Муниципальное бюджетное учреждение
Дополнительного образования
«Детско-юношеский центр» г. Колпашево

Учебное пособие по программированию в среде Lego Mindstorms EV3

(Учебное пособие предназначено для обучающихся и педагогов, изучающих программирование в среде Lego Mindstorms EV3)

Составитель: Шадрин Игорь Викторович,
педагог дополнительного образования

Колпашево - 2017

Содержание.

Глава 1. Технические характеристики Lego Mindstorms Education EV3

Набор Lego Mindstorms EV3 предназначен для конструирования и программирования роботов в средней и старшей школе, а также в кружках робототехники.

Существует две версии Lego Mindstorms EV3: версия для образовательных учреждений и домашняя версия. В технической части они идентичны, отличаются только комплектом поставки.

Базовый набор LEGO Mindstorms Education EV3 оптимизирован для использования в классе или кружке робототехники и содержит все необходимое для обучения с помощью технологий LEGO® Mindstorms®. Набор Lego-45544 позволяет ученикам конструировать, программировать и тестировать их решения, используя настоящие технологии робототехники.

Данная модель конструктора включает в себя мощный микрокомпьютер EV3, контролирующий моторы и собирающий данные с датчиков. Микрокомпьютер EV3 набора также поддерживает протоколы Bluetooth и Wi-Fi и функционал регистрации данных.

Программное обеспечение доступно для свободного скачивания на сайте производителя.

Возможности робота LEGO Mindstorms Education EV3 Артикул 45544:

- Различает семь основных цветов, реагирует на степень освещенности помещения;
- "Видит" на расстоянии до 2,5 метра с точностью до 1 мм, "слышит" ультразвуковые волны;
- Еще быстрее "соображает" и реагирует на изменения программ за счет мощного микрокомпьютера (300 MHz против 48 MHz у моделей поколения NXT!) и увеличенного объема оперативной памяти;
- "Общается" с компьютером и другими роботами по Wi-Fi и Bluetooth;
- Интегрирование с мобильными устройствами систем Android и iOS;
- Поддержка карт памяти формата microSD объемом до 32 Гб.

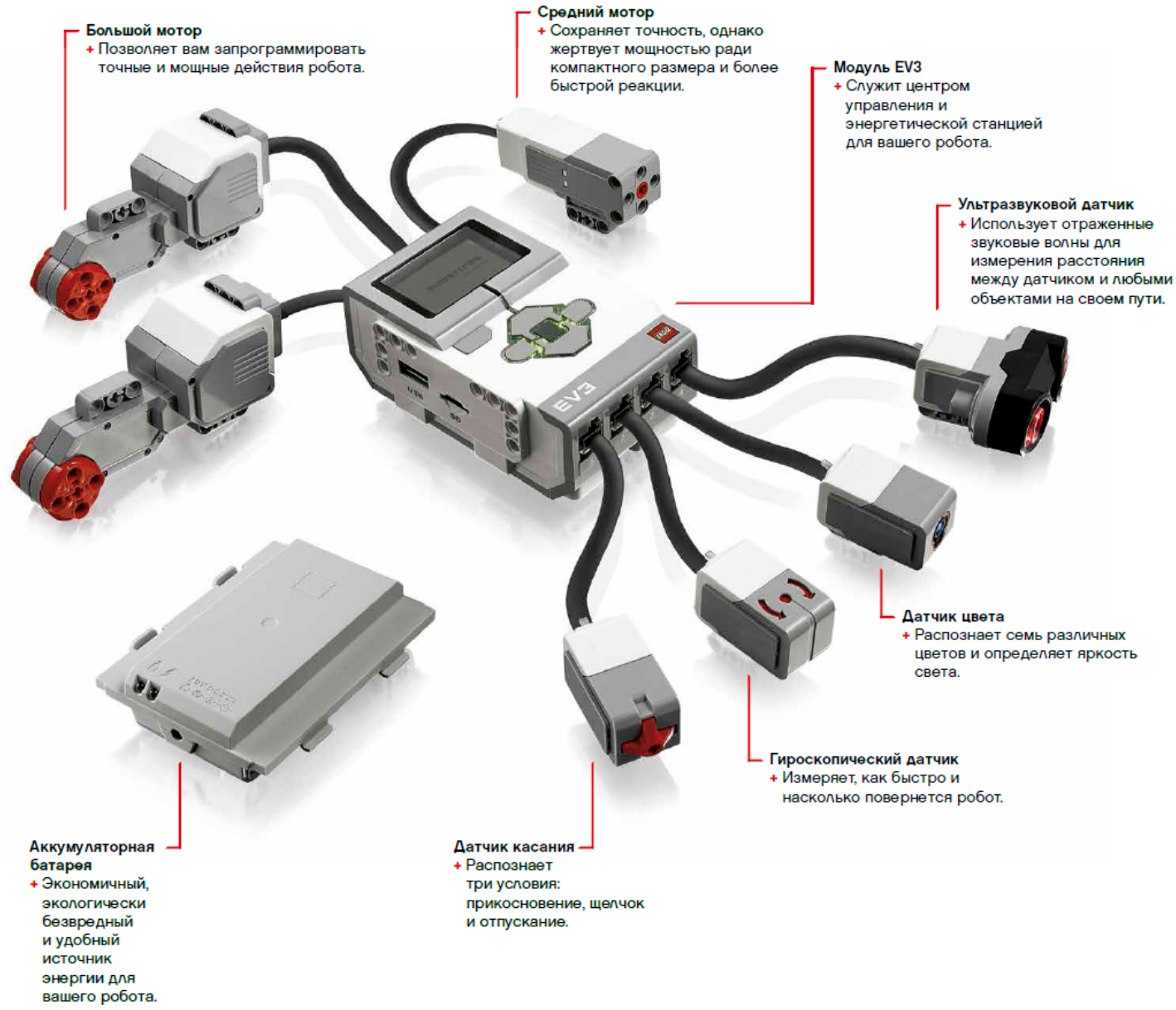
Этот набор с легкостью вдохновит ваших учеников на совместное обсуждение проблемы и поиск креативного решения, которое затем можно будет претворить в жизнь - построить и протестировать, - используя набор моторов, датчиков и строительных элементов LEGO.

Стартовый набор поставляется в удобной коробке, идеальной для хранения элементов и использования в классе.

Можно расширить возможности комплекта, используя ресурсный набор LEGO Education Mindstorms EV3 45560. Зарядное устройство к EV3 продается отдельно.

В набор входят:

- Три электросервомотора;
- Встроенные в моторы датчики вращения и ультразвуковой датчик;
- Датчик цвета, гироскопический датчик и два датчика касаний;
- Перезаряжаемая аккумуляторная батарея;
- Колеса;
- Соединительные кабели;
- Инструкции по сборке;
- Элементы LEGO® Technic для создания множества моделей



Обзор среды программирования

Глава 2. Программирование роботов

2.1 Основы алгоритмизации

Решение задач на компьютере основано на понятии алгоритма. Алгоритм – это точное предписание, определяющее вычислительный процесс, ведущий от начальных данных к исходному результату. Алгоритм означает точное описание некоторого процесса, инструкцию по его выполнению. Алгоритмизация – это процесс построения алгоритма для решения задач на компьютере.

Свойства алгоритмов:

1. Универсальность (массовость) – алгоритм может применяться к различным наборам исходных данных.
2. Дискретность - процесс решения задачи по алгоритму разбит на отдельные простые действия.
3. Однозначность - правила и порядок выполнения действий алгоритма должны пониматься однозначно.
4. Конечность - каждое из действий и весь алгоритм в целом обязательно завершаются.
5. Результативность - по завершении выполнения алгоритма обязательно получается верный результат.

Это только некоторые, самые основные из свойств алгоритма. На самом деле их можно рассмотреть больше.

Алгоритмы могут быть представлены разными способами:

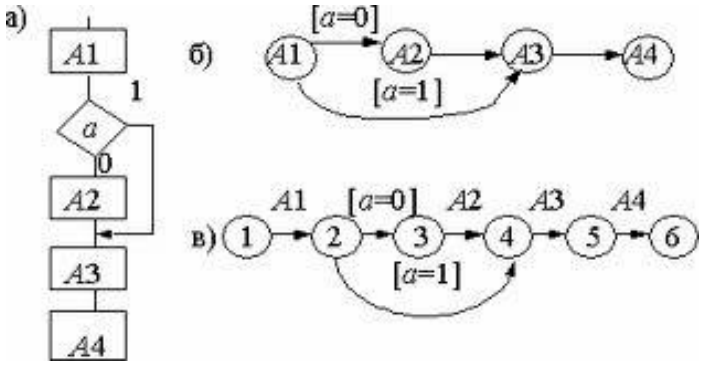
- словесно-формульное описание;
- блок-схема (схема из графических символов);
- алгоритмические языки;
- операторные схемы;
- псевдокод.

Словесно-формульный способ записи отличается тем, что описание осуществляется с помощью слов и формул. Т. е. человек записывает алгоритм словами с использованием профессиональных терминов, знаков и формул вычислений.

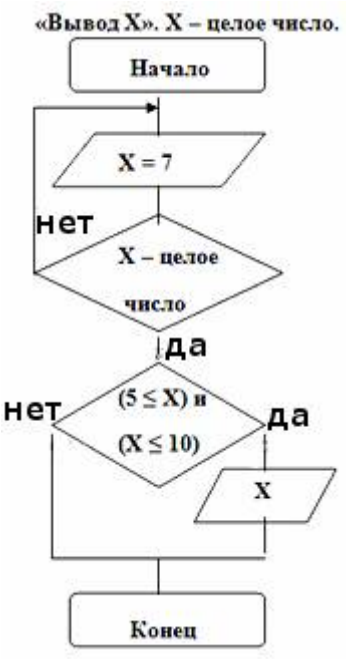
Графический способ описания алгоритма (используется в программном обеспечении Lego Mindstorms EV3) получил самое широкое распространение. Для описания используются блоки, которые соединяются между собой линиями связи.

Алгоритмические языки - это специальное средство, предназначенное для записи алгоритмов. Алгоритмические языки близки к математическим выражениям и к естественным языкам. Каждый алгоритмический язык имеет свой словарь. Алгоритм, записанный на алгоритмическом языке, выполняется по строгим правилам этого конкретного языка.

Использование операторных схем алгоритмов заключается в том, что каждый оператор обозначается буквой (например, А – арифметический оператор, Р – логический оператор и т.д.). Операторы записываются слева направо в последовательности их выполнения, причем, каждый оператор имеет индекс, указывающий порядковый номер оператора. Алгоритм записывается в одну строку в виде последовательности операторов



Пример операторной схемы.



Пример кода на языке блок схем.

Языки программирования – это искусственные языки записи алгоритмов для исполнения их на компьютере. Программирование – это процесс составления программы по заданному алгоритму.

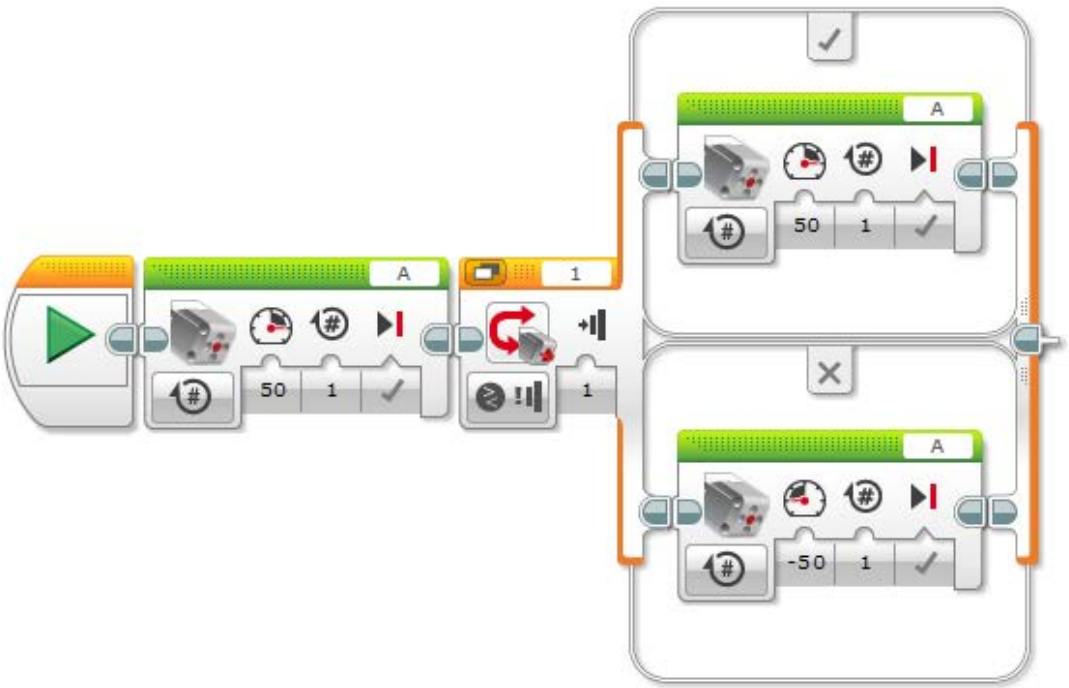
По структуре выполнения алгоритмы делятся на три вида:

- линейные;
- ветвления;
- циклические.

Линейный алгоритм (линейная структура) – это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз. Схема представляет собой последовательность блоков, которые располагаются сверху вниз или слева направо в порядке их выполнения.

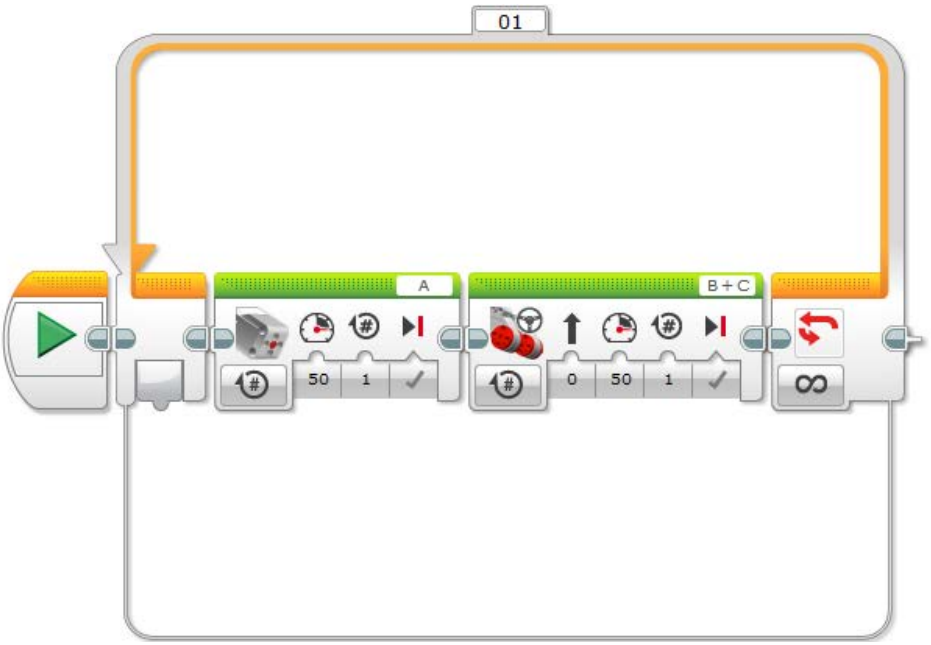
Пример записи линейного алгоритма на языке программирования EV3.

Но на практике часто встречаются задачи, в которых необходимо при различных условиях действовать по-разному. Такие задачи можно описать с помощью алгоритмов разветвляющейся структуры. Выбор направления продвижения по схеме алгоритма осуществляется по итогам проверки заданного условия. Ветвящиеся процессы в EV3 описываются оператором Переключатель.



Пример записи алгоритма с ветвлением.

Для решения некоторых задач нужно повторение отдельных участков вычислений. В таких задачах применяются алгоритмы циклической структуры (циклические алгоритмы). Цикл – последовательность команд, которая повторяется до тех пор, пока не будет выполнено заданное условие.



Пример записи циклического алгоритма в среде EV3.

Существуют циклы с известным и с неизвестным числом повторений. В цикле с неизвестным числом повторений выход из тела цикла, как правило, происходит при выполнении записанного условия.

Для того чтобы программист и робот понимали друг друга, роботу необходимо уметь подавать некоторые сигналы. Они служат сообщением о том, что робот выполнил или не выполнил то или иное действие. В EV3 такими сигналами служат звуковые сообщения, которые может воспроизводить встроенный динамик микроконтроллера. Сообщение также может быть выведено в виде текста или изображения на дисплее микроконтроллера.

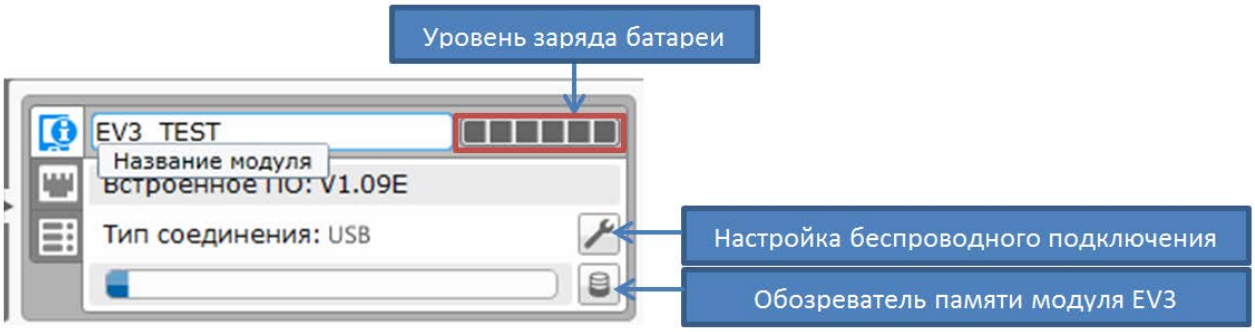
2.2 Способы подключения робота к компьютеру. Загрузка программ

Существует несколько способов подключения модуля EV3 к компьютеру.

- Через кабель USB.
- Через беспроводное соединение Bluetooth.
- Через беспроводное соединение Wi-Fi.

При подключении модуля на странице аппаратных средств надпись EV3 загорается красным.

При подключении на странице аппаратных средств можно посмотреть:



2.3 Палитры программирования и программные блоки.

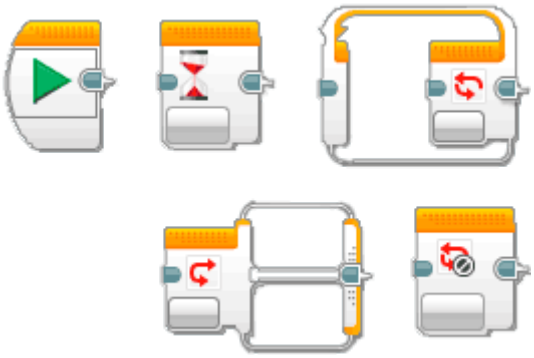
Давайте теперь обратим свой взгляд в нижний раздел среды программирования. Здесь находятся команды для программирования робота. Разработчики применили оригинальный прием и, сгруппировав программные блоки, присвоили каждой группе свой цвет, назвав группы палитрами.

Все программы в среде Lego Mindstorms EV3 состоят из блоков. У каждого блока есть один или несколько регулируемых параметров.

Блоки действий (Зеленый)

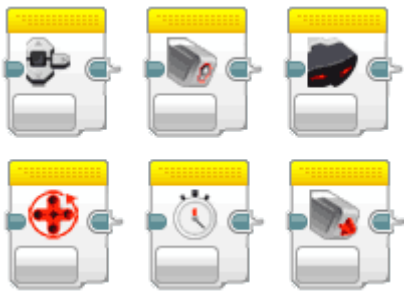
Блоки действий управляют действиями в рамках программы. Они контролируют вращение моторов, а также изображения, звук и подсветку модуля EV3.

Блок выполнения программ (Оранжевый)



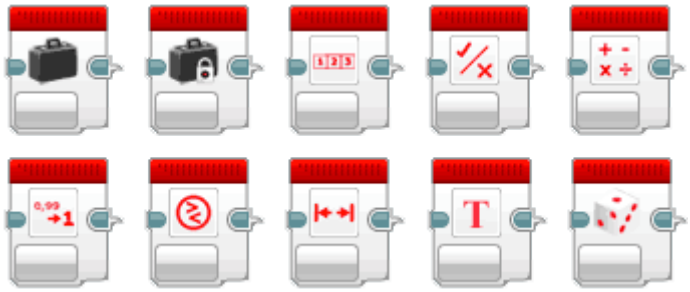
Блок выполнения программы управляют процессом выполнения программ. Все создаваемые тобой программы будут начинаться со стартового блока.

Блоки датчиков (Желтый)



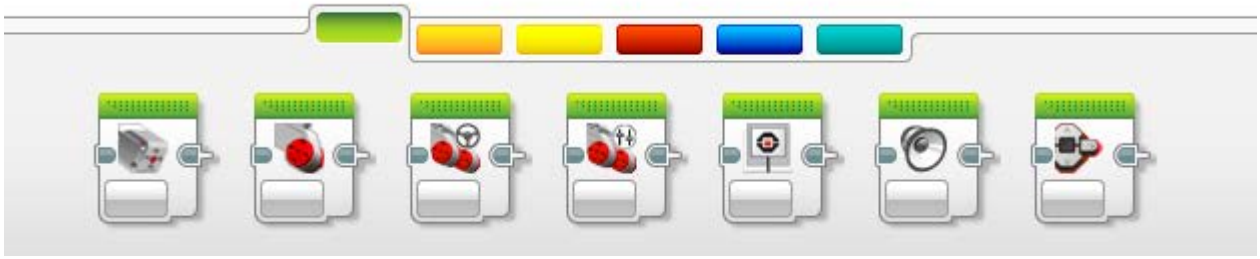
Блоки датчиков позволяют программе считывать входящие данные с датчика цвета, ИК-датчика, датчика касания и многое другое.

Блоки операции над данными (Красный)



Блоки операций над данными позволяют вводить и считывать переменные величины, сравнивать характеристики и многое другое.

2.4 Зеленая палитра «Действие». Моторы, изображения и звуки.



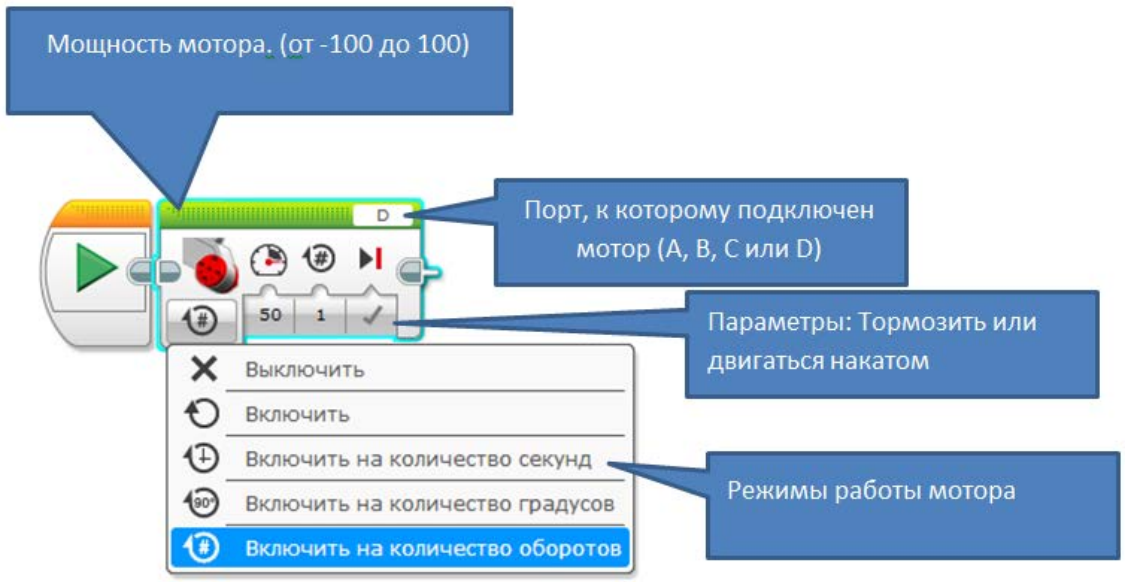
Здесь расположены **программные блоки управления моторами:**

- Средний мотор – предназначен для управления средним мотором.
- Большой мотор – для управления большим мотором.
- Рулевое управление
- Независимое управление моторами

А также блоки:

- Вывод на экран
- Вывод звука
- Индикатор состояния модуля

Рассмотрим настройку моторов на примере большого мотора:



Режим: "Включить" включает мотор с заданным параметром "Мощность" и после этого управление передается следующему программному блоку программы. Мотор будет продолжать вращаться, пока не будет остановлен следующим блоком "Большой мотор" с режимом "Выключить" или следующий блок "Большой мотор" не будет содержать другие параметры выполнения. Режим "Включить на количество секунд" включает большой мотор с установленной мощностью на указанное количество секунд, и только по завершению времени мотор остановится, а управление в программе перейдет к следующему программному блоку. Аналогично поведет мотор себя в режимах "Включить на количество градусов" и "Включить на количество оборотов": только после выполнения установленного вращения мотора он остановится, и управление в программе перейдет к следующему блоку.

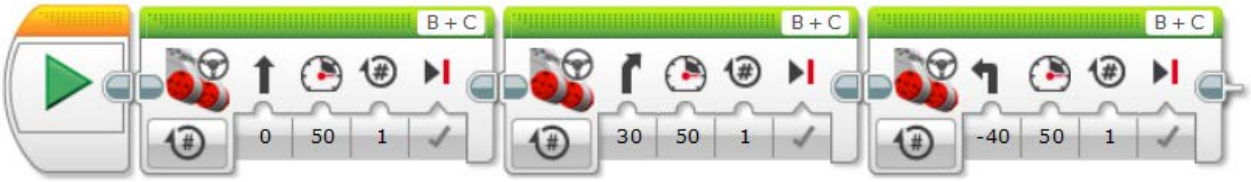
Параметр мощность может принимать значения от -100 до 100. Положительные значения мощности задают вращение мотора по часовой стрелке, отрицательные - против часовой. При значении мощности равном 0 мотор вращаться не будет, чем "выше" значение мощности, тем быстрее вращается мотор.

Параметр мощность задается только целыми значениями, параметры: секунды, градусы, обороты могут принимать значения с десятичной дробью. Но следует помнить, что минимальный шаг вращения мотора равен одному градусу.

Отдельно следует сказать о параметре "Тормозить в конце". Данный параметр, если установлен в значение "Тормозить" заставляет мотор тормозить

после выполнения команды, а если установлен в значение "Двигаться накатом", то мотор будет вращаться по инерции, пока сам не остановится.

Следующие два программных блока "Рулевое управление" и "Независимое управление моторами" реализуют управление парой больших моторов. По умолчанию левый большой мотор подключается к порту "B", а правый - к порту "C". Но вы можете в настройках блока поменять порты подключения в соответствии с требованиями вашей конструкции



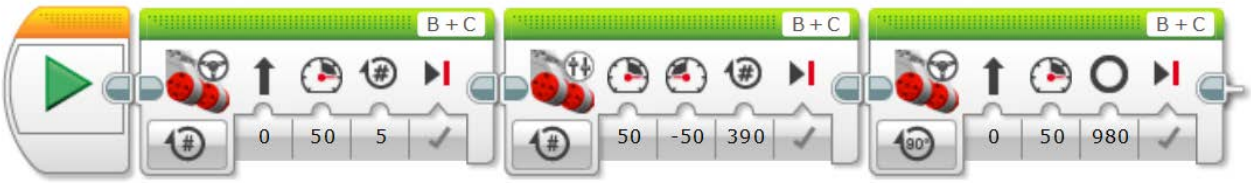
Программный блок "Независимое управление моторами" похож на программный блок "Рулевое управление". Он также управляет двумя большими моторами, только вместо параметра "Рулевое управление" появляется возможность независимого управления мощностью каждого мотора. При равном значении параметра "Мощность" для левого и правого мотора робот будет двигаться прямолинейно. Если на один мотор подать отрицательное значение мощности (например -50), а на второй - положительное значение (например 50), то робот будет разворачиваться на месте.

Например: Проехать прямолинейно вперед на 5 оборотов двигателя. Развернуться. Проехать на 980 градусов.

Используя программный блок «Рулевое управление» проехать вперед на 4 оборота.

Используя программный блок «Независимое управление моторами» развернуться на месте (также можно использовать блок «Рулевое управление», значение градусов придется подобрать экспериментально).

Используя программный блок «Рулевое управление» проехать вперед на 980 градусов.



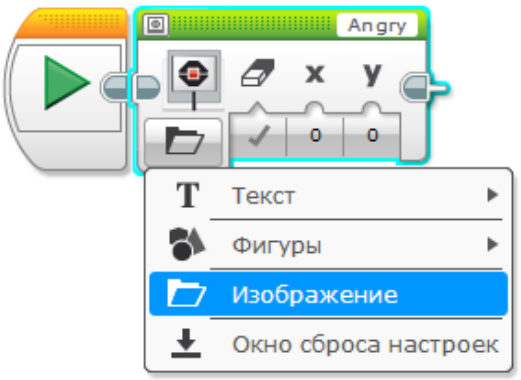
Значение параметра "Градусы" равно 390. Данное значение позволяет роботу развернуться вокруг своей оси. Если у вас другой робот, то вам придется подобрать другое значение.

Задание: Робот должен объехать препятствие (банка, мячик и т.п.) и вернутся к месту старта.

Задание: Робот должен проехать по траектории квадрат.

Экран, звук, индикатор состояния модуля

Для работы с дисплеем в EV3 предусмотрен отдельный блок – “Экран” (рис. 129).



Блок для работы с дисплеем

Этот блок имеет несколько режимов:

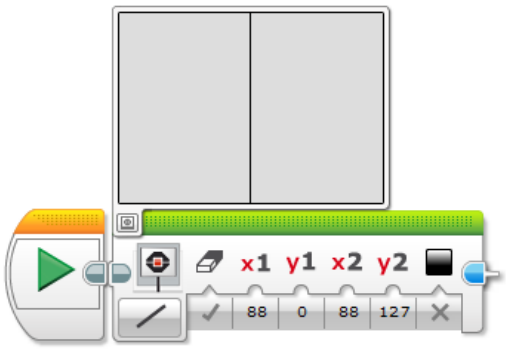
- вывод текста;
- вывод фигур
- вывод готового изображения;
- сброс настроек и вывод стандартного изображения;
- вывод пользовательского изображения.

Вы можете создавать более сложные изображения, поместив в программе подряд несколько блоков «Экран». Изображение может быть динамическим, т.е. изменяться во время выполнения программы, если подключить шины к

концентратору данных и передавать по ним новые значения координат. Запомните, что при выводе готового изображения в параметре «расположение» указываются координаты его левого нижнего угла.

Например:

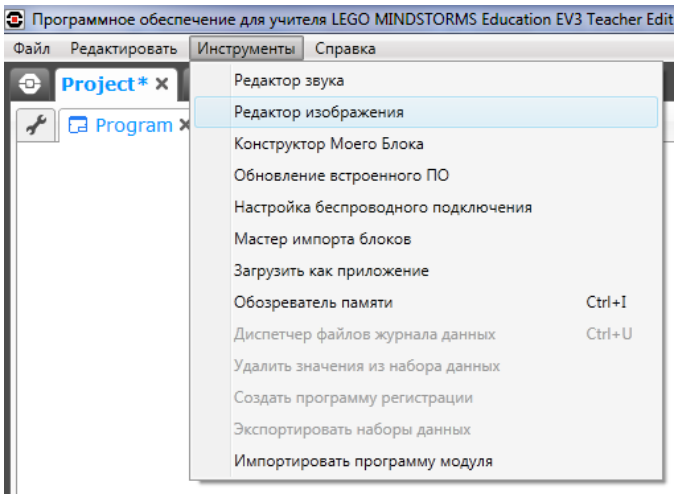
Напишем программу вывода на дисплей прямой линии, которая разделит его вертикально на две равные части. Для этого необходимо задать в параметрах команды координаты начальной и конечной точек прямой.



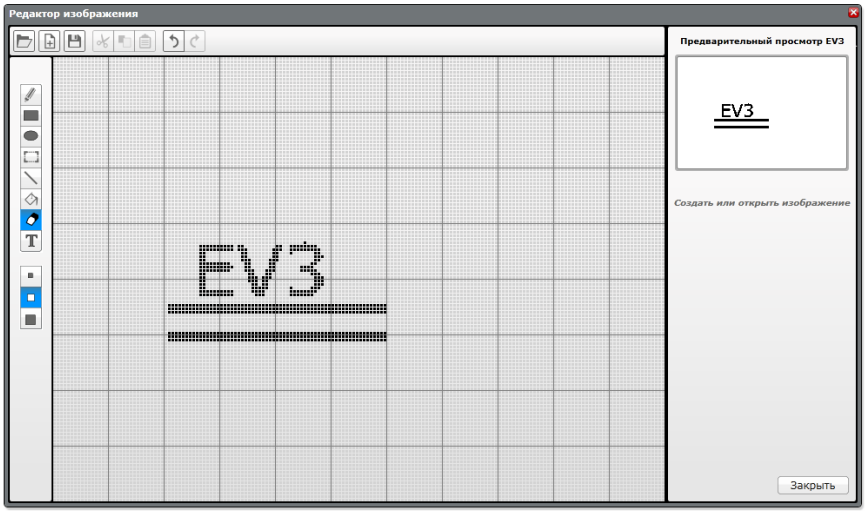
Выполняя данную программу, состоящую всего из одной команды, микроконтроллер очистит экран дисплея, а затем выведет на него изображение вертикальной прямой линии.

Задание: Самостоятельно измените данную программу так, чтобы она делила экран по горизонтали пополам.

Существует еще один способ создания собственного изображения на экране дисплея – для этого нужно включить редактор изображений. В этом режиме можно добавлять на экран готовые изображения. Однако среда EV3 предоставляет возможность самостоятельно создать изображение при помощи редактора изображений (Редактор изображений). Открыть редактор можно в меню “Инструменты” среды EV3.



Открываем редактор изображений



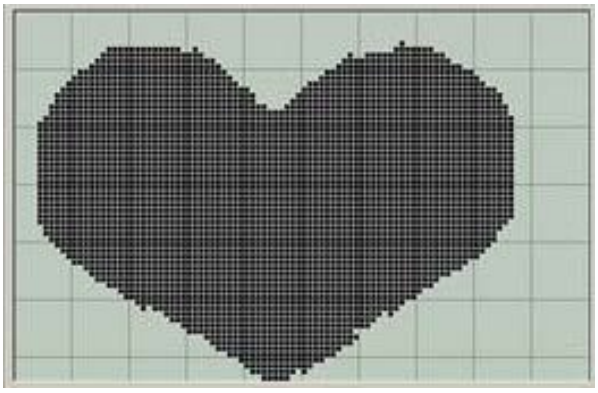
Поскольку дисплей блока EV3 черно-белый, редактор не поддерживает цветные палитры.

Помимо ручной отрисовки, редактор поддерживает импорт готовых изображений. Для этой цели служит кнопка “Открыть”, кнопка “Сохранить” позволяет сохранить готовое изображение и добавить его к списку изображений проекта. Редактор поддерживает форматы ric, jpg, jpeg, bmp, png.

Создадим собственное изображение в «Редакторе изображений» и заставим робота показывать его на дисплее. Пусть этим изображением будет бьющееся сердце, поскольку микроконтроллер, по сути, и есть сердце робота.

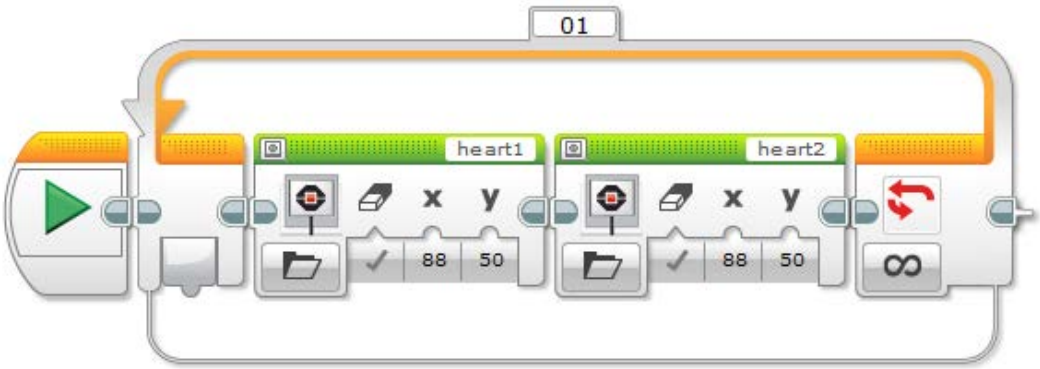
Нам необходимо два изображения сердца – одно большое, другое – маленькое. Когда Вы запустите в программе их циклическую смену, будет казаться, что у робота действительно бьется сердце.

Задание: Откройте «Редактор изображений». Нарисуйте в рабочей области сердце так, как вам больше нравится. Чтобы сделать его заметнее, можно воспользоваться заливкой. У Вас может получиться что-то похожее на рисунок.

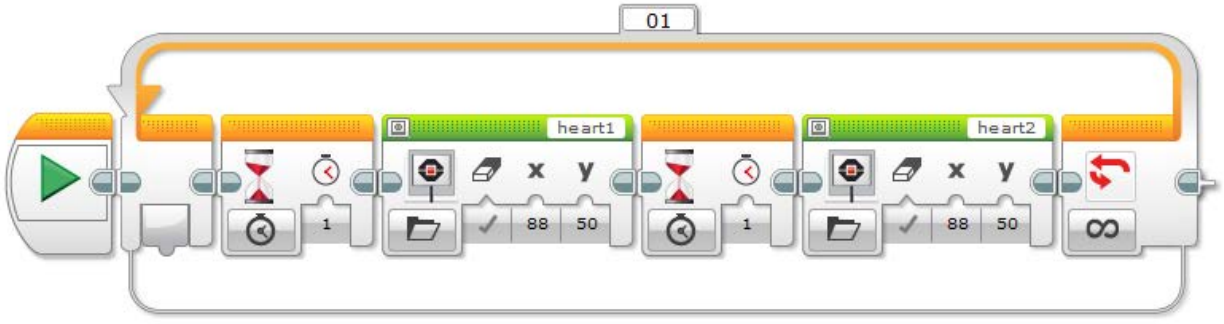


Сохраните полученное изображение с именем Heart1, нажав на кнопку сохранить.

Далее необходимо нарисовать еще одно сердце, но меньшего размера. Постарайтесь при этом оставить без изменений его форму. Сохраните второе изображение с именем Heart2 и закройте редактор. Теперь составим программу. Для этого поместите в рабочую область подряд два блока «Экран», заключив их в рамку команды «Цикл». Цикл необходим для того, чтобы изображения сменяли друг друга все время, пока работает программа. У Вас должна получиться структура, показанная на рисунке.



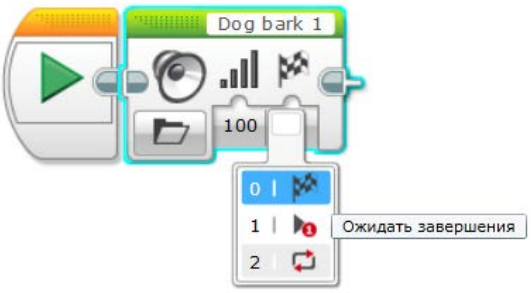
Но при запуске данная программа не работает. На экране Вы видите статическое изображение сердца, которое не изменяется. В чем же дело? Все дело в том, что программа работает правильно, но Вы не видите этого, потому что смена изображений происходит слишком быстро. Добавим в программный код два блока ожидания «Ожидание», установив в параметрах ожидание истечения времени в 1 секунду. Окончательный вариант программы представлен на рисунке.



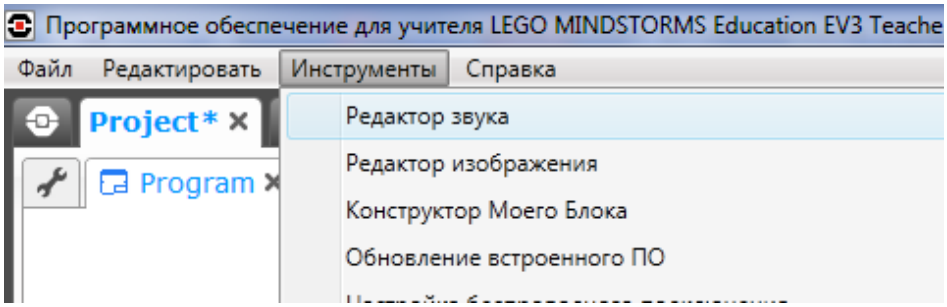
Звук. Работа с динамиком

Еще одним важным средством коммуникации робота является воспроизведение звуковых сигналов. Робот может также воспринимать звуки из окружающей среды при помощи звукового датчика. Для работы со звуком в языке EV3 предназначен блок «Звук».

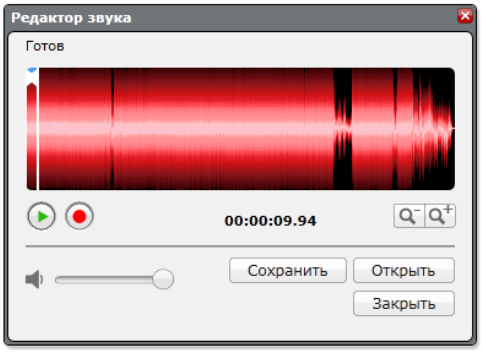
Звуки можно выбирать из стандартной библиотеки звуков. Если в блоке “Звук” установлен флажок “Ожидать завершения”, то звуковой файл будет воспроизведен полностью, и только потом осуществлён переход к следующей команде.



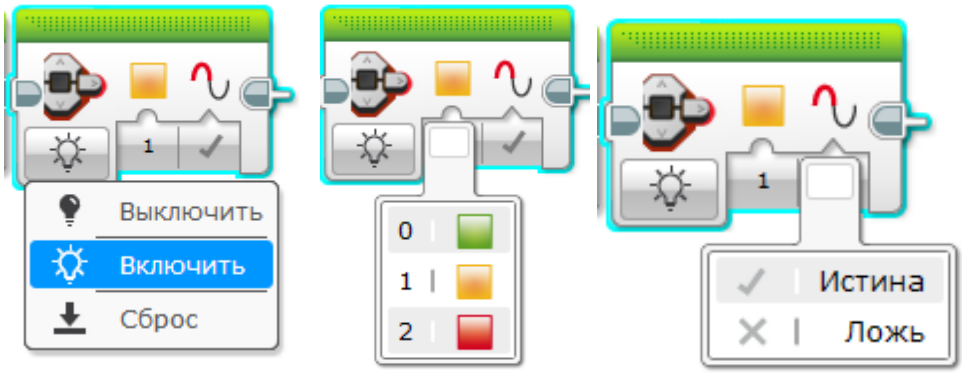
Так же можно создать собственную звуковую дорожку. Этот способ аналогичен подобному для блока «Экран»



Редактор звука позволяет открыть для редактирования звуковой файл формата rso, rsf, mp3 или wav. Нажав на кнопку “Открыть”, Вы можете выбрать файл произвольной длительности. В отличие от микрофонной записи, длина импортированного из операционной системы файла не ограничена. Но, как и в случае записи, сохранить для загрузки в команду «Звук» можно только 10 секунд из любого места файла.

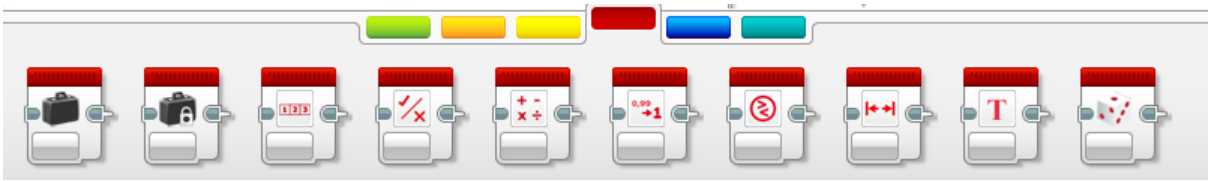


Нам осталось познакомиться с последним программным блоком зеленой палитры - блоком «Индикатор состояния модуля». Вокруг кнопок управления модулем EV3 смонтирована цветовая индикация, которая может светиться одним из трех цветов: зеленым, оранжевым или красным. За включение - выключение цветовой индикации отвечает соответствующий режим. Параметр "Цвет" задает цветовое оформление индикации. Параметр "Импульсный" отвечает за включение - отключение режима мерцания цветовой индикации. Как можно использовать цветовую индикацию? Например, можно во время различных режимов работы робота использовать различные цветовые сигналы. Это поможет понять: так ли выполняется программа, как мы запланировали.



2.5 Операции с данными.

В красной палитре среды программирования Lego Mindstorms EV3 сосредоточены программные блоки, необходимые для выполнения различных операций над числовыми, логическими или текстовыми данными.

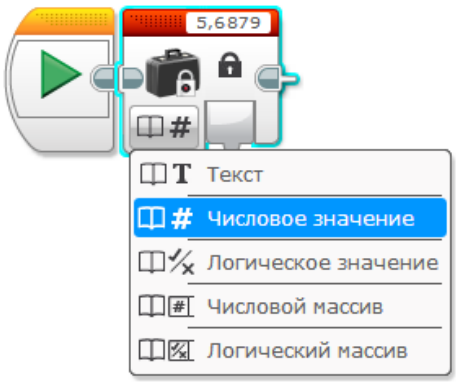


Среда программирования Lego mindstorms EV3 позволяет нам обрабатывать в своих программах пять различных типов данных: «Текст», «Числовое значение», «Логическое значение», «Числовой массив», «Логический массив». Тип данных «Числовое значение» позволяет нам выполнять различные математические операции над числами. Числа в программе могут быть как положительными, так и отрицательными, быть целыми значениями или содержать десятичную дробь. Например: 3,14, -2,5, 20.

Перед тем, как начать обрабатывать различные типы данных в наших программах, нам надо научиться их создавать и хранить. Для этих целей среда программирования предоставляет два вида программных

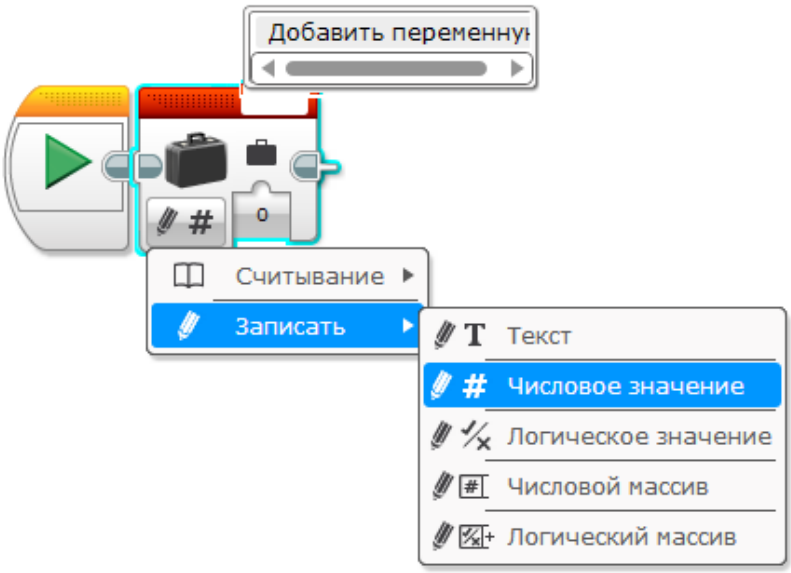
блоков: «Переменная» и «Константа». Эти блоки позволяют создать в памяти робота специальные ячейки, позволяющие записывать, извлекать и редактировать различные типы данных.

Программный блок «Константа» позволяет создавать ячейку памяти для хранения одного из пяти типов данных. Требуемое значение записывается в ячейку на этапе создания программы и остается неизменным во время выполнения всей программы. Для получения значения, записанного в блок



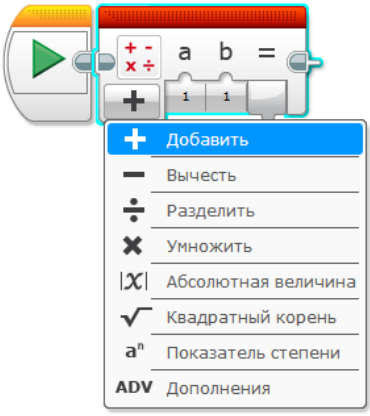
«Константа» используется «Вывод».

В программном блоке «Переменная» существует два режима «Считывание» и «Записать». Прежде чем использовать переменную, ей необходимо задать имя, выбрав параметр блока «Добавить переменную». Имя переменной может содержать только заглавные и строчные буквы латинского алфавита, цифры, а также символы _ и -. Задать значение переменной можно, записав или передав число в параметр «Значение».

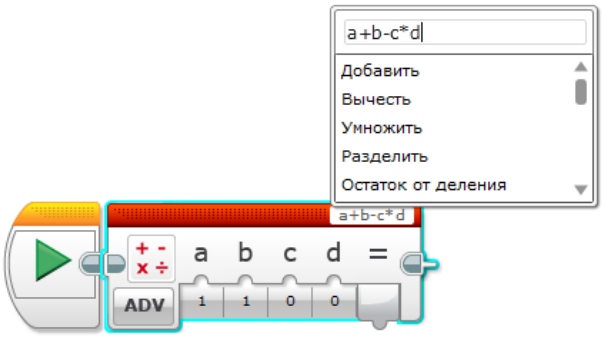


Блок математика

Блок математика служит для выполнения математических вычислений. Он позволяет выполнить выбранную математическую операцию над двумя числами, заданными параметрами "a" и "b".

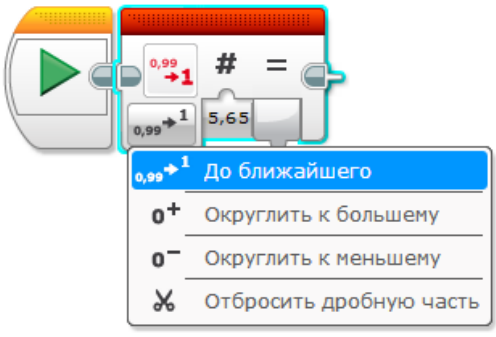


В режиме «Дополнения» количество параметров для расчета увеличивается до четырех. В параметр «Уравнение» можно вписать любую произвольную формулу.



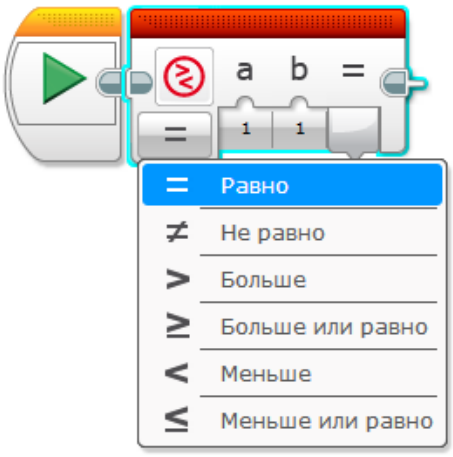
Блок «Округление»

Режимы «До ближайшего», «округлить к большему» и «округлить к меньшему» производят округление до целого значения. В режиме «Отбросить дробную часть» можно задать количество остающихся знаков дробной части после запятой.



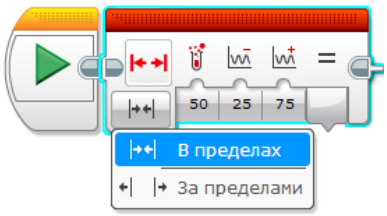
Блок «Сравнение»

Блок предназначен для сравнения двух числовых значений, которые могут вписываться в окна блока или приходить по проводникам. На выходе появляется логическое значение «Истина» или «Ложь», в зависимости от результата сравнения.

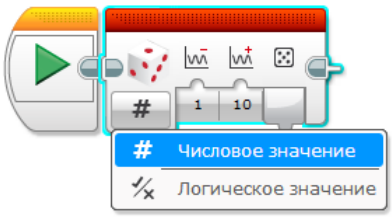


Блок «Интервал»

Блок предназначен для определения нахождения числа относительно диапазона: внутри или снаружи. Значения могут, как непосредственно вписываться в окна блока, так и приходить по проводника. На выходе блока появляется логическое значение «Истина» или «Ложь», в зависимости от полученного результата.



Блок «Случайное значение»

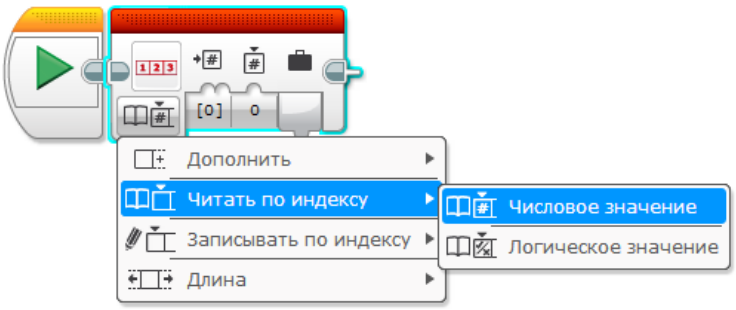


Блок предназначен для генерирования случайного значения (числового или логического) в указанном диапазоне.

Блок «Операции над массивом»

Массив – это набор однотипных элементов, расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу. Индекс первого элемента равен 0. В среде EV3 можно работать с одномерными числовыми и логическими массивами.

Блок Операции над массивом позволяет определять длину массива, читать, записывать и удалять требуемые элементы массива.



Прежде чем начать работу с массивами, необходимо их инициировать, т.е. указать тип (числовой или логический) и присвоить имя. Данные в массив можно вносить в ручном или автоматическом режиме (считывая показания с датчиков). Для создания массива необходимо использовать блок «Переменная».

Например: необходимо написать программу прямолинейного движения для проезда роботом расстояния в 0,5 метра.

Решение:

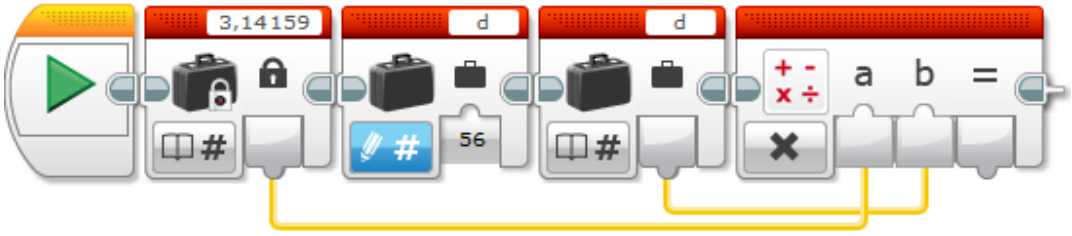
За один полный оборот мотора робот проезжает расстояние, равное длине окружности колеса. Это расстояние можно найти, умножив число Пи (=3,14159) на диаметр колеса. Диаметр колеса из образовательного набора Lego mindstorms EV3 равен 56 мм. Если переведем расстояние в 0,5 метра в миллиметры (500 мм) и разделим на расстояние, которое робот проходит за один оборот мотора, то узнаем: сколько оборотов мотора необходимо для проезда всего заданного расстояния.

Приступим к созданию программы:

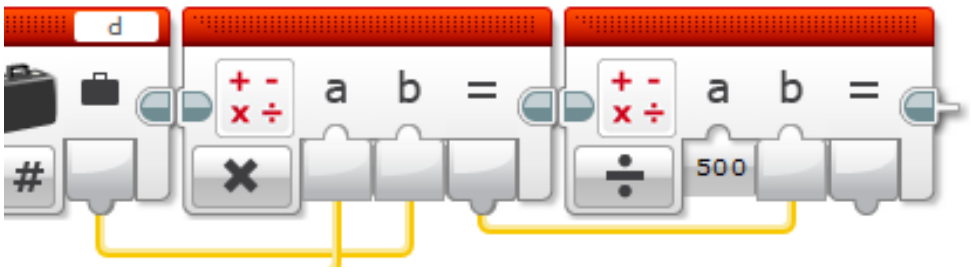
1. Используя программный блок «Константа», заведем в программу постоянное число Пи, равное примерно 3,14159.

2. Используя программный блок «Переменная», создадим в программе переменную «d» и занесем в нее значение диаметра колеса.

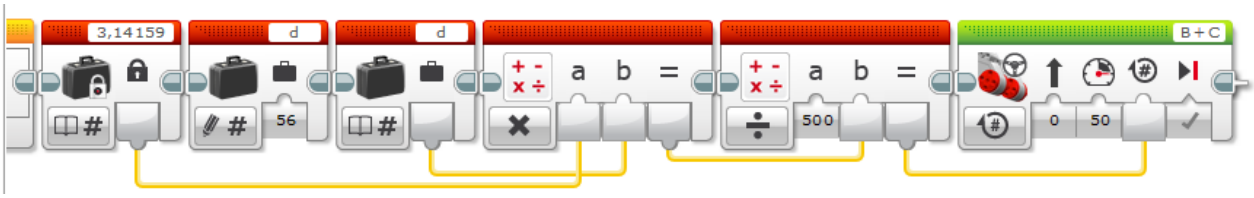
3. Используя программный блок «Математика», умножим значение блока «Константа» на значение переменной d. Для передачи значения из переменной d в программный блок «Математика» используем второй программный блок «Переменная» в режиме «Считывание». Для передачи значений между программными блоками используются шины данных. Чтобы установить шину данных, необходимо «потянуть» выходной параметр одного программного блока и «присоединить» его к входному параметру другого программного блока.



4. Используя программный блок «Математика», разделим значение пути (500 мм) на значение, полученное в шаге 3.



5. Полученное в шаге 4 значение подадим в параметр «Обороты» блока «Рулевое управление».



Загрузим полученную программу в нашего робота. Поставим робота на ровную свободную площадку и запустим программу. Измерив расстояние, пройденное роботом, убедимся в правильности нашей программы.

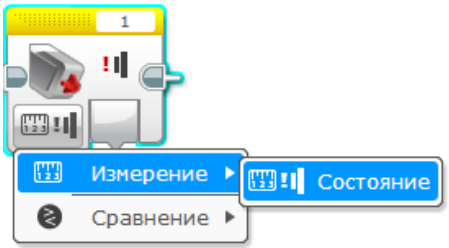
2.6. Работа с датчиками

Датчик или сенсор (англ. Sensor) – это первичный преобразователь, элемент измерительного, сигнального, регулирующего или управляющего устройства системы, который преобразует контролируемую величину в сигнал, удобный для измерения, передачи, преобразования хранения и регистрации информации о состоянии объекта измерений. Датчики широко используются при построении систем автоматизированного управления.

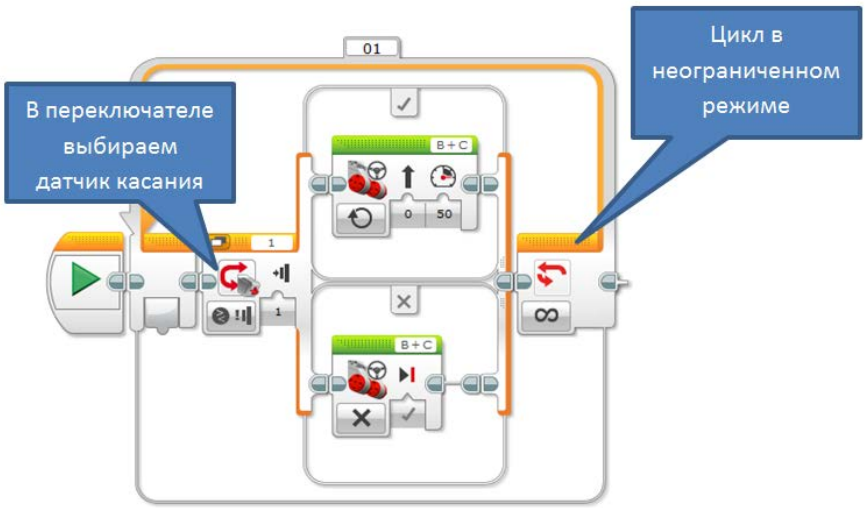
Датчики в среде программирования EV3 представлены жёлтой палитрой.

Датчик касания

Аналоговый датчик касания позволяет определить, нажата, отпущена или нажата и отпущена кнопка датчика. Датчик работает в трёх режимах: измерение, сравнение и ожидание.



Например: Создадим программу, при нажатии на датчик касания наш робот начинает двигаться, при отпускании кнопки робот останавливается.



Задача: Создать программу: при нажатии на датчик касания робот крутится на месте.

Датчик цвета

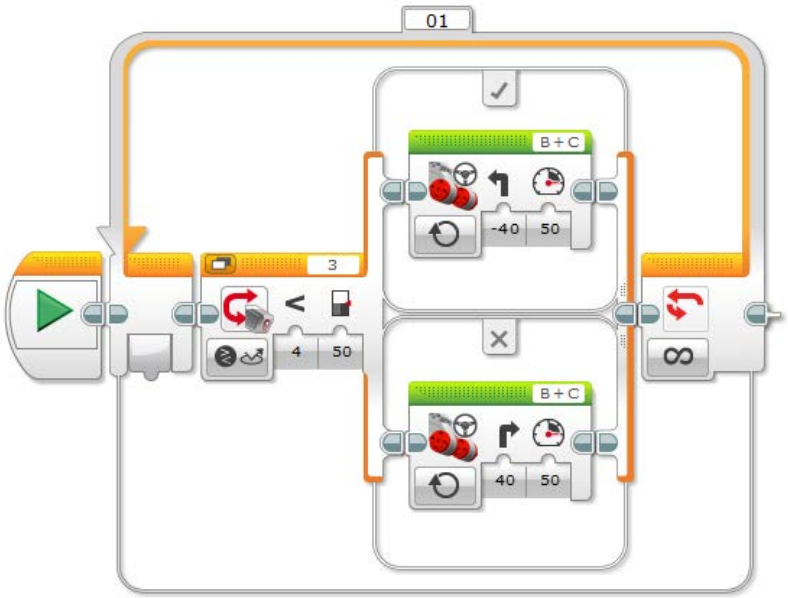
Цифровой датчик цвета EV3 способен:

Различить семь различных цветов, обозначенных цифрами (1 – чёрный, 2 – синий, 3 – зелёный, 4 – желтый, 5 – красный, 6 – белый, 7 – коричневый). А также:

- определить отсутствие цвета перед датчиком (код – 0);
- определить яркость отраженного от поверхности света по 100-бальной шкале;
- определить яркость внешнего освещения по 100-бальной шкале.

Этот датчик помогает в решении задачи движения по черной линии, сортировки и т.д.

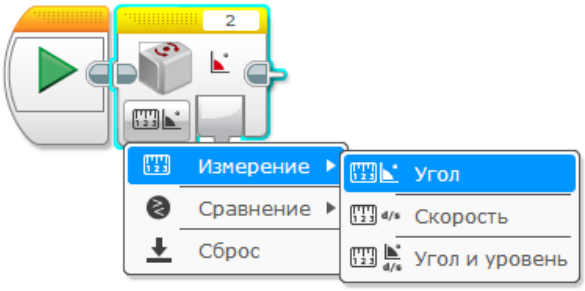
Например: создадим программу для движения по черной линии с одним датчиком цвета.



Задача: создать программу, чтобы робот озвучивал название цветов.

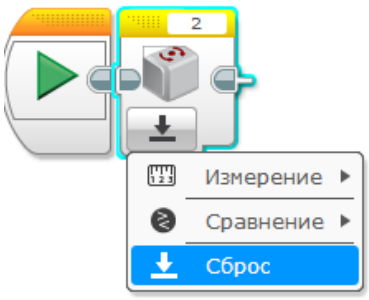
Гироскопический датчик

Цифровой гироскопический датчик предназначен для измерения угла и направления вращения робота, а также скорости его вращения. Точность измерения составляет +/-30 , максимальная скорость проведения измерений 4400/сек., частота опроса датчика 1кГц.

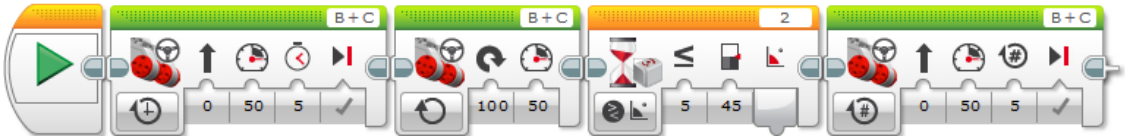


Гироскопический датчик определяет движение вокруг одной оси вращения. Это направление указано стрелкой на корпусе датчика. Угол и направление вращения может быть положительным или отрицательным. Вращение по часовой стрелке считается положительным, против часовой стрелки – отрицательным. Единица измерения скорости – градусы в секунду.

Будучи включенным, при длительном нахождении робота в неподвижном состоянии, определяемое значение угла не остаётся постоянным, а ошибочно меняется или дрейфует. Поэтому, чем больше времени проходит от начала первого обращения к гироскопическому датчику до чтения показаний, тем менее точным становятся результаты за счёт систематического накопления ошибки. Поэтому перед началом каждого измерения всегда необходимо производить обнуление угла при помощи режима Сброс.



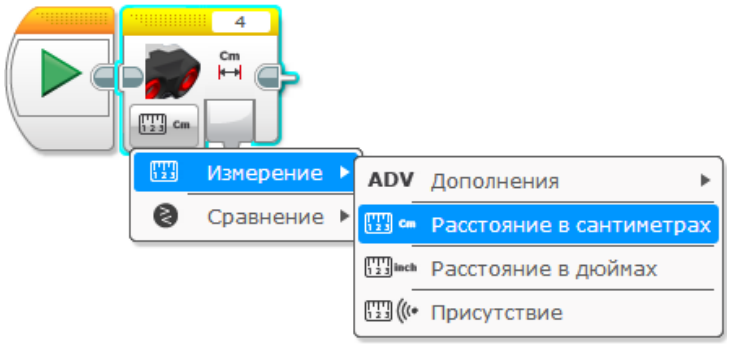
Например: создадим программу, что бы робот двигался вперед, поворачивался на 45 градусов и двигался дальше.



Задача: создать программу, чтобы робот проехал по траектории в виде квадрата.

Ультразвуковой датчик

Цифровой ультразвуковой датчик генерирует ультразвуковые волны и считывает их отражение для обнаружения и измерения расстояния до объектов. Он также может излучать или принимать волны от других ультразвуковых датчиков, например, для инициирования начала работы программы.

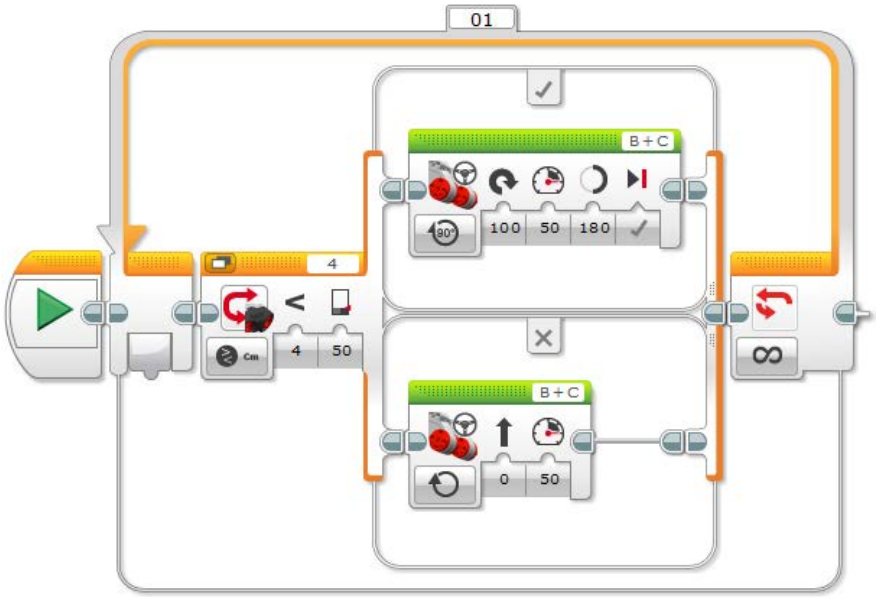


Минимальное определяемое расстояние – 3 см, максимальное – 250 см. Точность определения расстояния +/- 1 см. Если смотреть на датчик спереди, то левая сторона содержит передатчик сигнала, правая – приемник сигнала.

Угол излучения пучка лучей составляет примерно 200. Если мы поместим какой-либо объект на расстоянии 1 м перед датчиком, то ширина пучка лучей у объекта составит 65 см.

Ультразвуковой датчик работает в режиме измерения, сравнения и ожидания.

Например: создадим программу, что бы робот мог объезжать препятствия.

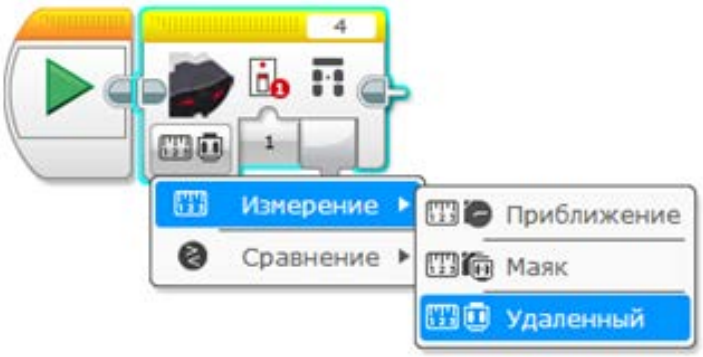


На рисунке, если расстояние до препятствия меньше 50 см. то робот поворачивает. Если расстояние от 50 и больше, то робот едет прямо.

Задача: создать программу для робота, который едет по прямой и останавливается перед препятствием и воспроизводит любой звук.

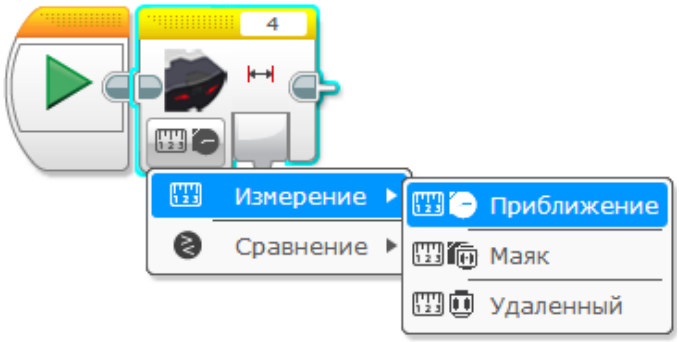
Инфракрасный датчик и маяк

Цифровой инфракрасный датчик (ИК-датчик) определяет расстояние до любого объекта, расстояние и расположение инфракрасного маяка, считывает и распознает сигналы от инфракрасного маяка. Маяк (ИК-маяк) позволяет осуществлять дистанционное управление роботом. Радиус действия маяка – до 2 метров.



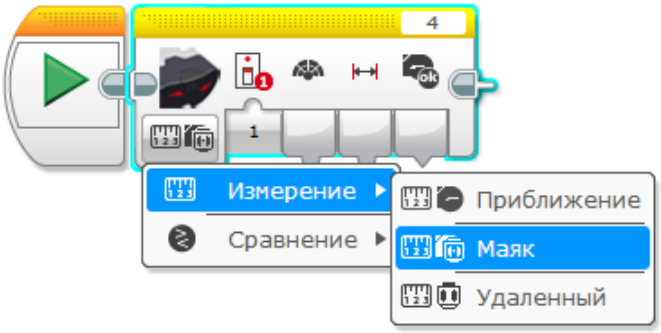
Режим определения относительного расстояния до объекта (Приближение).

В этом режиме датчик посылает инфракрасные сигналы и оценивает мощность отражённого сигнала. Если мы измеряем расстояние до объекта в том случае, когда он находится очень далеко, результат будет иметь значение 100,если очень близко (минимум 1 см) – 0. Мощность отражённого излучения сильно зависит от материала и цвета. Для определения точного расстояния используйте ультразвуковой датчик.



Режим определения расстояния и углового положения маяка

В режиме совместной работы с маяком инфракрасный датчик может обнаружить приблизительное относительное расстояние и угол отклонения маяка.

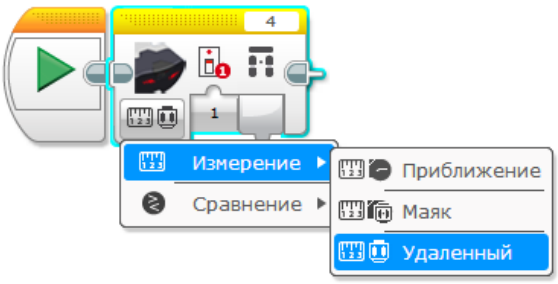


В случае, когда маяк расположен прямо перед датчиком, относительный результат измерения угла будет равен 0, максимальное расположение маяка слева, против часовой стрелки -25, справа по часовой стрелке 25.

Режим дистанционного управления.

Инфракрасный датчик способен определять состояние кнопок на маяке. Маяк имеет четыре канала, следовательно, можно организовать дистанционное управление одновременно четырьмя роботами.

Допускается одновременное нажатие двух управляющих кнопок. Каждый из этих вариантов пронумерован.



Например:

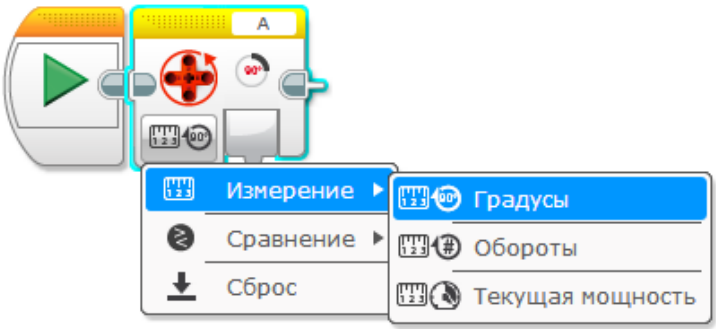
Инфракрасные лучи – это электромагнитное излучение, подчиняющееся законам оптики. Следовательно, чем светлее объект, тем большее количество лучей будет от него отражаться, чем темнее – тем меньшее количество лучей будет от него отражаться. Используя этот закон физики, мы можем проградуировать наш датчик так, чтобы он показывал не расстояние до объекта (будем полагать, что расстояние до всех объектов равны), а цвет объекта.

Установим робота с ИК-датчиком перед объектом, например, белого цвета, и зафиксируем показание, допустим 70 единиц. Затем направим датчик в сторону чёрного объекта (расстояние должно быть таким же) и зафиксируем результат, допустим 40 единиц. Повторим действия с объектами других цветов. Объекты должны отличаться только цветом, форма, размер и материал должны быть одинаковыми.

В программе пропишем: если измеренное расстояние попадает в диапазон от X до Y единиц, значит, цвет белый и т.д. Практика показывает, что разница между черными и белыми объектами на расстоянии около 30 см может достигать 20 единиц. В случае, когда необходимо определять цвета, например, красный, зелёный, синий, разброс значений будет не таким большим, но его можно обнаружить, Значения также будут различаться в зависимости от материала и температуры объекта.

Датчик Вращение мотора

Большой и средний моторы содержат встроенные датчики вращения, которые могут определять количество совершённых оборотов или градусов, а также мгновенную мощность мотора.



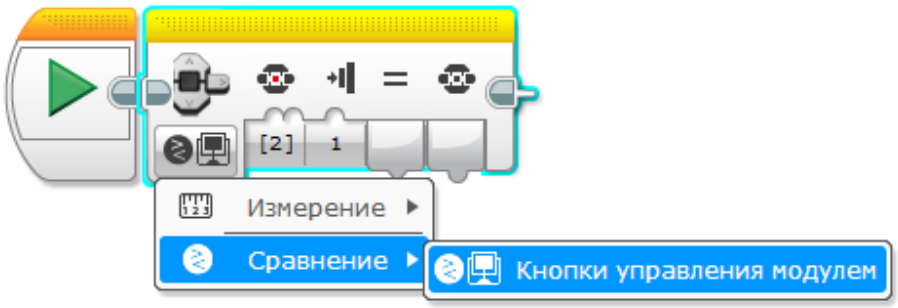
Подобно любому другому датчику датчик вращения используется в режимах измерения, ожидания и сравнения. Отсчёт сделанных оборотов (или градусов) начинается с момента запуска программы. Поэтому если необходим точный отсчёт оборотов или градусов, начиная с определенного момента, необходимо произвести сброс значений датчика. Если отсчёт начинается с начала выполнения программы, сброс не обязателен.

При повороте оси мотора по часовой стрелке происходит суммирование всех оборотов, при повороте оси против часовой стрелки – вычитание.

Кнопки управления модулем

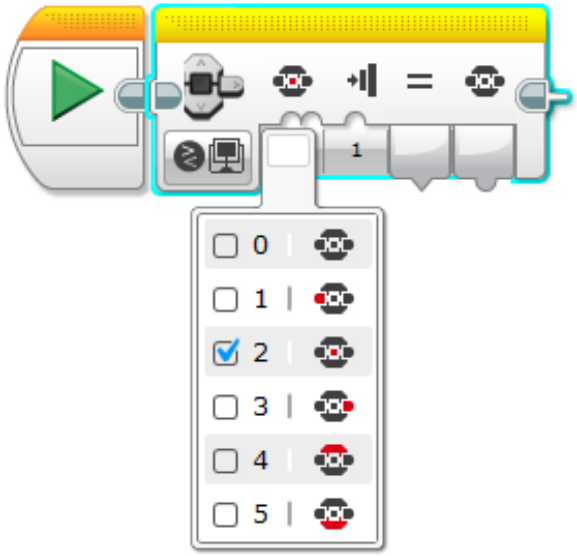
Модуль EV3 анализирует данные от пяти кнопок управления модулем (влево, центр, вправо, вверх и вниз), расположенных на передней панели. Существует возможность узнать, какая кнопка нажата, проверить режим нажатия (Нажатие, освобождение или щелчок) и получить логический вывод о нажатии (Истина или Ложь).

На рисунке программный блок «Кнопки управления модулем EV3».



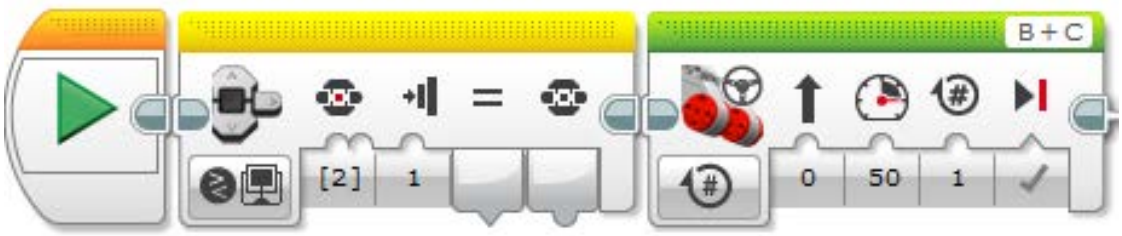
Кнопка «Назад» на модуле EV3 не относится к кнопкам управления модулем, а прерывает выполнение программы. Одновременное нажатие двух кнопок недопустимо.

У каждой кнопки есть свой номер.



Как и в других датчиках кнопки имеют режимы Сравнение, Измерение и Изменение.

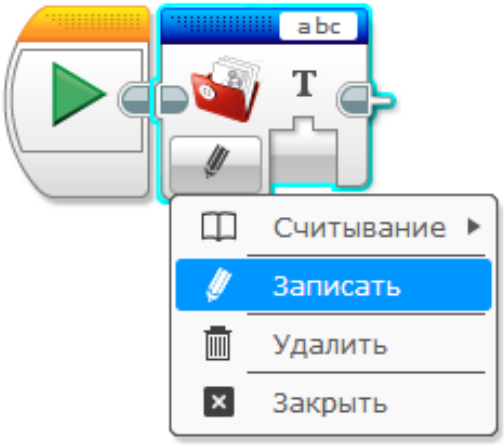
Например: создадим программу, которая активируется нажатием на среднюю кнопку EV3.



2.7 Синяя палитра «Дополнения».

Работа с файлами

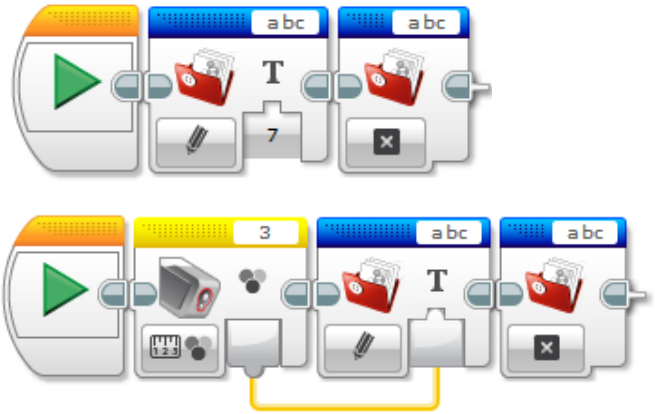
Работа с файлами позволяет хранить информацию после выключения питания робота.



Алгоритм работы с файлами:

- Запись данных в файл (если до этого момента файл не существовал, программа создаст его);
- Закрытие файла;
- Чтение данных из файла.

Записать данные в файл можно непосредственно в блоке или передать через проводник. Третий блок в каждом варианте – блок «Доступ к файлу» в режиме «Закрыть».



На рисунке ввод данных непосредственно в файл и через проводник

Файлы сохраняются и загружаются вместе с текущим проектом. Для того чтобы передать файл на компьютер, необходимо выполнить следующее:

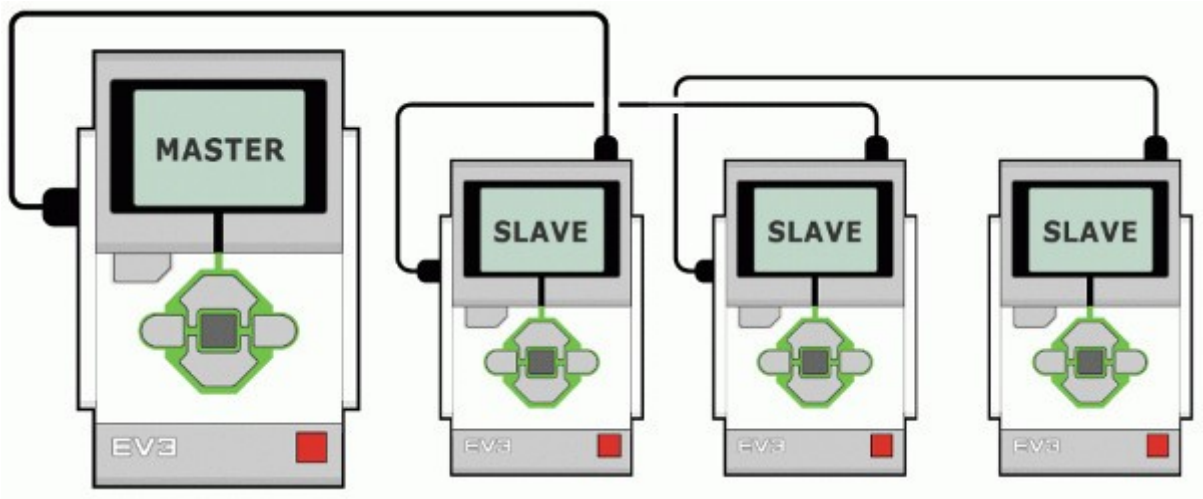
Выберите пункт «Инструменты – Обозреватель памяти».

Дважды щелкните левой кнопкой мыши по названию проекта и выберите название файла. Выберите «Загрузить».

Совместная работа нескольких роботов

Соединение роботов кабелем USB

Существует возможность связывать вместе до четырёх блоков EV3 USB-кабелем. После этого можно загружать программу в главный блок EV3, управлять моторами и считывать показания с датчиков всех связанных блоков.



На рисунке представлена схема последовательного подключения блоков EV3 – USB-порт сбоку на первом модуле EV3 соединяется с мини-портом USB следующего модуля EV3 с помощью USB-кабеля, входящего в комплект робота. Порт USB сбоку второго блока EV3 соединяется с мини-портом USB третьего блока EV3 и т.д.

Для включения режима последовательного подключения необходимо вызвать окно «Свойства проекта», щёлкнув по значку гаечный ключ в левом верхнем углу окна проекта и установить флажок «Режим подключения шлейфом».

При подключении робота к компьютеру в правом нижнем углу окна проекта в режиме Просмотр портов появятся четыре слоя, показывающие состояние моторов и датчиков всех подключенных роботов.

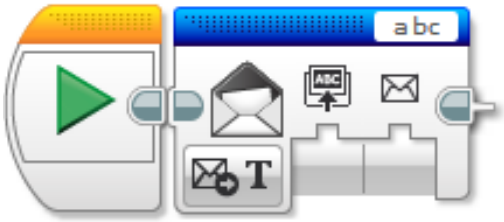
Далее создаем программу и загружаем её в первый (главный «master») блок EV3. Каждый программный блок датчика или мотора теперь имеет дополнительное окно для выбора номера блока EV3. Такие действия, как проигрывание звука, вывод текста или изображения на экран, подсветка кнопок и анализ состояния кнопок, возможны только на первом блоке EV3.

Связь роботов с помощью Bluetooth-соединения

Что бы связать роботов необходимо включить Bluetooth и сделать их видимым. Пусть мы связываем два робота – EV31 и EV32. Для этого последовательно ставим флажки в пунктах Bluetooth и Visibility (видимость). Затем на каждом роботе выбираем Connections и Search (поиск).

Робот EV31 найдет робота EV32, а робот EV32 обнаружит робота EV31. После ввода одинаковых паролей (по умолчанию 1234) роботы будут соединены и готовы к совместной работе.

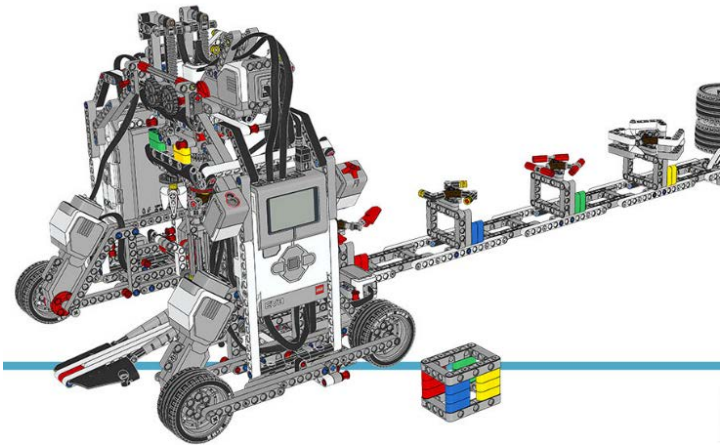
Для работы нам будет необходим программный блок Обмен сообщениями, находящийся на синей программной палитре.



Для обмена сообщениями необходимо указать данные, подобно тем, которые мы указываем при отправлении писем на почте:

- Куда (название блока EV3, в который мы посылаем сообщение или от кого (название EV3-блока, от которого мы ждём сообщение);
- Название сообщения;
- Содержание сообщения;
- тип сообщения (текстовое, числовое, логическое).

Пример совместной работы двух роботов «Фабрика спиннеров»



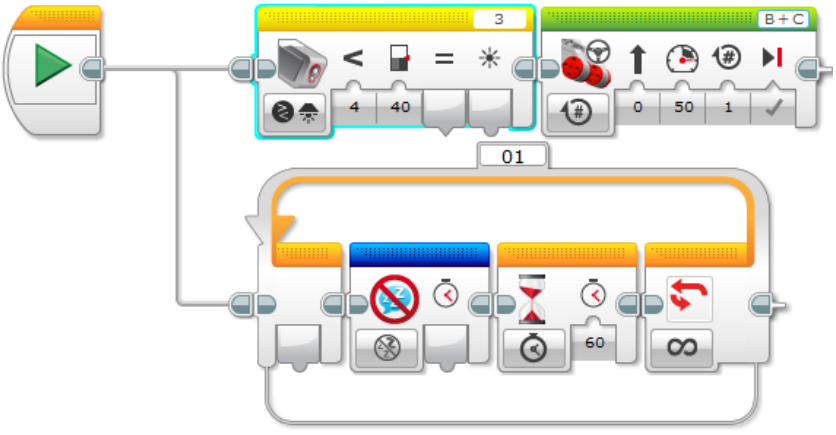
Полезные блоки и инструменты

Блок «Поддерживать в активном состоянии»

Когда мы не обращаемся к роботу и робот не выполняет никаких операций, по истечении определённого времени он выключается. Это вызывает неудобство при работе с программами, рассчитанными на длительные ожидания каких-либо процессов.

При каждом обращении к данному программному блоку происходит сброс значения таймера выключения робота на определённое число минут, продлевая время работы без выключения.

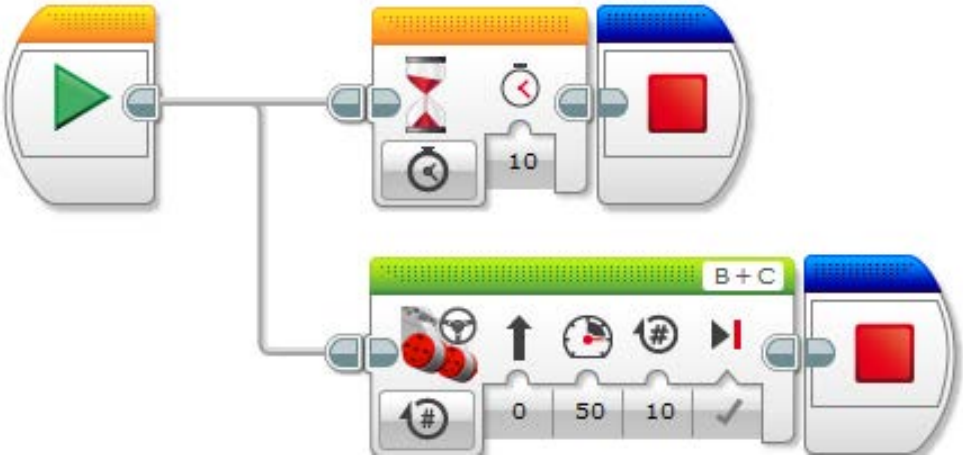
Например: Если яркость внешнего освещения меньше 40, тогда робот открывает жалюзи. А чтобы робот не отключился, мы каждую минуту сбрасываем таймер выключения с помощью блока «Поддерживать в активном состоянии».



Блок «Остановить программу»

Блок вызывает завершение работы всей программы, может быть установлен в любом месте программы.

Например: Мотор, подключенный в порт А будет крутиться 10 оборотов или 10 секунд

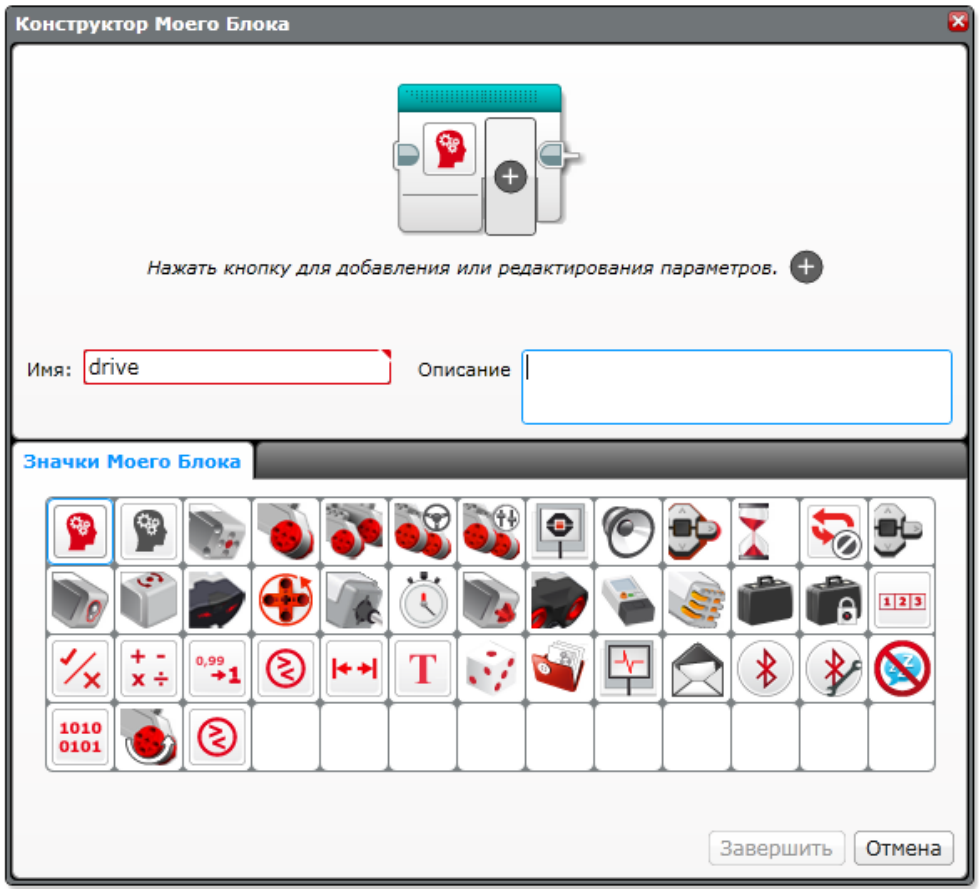


Создание подпрограмм

Подпрограмма – это поименованная или иным образом идентифицированная часть компьютерной программы, содержащая описание определённого набора действий. Подпрограмма может быть многократно вызвана из разных частей программы. Подпрограмма может быть многократно вызвана из разных частей программы.

Использование блоков подпрограмм позволяет сократить запись программы, сделать её более читабельно и простой в восприятии.

Для создания подпрограммы необходимо выделить блоки которые будут составлять подпрограмму и нажать «инструменты – конструктор моего блока»



Необходимо вписать имя блока, также можно добавить описание и выбрать значок.

Заключение

Мы рассмотрели основы для программирования EV3 роботов. Все программы, представленные в качестве примеров, являются действующими. Робота для программирования данных примеров можно собрать на творческой основе.

Для тестирования моделей использовались в качестве поля листы ламинированного ДСП белого цвета. Освещение – верхнее искусственное, основанное на энергосберегающих лампах.

Для правильной работы программ, необходимо корректировать значения соответствующих коэффициентов, в зависимости от получаемых вами реальных данных. Не бойтесь экспериментировать и создавать свои версии программ, ведь в экспериментах рождаются новые идеи.

При написании учебного пособия использовались материалы с сайтов:

education.lego.com

www.lego.com

robot-help.ru