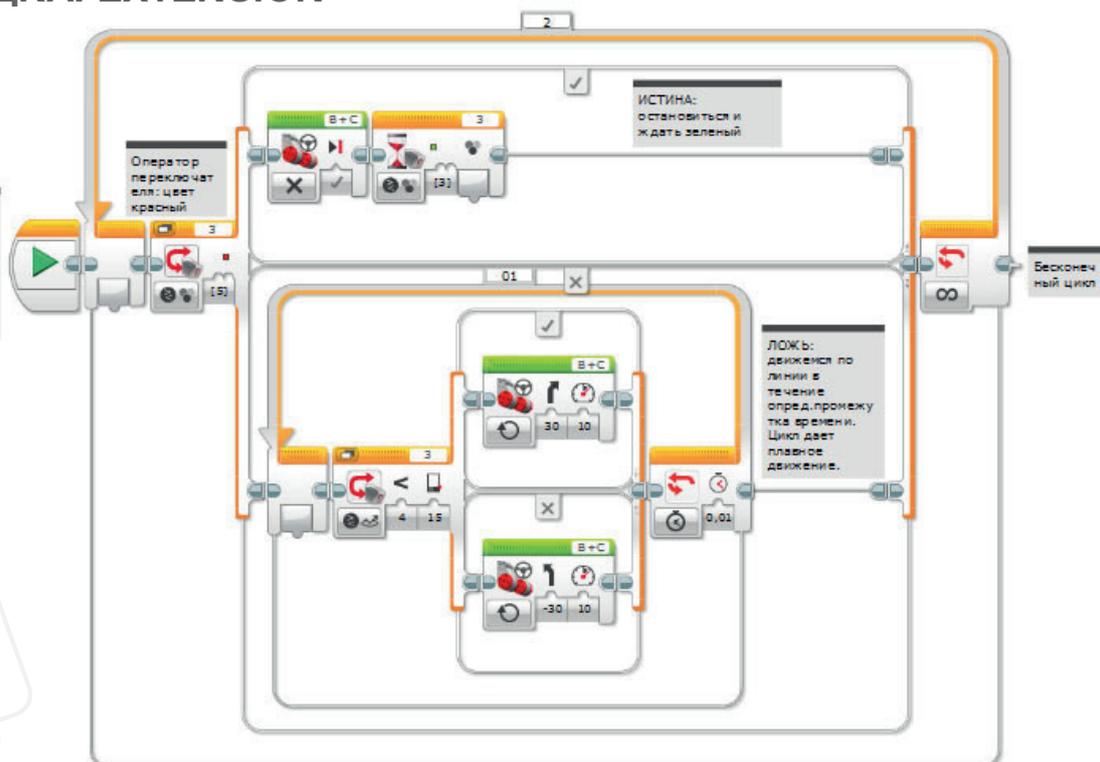
Используем датчик цвета

Построен на понимании рассеянного света, использует яркость отраженного света для движения по линии



ИМЯ ФАЙЛА С ВОЗМОЖНЫМИ РЕШЕНИЯМИ ПО УСЛОЖНЕНИЮ ЗАДАЧИ ИМИТАЦИЯ ДОКЛЕНДСКОГО ЛЕГКОГО МЕТРО): CS LESSON 5
ВКЛАДКА: EXTENSION

Комбинация предыдущих решений: движение по линии с возможностью остановки по дорожным сигналам



Занятие 5

Рабочие карточки учеников

ЗАДАЧИ НА СЕГОДНЯ

Сегодня вы будете использовать датчик цвета и блок ветвлений для принятия решений (булева логика). Два блока позволят роботу делать выбор в зависимости от обнаруженного цвета.

ЗАДАЧА 1

При управлении автомобилем важно знать правила дорожного движения и соблюдать их. Что должен сделать водитель, подъезжая к светофору?

Автоматизированным автомобилям нужен датчик, позволяющий автоматически распознавать сигналы светофора и реагировать на них.

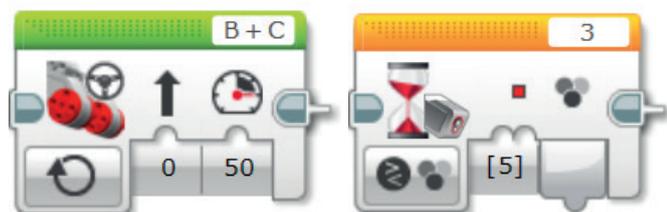
Для решения такой задачи вы должны запрограммировать колесного робота на выполнение команды "стоять". Какой цвет следует использовать в программе?

Запрограммируйте датчик цвета на распознавание красного цвета и остановку колесного робота с помощью блока ожидания.

Усовершенствуйте программу, научив робота останавливаться на определенном расстоянии от светофора.

Убедитесь, что робот реагирует только на красный цвет, исключая другие цвета.

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 5

Рабочие карточки учеников

ЗАДАЧА 2

После того, как робот запрограммирован на остановку по сигналу светофора, его необходимо научить снова трогаться.

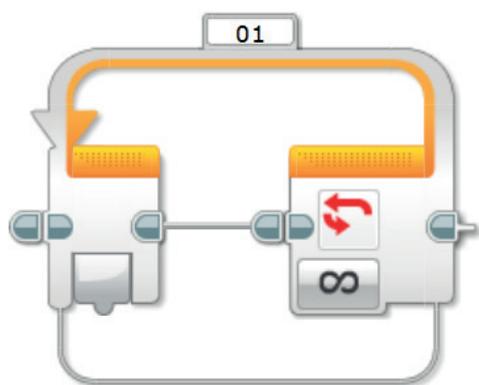
Составьте программу, где датчик цвета распознает команды "стоять" и "двигаться" и реагирует на них.

Какой цвет вы используете?

Что если вдоль улиц стоит несколько светофоров? Измените программу так, чтобы алгоритм "стоять - двигаться" мог повторяться.

Используемые блоки

Используйте те же блоки, что и в задаче на "программирование 1", но также не забывайте об этом:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 5

Рабочие карточки учеников

ЗАДАЧА 3

Для решения этой задачи колесный робот должен стать еще более автономным. Немного измените модель, чтобы датчик цвета был направлен вниз. Представьте себе, что автомобили могут двигаться на “автопилоте” по заданному маршруту, подобно автоматическим поездам московской монорельсовой дороги.

А теперь вы должны запрограммировать колесного робота именно на такие действия. Составьте программу, которая распознает специально проложенную черную линию и реагирует на нее. Соответственно, нужно составить программу следования по линии. Колесный робот должен следовать вдоль линии и не терять контакта с ней. Программу необходимо постоянно отлаживать, чтобы робот двигался вдоль линии как можно равномернее.

Полезная информация: Измените настройки датчика цвета в функции просмотра порта, чтобы датчик измерял интенсивность отраженного света.

Используемые блоки

Используйте те же блоки, что и в задачах на программирование 1 и 2, но также не забывайте о необходимых изменениях кода:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 5

Рабочие карточки учеников

По завершении занятия по программированию важно записать свои мысли и наблюдения. Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?



Мысли и наблюдения

Занятие 5

Примечания для учителя

РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

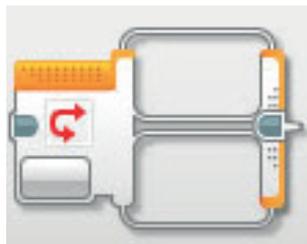
Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами.

НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Основы > Остановка на линии

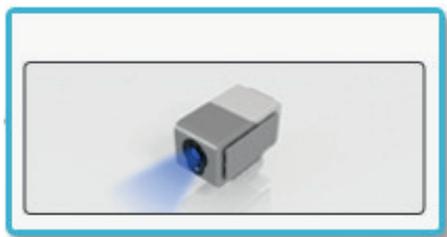


Расширенные знания > Ветвление

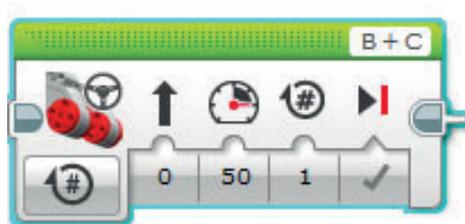


РАНЕЕ РАССМОТРЕННЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

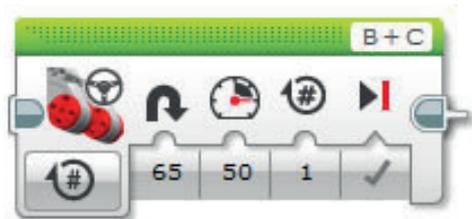
Аппаратное обеспечение >
Датчик цвета – Цвет



Основы > Движение по прямой



Основы > Движение по дуге



Расширенные знания > Цикл



Занятие 5

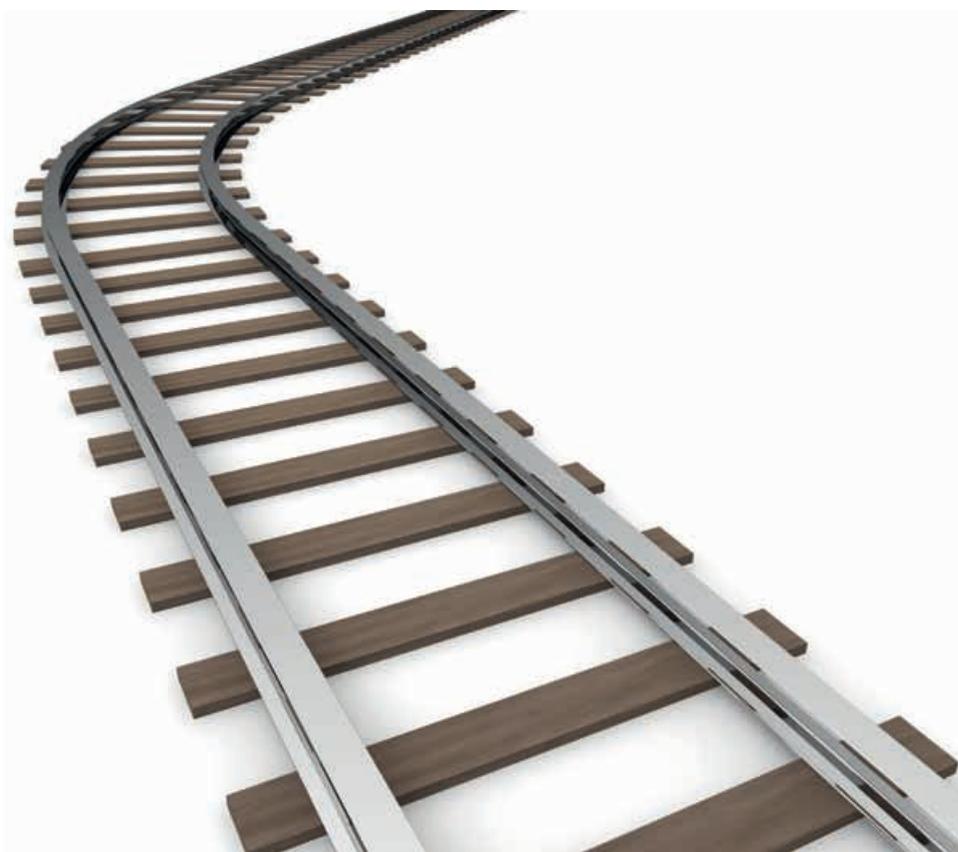
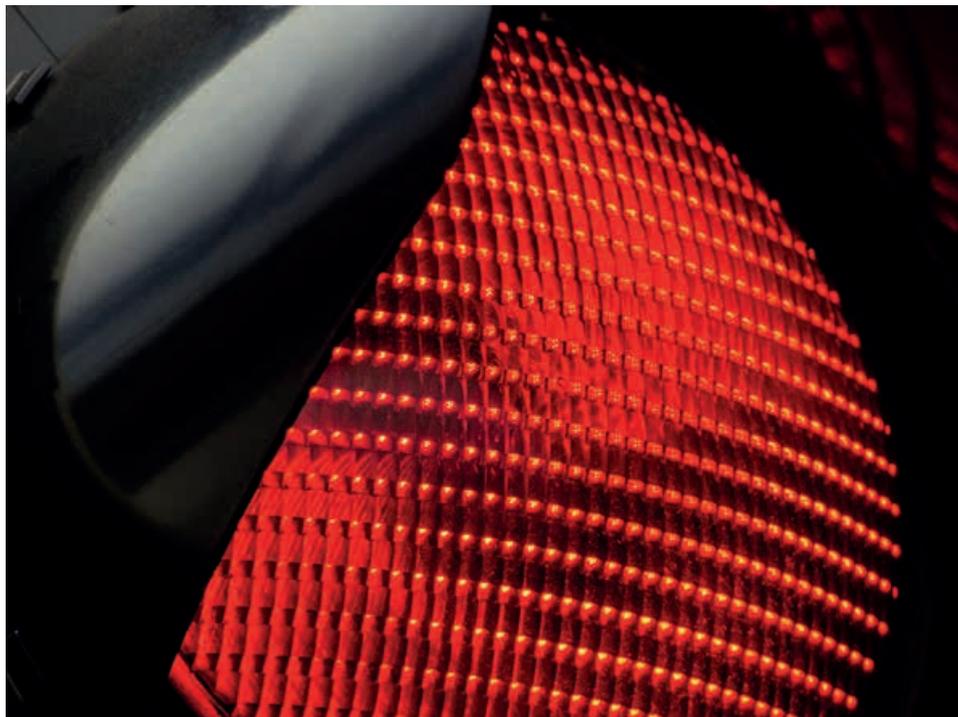
Приложение

ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 5:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 5

Приложение



Занятие 5

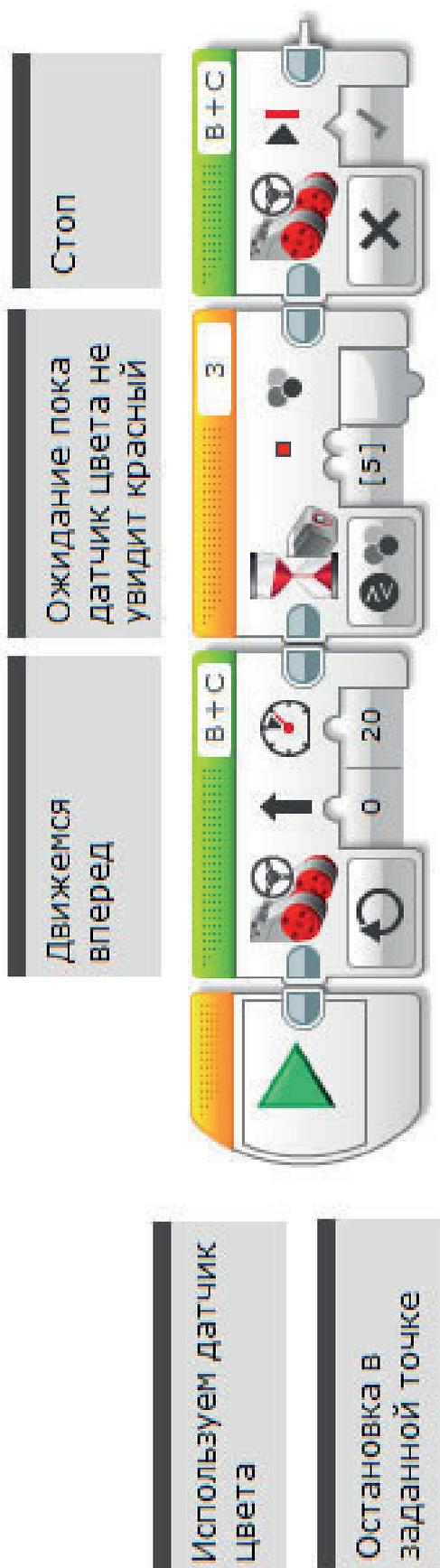
Приложение



Занятие 5

Приложение

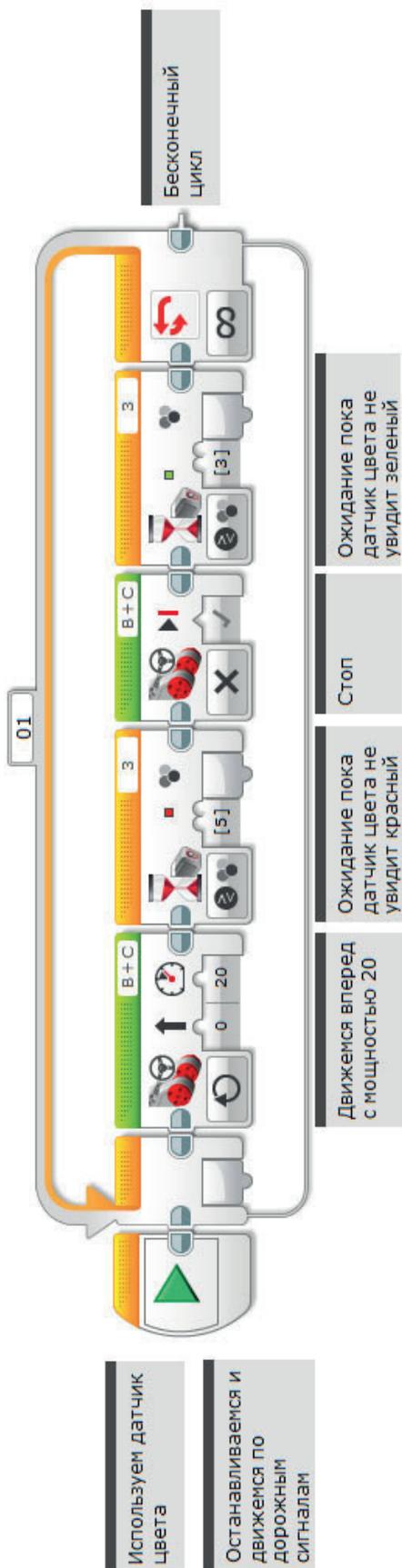
ИМЯ ФАЙЛА С
ВАРИАНТОМ РЕШЕНИЯ:
CS LESSON 5
ВКЛАДКА: MAIN 1



Занятие 5

Приложение

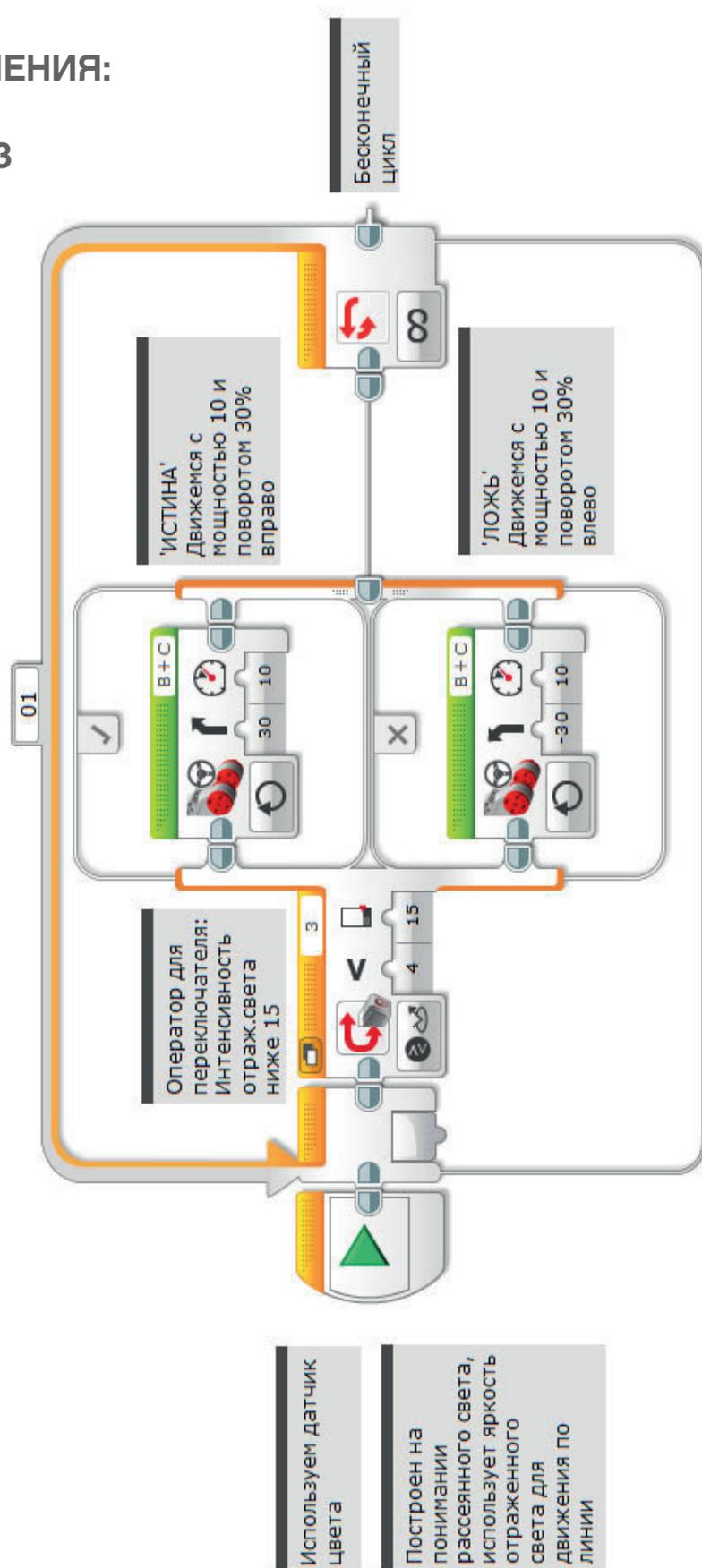
ИМЯ ФАЙЛА С
ВАРИАНТОМ РЕШЕНИЯ:
CS LESSON 5
ВКЛАДКА: MAIN 2



Занятие 5

Приложение

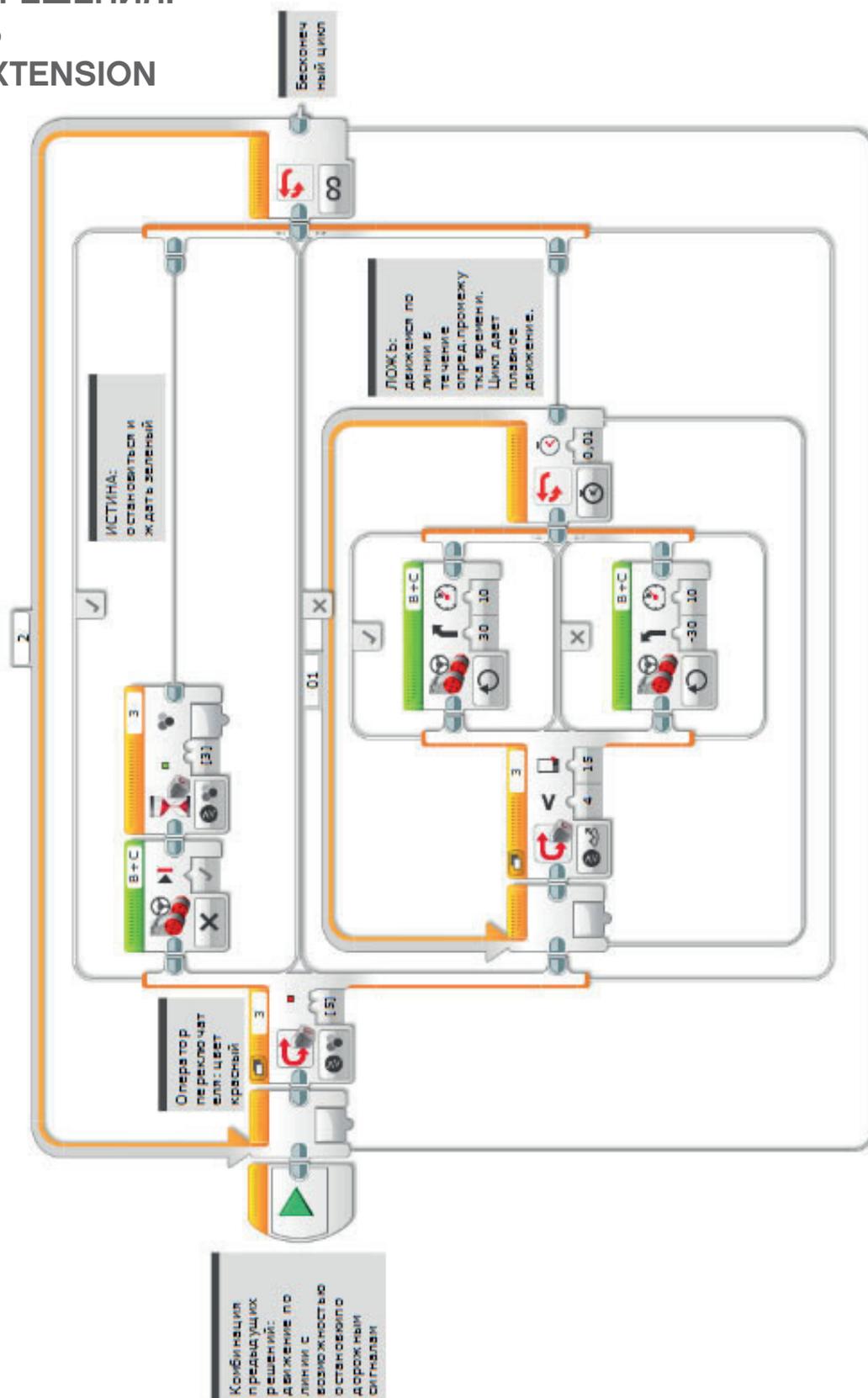
ИМЯ ФАЙЛА С
ВАРИАНТОМ РЕШЕНИЯ:
CS LESSON 5
ВКЛАДКА: MAIN 3



Занятие 5

Приложение

ИМЯ ФАЙЛА С
ВАРИАНТОМ РЕШЕНИЯ:
CS LESSON 5
ВКЛАДКА: EXTENSION



Занятие 5

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session5_1.c

```
Activity5_1.c
1  #pragma config(Sensor, S1, touchSensor, sensorEV3_Touch)
2  #pragma config(Sensor, S2, gyroSensor, sensorEV3_Gyro)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color, modeEV3Color_Color)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorA, armMotor, tmotorEV3_Large, PIDControl, encoder)
6  #pragma config(Motor, motorB, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  #pragma config(Motor, motorC, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
8  /**!Code automatically generated by 'ROBOTC' configuration wizard !!*/
9
10 /*
11 Create a program that drives the robot forward until the Color Sensor sees red.
12 The robot then stops.
13 */
14
15 task main()
16 {
17 //Set motor speed at 20% (Drive Forwards).
18 setMotorSpeed(motorB, 20);
19 setMotorSpeed(motorC, 20);
20
21 //Loop while the Color Sensor does not see red.
22 while(getColorName(colorSensor) != colorRed)
23 {
24 //Keep driving while the Color Sensor does not see red.
25 sleep(10);
26 }
27
28 //Set motor speed to 0% (Stop).
29 setMotorSpeed(motorB, 0);
30 setMotorSpeed(motorC, 0);
31 }
32
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 5

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session5_2.c

```
Activity5_2.c
1  #pragma config(Sensor, S1, touchSensor, sensorEV3_Touch)
2  #pragma config(Sensor, S2, gyroSensor, sensorEV3_Gyro)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color, modeEV3Color_Color)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorB, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
6  #pragma config(Motor, motorC, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
8
9  /*
10 Create a program where the robot drives forward stopping at the red brick and
11 moving off when the Color Sensor sees the green brick. This happens continuously.
12 */
13
14 task main()
15 {
16 //Repeat Forever.
17 while(true)
18 {
19 //Set motor speed to 20%. (Drive Forwards).
20 setMotorSpeed(motorB, 20);
21 setMotorSpeed(motorC, 20);
22
23 //Loop while the sensor does not see red.
24 while(getColorName(colorSensor) != colorRed)
25 {
26 //Keep driving while the sensor does not see red.
27 sleep(10);
28 }
29
30 //Set the motor speed to 0% (Stop).
31 setMotorSpeed(motorB, 0);
32 setMotorSpeed(motorC, 0);
33
34 //After we've seen red, we need to wait until we see green.
35 while(getColorName(colorSensor) != colorGreen)
36 {
37 sleep(10);
38 //Do nothing while the sensor does not see green.
39 //When the robot sees green loop back to start of program.
40 }
41 }
42 } //End of program - turn off all motors automatically.
43
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 5

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session5_3.c

```
Activity5_3.c
1  #pragma config(StandardModel, "EV3_REMBOТ")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard          !!*/
3
4  /*
5  Create a line following program that uses the Color Sensor in reflected mode to
6  'wiggle' along a line.
7  NOTE: ***THERE ARE SLIGHT MOTOR POWER DIFFERENCES BETWEEN ROBOTC & EV3 PROGRAMS***
8  */
9
10 task main()
11 {
12     //Repeat Forever.
13     while(true)
14     {
15         //If we see a dark value.
16         if(getColorReflected(colorSensor) < 15)
17         {
18             //Steer to the right.
19             setMotorSpeed(motorB, 30);
20             setMotorSpeed(motorC, 12);
21         }
22         //If we see a light value (or not dark).
23         else
24         {
25             //Steer to the left.
26             setMotorSpeed(motorB, 12);
27             setMotorSpeed(motorC, 30);
28         }
29     }
30 }
31
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 6

Звуковой сигнал заднего хода



Занятие 6

Звуковой сигнал заднего хода

Ученики вспомнят материал по ультразвуковому датчику и повторят принципы работы парковочных датчиков (датчиков, подающих предупреждающий звуковой сигнал при приближении к препятствию).

В процессе решения задач ученики создадут программы, имитирующие работу датчиков парковки, которыми оборудованы современные автомобили. Последняя из задач предусматривает составление программы, подающей предупреждающие звуковые сигналы, частота повторения которых должна возрастать по мере приближения робота к препятствию.

Обучающимся потребуется ряд методов программирования и программных блоков, включая параллельное программирование (многозадачность), циклы и переключатели. Также они узнают о математическом блоке и каналах передачи данных.

Интернет-ресурсы:
можно поискать в интернете видео, демонстрирующие работу датчиков парковки.



Занятие 6

План урока

РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание, что с помощью алгоритмов можно выполнять определенную последовательность команд;
- расширение знаний о булевой логике и ее применении;
- использование блока ожидания для программирования датчика цвета;
- понимание принципа работы ультразвукового датчика за счет отражения волн от предметов и умение запрограммировать датчик на определение расстояния;
- программирование колесного робота на задний ход, подачу звукового сигнала на определенном расстоянии от объекта, а также на остановку на заданном расстоянии от этого объекта;
- расширенное понимание блока цикла;
- изучение понятия переключателя и его использования для команд "истинно" и "ложно";
- освоение математических программных блоков и функций;
- освоение возможности переноса показаний с одного блока в другой через канал передачи данных.

ОСНОВНЫЕ ПОНЯТИЯ

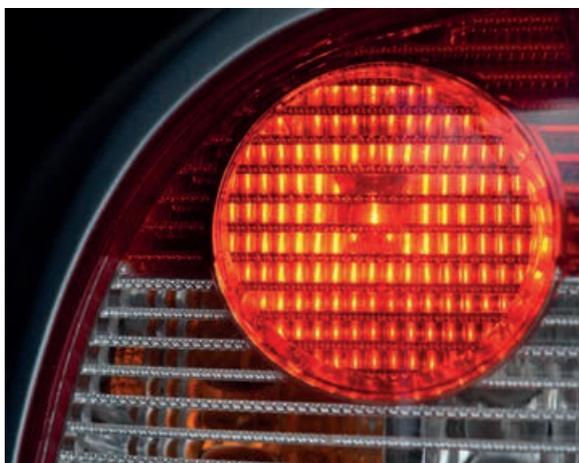
Вход, выход, алгоритм, ожидание, ультразвуковой датчик, наладка, цикл, булева логика, переключатель, математический блок, канал передачи данных, прерывание.

ВВЕДЕНИЕ

- Объясните ученикам, что на занятии они будут составлять и совершенствовать программы, подробно имитирующие работу датчиков парковки.
- Объясните, что задние датчики парковки действуют, испуская предупреждающий звуковой сигнал, который сначала имеет низкую частоту повторений, но по мере приближения автомобиля к препятствию (обычно стене) учащается. Это возможно благодаря использованию ультразвукового датчика.
- Расскажите ученикам, что на этом занятии они будут работать с тем же ультразвуковым датчиком, который им уже знаком по занятию 1, но теперь они существенно расширят свои знания и понимание принципов его работы.
- Попросите учеников объяснить, как работает этот датчик. Они должны рассказать, что ультразвуковые волны выходят через излучатель датчика и отражаются от объектов. По времени, за которое отраженная волна возвращается обратно к датчику, определяется расстояние.
- Можно выделить классу время на эксперименты с датчиком и проверку его точности. Влияет ли скорость на точность датчика?

Занятие 6

План урока



ОСНОВНОЕ ЗАДАНИЕ 1

- Для решения задачи необходимо составить программу, которая заставит колесного робота испускать звуковой сигнал при приближении к препятствию во время заднего хода.
- Чем ближе колесный робот подъезжает к препятствию, тем чаще должны становиться гудки. Робот должен останавливаться автоматически при достижении определенного расстояния от препятствия.
- Предупреждающие гудки (и особенно их частота) генерируются благодаря математическому блоку и каналу передачи данных.
- Такой принцип работы объясняется в примере программы ниже.
- Познакомьте класс с математическим блоком и продемонстрируйте его работу.
- Ученики должны прикрепить ультразвуковой датчик к заднему торцу базовой модели.
- При выполнении данного упражнения обучающиеся повторяют материал по параллельному программированию (многозадачности). Можно напомнить им, как перетаскивать изображение канала передачи данных от блока пуска для активации такой возможности.



Занятие 6

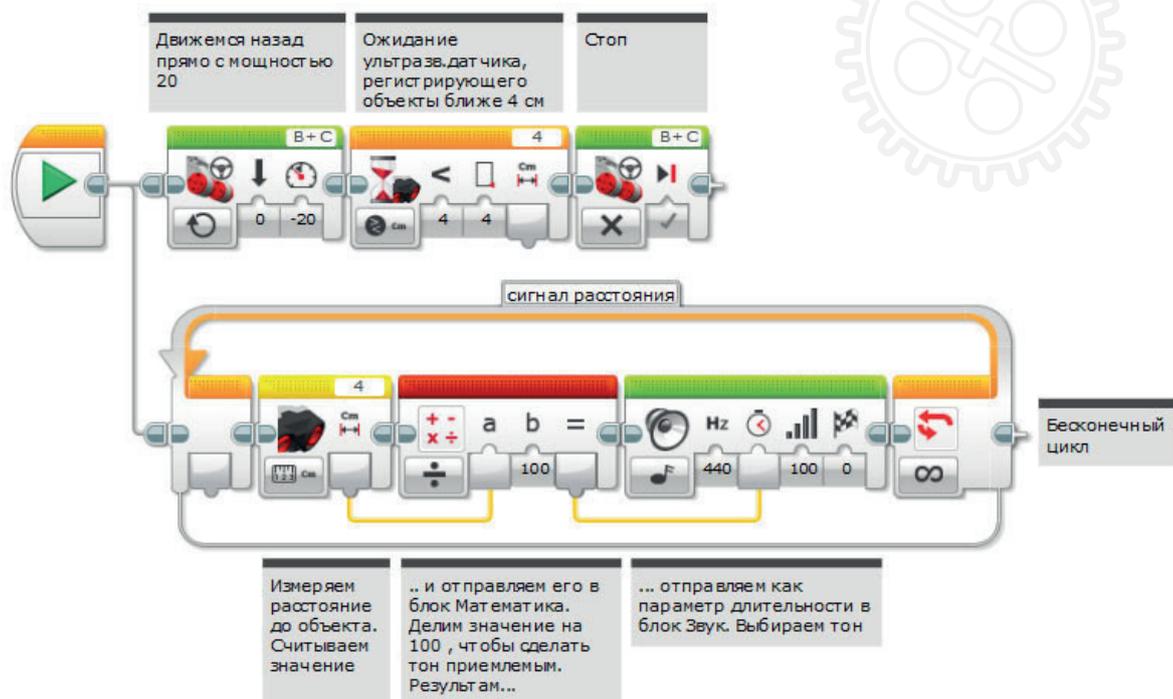
План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 6
ВКЛАДКА: MAIN 1

Моделирование задних парковочных сенсоров

При движении назад гудок звучит чаще по мере приближения к помехе

Мы используем ультразвуковой датчик динамически и измеряем приближение. Делим значения на 100, чтобы сделать тон приемлемым в секундах (чем ближе, тем чаще звучит сигнал)



ОСНОВНОЕ ЗАДАНИЕ 2

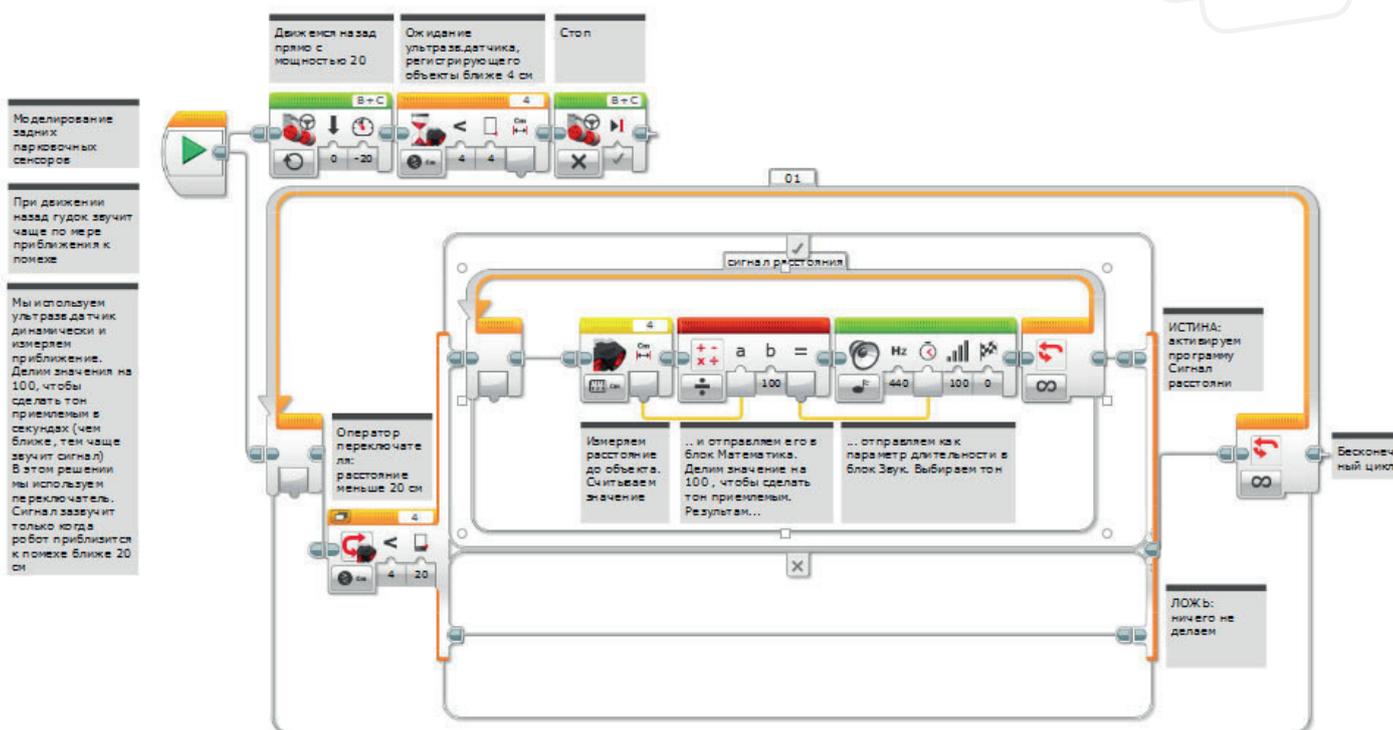
- Это задание непосредственно базируется на материале из задания 1.
- Спросите учеников, заметили ли они, что "не так" в их программах.
- Они должны ответить, что звуковой сигнал, подаваемой колесным роботом, включается сразу, независимо от того, на каком расстоянии от препятствия находится робот.
- Как усовершенствовать программу, чтобы действия робота были больше похожи на действия реального автомобиля?
- Можно сделать так, чтобы звуковой сигнал / гудки включались на определенном расстоянии от препятствия.
- Для этого потребуется блок переключения с оператором "истинно / ложно".



Занятие 6

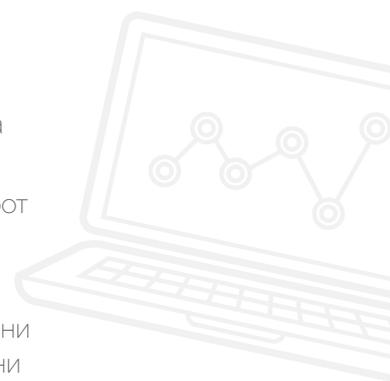
План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 6
ВКЛАДКА: MAIN 2



ОСНОВНОЕ ЗАДАНИЕ 3

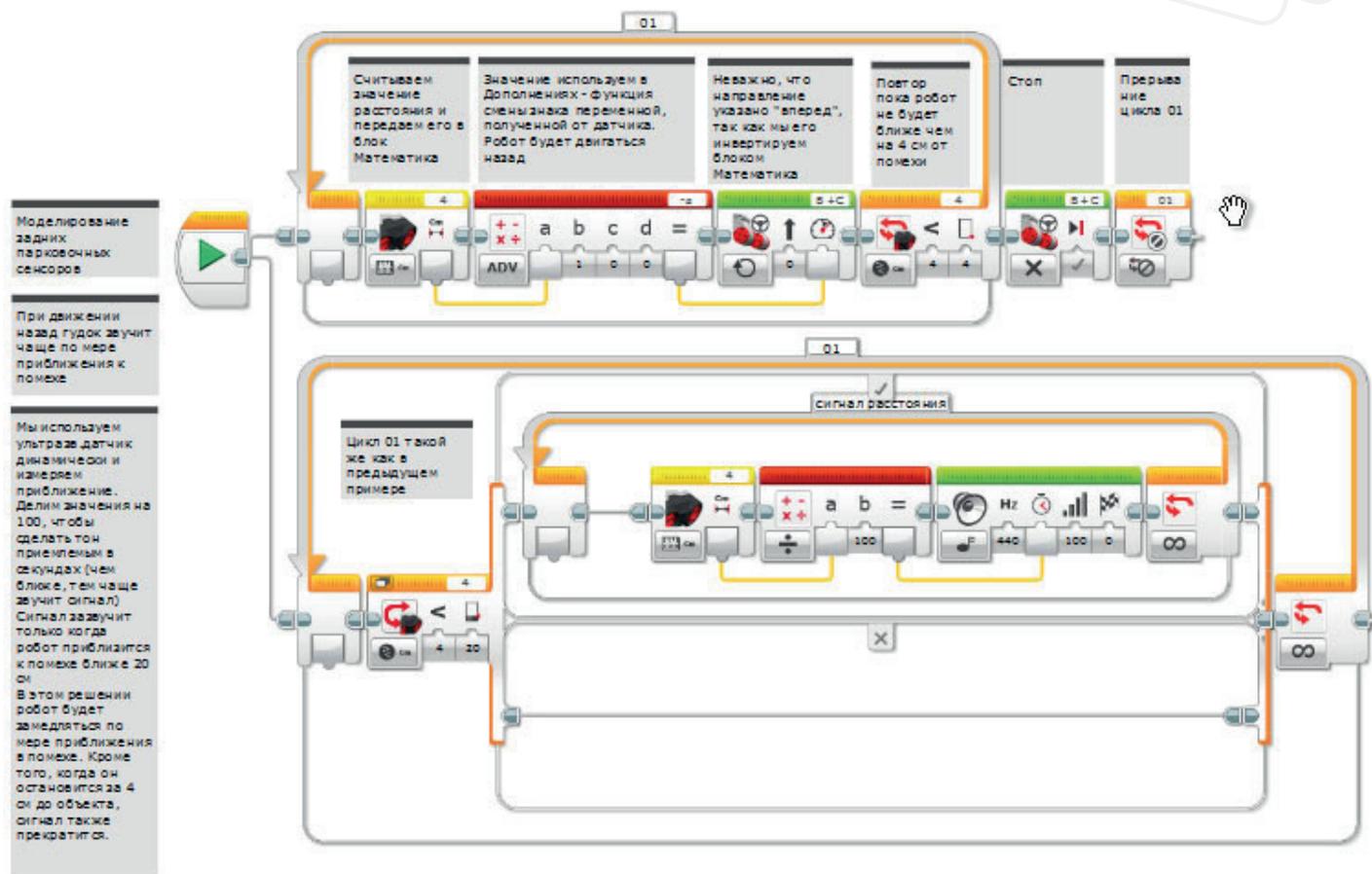
- Теперь колесные роботы, созданные учениками, должны действовать очень похоже на реальные автомобили при движении задним ходом.
- Гудки должны начинаться на определенном расстоянии от препятствия, при этом робот должен останавливаться на заданном расстоянии.
- Спросите учеников, как еще можно усовершенствовать программу.
- Обратите их внимание на то, что когда автомобили сдают назад и останавливаются, они выполняют такое действие, которое их колесные роботы пока выполнить не могут - они замедляют ход по мере приближения к препятствию, и при торможении предупреждающий звуковой сигнал выключается.
- Таким образом, обучающиеся должны далее усовершенствовать программу с включением в нее этих двух функций.
- Для этого следует использовать еще один математический блок и связать ультразвуковой датчик со скоростью, чтобы он инициировал замедление хода колесного робота по мере сокращения расстояния до препятствия.
- Также потребуется блок прерывания цикла, который выключит звуковой сигнал при остановке колесного робота.



Занятие 6

План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 6
ВКЛАДКА: MAIN 3



ПОДВЕДЕНИЕ ИТОГОВ

- Это отличная возможность для всех групп в классе поупражняться в программировании совместно.
- Например, можно одновременно направить всех роботов к одному препятствию и проверить, какой наиболее точно передает действия автомобиля.
- Предложите классу заснять колесных роботов на видео, чтобы оценить их работу позднее.



Занятие 6

Рабочие карточки учеников

ЗАДАЧИ НА СЕГОДНЯ

Сегодня мы рассмотрим, как желтый блок датчика функционирует совместно с математическим блоком. Также мы будем использовать блок цикла.

Затем нужно перенести первый параграф задачи 1, начинающийся со слов "в ходе решения задач". в конец Задач к сегодняшнему тексту, представленному выше.



ЗАДАЧА 1

В ходе решения задач вы будете программировать колесного робота таким образом, чтобы он имитировал работу парковочного датчика автомобиля.

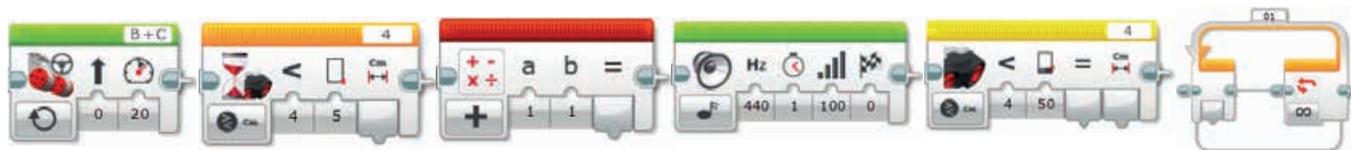
Что происходит, когда некоторые автомобили сдают назад? Слышатся гудки, которые учащаются по мере приближения автомобиля к препятствию.

Составьте программу, которая направляет колесного робота задним ходом, издает предупреждающие гудки при приближении к препятствию и затем автоматически останавливает его на заданном расстоянии.

Полезная информация 1: необходимо применить метод параллельного программирования (многозадачность).

Полезная информация 2: для увеличения частоты гудков при приближении колесного робота к препятствию используйте уже полученные знания о математическом блоке и канале передачи данных.

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 6

Рабочие карточки учеников

ЗАДАЧА 2

Какие особенности вы заметили у своей программы и у звукового сигнала в частности? Гудки должны становиться чаще по мере приближения колесного робота к препятствию.

Однако в реальности предупреждающий сигнал включается только когда автомобиль находится на определенном расстоянии от препятствия.

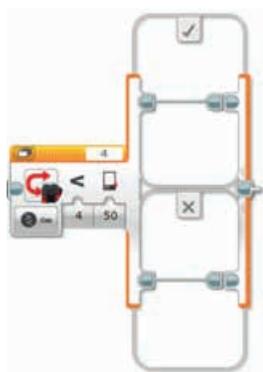
Сымитируйте такие действия при помощи вашей программы.

Нужно усовершенствовать уже созданную программу: внесите в нее небольшие изменения, чтобы гудки начинались на заданном расстоянии от препятствия.

Полезная информация: используйте оператор “истинно/ложно” и булеву логику. Какой программный блок требуется для этого?

Используемые блоки

Используйте те же блоки, что и в задаче на программирование 1, но также не забывайте о следующем:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 6

Рабочие карточки учеников

ЗАДАЧА 3

К настоящему моменту колесный робот должен достаточно правдоподобно имитировать работу задних датчиков парковки. Настало время перейти на следующий уровень программирования.

Добавьте больше функций.

1. Сделайте так, чтобы гудки прекращались, когда колесный робот останавливается на заданном расстоянии от препятствия.
2. Измените программу таким образом, чтобы робот замедлял движение с началом

гудков.

Полезная информация 1: для прекращения звукового сигнала необходимо прервать цикл.

Полезная информация 2: необходимо связать скорость с расстоянием через второй математической блок на определенном этапе программы. Определите этот этап.

Используемые блоки

Используйте те же блоки, что и в задачах на программирование 1 и 2, но также не забывайте о следующих пунктах:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 6

Рабочие карточки учеников

По завершении занятия по программированию важно записать свои мысли и наблюдения. Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?



Мысли и наблюдения

Занятие 6

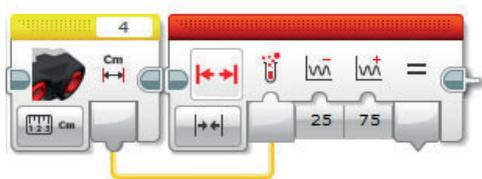
Примечания для учителя

РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

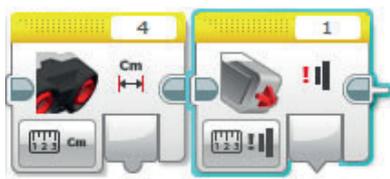
Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами..

НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

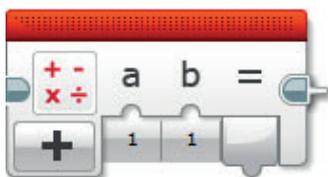
Расширенные знания > Каналы передачи данных



Расширенные знания > Блоки датчиков

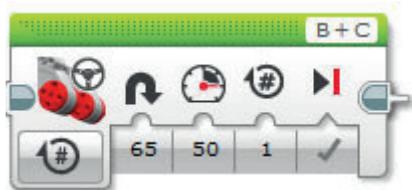


Расширенные знания > Математика – Основы



РАНЕЕ РАССМОТРЕННЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

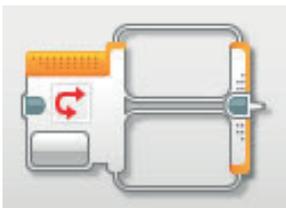
Основы > Движение по дуге



Расширенные знания > Цикл



Расширенные знания > Переключатель



Расширенные знания > Многозадачность



Занятие 6

Приложение

ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 6:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 6

Приложение



Занятие 6

Приложение



Занятие 6

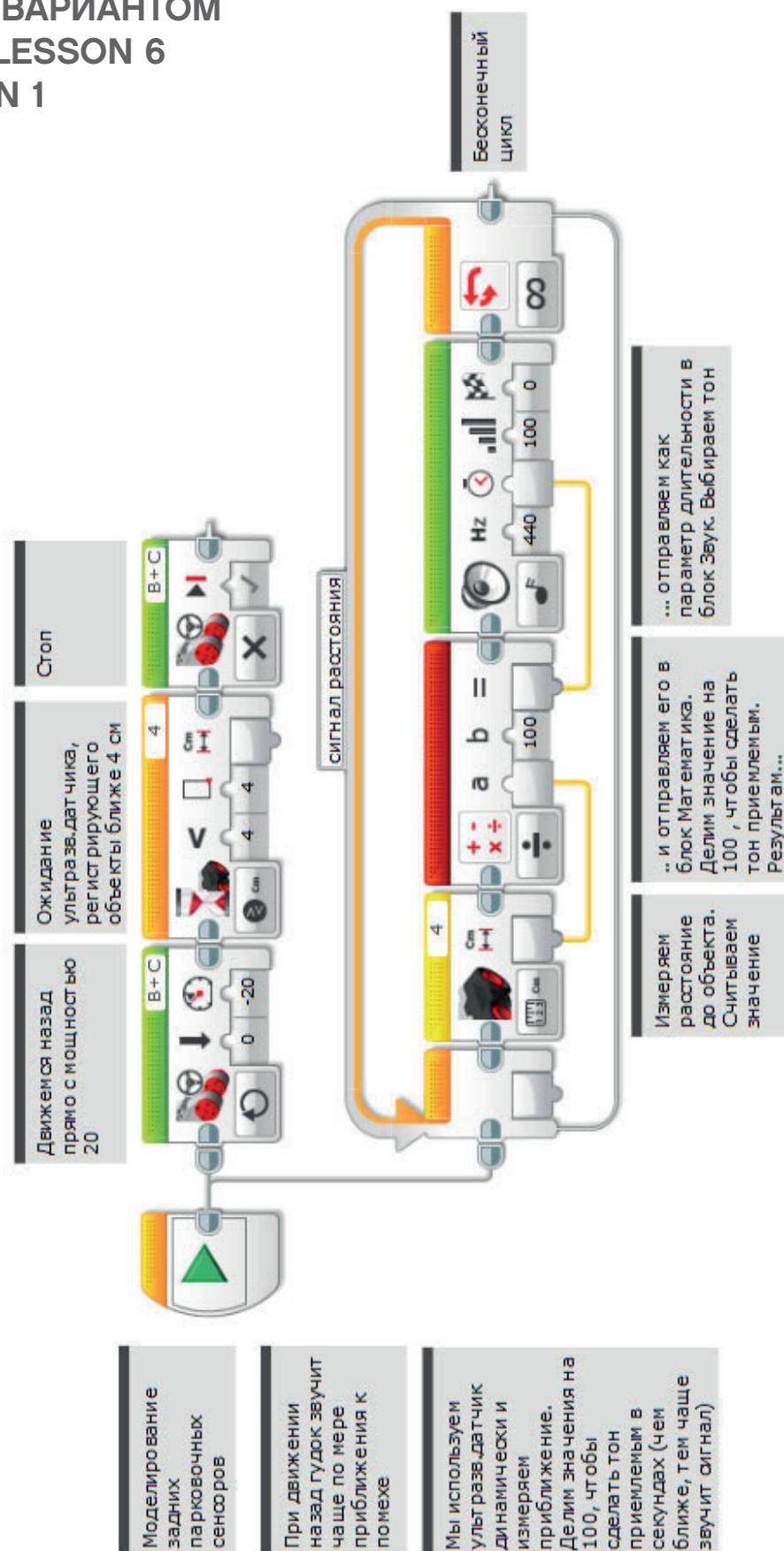
Приложение



Занятие 6

Приложение

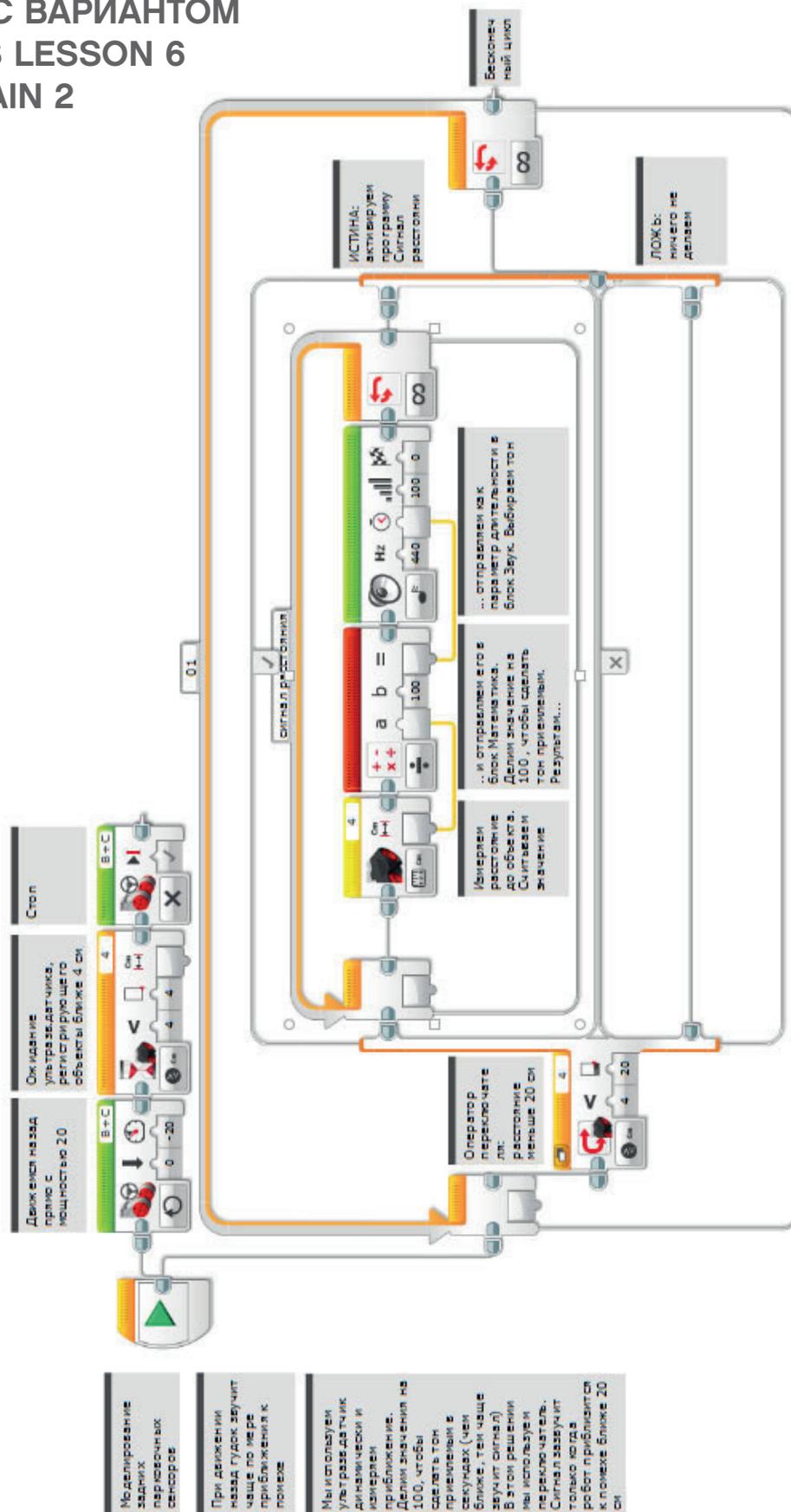
ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 6
ВКЛАДКА: MAIN 1



Занятие 6

Приложение

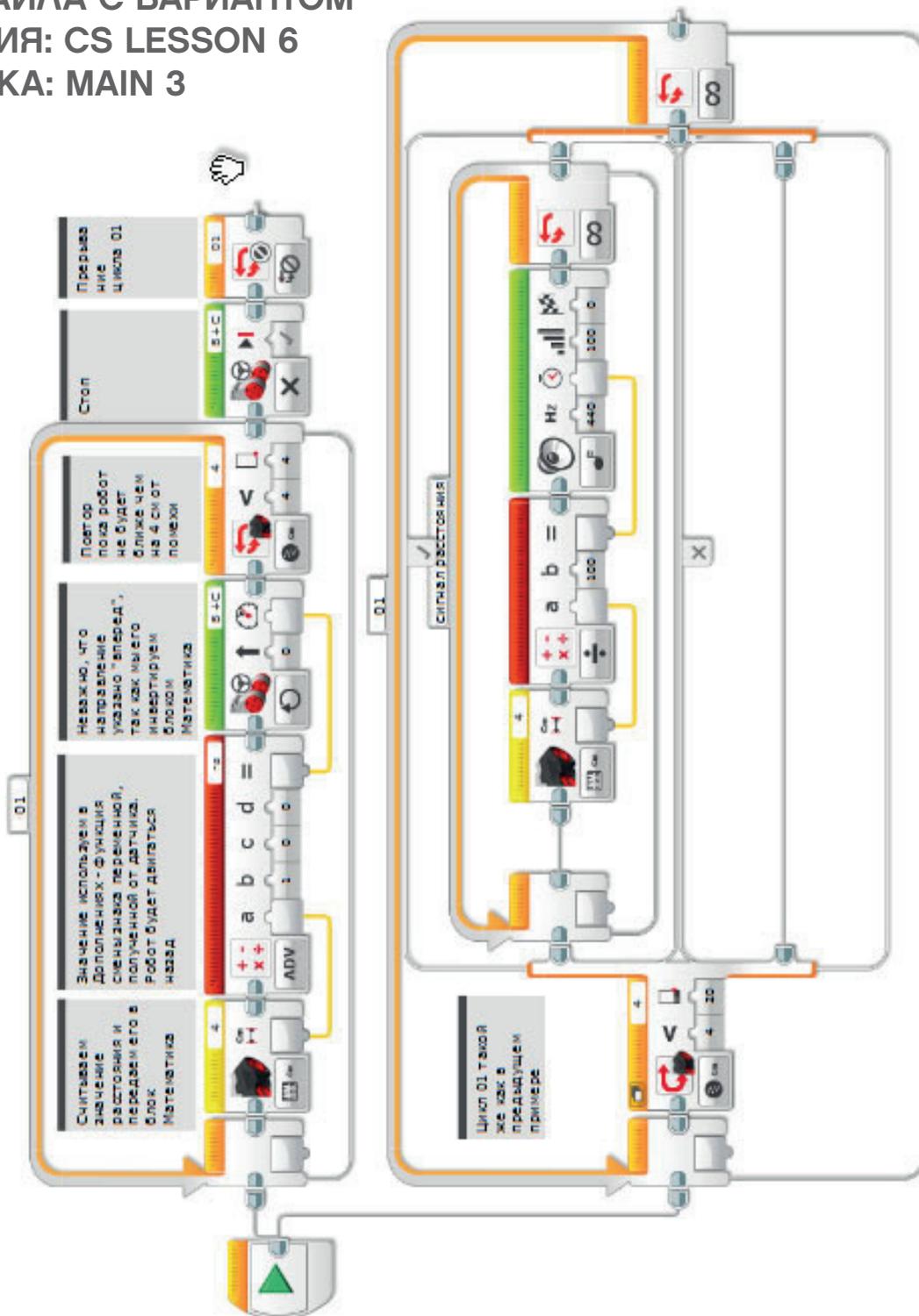
ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 6
ВКЛАДКА: MAIN 2



Занятие 6

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 6
ВКЛАДКА: MAIN 3



Моделирование задних парковочных сенсоров

При движении назад гудок звучит чаще по мере приближения к помехе.

Мы используем ультразвуковой датчик для измерения и в режиме приближения. Делим значения на 100, чтобы сделать тон приятным в секундах (чем ближе, тем чаще звучит сигнал). Сигнал звучит только тогда, когда робот приближается к помехе ближе 20 см.

В этом решении робот будет двигаться по мере приближения к помехе. Кроме того, когда он остановится за 4 см до объекта, сигнал также преобразит в.

Занятие 6

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session6_1.c

```
Activity6_1.c
1  #pragma config(StandardModel, "EV3_REMOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard          !**//
3
4  /*
5  Create a program to drive the robot backwards with a warning beep getting quicker
6  as the Ultrasonic Sensor sees an approaching object. The robot stops when a distance
7  of less than 4cm is reached.
8  */
9
10 task distanceBeeper()
11 {
12     while(true)
13     {
14         //Start playing a tone (in tens of milliseconds).
15         playTone(440, getUSDistance(sonarSensor));
16
17         // Delay while the sound plays.
18         while(bSoundActive)
19         {
20             sleep(10);
21         }
22
23         //Add 50ms of "silence" to avoid a constant tone.
24         sleep(50);
25     }
26 }
27
28 task main()
29 {
30     //Start the "distance beeper" task.
31     startTask(distanceBeeper);
32
33     // Set the motor speed.
34     setMotorSpeed(motorB, -20);
35     setMotorSpeed(motorC, -20);
36
37     // Wait while the distance is great than 4.
38     while(getUSDistance(sonarSensor) >= 4)
39     {
40         sleep(10);
41     }
42 }
43
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 6

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session6_2.c

```
Activity6_2.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard      !!**//
3
4  /*
5  Create a program to drive the robot backwards with a warning beep getting quicker
6  as the Ultrasonic Sensor sees an approaching object. The beep only sounds at a distance
7  less than 20cm. The robot stops when a distance of less than 4cm is reached.
8  */
9
10 task distanceBeeper ()
11 {
12     while(true)
13     {
14         if (getUSDistance(sonarSensor) < 20){
15             //Start playing a tone (length determined by distance).
16             playTone(440, (getUSDistance(sonarSensor)));
17
18             // Wait for the tone to be done playing.
19             while(bSoundActive)
20             {
21                 sleep(10);
22             }
23
24             //Add 50ms of "silence" to avoid a constant tone.
25             sleep(50);
26         }
27     }
28 }
29
30 task main()
31 {
32     //Start the "distance beeper" task.
33     startTask(distanceBeeper);
34
35     setMotorSpeed(motorB, -20);
36     setMotorSpeed(motorC, -20);
37
38     while(getUSDistance(sonarSensor) >= 4)
39     {
40         //Keep driving.
41         sleep(10);
42     }
43 }
44
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 6

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session6_3.c

```
Activity6_3.c
1  #pragma config(StandardModel, "EV3_REMBO")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard  !!*/
3
4  /*
5  Create a program to drive the robot backwards with a warning beep
6  getting quicker as the Ultrasonic Sensor sees an approaching object.
7  The beep only sounds at a distance less than 20cm. The robot
8  slows down as it nears the object. The robot stops when a distance
9  of less than 4cm is reached as does the beeper.
10 */
11
12 task distanceBeeper()
13 {
14     while(true)
15     {
16         if (getUSDistance(sonarSensor) < 20){
17
18             //Start playing a tone (length determined by distance).
19             playTone(440, (getUSDistance(sonarSensor)));
20
21             //Delay for the length of the tone.
22             //Wait while the sound is playing.
23             while (bSoundActive)
24             {
25                 sleep(10);
26             }
27
28             //Add 50ms of "silence" to avoid a constant tone.
29             sleep(50);
30         }
31     }
32 }
33
34 task main()
35 {
36     //Start the "distance beeper" task.
37     startTask(distanceBeeper);
38
39     while(SensorValue(sonarSensor) >= 4)
40     {
41         setMotorSpeed(motorB, -getUSDistance(sonarSensor));
42         setMotorSpeed(motorC, -getUSDistance(sonarSensor));
43         sleep(50);
44     }
45
46     //Set motor speed to 0 and stop the distanceBeeper loop.
47     setMotorSpeed(motorB, 0);
48     setMotorSpeed(motorC, 0);
49     stopTask(distanceBeeper);
50 }
51
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 7

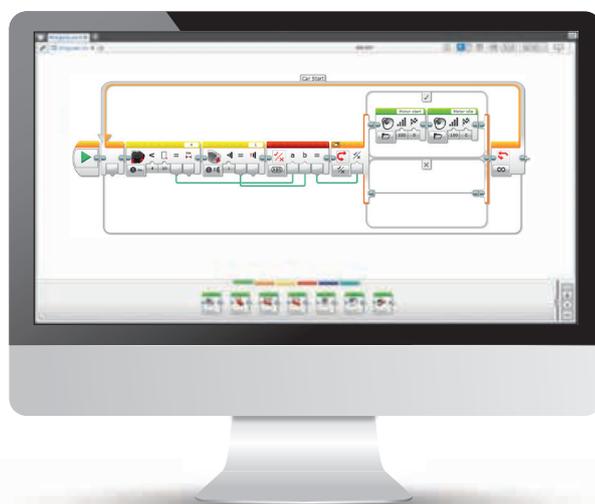
Пуск автомобиля без ключа



Занятие 7

Пуск автомобиля без ключа

Знаете ли вы, что автомобиль теперь можно завести без ключа? "Ключ зажигания", или брелок остается в вашем кармане или сумочке, а двигатель запускается дистанционно. Водителю нужно всего лишь нажать на несколько кнопок и выжать сцепление для запуска автомобиля. Ученики ищут способы реализовать такие же возможности для своего робота при помощи комбинации датчиков и алгебраической логики. Например, находится ли что-либо в диапазоне чувствительности ультразвукового датчика и было ли нажатие на датчик касания? Если ответ на оба вопроса "истинно", программа запустит соответствующее действие. Ученики будут использовать такую логику на занятии для управления колесным роботом.



Занятие 7

План урока



РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание нескольких ключевых алгоритмов, отражающих алгоритмическое мышление;
- понимание простейшей булевой логики (например, операций “И”, “ИЛИ” и “НЕТ”) и некоторые варианты ее применения в схемах и программах;
- использование блока логики в сочетании с блоком переключения;
- применение сочетания нескольких датчиков для запуска программы интеллектуального блока EV3.

ОСНОВНЫЕ ПОНЯТИЯ

Вход, выход, булева логика, алгебраический, переключатель, истинно, ложно.

ВВЕДЕНИЕ

- Объясните ученикам, что на уроке они должны настроить робота так, чтобы он двигался, как настоящий автомобиль, который запускается нажатием кнопки.
- Ранее при программировании мы в основном использовали один датчик за раз. Но как насчет использования двух датчиков? Для чего это может понадобиться? Напомните детям, что в охранной системе автомобиля сигнализация может активироваться разными датчиками. Но что делать, если два датчика одновременно посылают сообщения?
- Современные машины, запускающиеся без ключа, требуют трех действий для запуска. Каковы они? Ученики должны назвать следующие три условия запуска: ключ, выжимание сцепления и нажатие кнопки. Как можно реализовать это для колесного робота? Для этого необходимо изучить логику.
- Наша задача сегодня – составить программу, которая включает управление роботом только после активации правильной пусковой комбинации.

ВВЕДЕНИЕ В ЗАДАНИЕ НА ИССЛЕДОВАНИЕ

- Ученики в парах или маленьких группах собирают модуль датчика касания (Учебник по робототехнике > Самоучитель > Инструкции по сборке > Датчик касания – Мобильная база)
- Также в парах или маленьких группах ученики собирают модуль ультразвукового датчика (Учебник по робототехнике > Самоучитель > Инструкции по сборке > Ультразвуковой датчик – Мобильная база)



Занятие 7

План урока



ОСНОВНОЕ ЗАДАНИЕ 1

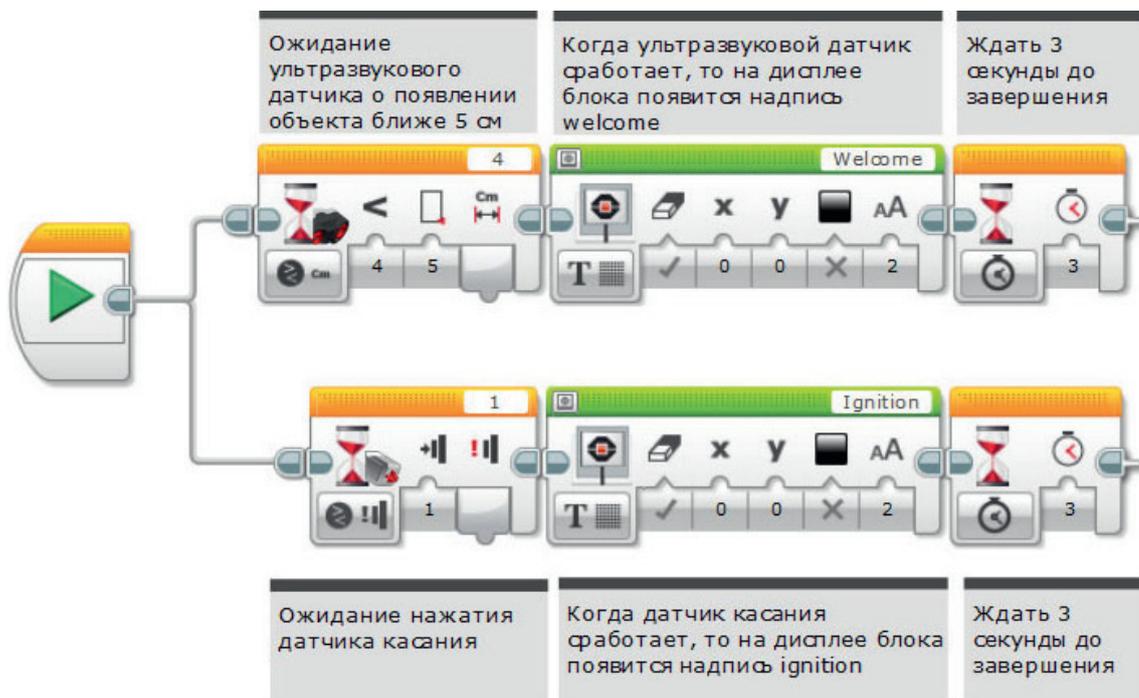
- Обучающимся требуется некоторый опыт работы с несколькими датчиками. Выделите время на уроке для этой цели.
- Убедитесь, что к базовой модели подключен как ультразвуковой датчик, так и датчик касания. Соответствующие схемы сборки можно найти в руководстве по сборке или в руководствах по сборке учебника по робототехнике. Поясните классу, что они должны составить программу, которая будет отображать сообщения на дисплее интеллектуального блока EV3 при срабатывании одного из датчиков.
- После того, как робот начнет отображать сообщения, попросите учеников поместить программу в цикл, чтобы сообщения повторялись.



Занятие 7

План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 7
ВКЛАДКА: MAIN 1



ОСНОВНОЕ ЗАДАНИЕ 2

- Опросите класс по первой программе. Спросите учеников, как они смогли включить два датчика в последовательность программы. Объясните ученикам, что им нужно использовать блок под названием "блок логических операций".
- Для примера необходимо посмотреть на диаграмму Венна. Расскажите о том, что блок логических операций похож по принципу действия на диаграмму Венна. Начертите диаграмму на доске в виде двух пересекающихся кругов, либо найдите пример в интернете. Объясните принцип действия. Обратите внимание класса на ряд условий, заданный в диаграмме, и связь с логическими операциями в программе. Поясните следующую иллюстрацию и принцип получения результатов.



Занятие 7

План урока

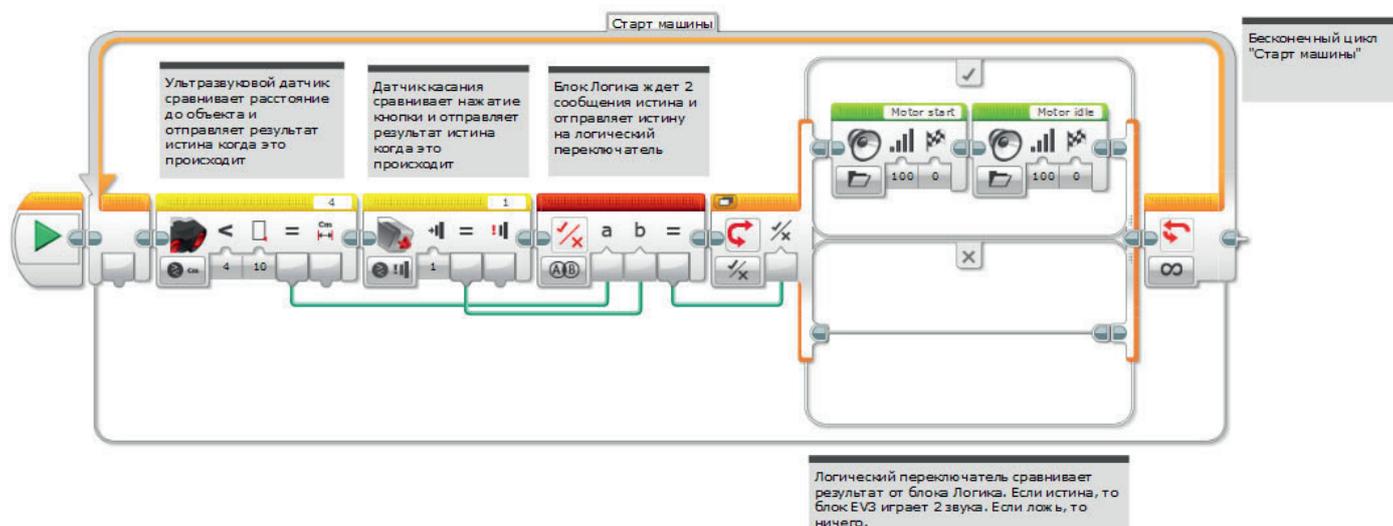


Modes	Inputs Used	Result
И	A, B	Истинно, если A и B являются истинными, в противном случае – ложно
ИЛИ	A, B	Истинно, если A или B (или и то, и другое) являются истинными, ложно, если A и B являются ложными
исключающее ИЛИ	A, B	Истинно, если только одно условие из A и B истинно, ложно, если A и B являются истинными, ложно, если A и B являются ложными
НЕ	A	Истинно, если A ложно, ложно, если A истинно

ОСНОВНОЕ ЗАДАНИЕ 2

- Блок логических операций программного обеспечения EV3 генерирует истинный или ложный результат на выходе, в зависимости от условий, установленных в блоке. Большинство пользователей будет применять его для A и B, задавая истинный результат. Именно так будем делать сегодня и мы.
- В примере ниже ультразвуковой датчик имитирует кнопки на брелоке в кармане, которыми пользуется находящийся в машине водитель. Такое включение напоминает проникновение в мыльный пузырь. Датчик касания используется для запуска автомобиля. Попросите класс составить такую программу.

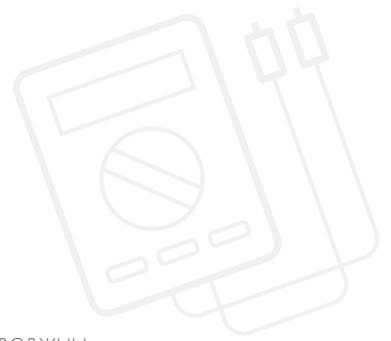
ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 7 ВКЛАДКА: MAIN 2



- Спросите учеников, так ли все происходит в реальной жизни.

Занятие 7

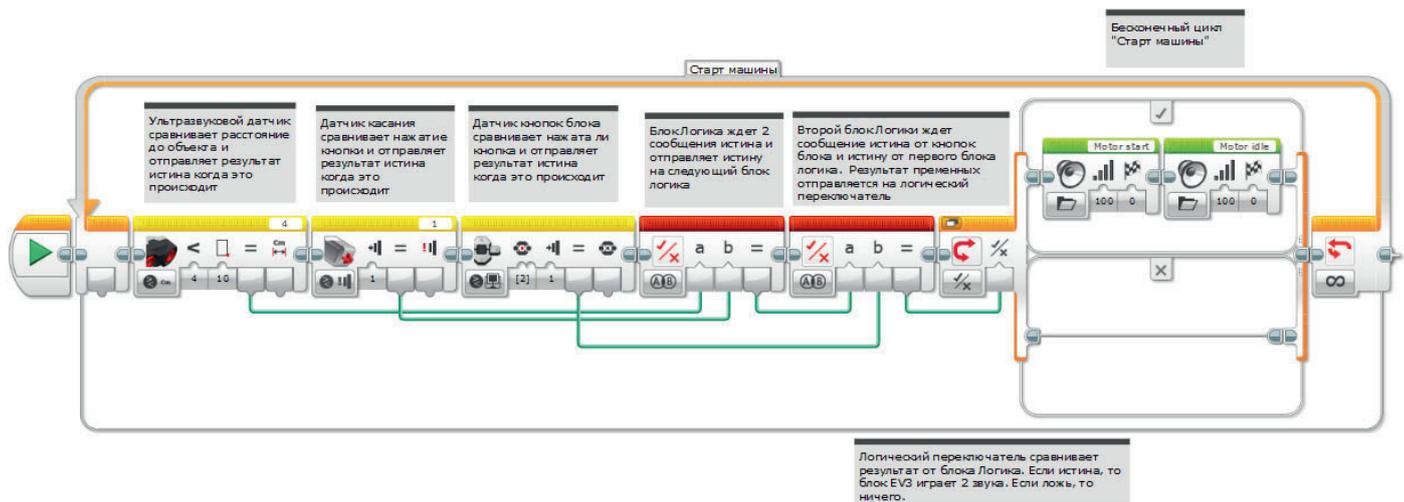
План урока



ОСНОВНОЕ ЗАДАНИЕ 3

- Проверьте, понимают ли обучающиеся, что для запуска автомобиля три условия должны быть истинными.
- В первую очередь, ключ или брелок должен оказаться в машине.
- Во-вторых, необходимо отжать сцепление.
- В-третьих, кнопкой включается зажигание.
- Дайте классу задание составить программу, имитирующую эти действия. Какие блоки нам нужны?
- Примечание: для пуска некоторых автомобилей требуется только два из вышеперечисленных условий. Можно также дать задание классу /группам составить программу только с двумя условиями.

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 7 ВКЛАДКА: MAIN 3



ПОДВЕДЕНИЕ ИТОГОВ

- Что ученики могут сказать о своем опыте использования каналов передачи данных? Попросите их поделиться своими впечатлениями. Обратите их внимания на то, что условные изображения каналов передачи данных в виде проводов можно перемещать и упорядочивать, что помогает сделать программы более аккуратными.
- Возможный вариант усложнения: составить программу, которая выводит на экран сообщение об ошибке, если условия включения автомобиля не выполняются.
- Повторите с классом материал по применению блока ожидания в программировании, отметив, что он в основном используется для одного датчика одновременно. Блок логических операций позволяет одновременно использовать несколько датчиков.
- Попросите класс сравнить текстовые версии задач с версиями в программе EV3. Пусть они отслежат ход каждого из решений, чтобы понять, как работают обе версии. Попросите учеников записать мысли по поводу использования двух разных языков в рабочих карточках.

Занятие 7

Рабочие карточки учеников



ЗАДАЧИ НА СЕГОДНЯ

Вы должны создать для колесного робота систему ввода информации без использования ключа. При активации определенной комбинации датчиков должна выполняться программа движения. Для успешного решения задачи вам потребуется ряд датчиков, а задачи 2 и 3 предусматривают использование одного или нескольких блоков логических операций. На это листе ответов нет. Здесь предложены программные блоки для вашего рассмотрения, с помощью которых можно решить задачу. На странице есть свободное место, куда где вы можете записать псевдокод, а также свои наблюдения.

ЗАДАЧА 1

Запрограммируйте робота на отображение текста "Welcome" ("Добро пожаловать!") при обнаружении предмета ультразвуковым датчиком и текста "Ignition" ("Зажигание") при нажатии на датчик касания.

Полезная информация: ультразвуковой датчик должен быть установлен на параметр "меньше" (<).

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 7

Рабочие карточки учеников



ЗАДАЧА 2

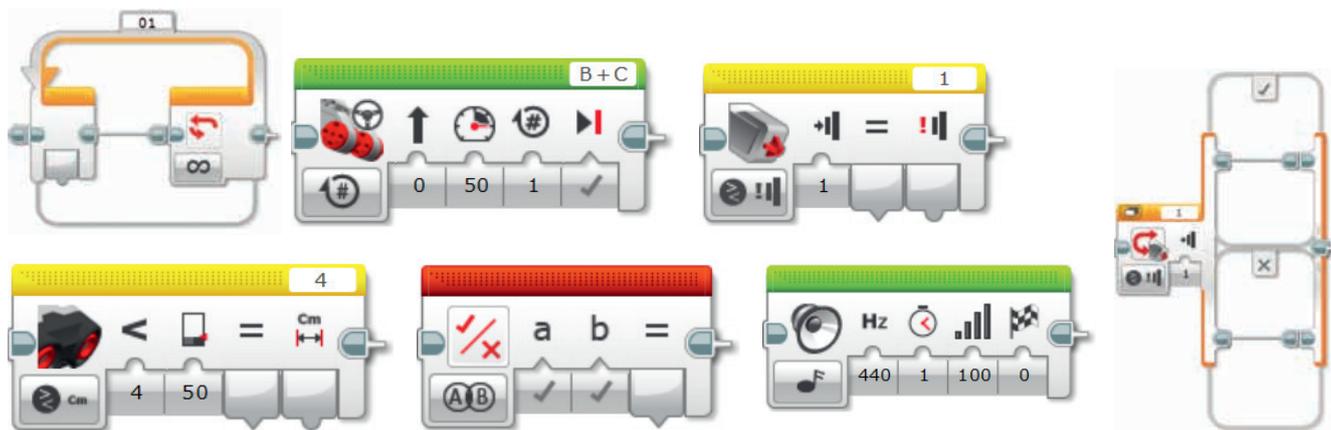
Задача 2 основывается на использовании блока логических операций. Также она требует, чтобы два датчика работали вместе, направляя данные в другой блок.

Как же действуют автомобили, запускающиеся без ключа? Для решения этой задачи примем датчик касания за “зажигание”, а ультразвуковой датчик будет определять наличие ключа в машине. Для пуска колесного робота необходимо правильно активировать оба датчика.

Теперь вы знаете, как использовать несколько датчиков в программах. Вам понадобятся блоки датчиков (желтые) для создания логики для блока логических операций. Каждый блок датчика используется для создания выходных данных со статусом “истинно”. Эти выходные данные направляются от блока датчика к блоку логических операций. При выполнении нужных условий выходные данные из блока логических операций могут направляться в блок переключения. В этой программе два блока датчика посылают данные в блок логических операций.

Используемые блоки

Используйте те же блоки, что и в задаче на программирование 1, но также не забывайте об этих:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 7

Рабочие карточки учеников



ЗАДАЧА 3

Теперь колесного робота необходимо запрограммировать так, чтобы он реагировал на выполнение условий тремя разными датчиками. Используются следующие датчики:

- датчик касания = зажигание;
- ультразвуковой датчик = обнаружение ключа в машине;
- кнопки интеллектуального блока = сцепление.

Блок логических операций может принимать входные данные из двух источников. Но что если нам нужны три источника? Подумайте, как можно использовать два логических блока для реализации такого варианта.

Данные от двух датчиков направляются в первый блок логических операций. Далее он посылает выходные данные в следующий блок логических операций, который обслуживает третий источник (датчик).

Далее полученный результат направляется к блоку переключения.

Используемые блоки

Используйте те же блоки, что и в задачах на программирование 1 и 2, но также не забывайте о следующих пунктах:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 7

Рабочие карточки учеников



По завершении занятия по программированию важно записать свои мысли и наблюдения. Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?
- Если сравнить текстовое и визуальное программирование, структуру каких программ легче отслеживать? Попробуйте написать программу в другой среде и сравните, какой вариант эффективнее.

Мысли и наблюдения

Занятие 7

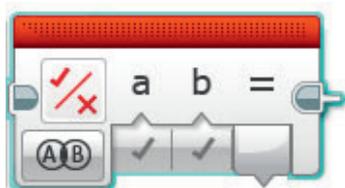
Примечания для учителя

РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами.

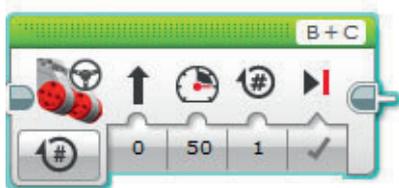
НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Расширенные знания > Логика

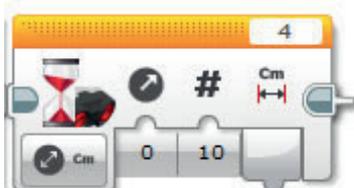


РАНЕЕ РАССМОТРЕННЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Основы > Движение по прямой



Основы > Остановиться у объекта



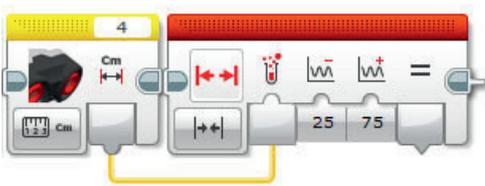
Расширенные знания > Цикл



Расширенные знания > Переключатель



Расширенные знания > Шины данных



Расширенные знания > Блоки датчиков



Занятие 7

Приложение

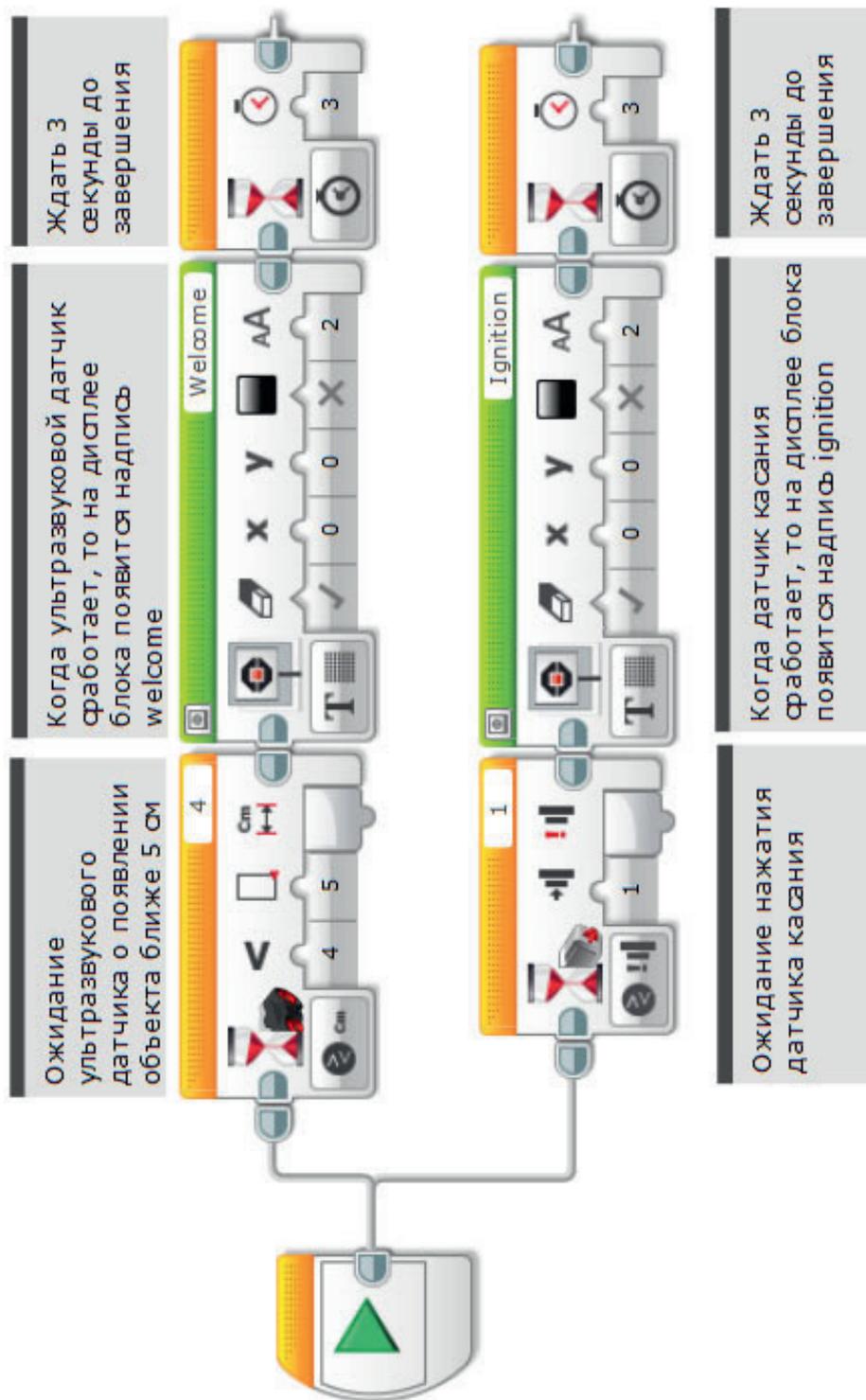
ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 7:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 7

Приложение

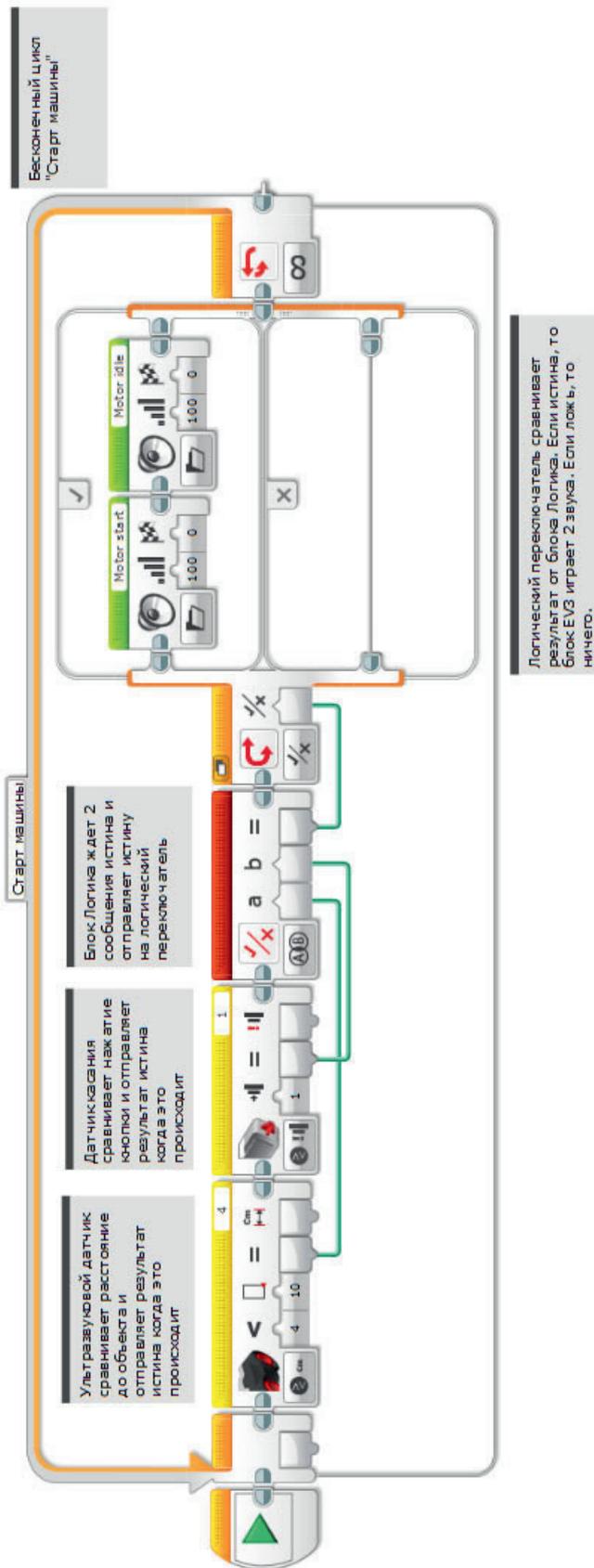
ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 7
ВКЛАДКА: MAIN 1



Занятие 7

Приложение

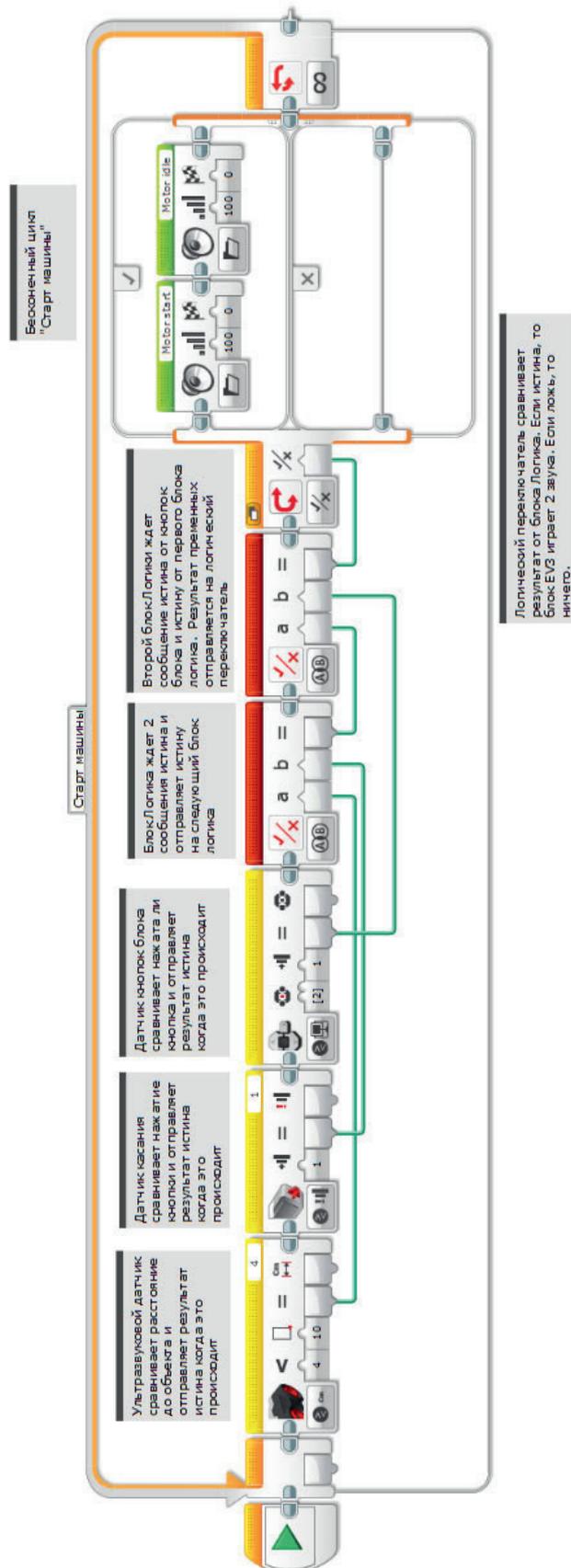
ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 7
ВКЛАДКА: MAIN 2



Занятие 7

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 7
ВКЛАДКА: MAIN 3



Занятие 7

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session7_1.c

```
Activity7_1.c*
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!*/
3
4  /*
5  Create a program that shows how a keyless ignition works.
6  The Ultrasonic Sensor welcomes the driver. The Touch Sensor works
7  the ignition.
8  */
9
10 task touchTask()
11 {
12     //Wait for the touch sensor to be pressed.
13     while(getTouchValue(touchSensor) == 0)
14     {
15         //Do Nothing.
16         sleep(10);
17     }
18
19     //Display Text on the LCD Screen.
20     eraseDisplay();
21     displayCenteredBigTextLine(4, "Ignition");
22
23     //Display text for 3 seconds.
24     sleep(3000);
25 }
26
27 task main()
28 {
29     //Start the second task to monitor the touch sensor.
30     startTask(touchTask);
31
32     //Wait for the sonar sensor to see an object that is less than
33     //or equal to 5cm away.
34     while(getUSDistance(sonarSensor) >= 5)
35     {
36         //Do Nothing.
37         sleep(10);
38     }
39
40     //Display text on the LCD Screen.
41     eraseDisplay();
42     displayCenteredBigTextLine(4, "Welcome");
43
44     //Display text for 3 seconds.
45     sleep(3000);
46 }
47
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 7

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session7_2.c

```
Activity7_2.c*
1  #pragma config(StandardModel, "EV3_REMBOТ")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard  !!*/
3
4  /*
5  Create a program that starts the engine running when two conditions
6  are met. The Sonar Sensor detects the driver in the seat and
7  the Touch Sensor is the press of the ignition.
8  */
9
10 task main()
11 {
12     //Repeat continuously.
13     while(true)
14     {
15         //If Sonar Sensor is less than 10 AND Touch Sensor is pressed.
16         if(getUSDistance(sonarSensor) < 10 && getTouchValue(touchSensor) == 1)
17         {
18             playSoundFile("Motor start");
19
20             // Wait for the tone to be done playing.
21             while(bSoundActive) sleep(10);
22
23
24             //After the engine starts - keep playing the sound.
25
26             playSoundFile("Motor idle");
27
28             // Wait for the tone to be done playing.
29             while(bSoundActive) sleep(10);
30         }
31     }
32 }
33
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Activity 7

Appendix

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session7_3.c

```
Activity7_3.c
1  #pragma config(StandardModel, "EV3_REMBO")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard  !!*/
3
4  /*
5  Create a program that starts the engine running when three conditions
6  are met. The Sonar Sensor detects the key in the car, the Button
7  press represents the clutch being pressed and the Touch Sensor
8  is the press of the ignition.
9  */
10
11 task main()
12 {
13     //Repeat continuously.
14     while(true)
15     {
16         //If Sonar Sensor is less than 10 AND Touch Sensor is pressed |
17         //AND Enter Button is pressed.
18         if(getUSDistance(sonarSensor) < 10 &&
19             getTouchValue(touchSensor) == 1 &&
20             getButtonPress(buttonEnter) == 1)
21         {
22             playSoundFile("Motor start");
23             // Wait for the tone to be done playing.
24             while(bSoundActive) sleep(10);
25
26             //After the engine starts - keep playing the sound.
27             playSoundFile("Motor idle");
28
29             // Wait for the tone to be done playing.
30             while(bSoundActive) sleep(10);
31         }
32     }
33 }
34
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 8

Круиз-контроль



Занятие 8

Крузи-контроль

На этом занятии ученики познакомятся с понятием переменных. К концу занятия они создадут систему крузи-контроля для колесного робота.

При нажатии на датчик касания скорость колесного робота будет увеличиваться.

Обратите внимание класса на то, что в настоящих машинах, несмотря на автоматический крузи-контроль, скорость можно легко увеличивать или уменьшать за счет функций ступенчатого управления скоростью (обычно водитель управляет ступенчатым изменением скорости с помощью специальных кнопок на руле). Такие ситуации можно моделировать при помощи блока переменных. В нашем случае скорость будет устанавливаться нажатием на датчик касания.



Занятие 8

План урока



РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- понимание нескольких ключевых алгоритмов, отражающих алгоритмическое мышление;
- использование блока переменных для хранения информации;
- разработка многоуровневых программ;
- создание раздела “Мои блоки”.

ОСНОВНЫЕ ПОНЯТИЯ

Вход, выход, переменная, постоянная, цикл, ожидание, мотор, датчик касания.

ВВЕДЕНИЕ

- Посмотрите с классом видеосюжет о движении автомобиля с применением круиз-контроля. Спросите учеников, что там происходит. Обратите их внимание на то, что бортовой компьютер автоматически регулирует скорость автомобиля, что немного похоже на работу автопилота в самолете.
- Используйте для пояснений иллюстрации ниже.
- Спросите детей, для чего, по их мнению, используются кнопки “плюс” и “минус”. Помогите им догадаться, что с их помощью можно регулировать скорость автомобиля.
- Объясните, что при решении сегодняшней задачи ученики будут использовать два датчика касания для регулирования и поддержания скорости колесного робота.



Рисунок 1



Рисунок 2

Занятие 8

План урока

- Посмотрите на рисунок 2. Для чего нужны кнопки “ускорение” и “замедление”? Ученики должны ответить, что они ускоряют и замедляют ход автомобиля.
- Сообщите им, что на занятии они научатся работать с блоком переменных.
- Может потребоваться объяснение ученикам разницы между постоянной и переменной: постоянные используются, когда одно и то же значение постоянно повторяется в программе. Эти фиксированные значения могут меняться только пользователем, в момент, когда программа не выполняется.
- Можно продемонстрировать классу блок постоянных в действии совместно с блоком движения и блоком управления дисплеем.
- Переменная – это способ хранения значений в программе, которые могут в этой программе использоваться. Различие состоит в том, что в данном случае значение может многократно заменяться на другое по мере выполнения программы.
- Покажите классу, как пользоваться блоком переменных, и попросите их составить программу с переменной.

ОСНОВНОЕ ЗАДАНИЕ 1

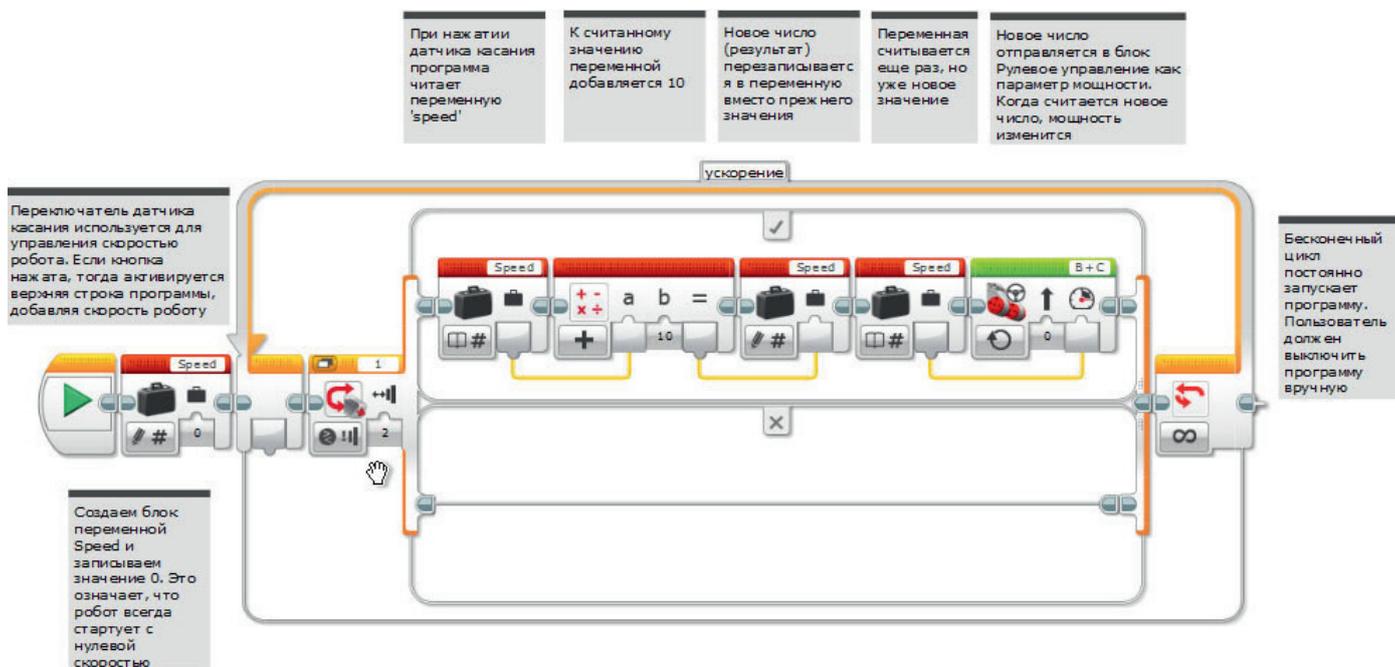
- На данный момент ученики должны быть знакомы с мобильной базовой моделью из учебника по робототехнике. Если она еще не собрана, то нужно ее собрать на текущем этапе. Помните, что ученикам нужно установить на модель два датчика касания, направленных вперед.
- Расширьте представление учеников о блоке переменных. Объясните, что это блок программирования, в котором могут храниться данные (текст, логика, число или массивы), которые могут быть переписаны в любой момент по ходу выполнения программы.
- Информацию в блоке можно считывать и записывать при помощи одного из трех других блоков: математического блока, текстового блока или блока операций над массивами.
- Работа необходимо запрограммировать так, чтобы после начала движения его можно было ускорить, нажав на датчик касания. Требуется включить в цикл функцию ожидания для датчика касания.



Занятие 8

План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 8
ВКЛАДКА: MAIN 1



ОСНОВНОЕ ЗАДАНИЕ 2

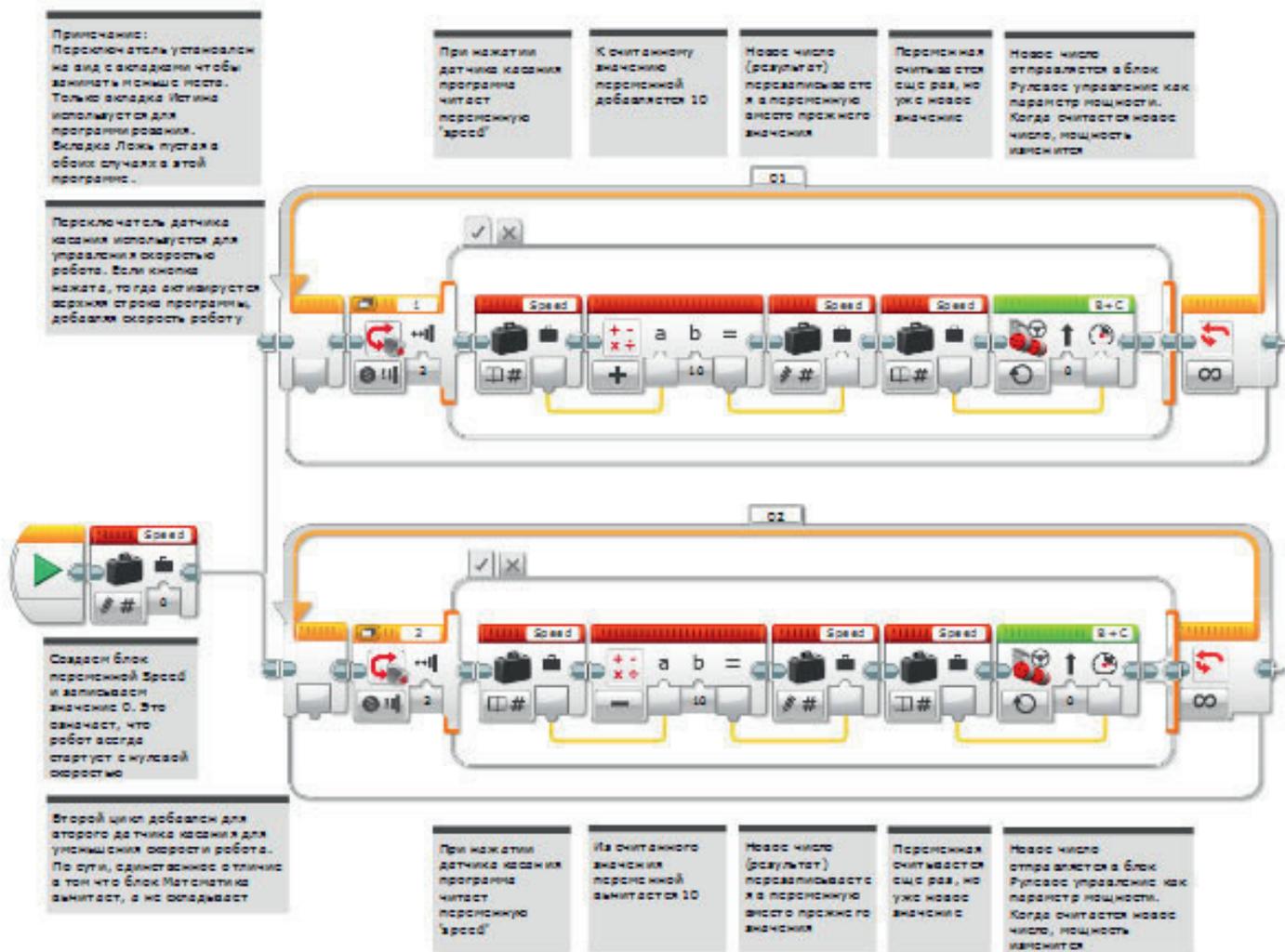
- После составления первой программы, когда робот сможет ускориться, предложите ученикам подумать, как изменить программу для замедления робота.
- Выслушав идеи учеников, помогите им догадаться, что одним из решений может быть внедрение второго бесконечного цикла, при этом просто изменится порт датчика касания (добавление другого датчика) с переходом математического блока на вычитание вместо сложения.
- Напомните ученикам о многозадачности и предупредите, что при выполнении задания они должны использовать собственные знания о таком типе программирования. Сообщите, что блок цикла необходимо перетащить в зону программирования до подсоединения второго канала передачи данных от блока пуска.



Занятие 8

План урока

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 8
ВКЛАДКА: MAIN 2



Занятие 8

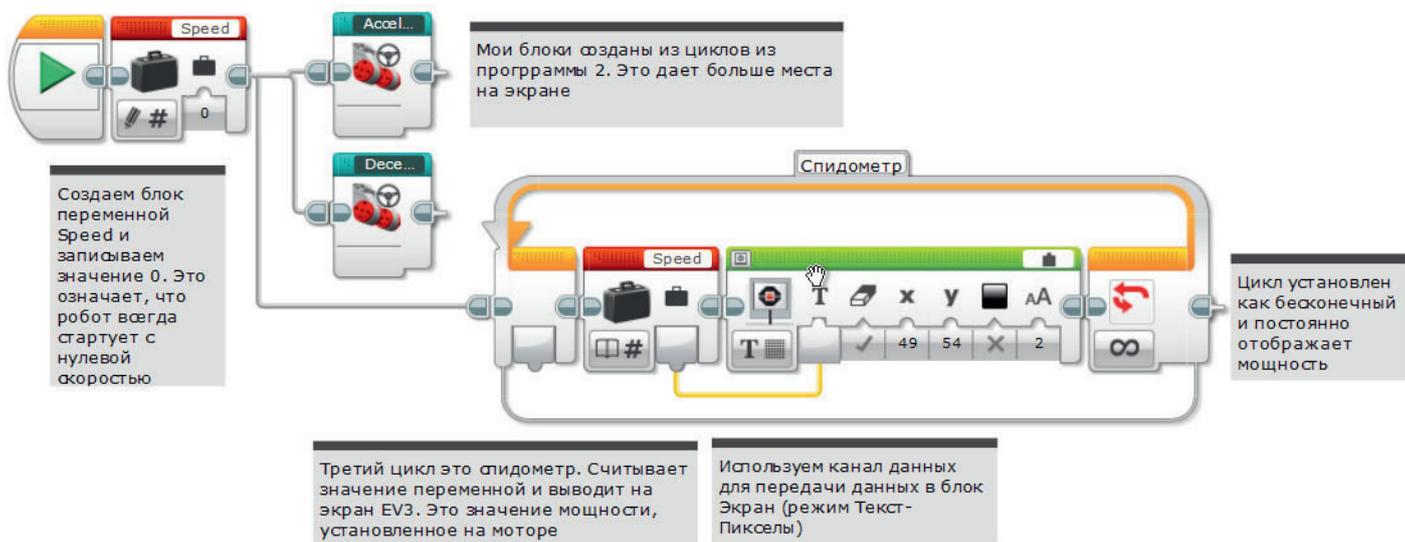
План урока



ОСНОВНОЕ ЗАДАНИЕ 3

- После того, как робот сможет ускорять и замедлять ход по нажатию на кнопку (или две), можно усложнить программу таким образом, чтобы на дисплей интеллектуального блока EV3 выводилась информация, насколько быстро движется колесный робот, что еще больше приблизит модель к реальному автомобилю.
- Попросите учеников подумать над использованием блока переменных. Могут ли они использовать его для отображения информации?
- Одно из возможных решений представлено ниже.
- На этом занятии класс научится создавать раздел "Мои блоки". Два таких раздела присутствуют в решении ниже. Раздел "Мои блоки" позволяет добавлять подпрограммы в уже написанные программы. В примере ниже были взяты контуры ускорения и замедления, из которых был составлен раздел "Мои блоки". Такое решение имеет две причины: во-первых, экономия пространства, во-вторых, обеспечения возможности использования данных подпрограмм в других программах.

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 8 ВКЛАДКА: MAIN 3



Занятие 8

План урока

ПОДВЕДЕНИЕ ИТОГОВ

- Спросите класс как можно использовать блок постоянных для регулирования скорости. Сравните функционирование этого блока с работой блока переменных и добейтесь от учеников ответа, что фиксированные значения могут пригодиться, чтобы заставить робота соблюдать "ограничения скорости".
- При активации первого датчика касания может использоваться переменная до 30%, что можно сравнить с пределом 30 км/час на дороге. Для второго датчика касания можно установить предел скорости 70% через другой блок постоянных. Пусть ученики также попробуют использовать константы вместо переменных.
- Попросите класс сравнить текстовые версии задач с версиями в программе EV3. Пусть они отслежат ход каждого из решений, чтобы понять, как работают обе версии. Попросите учеников записать мысли по поводу использования двух разных языков в рабочих карточках.



Занятие 8

Рабочие карточки учеников



ЗАДАЧИ НА СЕГОДНЯ

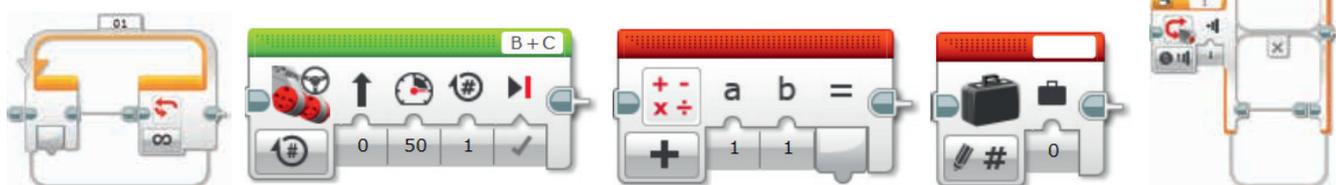
Сегодня вы создадите для вашего робота систему круиз-контроля, подобную тем, что присутствуют во многих современных машинах. Вам потребуются два датчика касания из конструктора EV3 для имитации кнопок на руле автомобилей с круиз контролем.

ЗАДАЧА 1

Запрограммируйте автомобиль на ускорение шагами по 10. Используйте блок переменных как уставку скорости, допускающую прибавление к значению.

Полезная информация: убедитесь, что блок движения находится в режиме "включено" (On).

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 8

Рабочие карточки учеников



ЗАДАЧА 2

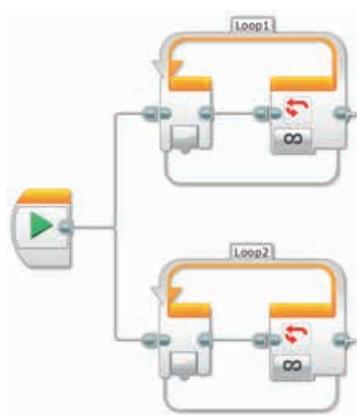
У вас уже есть программа, которая может ускорять движение колесного робота. Теперь нужно составить новый подраздел, который будет его замедлять. Для этого нужно просто добавить второй цикл и блок переключения.

В дополнительный цикл включается блок второго датчика касания и математической блок, настроенный на вычитание, а не на сложение.

Не забывайте, что вы должны использовать многозадачность, когда две ветки программы выполняются одновременно.

Используемые блоки

Используйте те же блоки, что и в задаче на программирование 1, но также не забывайте об этом:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 8

Рабочие карточки учеников



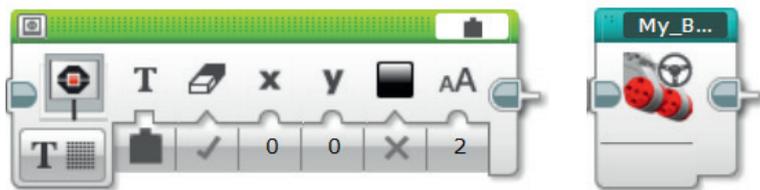
ЗАДАЧА 3

Скорость колесного робота теперь можно регулировать двумя датчиками касания, но также удобной была бы функция определения скорости (мощности мотора) с отображением ее значения на дисплее интеллектуального блока EV3. Учитель покажет вам, как составить раздел “Мои блоки” из уже написанных вами программ. Такое решение имеет два полезных качества. Во-первых, оно позволяет экономить место на экране программирования, во-вторых, эти подпрограммы можно использовать повторно в других ваших программах, поскольку они хранятся в отдельной категории на панели программирования.

Чтобы составить программу для визуального отображения значений мощности, снимите значение переменной, управляющей мощностью мотора, и отобразите его на интеллектуальном блоке EV3 при помощи блока управления дисплеем, настроенного на режим “текст – пиксели”.

Используемые блоки

Используйте те же блоки, что и в задачах на программирование 1 и 2, но также не забывайте о следующих двух:



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 8

Рабочие карточки учеников



По завершении занятия по программированию важно записать свои мысли и наблюдения. Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?

Мысли и наблюдения

Занятие 8

Примечания для учителя

РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

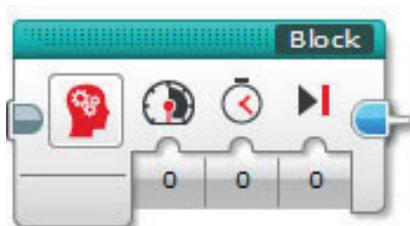
Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами.

НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Более сложные действия >
Переменные

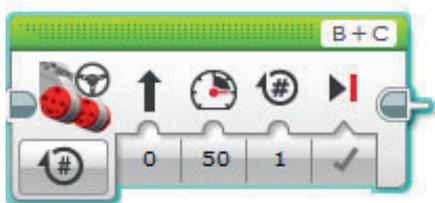


Инструменты > Мои блоки



РАНЕЕ РАССМОТРЕННЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Основы > Движение по прямой



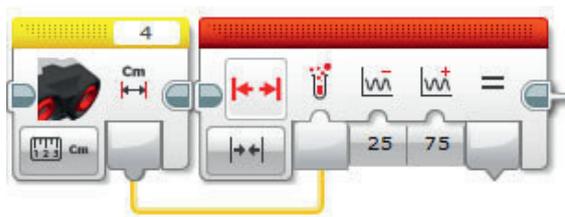
Более сложные действия > Многозадачность



Более сложные действия > Цикл



Более сложные действия > Шины данных



Более сложные действия > Математика – Основы



Занятие 8

Приложение

ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 8:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 8

Приложение

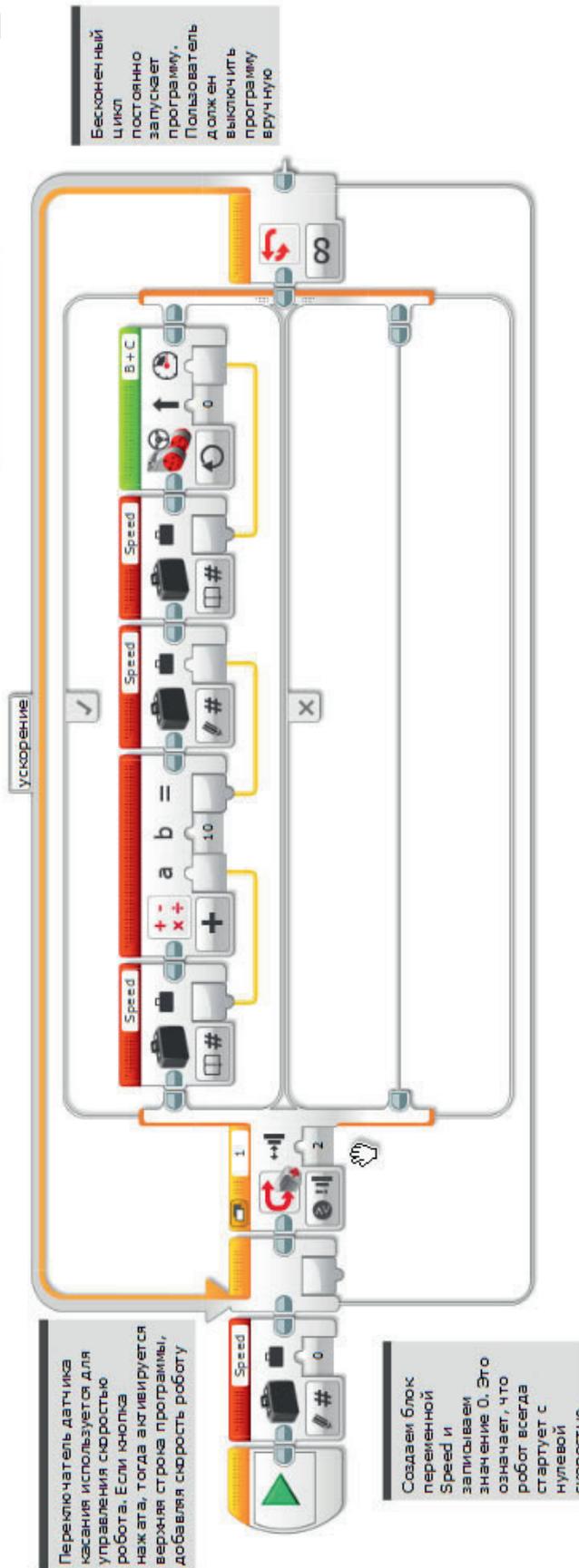


Занятие 8

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 8
ВКЛАДКА: MAIN 1

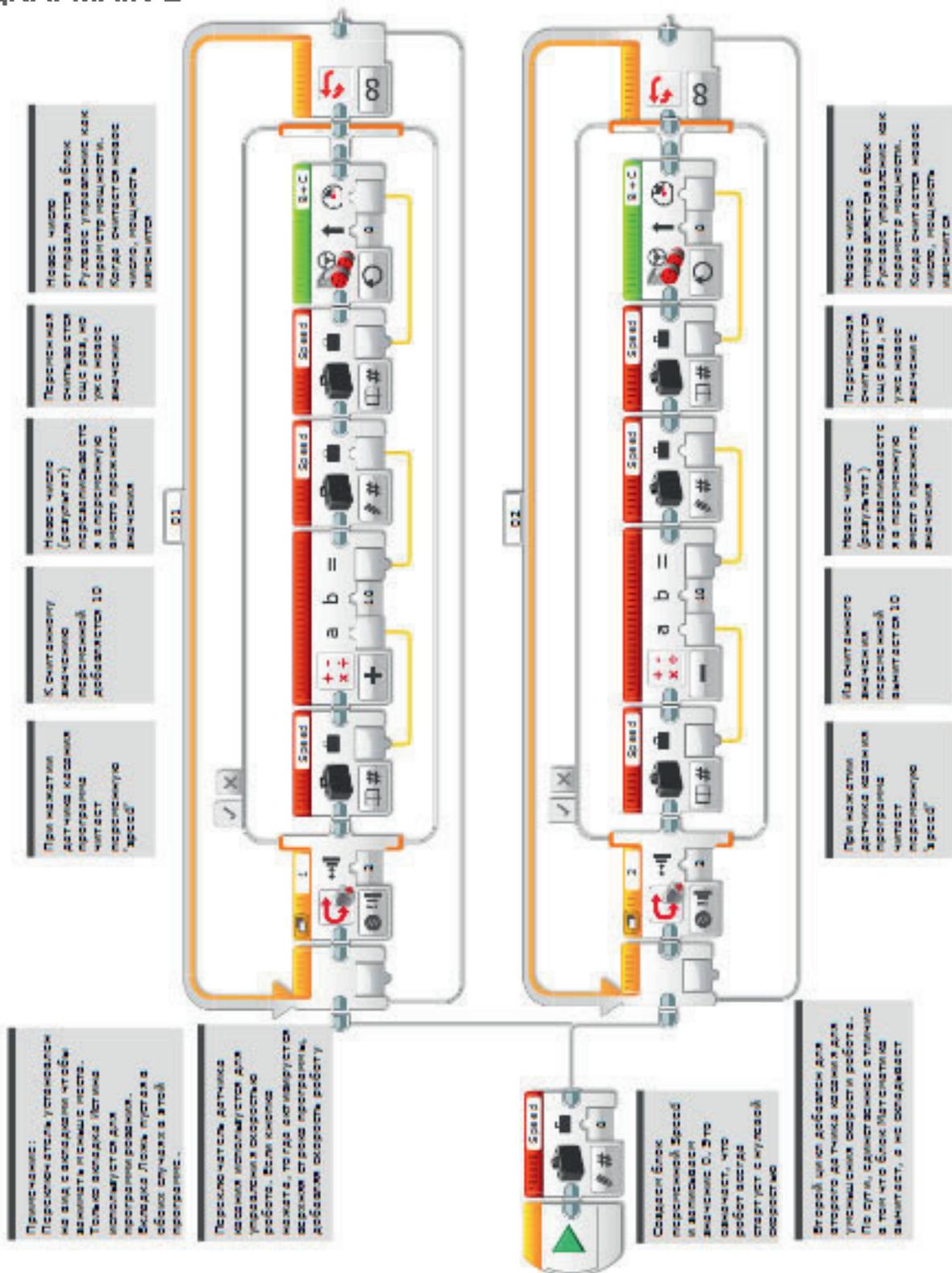
- При нажатии датчика касания программа читает переменную 'speed'
- К считанному значению переменной добавляется 10
- Новое число (результат) перезаписывается в переменную вместо прежнего значения
- Переменная считывается еще раз, но уже новое значение
- Новое число отправляется в блок Рулевого управления как параметр мощности. Когда считывается новое число, мощность изменится



Занятие 8

Приложение

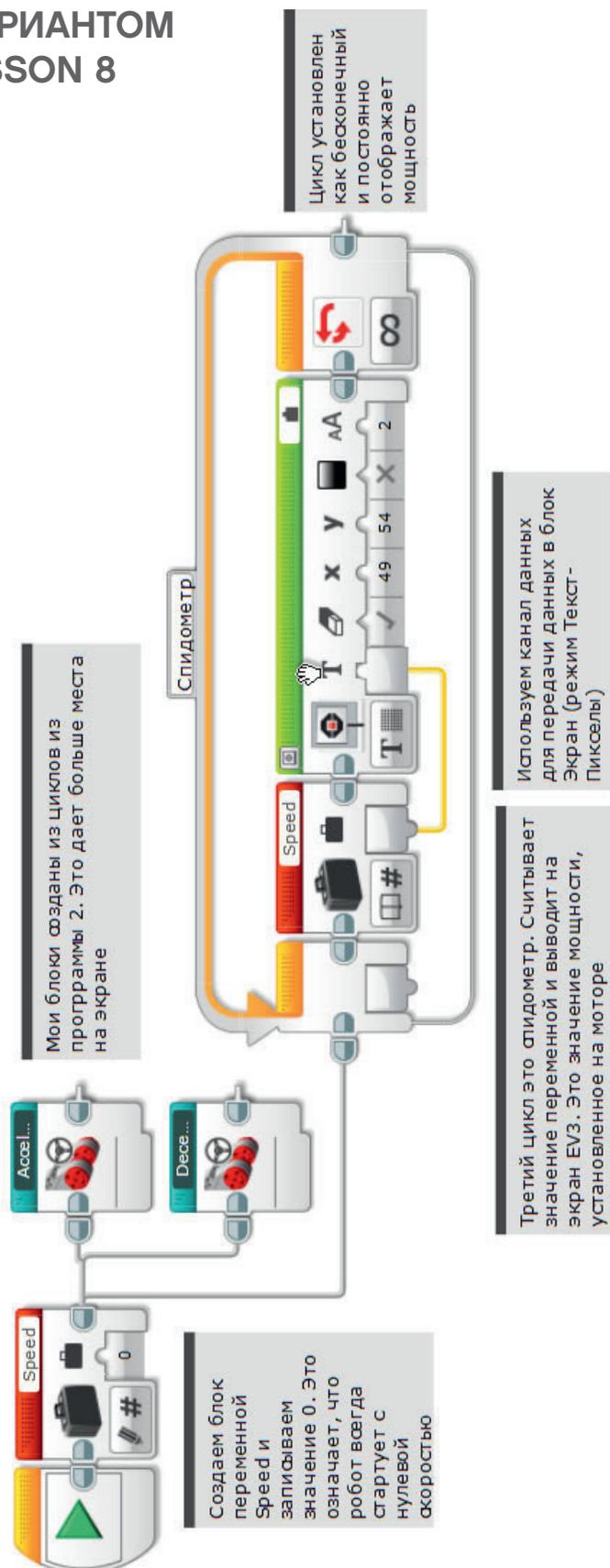
ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 8
ВКЛАДКА: MAIN 2



Занятие 8

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 8
ВКЛАДКА: MAIN 3



Занятие 8

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session8_1.c

```
Activity8_1.c*
1  #pragma config(Sensor, S1, touchSensor, sensorEV3_Touch)
2  #pragma config(Sensor, S2, touchSensor2, sensorEV3_Touch)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorB, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
6  #pragma config(Motor, motorC, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  /**!Code automatically generated by 'ROBOTC' configuration wizard      !**//
8
9  /*
10 Create a program to control the positive speed of the robot by a press of a
11 Touch Sensor.
12 */
13
14 task main()
15 {
16 //Create an integer (whole number) variable to store our speed value.
17 int speed = 0;
18
19 //Repeat our control loop forever.
20 while(true)
21 {
22 //When I press the touch sensor button.
23 if(getTouchValue(touchSensor) == 1)
24 {
25
26 //Add 10 to our 'speed' variable
27 if(speed < 100) speed = speed + 10;
28
29 //Create a loop to wait for the touch sensor button to be released.
30 while(getTouchValue(touchSensor) == 1)
31 {
32 //Wait for button to be released.
33 sleep(10);
34 }
35
36 //Set motorB and motorC speed to the value of the 'speed' variable
37 setMotorSpeed(motorB, speed);
38 setMotorSpeed(motorC, speed);
39 }
40 }
41 }
42
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 8

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session8_2.c

```
Activity8_2.c*
1  #pragma config(Sensor, S1, touchSensor, sensorEV3_Touch)
2  #pragma config(Sensor, S2, touchSensor2, sensorEV3_Touch)
3  #pragma config(Sensor, S3, colorSensor, sensorEV3_Color)
4  #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5  #pragma config(Motor, motorB, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
6  #pragma config(Motor, motorC, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7  /**!!Code automatically generated by 'ROBOTC' configuration wizard      !**//
8
9  /*
10 Create a program to control the positive and negative speed of the robot
11 by the press of two Touch Sensors.
12 */
13
14 task main()
15 {
16     //Create an integer (whole number) variable to store our speed value.
17     int speed = 0;
18
19     //Repeat our control loop forever.
20     while(true)
21     {
22         //When I press the touch sensor button.
23         if(getTouchValue(touchSensor) == 1)
24         {
25             //Add 10 to our 'speed' variable
26             if(speed < 100) speed = speed + 10;
27
28             //Create a loop to wait for the touch sensor button to be released.
29             while(getTouchValue(touchSensor) == 1)
30             {
31                 //Wait for button to be released
32                 sleep(10);
33             }
34             //Set motorB and motorC speed to the value of the 'speed' variable.
35             setMotorSpeed(motorB, speed);
36             setMotorSpeed(motorC, speed);
37         }
38
39         //When I press the touch sensor #2 button.
40         if(getTouchValue(touchSensor2) == 1)
41         {
42             //Subtract 10 from our 'speed' variable.
43             if(speed > -100) speed = speed - 10;
44
45             //Create a loop to wait for the touch sensor #2 button to be released.
46             while(getTouchValue(touchSensor2) == 1)
47             {
48                 //Wait for button #2 to be released.
49                 sleep(10);
50             }
51             //Set motorB and motorC speed to the value of the 'speed' variable.
52             setMotorSpeed(motorB, speed);
53             setMotorSpeed(motorC, speed);
54         }
55     }
56 }
57
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 8

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session8_3.c

```
Activity8_3.c
1 #pragma config(Sensor, S1, touchSensor, sensorEV3_Touch)
2 #pragma config(Sensor, S2, touchSensor2, sensorEV3_Touch)
3 #pragma config(Sensor, S3, colorSensor, sensorEV3_Color)
4 #pragma config(Sensor, S4, sonarSensor, sensorEV3_Ultrasonic)
5 #pragma config(Motor, motorB, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
6 #pragma config(Motor, motorC, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)
7 /**!Code automatically generated by 'ROBOTC' configuration wizard !!!*/
8
9 /*
10 Create a program to control the positive and negative speed of the robot
11 by a press of two Touch Sensors. Display the speed on the EV3 screen.
12 */
13
14 //Create an integer (whole number) variable to store our speed value.
15 int speed = 0;
16
17 void Accelerate()
18 {
19     //When I press the touch sensor button.
20     if(getTouchValue(touchSensor) == 1)
21     {
22         //Add 10 to our 'speed' variable
23         if(speed < 100) speed = speed + 10;
24
25         //Create a loop to wait for the touch sensor button to be released.
26         while(getTouchValue(touchSensor) == 1)
27         {
28             //Wait for button to be released
29             sleep(10);
30         }
31         //Set motorB and motorC speed to the value of the 'speed' variable.
32         setMotorSpeed(motorB, speed);
33         setMotorSpeed(motorC, speed);
34     }
35 }
36 void Decelerate()
37 {
38     //When I press the touch sensor #2 button.
39     if(getTouchValue(touchSensor2) == 1)
40     {
41         //Subtract 10 from our 'speed' variable.
42         if(speed > -100) speed = speed - 10;
43
44         //Create a loop to wait for the touch sensor #2 button to be released.
45         while(getTouchValue(touchSensor2) == 1)
46         {
47             //Wait for button to be released
48             sleep(10);
49         }
50         //Set motorB and motorC speed to the value of the 'speed' variable.
51         setMotorSpeed(motorB, speed);
52         setMotorSpeed(motorC, speed);
53     }
54 }
55 task main()
56 {
57     //Create a string to store our text to be displayed to the LCD.
58     string displaySpeed;
59
60     //Repeat our control loop forever.
61     while(true)
62     {
63         Accelerate();
64         Decelerate();
65         //Format our string to display "Speed: 50" or similar values.
66         stringFormat(displaySpeed, "Speed: %d ", speed);
67         //Draw the string on the LCD screen.
68         drawTextAt(49, 54, displaySpeed);
69         //Set motorB and motorC speed to the value of the 'speed' variable.
70         setMotorSpeed(motorB, speed);
71         setMotorSpeed(motorC, speed);
72     }
73 }
74
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ЗАНЯТИЕ 9

Блуждающие роботы



Занятие 9

Блуждающие роботы

На этом занятии будут рассматриваться массивы и способы их применения для управления колесными роботами, созданными учениками.

В ходе занятия обучающиеся узнают, что такое массивы, как они работают и почему они важны для компьютерного программирования. Также они научатся использовать массивы в своих программах.

Обучающиеся должны составить программы, подобные программам роботизированных игрушек, таких как BeeBot и Roamer.

На этапе введения в тему вы будете демонстрировать массив в действии при помощи сортировщика по цвету. Ученики могут посмотреть видео, включенное в основной набор программного обеспечения модели, которую они собираются собрать. В случае сборки сортировщика, сборку и программирование необходимо выполнить до начала занятия.

Инструкцию по сборке и описание модели «Сортировщик цвета» можно найти в Базовом ПО LME EV3. Для сборки данной модели требуется Базовый набор LME EV3 (артикул 45544). Нажмите на «Инструкции к модели» (Model Instructions), и в верхней части списка вы увидите нужную модель.

Нажмите на эту вкладку и затем нажмите на «Открыть» (Open) для запуска инструкций по сборке модели (также можно открыть вкладку, просто щелкнув по ней два раза). Здесь вы найдете инструкции по сборке модели и пример программы. По завершении сборки модели загрузите программу и запустите ее, опираясь на видео на первой странице руководства.

Это занятие отличается от предыдущих, поскольку в нем предусмотрено всего два основных задания, поскольку выполнение каждого из них займет у класса много времени.



Занятие 9

План урока



РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- правильное использование способов организации данных, таких как списки, таблицы и массивы;
- понимание простейшей булевой логики (например, операций "И", "ИЛИ" и "НЕТ") и некоторые варианты ее применения в контурах и программировании;
- использование кнопок интеллектуального блока для управления перемещениями колесного робота;
- использование блока переменных для хранения информации;
- использование блока операций над массивами.

ВВЕДЕНИЕ

- Объясните ученикам, что они должны запрограммировать колесного робота на действия (например, движение) по командам, которые подаются через кнопки на интеллектуальном блоке EV3 (вперед, назад, вправо, влево)
- Расскажите, что массив является временной формой хранения числовых данных в определенной последовательности. Эти числовые данные можно впоследствии использовать в программе.
- Запустите сортировщик по цвету несколько раз и попросите учеников записать его действия в форме псевдокода.
- Свяжите программирование учебного колесного робота с программированием устройства спутниковой навигации (навигатор). Как можно реализовать такую функцию для колесного робота? Обучающиеся обсуждают в группах, как можно запрограммировать робота на движение по маршруту. Спросите, как работает навигатор.



Занятие 9

План урока

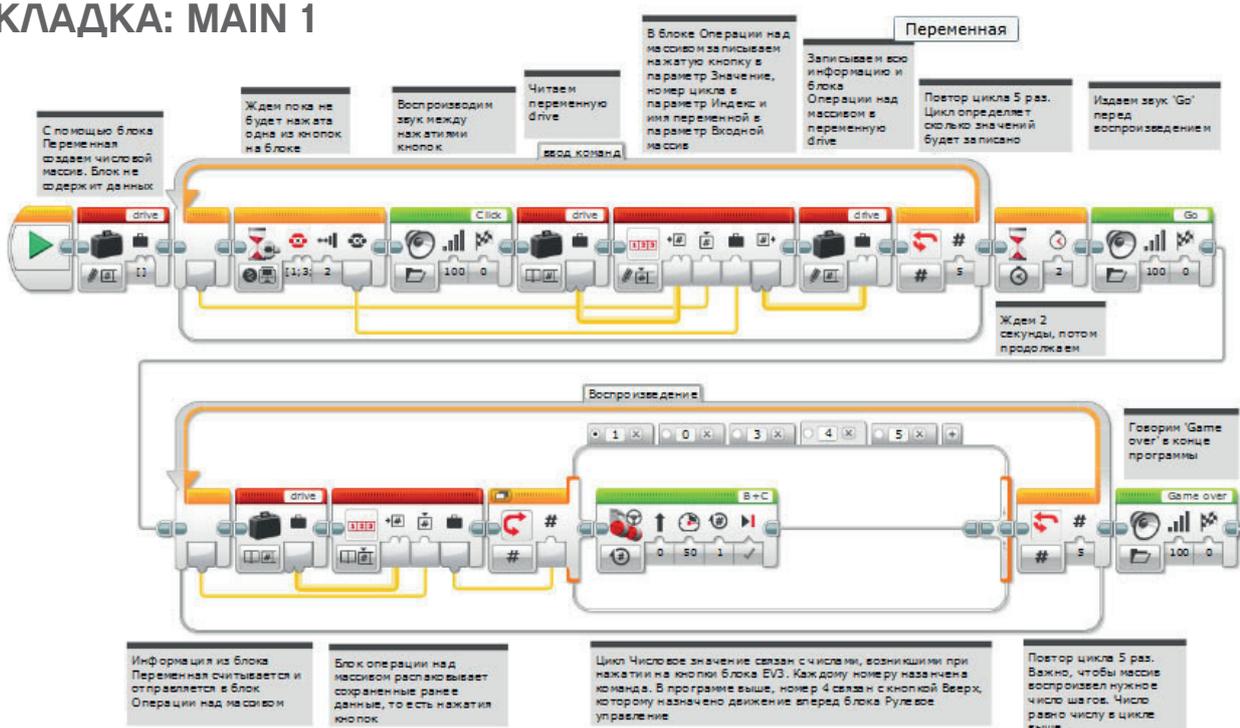


- Помогите ученикам догадаться, что им потребуется блок операций над массивами. Поясните, что этот блок также используется для хранения данных в блоке переменных, который тоже понадобится. Впоследствии данные можно извлекать и использовать в программе.
- Покажите снова модель сортировщика по цвету. Спросите учеников, как, по их мнению, она работает. Они должны ответить, что программа запоминает порядок цветных кубиков и помещает их в указанные в ней емкости.
- Реализуйте подобный принцип в программе колесного робота.

ОСНОВНОЕ ЗАДАНИЕ 1

- Исследуйте программу сортировщика по цвету. Дайте задание группам учеников объяснить, что происходит в программе. Понимают ли они это? Попросите класс составить такую программу.
- Рассмотрите программу с учениками, убедившись, что они понимают основы создания переменной, применения блока операций над массивами и необходимости считывания и последующей записи переменной. Ученикам нужно также понять, что процесс происходит в следующем порядке:
считывание переменной, запись, еще одно считывание и использование информации.
- Дайте ученикам задание написать программу (подобную программе роботизированных игрушек VeeBot, Roamer, и т.д.), которая заставит робота двигаться по определенному маршруту. Для начала, ограничьте программу пятью шагами.
- Объясните, что программу необходимо разделить на две части. Сначала необходимо собрать информацию, затем использовать собранную информацию, подобно тому, как это делает сортировщик по цвету в примере

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 9 ВКЛАДКА: MAIN 1



Занятие 9

План урока

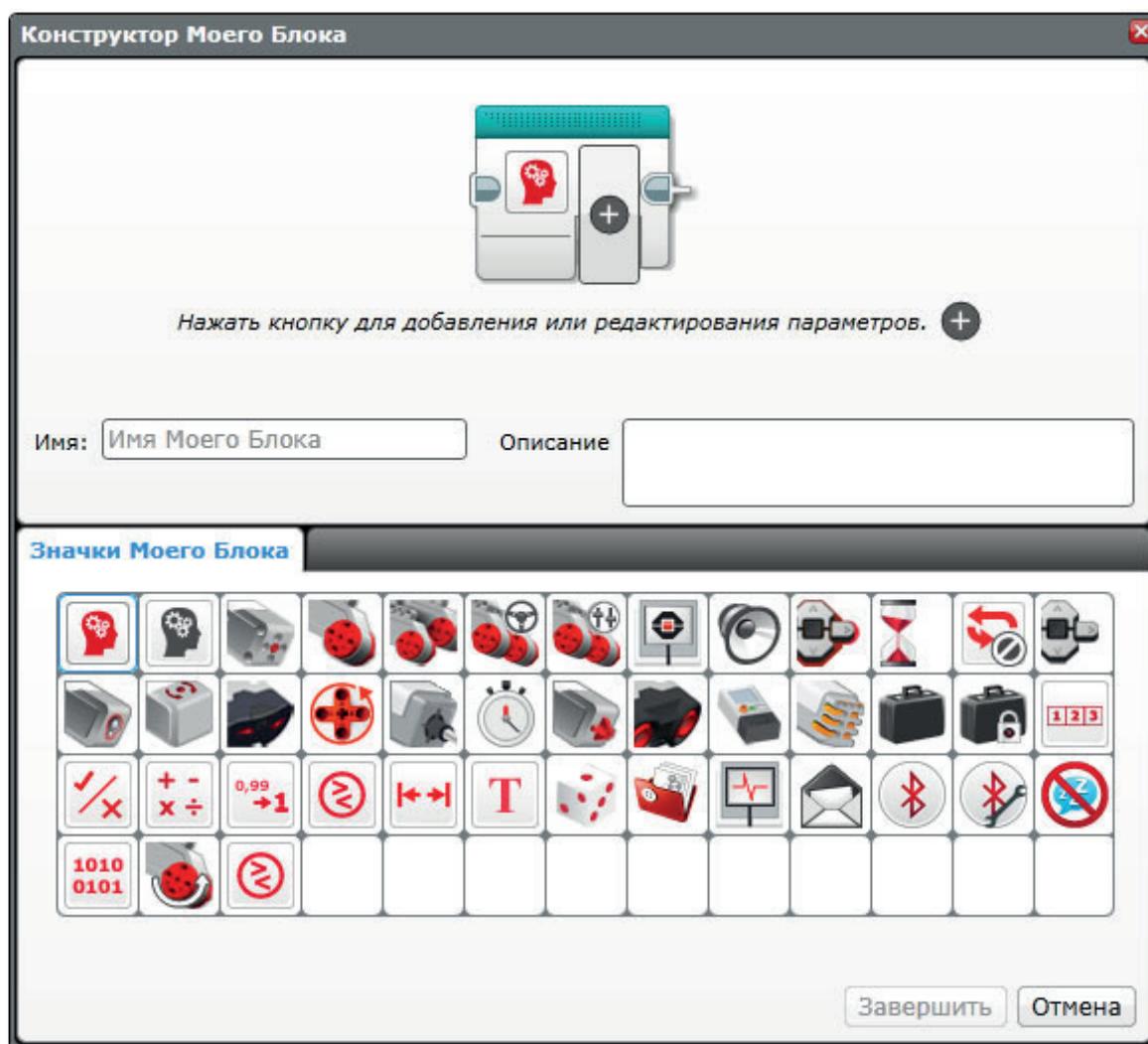


ОСНОВНОЕ ЗАДАНИЕ 2

- При выполнении задания 1 мы ограничиваемся числом, включенным в цикл.
- Покажите классу, как создать раздел "Мой блок" и затем добавить к нему параметр.
- Обратите внимание, что к панели программирования добавлена новая вкладка, на которой показано содержание раздела "Мой блок". Для завершения раздела "Мой блок" пользователь должен соединить параметр "а" с двумя входами

в составе программы. В нашей программе этот параметр соединяется с двумя блоками цикла. Благодаря этому данные можно вводить непосредственно в раздел "Мой блок", а не в содержащуюся в нем программу.

- Обратите внимание учеников на то, что такая структура поможет им легко вводить желаемое количество шагов. Проверьте, понимают ли обучающиеся, что добавление параметра в "Конструктор моего блока" позволяет им вводить данные в созданный "Мой блок".

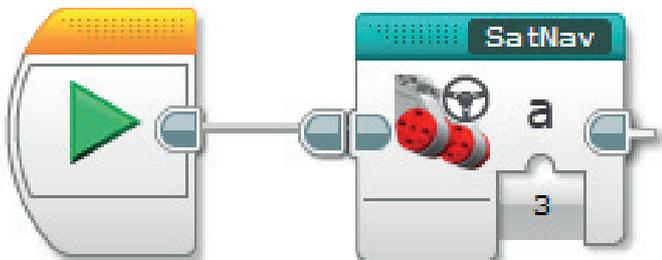


Занятие 9

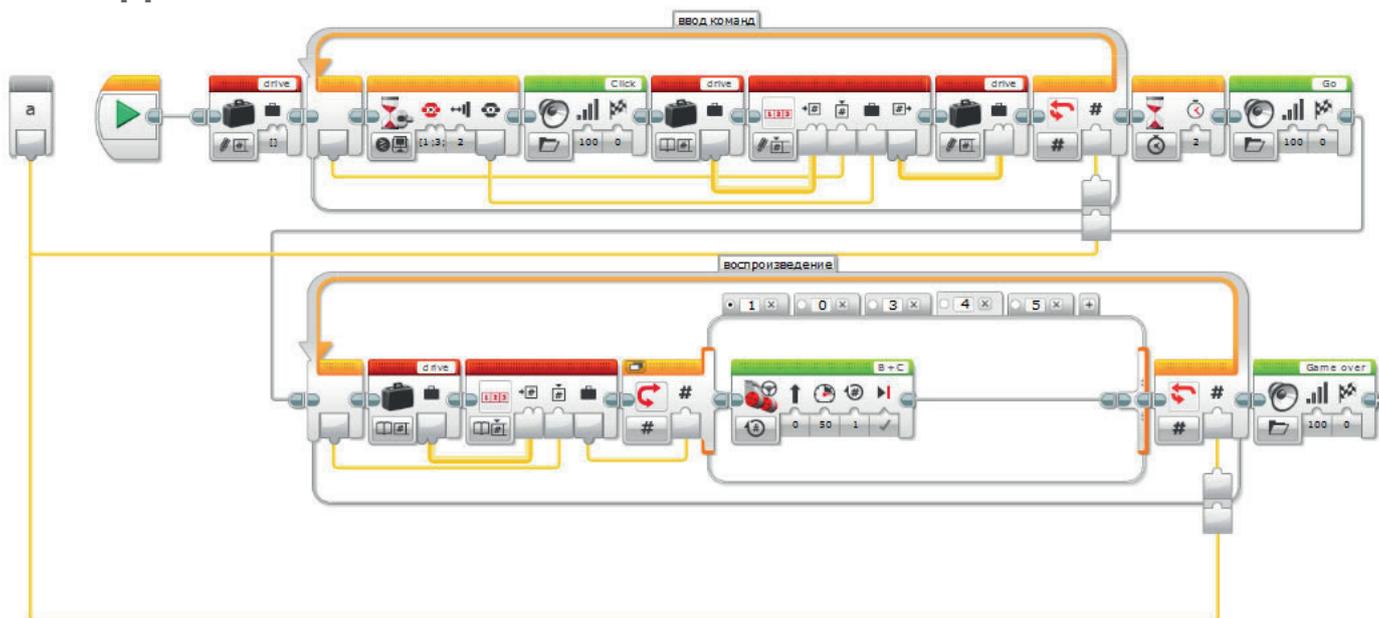
План урока



ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 9
ВКЛАДКА: MAIN 2

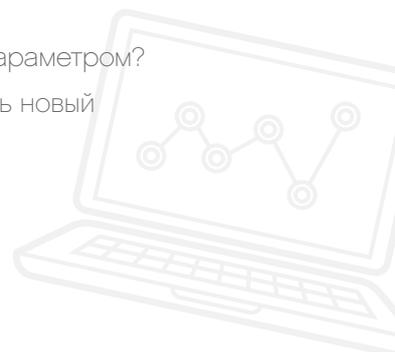


ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 9
ВКЛАДКА: SATNAV



ПОДВЕДЕНИЕ ИТОГОВ

- Соберите группу, чтобы ученики могли поделиться своими успехами в программировании. Спросите, как можно было бы улучшить их программы.
- Легче ли ученикам водить нужное количество шагов при создании "Моего блока" с параметром?
- Попросите класс рассмотреть альтернативные решения. Могут ли ученики придумать новый способ программирования колесного робота на движение по классу?



Занятие 9

Рабочие карточки учеников



ЗАДАЧИ НА СЕГОДНЯ

Сегодня вы научитесь работать с массивами. Это важный блок, позволяющий хранить большие объемы информации и использовать при необходимости. Вы создадите автоматизированный автомобиль, запрограммированный на движение, разделенное на ряд шагов. Направление можно выбирать с 4

ЗАДАЧА 1

Вы наблюдали за работой сортировщика по цвету. Теперь вы должны создать массив, чтобы запрограммировать колесного робота на движение по классу с управлением кнопками на интеллектуальном блоке EV3. Четыре кнопки интеллектуального блока можно использовать для управления (влево, вправо, назад и вперед).

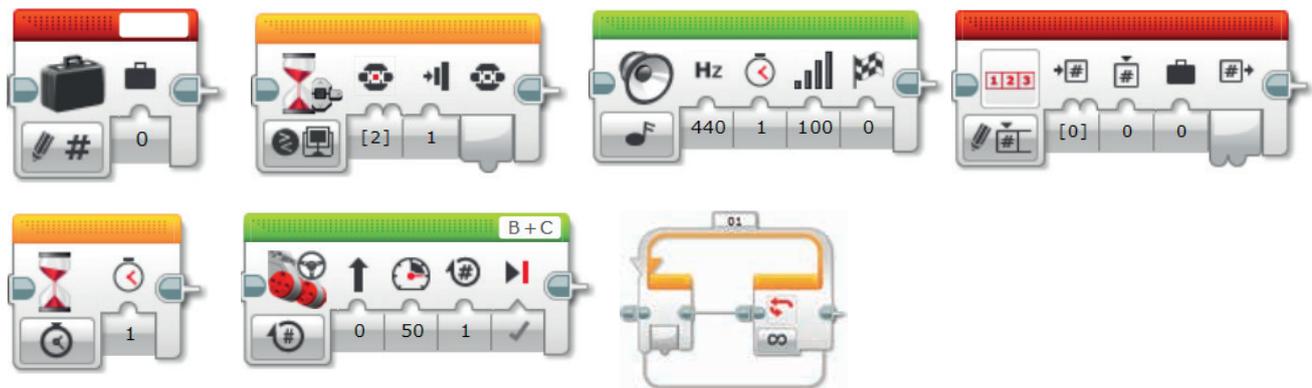
Для начала необходимо ограничить программу пятью командами, введя значение 5 в блок цикла. Полезная информация 1: ваша программа будет состоять из двух четких этапов:

1. сбор данных;
2. использование данных.

Полезная информация 2: для этого задания потребуются два блока цикла, необходимые для выполнения двух вышеуказанных этапов. Полезная информация 3: использование блока переменных часто требует трехшагового процесса:

считывание информации с блока переменных, добавление в него информации и затем запись в блок переменных для сохранения новых данных.

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 9

Рабочие карточки учеников



ЗАДАЧА 2

Создайте "Мой блок", чтобы легко изменять количество шагов в программе.

Чтобы изменить количество шагов движения с 5 на другое число потребуется редактировать оба блока цикла в программе. Это действие можно упростить, создав "Мой блок" с параметром.

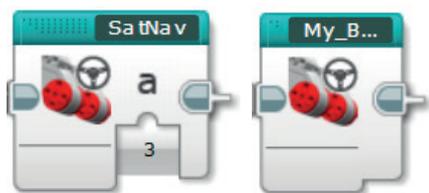
"Мой блок" позволяет легко и четко менять количество циклов. Ваша задача - создать "Мой блок" из программы, составленной при решении задачи 1.

Полезная информация 1: при создании "Моего блока" выделите блоки, которые должны быть включены, но НЕ блок пуска.

Полезная информация 2: при необходимости ввода параметров на более позднем этапе убедитесь, что параметр добавлен к "Моему блоку", как показано ниже. При создании блока используйте клавишу "+".

Полезная информация 3: Параметр необходимо подключить к входу блока в программе. В нашем случае два контура

Используемые блоки



Начните с плана программы. Запишите ее ниже в виде псевдокода:

Занятие 9

Рабочие карточки учеников



По завершении занятия по программированию важно записать свои мысли и наблюдения.

Рассмотрите следующие пункты и в таблице ниже опишите, как прошло занятие.

- Что можно было бы сделать для усовершенствования программы?
- Можно ли упростить структуру программы? Не слишком ли много программных блоков вы использовали? Существует ли более эффективный способ составления программы?
- Какие варианты применения в реальной жизни возможны для вашей программы?

Мысли и наблюдения

Занятие 9

Примечания для учителя

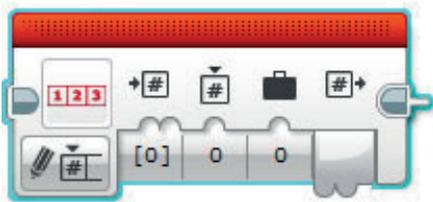


РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

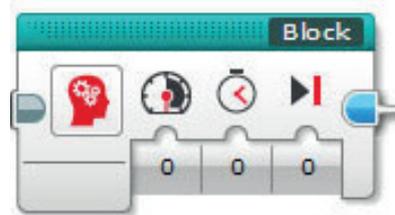
Указанные ниже руководства из учебника по робототехнике помогут учителям и ученикам справиться с задачами.

НОВЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Более сложные действия > Массивы



Инструменты > Мои блоки

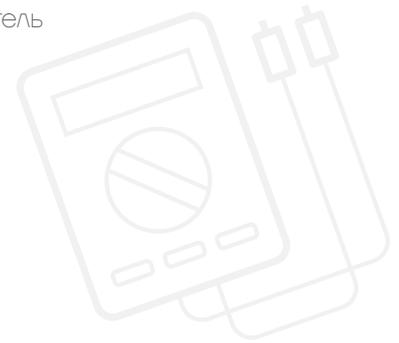
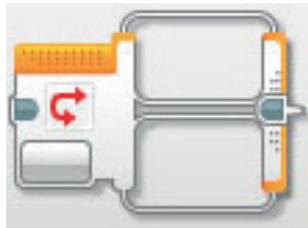


РАНЕЕ РАССМОТРЕННЫЕ РУКОВОДСТВА УЧЕБНИКА ПО РОБОТОТЕХНИКЕ

Более сложные действия > Цикл



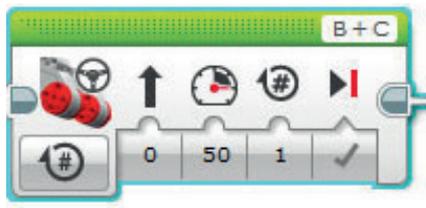
Более сложные действия > Переключатель



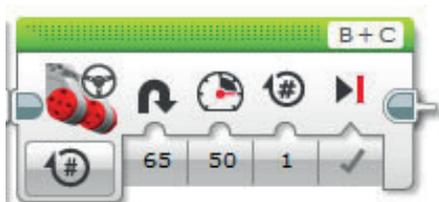
Более сложные действия > Переменные



Основы > Перемещение по прямой



Основы > Движение по дуге



Занятие 9

Приложение

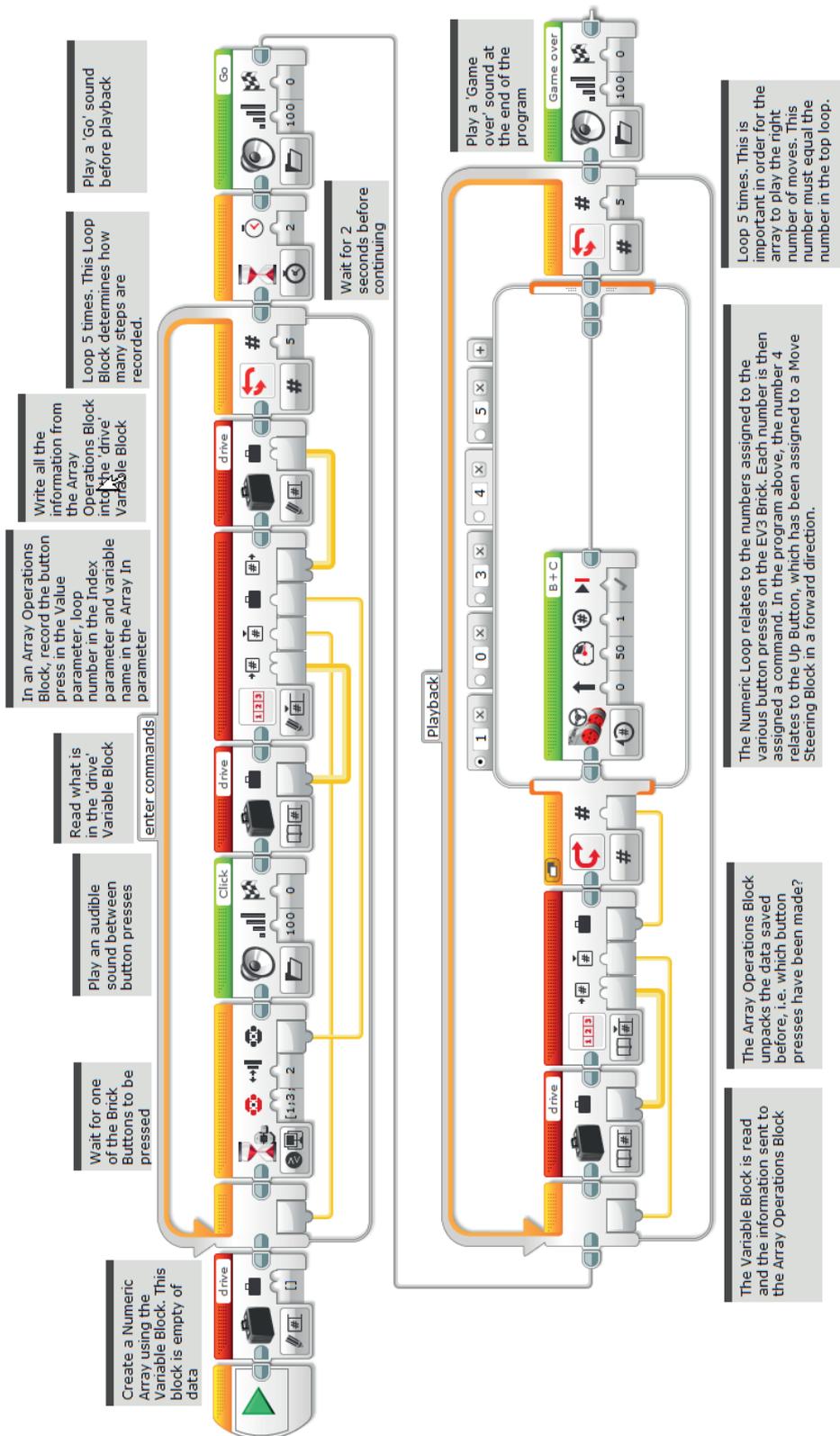
ПРИЛОЖЕНИЕ ДЛЯ ЗАНЯТИЯ 9:
КРУПНЫЕ ИЗОБРАЖЕНИЯ И ПРОГРАММЫ



Занятие 9

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ РЕШЕНИЯ: CS LESSON 9
ВКЛАДКА: MAIN 1



Занятие 9

Приложение

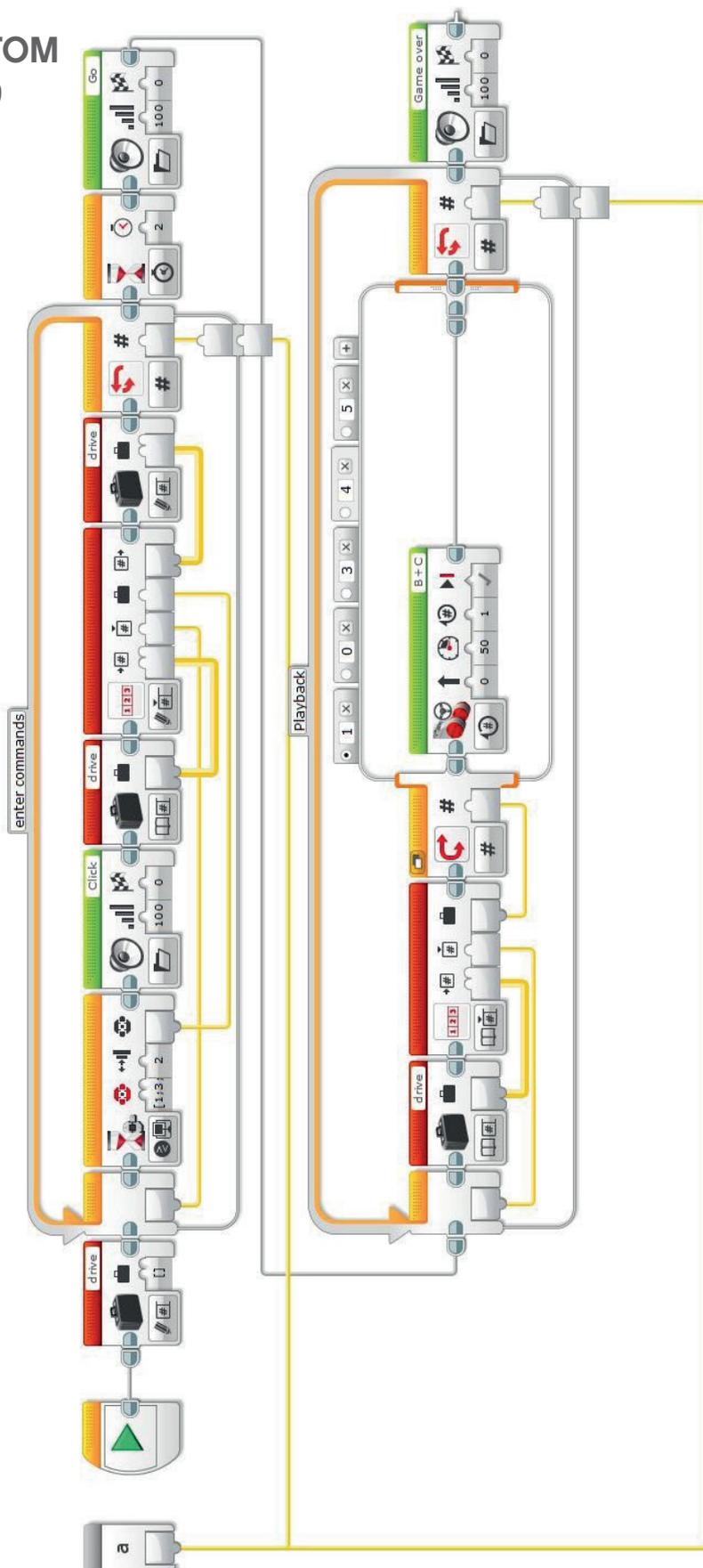
ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 9
ВКЛАДКА: MAIN 2



Занятие 9

Приложение

ИМЯ ФАЙЛА С ВАРИАНТОМ
РЕШЕНИЯ: CS LESSON 9
ВКЛАДКА: SATNAV



Занятие 9

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session9_1.c

```
Activity9_1.c
1 #pragma config(StandardModel, "EV3_REMBOBOT")
2 /**Code automatically generated by 'ROBOTC' configuration wizard. !!!*/
3
4 /*
5 Use the buttons on the EV3 brick to program the robot to move around.
6 3 commands can be entered into the EV3 brick.
7 Left Button = 1
8 Right Button = 3
9 Up Button = 4
10 Down Button = 5
11 */
12
13 int drive[5];
14
15 task main()
16 {
17     for(int i = 0; i < 5; i++) //i = i + 1
18     {
19         while(getButtonPress(buttonAny) == 0)
20         {
21             //Wait for Any Button to be pressed.
22
23             if(getButtonPress(buttonLeft) == 1) drive[i] = 1;
24             else if(getButtonPress(buttonRight) == 1) drive[i] = 3;
25             else if(getButtonPress(buttonUp) == 1) drive[i] = 4;
26             else if(getButtonPress(buttonDown) == 1) drive[i] = 5;
27
28             playSoundFile("Click");
29             while(bSoundActive)
30             {
31                 sleep(10);
32             }
33
34             while(getButtonPress(buttonAny) == 1)
35             {
36                 //Wait for All Buttons to be released.
37                 sleep(10);
38             }
39
40
41             sleep(2000);
42             playSoundFile("Go");
43             while(bSoundActive)
44             {
45
46                 sleep(10);
47             }
48
49             for(int i = 0; i < 5; i++)
50             {
51                 if(drive[i] == 1)
52                 {
53                     //Turn Left Code.
54                     moveMotorTarget(motorC, 360, 50);
55                     waitUntilMotorStop(motorC);
56                 }
57                 else if(drive[i] == 3)
58                 {
59                     //Turn Right Code.
60                     moveMotorTarget(motorB, 360, 50);
61                     waitUntilMotorStop(motorB);
62                 }
63                 else if(drive[i] == 4)
64                 {
65                     //Forward Code.
66                     moveMotorTarget(motorB, 360, 50);
67                     moveMotorTarget(motorC, 360, 50);
68                     waitUntilMotorStop(motorB);
69                     waitUntilMotorStop(motorC);
70                 }
71                 else if(drive[i] == 5)
72                 {
73                     //Backward Code.
74                     moveMotorTarget(motorB, -360, -50);
75                     moveMotorTarget(motorC, -360, -50);
76                     waitUntilMotorStop(motorB);
77                     waitUntilMotorStop(motorC);
78                 }
79             }
80
81             playSoundFile("Game over");
82             while(bSoundActive)
83             {
84                 sleep(10);
85             }
86
87         }
88     }
89 }
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

Занятие 9

Приложение

ИМЯ ФАЙЛА с вариантом решения на языке ROBOTC: Session9_2.c

```
Activity9_2.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration
3  //Wizard !!*/
4
5  //Left Button =1; Right Button =3; Up Button =4; Down Button =5
6  //Maximum of 100 steps
7  int drive[100];
8
9  void recordSteps (int numberOfSteps)
10 {
11   for(int i = 0; i < numberOfSteps; i++) //i = i + 1
12   {
13     while(getButtonPress(buttonAny) == 0)
14     {
15       //Wait for Any Button to be pressed
16       sleep(10);
17     }
18
19     if(getButtonPress(buttonLeft) == 1)   drive[i] = 1;
20     else if(getButtonPress(buttonRight) == 1) drive[i] = 3;
21     else if(getButtonPress(buttonUp) == 1)   drive[i] = 4;
22     else if(getButtonPress(buttonDown) == 1) drive[i] = 5;
23
24     playSoundFile("Click");
25     while(bSoundActive) sleep(10);
26
27     while(getButtonPress(buttonAny) == 1)
28     {
29       //Wait for All Buttons to be released.
30       sleep(10);
31     }
32   }
33 }
34
35 void playSteps(int numberOfSteps)
36 {
37   for(int i = 0; i < numberOfSteps; i++)
38   {
39     if(drive[i] == 0)
40     {
41       //No Step Found, End Code.
42       return;
43     }
44     else if(drive[i] == 1)
45
46
47     //Turn Left Code
48     moveMotorTarget(motorC, 360, 50);
49     waitUntilMotorStop(motorC);
50
51   }
52   else if(drive[i] == 3)
53   {
54     //Turn Right Code
55     moveMotorTarget(motorB, 360, 50);
56     waitUntilMotorStop(motorB);
57   }
58   else if(drive[i] == 4)
59   {
60     //Forward Code
61     moveMotorTarget(motorB, 360, 50);
62     moveMotorTarget(motorC, 360, 50);
63     waitUntilMotorStop(motorB);
64     waitUntilMotorStop(motorC);
65   }
66   else if(drive[i] == 5)
67   {
68     //Backward Code
69     moveMotorTarget(motorB, -360, -50);
70     moveMotorTarget(motorC, -360, -50);
71     waitUntilMotorStop(motorB);
72     waitUntilMotorStop(motorC);
73   }
74 }
75
76
77 task main()
78 {
79   recordSteps(8);
80
81   sleep(2000);
82   playSoundFile("Go");
83   while(bSoundActive) sleep(10);
84
85   playSteps(8);
86
87   playSoundFile("Game over");
88   while(bSoundActive) sleep(10);
89
90 }
```

Данный код может быть изменен в связи с регулярными обновлениями ROBOTC.

ИТОГОВЫЕ ПРОЕКТЫ

ЗАНЯТИЕ 10

Проектирование собственного мобильного автономного робота

ЗАНЯТИЕ 11

Конструирование и программирование собственного мобильного автономного колесного робота

ЗАНЯТИЕ 12

Анализ, доработка и представление собственного мобильного автономного колесного робота



ЗАНЯТИЕ 10

Проектирование собственного самоходного колесного робота



Занятие 10

Проектирование собственного самоходного колесного робота



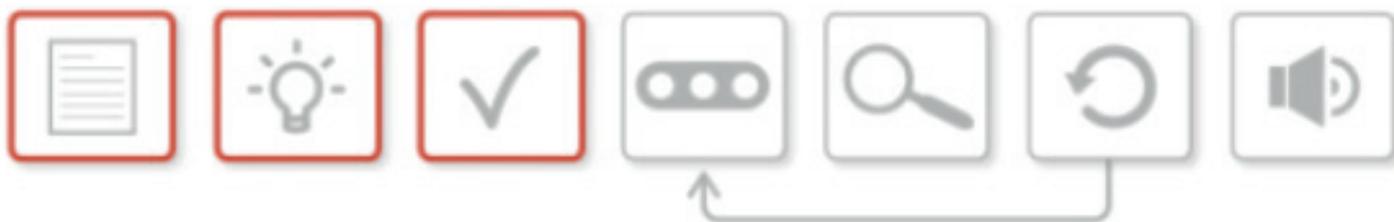
За девять проведенных занятий ученики узнали о различных аспектах автоматизации колесного робота, от сообщения информации пассажирам и пешеходам до способности автомобиля двигаться по заданному маршруту. Теперь они могут создать собственный “автоматизированный автомобиль”, воплотив в нем как можно больше из освоенных на занятиях функций.

Оценка созданных учениками колесных роботов будет производиться так же, как и оценка настоящих автомобилей – по уровню их оснащения. Не секрет, что даже одна и та же модель может иметь совершенно разный объем опций и уровень оснащения, обеспечивая водителя и пассажиров различным уровнем безопасности и комфорта.

Для выполнения такого проекта необходимо соблюдать инженерно-технический процесс, описанный в комплекте заданий “Инженерные проекты LEGO® MINDSTORMS® Education EV3” (артикул 2005544).

Это дополнительный комплект заданий, специально разработанный для инженерно-технических дисциплин (который также можно использовать для организации практических занятий по информатике).

На текущем занятии будут обобщены все аспекты программирования, рассмотренные на данный момент с классом. Дайте задание составить в группах “карту идей” с указанием того, что необходимо для самоходного автоматизированного колесного робота. Напомните ученикам о том, что желательно соблюдать инженерно-технический процесс, описанный в комплекте заданий “Инженерные проекты” для конструкторов EV3. На сегодняшнем занятии процесс включает следующие этапы:



- **Техническое задание**
- **Мозговой штурм**
- **Выбор лучшего решения**
- Конструирование и программирование выбранной модели
- Испытание и анализ
- Пересмотр и доработка решения
- Представление результатов

Первые три пункта должны быть выполнены на текущем занятии. Остальные пункты будут выполняться на последующих занятиях.



Занятие 10

План урока

РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- проектирование, применение и анализ вычислительных абстракций, моделирующих состояние и поведение физических систем в реальных задачах.

ОСНОВНЫЕ ПОНЯТИЯ

Проектирование, карта идей, мозговой штурм, решение, задание на разработку.

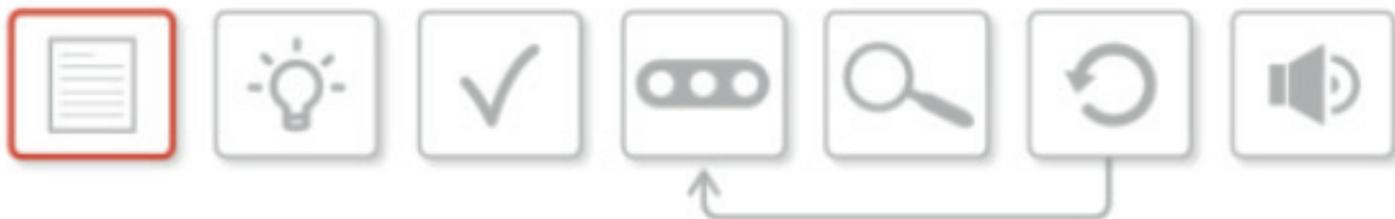
ВВЕДЕНИЕ – ЗАДАНИЕ НА РАЗРАБОТКУ

- Объясните инженерно-технический процесс по плану выше. Обратите внимание класса на его семь этапов и сообщите, что ученики будут выполнять этот процесс в течение трех последующих занятий.
- Занятие 10 (текущее занятие): техническое задание, мозговой штурм (карта идей) и выбор лучшего решения.
- Занятие 11: конструирование и программирование выбранной модели, испытание и анализ
- Занятие 12: пересмотр и доработка решения.
- Техническое задание по данному проекту:

проектирование и конструирование самоходного автоматизированного колесного робота, который может двигаться из пункта А в пункт В, обходя препятствия.

Если сравнивать такую концепцию с классификацией автомобилей, это будет стандартная, базовая модель. Однако к конструкции колесного робота можно добавить любое из нижеперечисленного, чтобы поднять его "класс". Классификацию можно изменить в соответствии с уровнем обучающихся.

- **Стандартная комплектация:** колесный робот обходит препятствия.
- **Расширенная комплектация:** стандартная плюс реагирование автомобиля на сигналы светофоров и предупреждение пешеходов.
- **Улучшенная комплектация:** расширенная комплектация плюс пуск без ключа.
- **Комплектация "люкс":** улучшенная комплектация плюс круиз-контроль.



Занятие 10

План урока



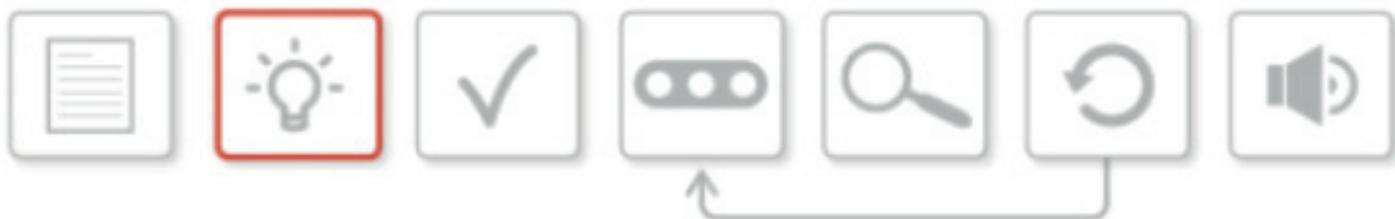
ИТОГОВЫЙ ПРОЕКТ: ОСНОВНОЕ ЗАДАНИЕ

МОЗГОВОЙ ШТУРМ ДЛЯ ПОИСКА РЕШЕНИЯ (СОСТАВЛЕНИЕ КАРТЫ ИДЕЙ)

- Рассмотрите техническое задание. Спросите обучающихся, какого класса автомобиль они хотят создать.
- Уточните, нужно ли им повторить материал по каким-либо аспектам программирования, необходимым для создания их модели.
- Пусть ученики зададут себе следующие вопросы: Какие датчики нужны? Имеется ли достаточно датчиков? Какие компромиссные решения необходимы?
- После этого ученики приступают к поиску, анализу идей по выполнению технического задания и обмену ими.
- Предложите им ознакомиться с современным самоходным транспортом:
- Московский Монорельс;
- автоматические миниавтобусы аэропорта Хитроу;
- Дубайский метрополитен.

Ученики могут поискать видео и статьи в интернете о каждой из этих систем и сравнить характеристики, а затем обсудить, какие их функции можно реализовать в собственных моделях.

- Предложите ученикам рассмотреть другие кирпичики в комплектах LEGO® EV3, что подтолкнет их к новым идеям. Большинство будет придерживаться модели из учебника по робототехнике, но некоторые могут пойти на эксперимент.



РАСПРЕДЕЛЕНИЕ ОБЯЗАННОСТЕЙ

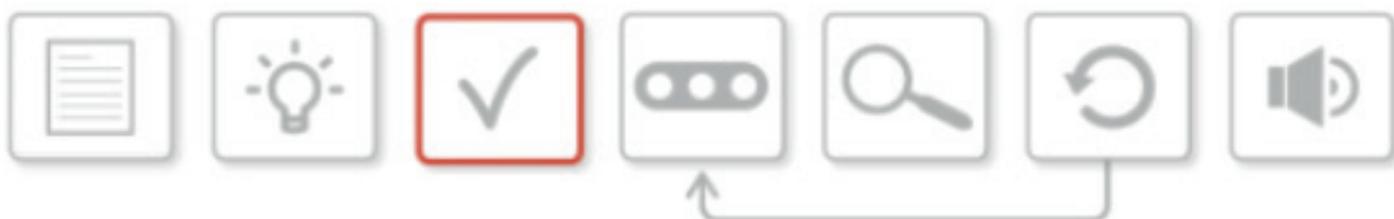
- Как необходимо распределить обязанности в группе, чтобы создать реальную модель по техническому заданию? Попросите класс определить обязанности, которые должны выполняться в группе. В каждой группе будут участники с разными навыками. У каждого из нас есть свои способности. Предложите детям обсудить свои сильные и слабые стороны. У кого лучше получается собирать конструкторы? Кто умеет хорошо программировать? По результатам обсуждения можно распределить обязанности.
- **Возможные обязанности в группе**
 - Конструктор
 - Сборщик модели
 - Испытатель
 - Продавец
 - Программист

Занятие 10

План урока

ВЫБОР ЛУЧШЕГО РЕШЕНИЯ

- Обучающимся необходимо взвесить все плюсы и минусы каждой идеи. Далее необходимо принять окончательный вариант конструкторского решения, который они будут реализовывать на практике.
- Обучающиеся представляют свое окончательное решение классу и поясняют свой выбор.



Ученики должны записывать каждый этап процесса в рабочих карточках или в редакторе контента в программном обеспечении EV3

ПОДВЕДЕНИЕ ИТОГОВ

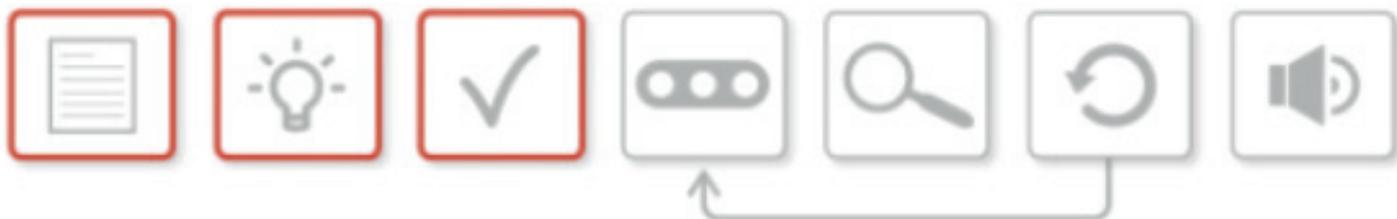
- Каковы следующие этапы? После выбора технического решения группы приступают к конструированию и программированию колесного робота..
- Сообщите ученикам, что на следующих занятиях они будут собирать, программировать и испытывать колесного робота.



Занятие 10

Рабочие карточки учеников

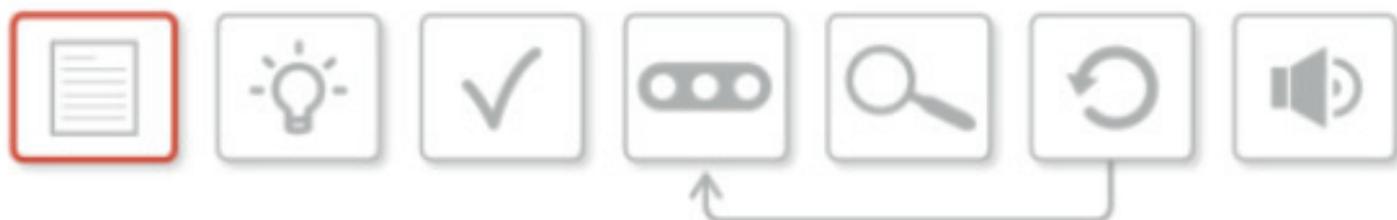
ЗАДАЧИ НА СЕГОДНЯ



Рассмотрите процесс проектирования.

Сегодня вы будете работать с тремя этапами: ознакомление с **техническим заданием**, **мозговой штурм** в составе группы и **выбор решения**.

Учитель даст вам техническое задание. Повторим его:



ПРОЕКТИРОВАНИЕ И КОНСТРУИРОВАНИЕ САМОХОДНОГО АВТОМАТИЗИРОВАННОГО КОЛЕСНОГО РОБОТА, КОТОРЫЙ МОЖЕТ ДВИГАТЬСЯ ИЗ ПУНКТА А В ПУНКТ В, ОБХОДЯ ПРЕПЯТСТВИЯ.

Первый этап процесса проектирования состоит в совместном выработывании самой лучшей идеи. После того, как вы сравнили все плюсы и минусы разных идей, нужно выбрать одну из них и обосновать свой выбор. Здесь важна коллективная работа, которая

зачастую является одним из самых трудных моментов проекта. Учтите, что выбрать могут не вашу идею. Вы должны прийти к общему мнению, чья идея лучше, и затем представить обоснования выбора.

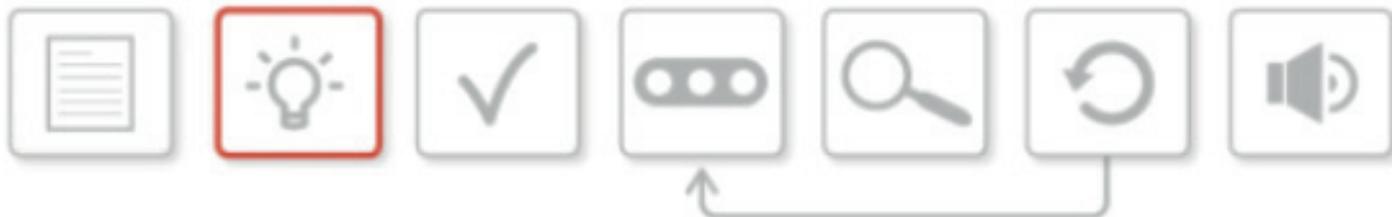


Занятие 10

Рабочие карточки учеников



МОЗГОВОЙ ШТУРМ ПО ТЕХНИЧЕСКОМУ ЗАДАНИЮ



Обсудите техническое задание. Какую из версий автоматизированного колесного робота хотела бы реализовать ваша группа?

- **Стандартная комплектация:** колесный робот обходит препятствия.
- **Расширенная комплектация:** стандартная плюс реагирование автомобиля на сигналы светофоров и предупреждение пешеходов.
- **Улучшенная комплектация:** расширенная комплектация плюс пуск без ключа.
- **Комплектация “люкс”:** улучшенная комплектация плюс круиз-контроль.

Теперь предлагайте конструктивные решения в режиме мозгового штурма. Какие функции вы хотели бы включить в конструкцию и далее в программу? Будут ли нужны изменения в физической конструкции робота? Запишите все ваши идеи и сделайте наброски ниже.

Занятие 10

План урока



РАСПРЕДЕЛЕНИЕ ОБЯЗАННОСТЕЙ

Для работы над каждым проектом нужна рабочая группа и вы - часть этой группы.
Какие задачи требуется выполнить?

Перечислите различные обязанности, которые необходимо распределить в группе, и назначьте их исполнителей.

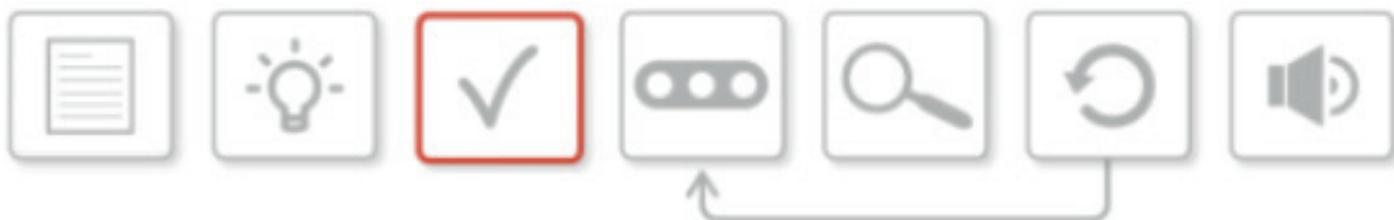
A large rectangular area defined by a dotted border, intended for students to write their answers to the questions above.

Занятие 10

Рабочие карточки учеников



ВЫБОР И ПРЕДСТАВЛЕНИЕ ЛУЧШЕГО РЕШЕНИЯ



Пришло время выбрать лучшую идею из предложенных путем мозгового штурма.

Подготовьте небольшую презентацию, чтобы объяснить учителю и товарищам, какое техническое решение вы выбрали, и приведите обоснования вашего выбора.

Также вы можете пояснить распределение обязанностей в группе.

Расскажите, как вы пришли к такому варианту, описав преимущества и слабости участников группы.

Продолжительность презентации должна быть не более пяти минут. Кто и как делает презентацию - решает группа.

Делайте записи в поле ниже.

ЗАНЯТИЕ 11

Конструирование и программирование собственного самоходного автоматизированного колесного робота



Занятие 11

Конструирование и программирование собственного самоходного автоматизированного колесного робота



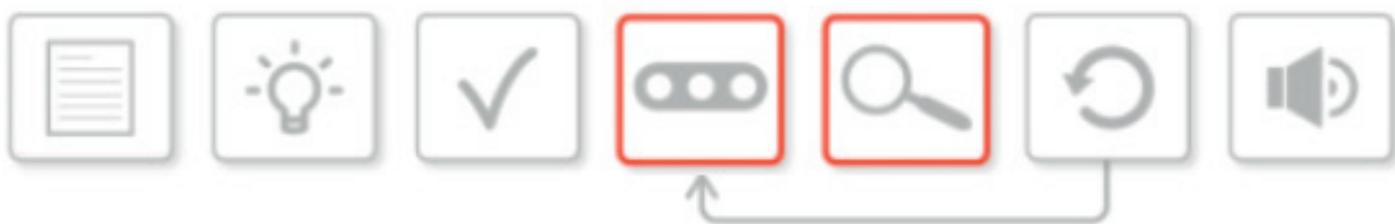
За десять проведенных занятий ученики узнали о различных аспектах автоматизации колесного робота, от сообщения информации пассажирам и пешеходам до способности автомобиля двигаться по заданному маршруту. Теперь они могут создать собственный "автоматизированный автомобиль", воплотив в нем как можно больше из освоенных на занятиях функций.

Оценка созданных учениками колесных роботов будет производиться так же, как и оценка настоящих автомобилей – по уровню их оснащения. Не секрет, что даже одна и та же модель может иметь совершенно разный объем опций и уровень оснащения, обеспечивая водителя и пассажиров различным уровнем безопасности и комфорта.

Для выполнения такого проекта необходимо соблюдать инженерно-технический процесс, описанный в комплекте заданий "Инженерные проекты LEGO® MINDSTORMS® Education EV3" (артикул 2005544).

Это дополнительный комплект заданий, специально разработанный для инженерно-технических дисциплин (который также можно использовать для организации практических занятий по информатике).

На текущем занятии будут обобщены все аспекты программирования, рассмотренные на данный момент с классом. Дайте задание составить в группах "карту идей" с указанием того, что необходимо для самоходного автоматизированного колесного робота. Напомните ученикам о том, что желательно соблюдать инженерно-технический процесс, описанный в комплекте заданий "Инженерные проекты" для конструкторов EV3. На сегодняшнем занятии процесс включает следующие этапы:



- техническое задание;
- мозговой штурм;
- выбор лучшего решения;
- **Конструирование и программирование выбранной модели**
- **Испытание и анализ**
- Пересмотр и доработка решения
- Представление результатов

Выделенные пункты должны быть выполнены на текущем занятии. Остальные пункты будут выполняться на последующих занятиях.



Занятие 11

План урока



РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- проектирование, применение и анализ вычислительных абстракций, являющихся моделями состояния и действия задач и физических систем реального мира;
- объективность при анализе собственной работы;
- сотрудничество в команде.

ОСНОВНЫЕ ПОНЯТИЯ

Проектирование, решение, техническое задание, испытание, анализ.

ВВЕДЕНИЕ – ЗАДАНИЕ НА РАЗРАБОТКУ

- Подведите итоги работы на занятии 10 и убедитесь, что ученики находятся на таком этапе, когда они могут уверенно перейти к требованиям инженерно-технического процесса данного занятия.
- Объясните ученикам, что на этом занятии они будут выполнять следующие два этапа процесса а именно конструирование и программирование колесного робота, а затем испытание и анализ соответствия робота его назначению.
- Напомните ученикам суть технического задания:

проектирование и конструирование самоходного автоматизированного колесного робота, который может двигаться из пункта А в пункт В, обходя препятствия.

Если сравнивать такую концепцию с классификацией автомобилей, это будет стандартная, базовая модель. Однако к конструкции колесного робота можно добавить любое из нижеперечисленного, чтобы поднять его "класс". Классификацию можно изменить в соответствии с уровнем обучающихся.

- **Стандартная комплектация:** колесный робот обходит препятствия.
- **Расширенная комплектация:** стандартная плюс реагирование автомобиля на сигналы светофоров и предупреждение пешеходов.
- **Улучшенная комплектация:** расширенная комплектация плюс пуск без ключа.
- **Комплектация "люкс":** улучшенная комплектация плюс круиз-контроль.



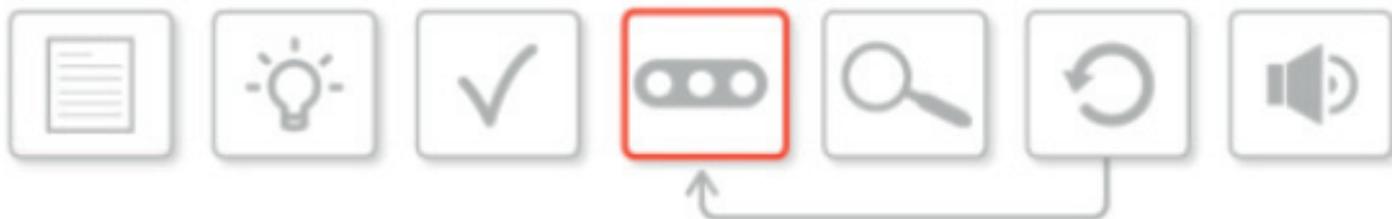
Занятие 11

План урока



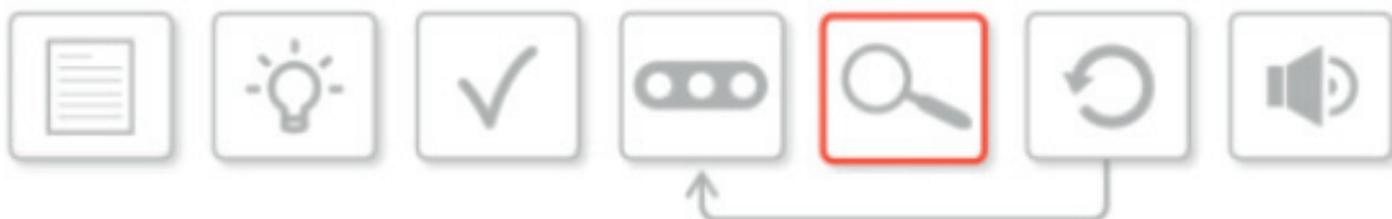
ИТОГОВЫЙ ПРОЕКТ: ОСНОВНОЕ ЗАДАНИЕ

КОНСТРУИРОВАНИЕ И ПРОГРАММИРОВАНИЕ ВЫБРАННОЙ МОДЕЛИ



- Ученики выполняют ретроспективный анализ своей работы на занятии 10, особенно своего конструктивного решения для модели колесного робота.
- Помогите классу обобщить и обсудить конструктивное решение и “класс” выбранной конструкции.
- Ученики могут захотеть уделить это время распределению задач в соответствии с обязанностями, которые они получили на занятии 10.
- Можно посоветовать группам разбиться на подгруппы, одни из которых будут заниматься конструированием, другие - программированием.
- Напомните ученикам, что они могут не достичь выполнения технического задания после первичной сборки и программирования, им нужно еще испытать и проанализировать то, что у них получилось.

ИСПЫТАНИЕ И АНАЛИЗ



- Это непрерывный процесс, о чем нужно напоминать ученикам.
- Важно обратить внимание обучающихся на то, что они должны постоянно анализировать конструкцию и программу.
- Кто-то может захотеть выполнить конструирование и программирование полностью до испытаний, другие могут применить пошаговый подход. Ученики могут использовать любой удобный для них метод. Это хорошее упражнение в самообучении.
- На этом этапе напомните ученикам, что им следует постоянно сверяться с техническим заданием и собственным проектом и идеями, и эти идеи могут измениться по ходу занятий, в особенности на занятии 12.

Занятие 11

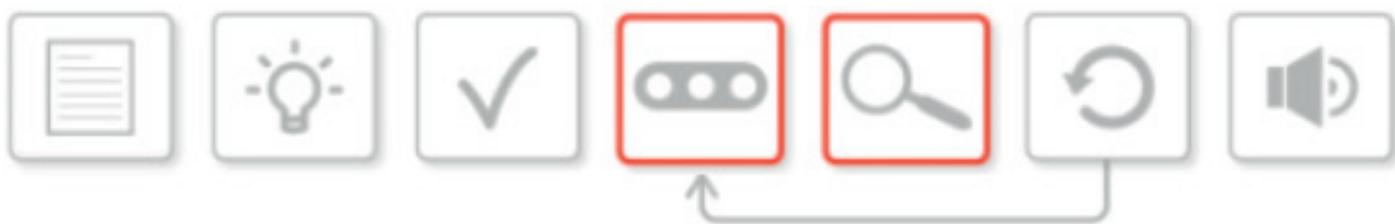
План урока

Ученики должны записывать каждый этап процесса в рабочих карточках или в редакторе контента в программном обеспечении EV3.

ПОДВЕДЕНИЕ ИТОГОВ

- Каковы следующие этапы? Группы завершили конструирование и программирование колесных роботов. На следующих занятиях они будут анализировать выполненную работу, внося при необходимости изменения в конструкцию и программу, после чего они представят результаты остальной группе.

ЗАДАЧИ НА СЕГОДНЯ



На этом занятии вам нужно будет постоянно сверяться с техническим заданием и вашими решениями по конструированию и программированию, над которыми вы работали на прошлом занятии.

Вспомните техническое задание:

проектирование и конструирование самоходного автоматизированного колесного робота, который может двигаться из пункта А в пункт В, обходя препятствия.

Посмотрите на схему инженерно-технического процесса. Сегодня мы будем работать над конструированием и программированием выбранной модели, а также заниматься ее испытаниями и анализом.

Не забывайте записывать ваши решения по конструированию и программированию в редакторе контента в программном обеспечении EV3, либо в рабочей карточке.

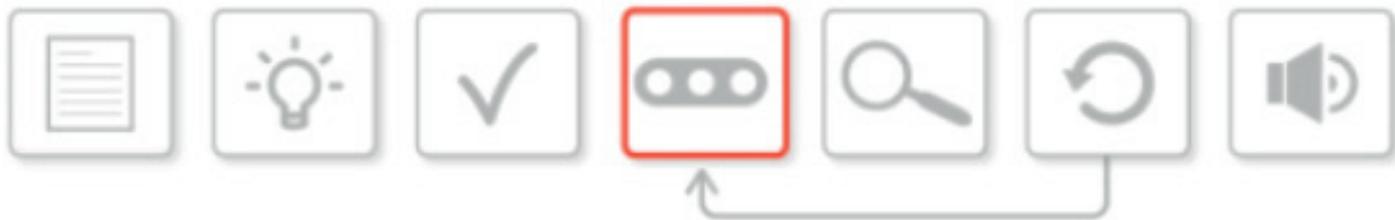


Занятие 11

Рабочие карточки учеников



КОНСТРУИРОВАНИЕ И ПРОГРАММИРОВАНИЕ ВЫБРАННОЙ МОДЕЛИ



Вспомните вашу работу на занятии 10, особенно конструктивное решение для модели колесного робота. Какую версию вы выбрали (стандартную, расширенную, улучшенную, или люкс)?

Обобщите свои идеи с занятия 10 и обсудите их выражение в техническом решении и программе. Можно поменяться обязанностями, распределенными на занятии 10. Также следует эффективно использовать время за счет распределения задач.

При желании можете разделить группу на подгруппы, чтобы одни занимались конструированием, а другие – программированием.

У вас может не получиться выполнить техническое задание, или даже реализовать ваше конструктивное решение с первой попытки. Чтобы достигнуть поставленных на сегодня целей, вам необходимо постоянно анализировать конструкцию и отлаживать программы.

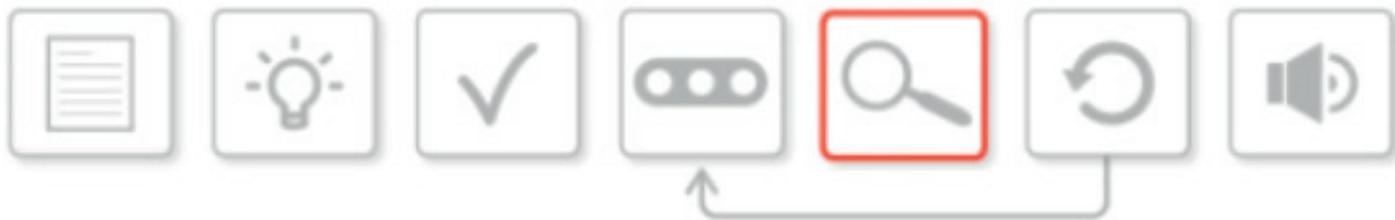
Делайте записи о вашей работе ниже или в редакторе контента:

Занятие 11

Рабочие карточки учеников



ИСПЫТАНИЕ И АНАЛИЗ



Это непрерывный процесс, который не должен прекращаться до конца занятия! Испытания и анализ - очень важные факторы в инженерно-техническом процессе.

Конструирование и программирование можно завершить до проведения испытаний. Либо, напротив, можно выполнять процесс в несколько более коротких шагов.

Обсудите методы с группой и выберите удобный.

На этом этапе необходимо постоянно сверяться с техническим заданием и собственным конструктивным решением и идеями.

При проведении испытаний задайте себе следующие вопросы:

- Соответствует ли наш колесный робот техническому заданию?
- Отражает ли он выбранное конструктивное решение?
- Выполняет ли он желаемые действия, т.е. работает ли составленная программа?

Ваши идеи могут изменяться и развиваться по ходу занятия. Это часть процесса, по ней необходимо вести записи либо в рабочей карточке, либо в редакторе контента EV3.

ЗАНЯТИЕ 12

Анализ, доработка и представление собственного самоходного автоматизированного колесного робота



Занятие 12



Анализ, доработка и представление собственного самоходного автоматизированного колесного робота

За одиннадцать проведенных занятий ученики узнали о различных аспектах автоматизации колесного робота, от сообщения информации пассажирам и пешеходам до способности автомобиля двигаться по заданному маршруту. Теперь они могут создать собственный "автоматизированный автомобиль", воплотив в нем как можно больше из освоенных на занятиях функций.

Оценка созданных учениками колесных роботов будет производиться так же, как и оценка настоящих автомобилей – по уровню их оснащения. Не секрет, что даже одна и та же модель может иметь совершенно разный объем опций и уровень оснащения, обеспечивая водителя и пассажиров различным уровнем безопасности и комфорта.

Для выполнения такого проекта необходимо соблюдать инженерно-технический процесс, описанный в комплекте заданий "Инженерные проекты LEGO® MINDSTORMS® Education EV3" (артикул 2005544).

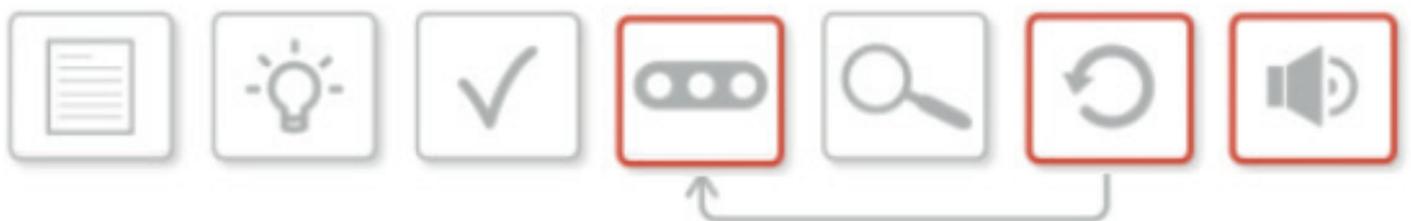
Это дополнительный комплект заданий, специально разработанный для инженерно-технических дисциплин (который также можно использовать для организации практических занятий по информатике).

На текущем занятии будут обобщены все аспекты программирования, рассмотренные на данный момент с классом. Дайте задание составить в группах "карту идей" с указанием того, что необходимо для самоходного автоматизированного колесного робота. Напомните ученикам о том, что желательно соблюдать инженерно-технический процесс, описанный в комплекте заданий "Инженерные проекты" для конструкторов EV3. На сегодняшнем занятии процесс включает следующие этапы:

Это занятие является продолжением занятия 11. На занятии ученикам будет предложено пересмотреть конструкторские и программные решения с прошлого занятия и критически проанализировать сделанное. В ходе коллективного анализа они должны определить, что получилось, и того, что требует улучшения. После этого им необходимо предоставить время для доработки и совершенствования конструкции и программы. В заключение, когда все работы по программированию и испытаниям будут выполнены, ученики должны составить презентацию для одноклассников,

в которой будет рассматриваться инженерно-технический процесс и то, как они добились выполнения технического задания.

Напомните ученикам о том, что желательно соблюдать инженерно-технический процесс, описанный в комплекте заданий по инженерно-техническим проектам. Сегодняшний процесс выглядит следующим образом:



Занятие 12

План урока



- техническое задание;
- мозговой штурм;
- выбор лучшего решения;
- **Конструирование и программирование выбранной модели**
- Испытание и анализ
- **Пересмотр и доработка решения**
- **Представление результатов**

Выделенные пункты должны быть выполнены на текущем занятии.

РЕЗУЛЬТАТЫ

Ученики должны приобрести следующие знания и умения:

- проектирование, применение и анализ вычислительных абстракций, моделирующих состояние и поведение физических систем в реальных задачах;
- объективность при анализе собственной работы;
- способность критически анализировать собственную работу и совместно совершенствовать ее;
- объяснение своего инженерно-технического процесса товарищам;
- сотрудничество в команде.

ОСНОВНЫЕ ПОНЯТИЯ

Проектирование, решение, техническое задание, испытание, анализ, пересмотр и доработка решения.

ВВЕДЕНИЕ

- Подведите итоги работы на занятии 11 и убедитесь, что ученики находятся на таком этапе, когда они могут уверенно перейти к требованиям инженерно-технического процесса данного занятия.
- Объясните ученикам, что в начале занятия они запустят программу, составленную для колесного робота на предыдущем занятии, и что их сегодняшние задачи следующие: анализ сделанного, выводы, на основе которых определяются изменения или усовершенствования в конструкции или программе.
- Обратите их внимание на то, что изменения в конструкции или программу могут привести к тому, что модель робота может измениться с базовой на более продвинутую.
- Попросите записывать все изменения конструкции или программы либо в рабочих карточках, либо в редакторе контента программного обеспечения EV3, поскольку это потребуется для итоговой презентации.



Занятие 12

План урока



- Напомните ученикам суть технического задания:

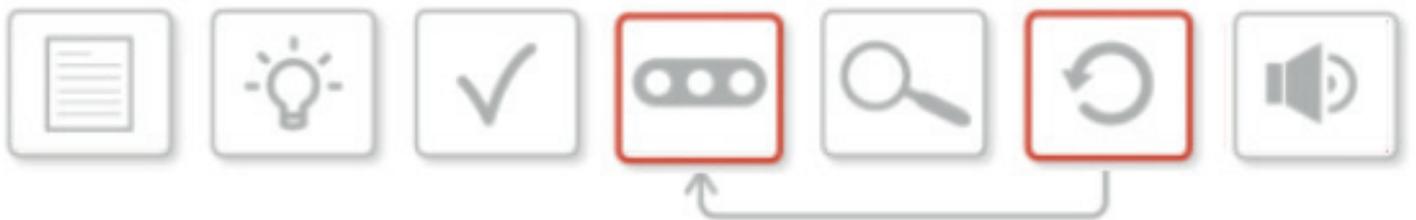
проектирование и конструирование самоходного автоматизированного колесного робота, который может двигаться из пункта А в пункт В, обходя препятствия.

Если сравнивать такую концепцию с классификацией автомобилей, это будет стандартная, базовая модель. Однако к конструкции колесного робота можно добавить любое из нижеперечисленного, чтобы поднять его "класс". Классификацию можно изменить в соответствии с уровнем обучающихся.

- **Стандартная комплектация:** колесный робот обходит препятствия.
- **Расширенная комплектация:** стандартная плюс реагирование автомобиля на сигналы светофоров и предупреждение пешеходов.
- **Улучшенная комплектация:** расширенная комплектация плюс пуск без ключа.
- **Комплектация "люкс":** улучшенная комплектация плюс круиз-контроль

ИТОГОВЫЙ ПРОЕКТ: ОСНОВНОЕ ЗАДАНИЕ 1

ПЕРЕСМОТР И ДОРАБОТКА / КОНСТРУИРОВАНИЕ И ПРОГРАММИРОВАНИЕ РЕШЕНИЯ



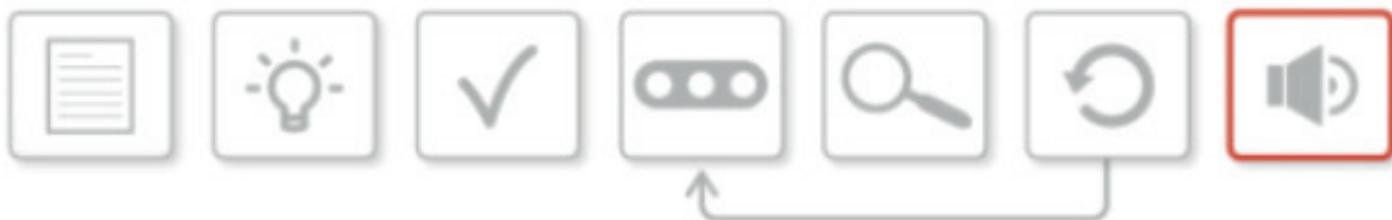
- Расскажите классу, что на этом этапе занятия они будут постоянно пересматривать, испытывать и (возможно) заново собирать колесных роботов.
- Вы можете установить критерии успеха, на которые будут ориентироваться обучающиеся. Критерии можно выбрать совместно с ними.
- Здесь важно напомнить ученикам об исходном техническом задании.
- Обучающиеся работают в группах, запуская программы и обсуждая способы усовершенствования действий колесного робота, а также соответствие исходному техническому заданию.
- Они должны рассмотреть варианты программирования и, где необходимо, внести изменения в конструкцию.
- Подскажите ученикам, что за счет непрерывного пересмотра, доработки, изменения конструкции колесный робот может усовершенствоваться и перейти в другой "класс".
- Выделите время к концу занятия, чтобы каждая группа смогла выполнить один окончательный запуск программы. Соберите весь класс вместе для этого мероприятия. Пока колесные роботы движутся, оценивайте их вместе с учениками по установленным критериям успеха.

Занятие 12

План урока

ИТОГОВЫЙ ПРОЕКТ: ОСНОВНОЕ ЗАДАНИЕ 2

ПРЕДСТАВЛЕНИЕ РЕЗУЛЬТАТА



- Ученики справились с инженерно-техническим процессом, создали окончательную версию колесного робота и использовали максимум знаний при его программировании. Настал завершающий этап - демонстрация проекта товарищам.
- Обучающиеся должны фиксировать свой прогресс с занятия 10 в рабочих карточках или в редакторе контента программного обеспечения EV3. Для этого можно делать текстовые записи, видео, фотографии и скриншоты программных решений.
- Объясните классу, что нужно подготовить краткую презентацию, демонстрирующую ход их работы над проектом и решения в ходе инженерно-технического процесса. В презентациях должно рассказываться об успехах и ошибках, а также о классе, которого достиг робот по мнению своих создателей (стандартная или люкс). Также они должны указать, в какой степени им удалось выполнить техническое задание.
- Презентацию можно сделать в любом желаемом формате. Можно использовать программное обеспечение для презентаций, например PowerPoint / Keynote / Prezi, либо редактор контента в программном обеспечении EV3.

ПОДВЕДЕНИЕ ИТОГОВ

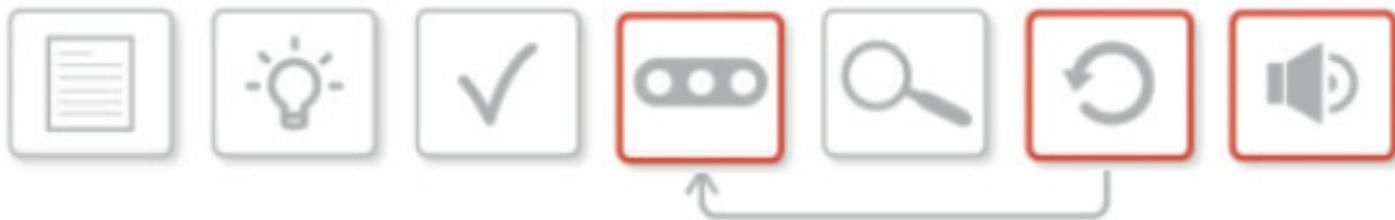
- Во многом презентации учеников служат подведению итогов последних трех занятий. В процессе представления своей работы и анализа выполнения технического задания ученики осуществляют оценку собственной работы.
- После каждой из презентаций группам необходимо давать отзывы, от класса и от учителя



Занятие 12

План урока

ЗАДАЧИ НА СЕГОДНЯ



Это последнее занятие по данной теме. На нем снова необходимо постоянно сверяться с техническим заданием и вашими решениями по конструированию и программированию, над которыми вы работали на прошлом занятии.

Вспомните техническое задание:

проектирование и конструирование самоходного автоматизированного колесного робота, который может двигаться из пункта А в пункт В, обходя препятствия.

Посмотрите на схему инженерно-технического процесса. Занятие посвящено **пересмотру, доработке (ведущей к новому конструированию и программированию) и представлению результатов.**

Не забывайте записывать ваши действия в редакторе контента в программном обеспечении EV3, либо в рабочей карточке.

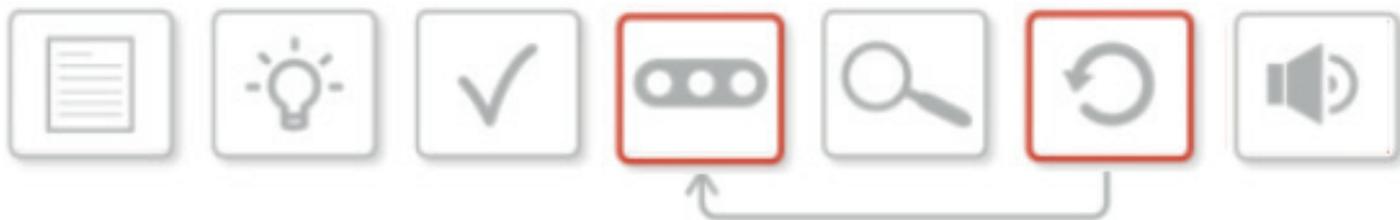


Занятие 12

Рабочие карточки учеников



ПЕРЕСМОТР И ДОРАБОТКА / КОНСТРУИРОВАНИЕ И ПРОГРАММИРОВАНИЕ РЕШЕНИЯ



При работе над задачей вы будете постоянно заниматься пересмотром, испытаниями колесного робота и (вполне вероятно) будете заново его конструировать.

Возможно, вы с товарищами и учителем установили критерии успешного решения задачи. Если да, не забывайте сверяться с ними время от времени.

Какой “класс” колесных роботов вы выбрали и воплотили в модели на двух предыдущих занятиях? В процессе такого пересмотра и доработки вы можете изменить конструкцию и программу колесного робота таким образом, что он перейдет в более высокий “класс”.

Работая в группах, запустите колесного робота в последний раз. Соответствует ли наш колесный робот техническому заданию? Можно ли его усовершенствовать путем изменения программы, конструкции или того и другого? Рассмотрите дополнительные возможности программы и, если необходимо, измените конструкцию.

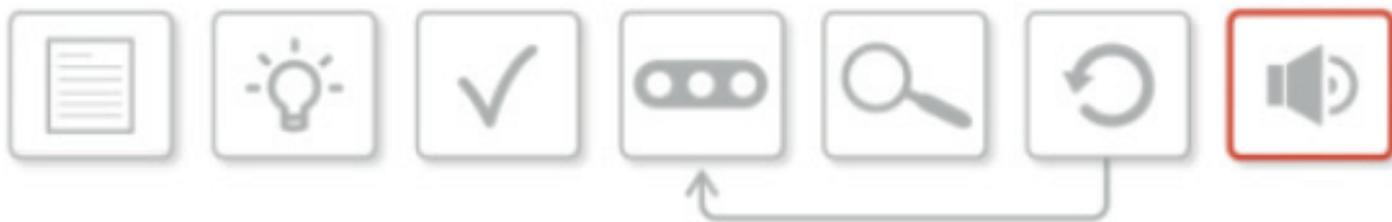
В конце занятия продемонстрируйте колесного робота остальной группе и оцените его относительно задания и ваших критериев успеха.

Делайте записи о вашей работе ниже или в редакторе контента:

Занятие 12

Рабочие карточки учеников

ПРЕДСТАВЛЕНИЕ



После того, как пройден весь инженерно-технической процесс, собрана и запрограммирована окончательная версия колесного робота, в заключение вы должны продемонстрировать вашу работу остальной группе.

Вы должны были фиксировать свою работу с занятия 10 либо в рабочей карточке, либо в редакторе контента в программном обеспечении EV3.

Для этого можно делать текстовые записи, видео, фотографии и скриншоты программных решений.

Подготовьте краткую презентацию, объясняющую ход вашей работы в рамках инженерно-технического процесса, и опишите решения, которые вы разработали в группе и которые помогли вам создать колесного робота и программу.

Презентация должна объяснять степень выполнения технического задания и какому классу соответствует конструкция и программа робота.

В ней должны быть показаны успехи и ошибки, и как эти ошибки были исправлены.

Презентацию можно сделать в любом желаемом формате. Можете использовать программное обеспечение для презентаций,

например, PowerPoint / Keynote / Prezi, редактор контента в программном обеспечении EV3 или иные средства по выбору.

