

Программирование моторов конструктора Lego mindstorms EV3

Содержание урока

Введение:

На втором занятии мы детальнее познакомимся со средой программирования и подробно изучим команды, задающие движение нашему роботу-тележке, собранному на первом занятии. Итак, давайте запустим среду программирования Lego mindstorms EV3, загрузим наш проект lessons.ev3, созданный ранее и добавим в проект новую программу - lesson-2-1. Программу можно добавить двумя способами:

- Выбрать команду "Файл"-**"Добавить программу"** (Ctrl+N).
- Нажать "+" на вкладке программ.

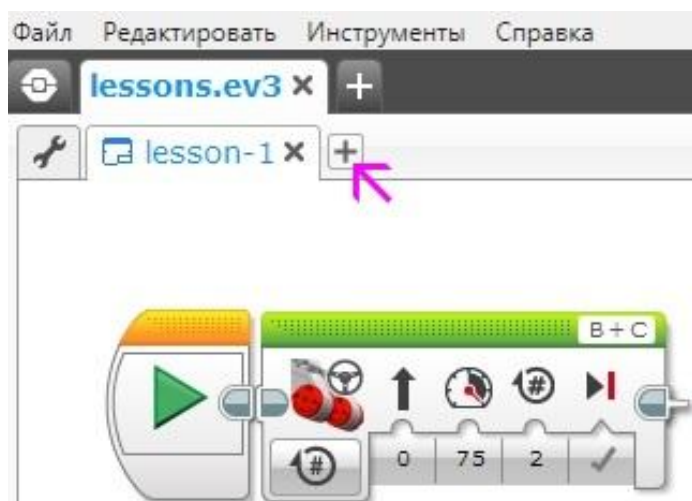


Рис. 1

2.1. Палитры программирования и программные блоки

Давайте теперь обратим свой взгляд в нижний раздел среды программирования. Из материала первого занятия мы уже знаем, что здесь находятся команды для программирования робота. Разработчики применили оригинальный прием и, сгруппировав программные блоки, присвоили каждой группе свой цвет, назвав группы палитрами.

Зеленая палитра называется: **"Действие"**:

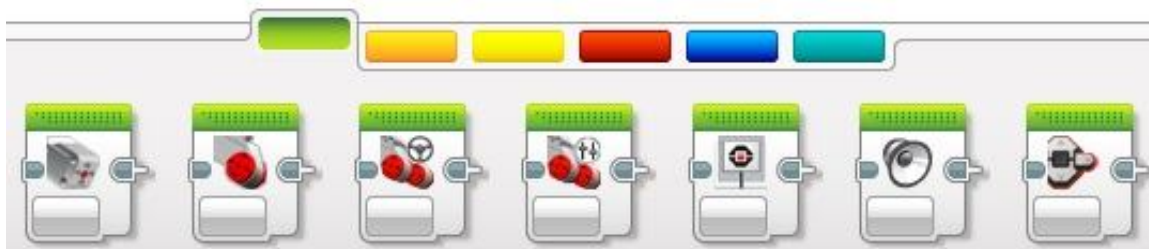
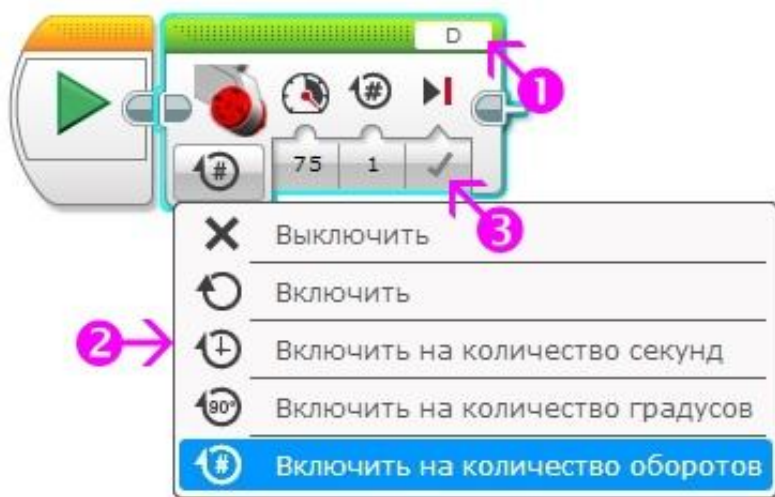


Рис. 2

На данной палитре расположены программные блоки управления моторами, блок вывода на экран, блок управления индикатором состояния модуля. Сейчас мы начнем изучение этих программных блоков.

2.2. Зеленая палитра – блоки действия

Первый программный блок зеленой палитры предназначен для управления средним мотором, второй блок - для управления большим мотором. Так как параметры этих блоков идентичны - рассмотрим настройку на примере блока - большой мотор.

**Рис. 3**

Для правильной настройки блока управления большим мотором мы должны:

1. Выбрать порт, к которому подключен мотор (**A, B, C или D**) (**Рис. 3 поз. 1**)
2. Выбрать режим работы мотора (**Рис. 3 поз. 2**)
3. Настроить параметры выбранного режима (**Рис. 3 поз. 3**)

Чем же отличаются режимы? 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 (на Рис. 3 мощность установлена в 75) 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 **"В"**, а правый - к порту **"С"**. Но вы можете в настройках блока поменять порты подключения в соответствии с требованиями вашей конструкции (**Рис. 4 поз. 1**).

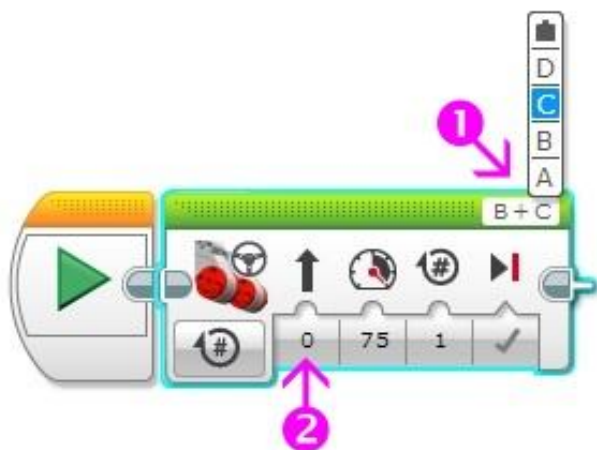


Рис. 4

Параметр **"Рулевое управление"** (**Рис. 4 поз. 2**) может принимать значения от -100 до 100. Отрицательные значения параметра заставляют робота поворачивать налево, при значении равном 0 робот движется прямо, а положительные значения заставляют робота поворачивать направо. Стрелка над числовым параметром меняет свою ориентацию в зависимости от значения, подсказывая тем самым направление движения робота (**Рис. 5**).

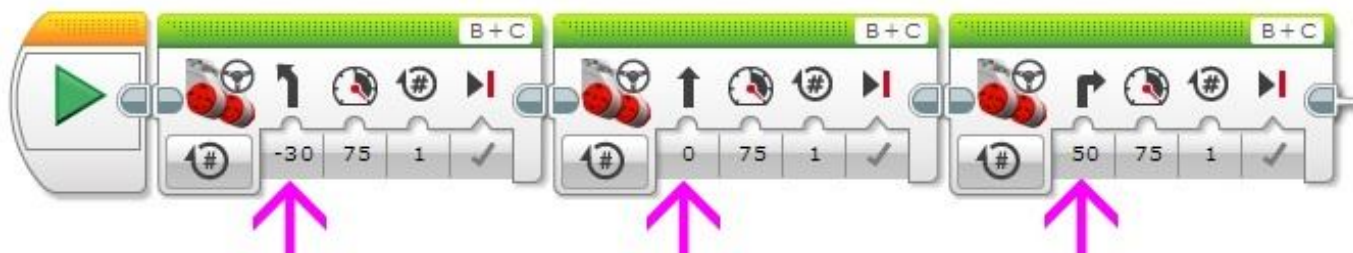


Рис. 5

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 (**Рис. 6**).

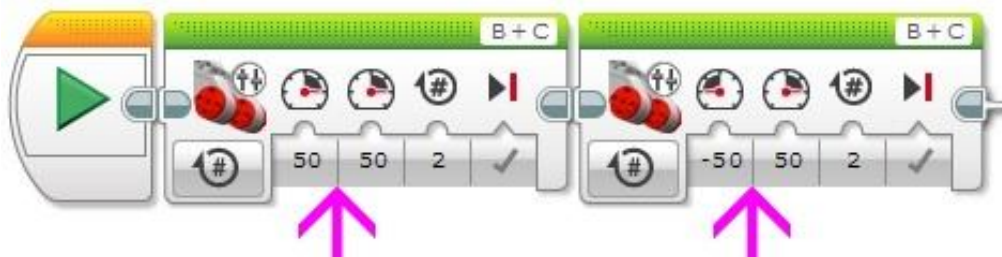


Рис. 6

Режимы работы этих блоков аналогичны режимам блока управления одним мотором, поэтому дополнительного описания не требуют...

2.3. Прямолинейное движение, повороты, разворот на месте остановка

Итак, теперь мы можем написать программу движения робота по какому-либо маршруту.

Задача 1: Проехать прямолинейно вперед на 4 оборота двигателя. Развернуться. Проехать на 720 градусов.

Решение (**Рис. 7**):

1. Используя программный блок "Рулевое управление" проехать вперед на 4 оборота.
2. Используя программный блок "Независимое управление моторами" развернуться на месте (значение градусов придется подобрать экспериментально).
3. Используя программный блок "Рулевое управление" проехать вперед на 720 градусов.

Примечание: Почему при развороте пришлось подбирать значение градусов в **блоке 2**? Разве не **360** градусов - искомая величина? Нет, если мы зададим значение параметра "**Градусы**" равным **360**, то тем самым заставим на искомую величину провернуться валы левого и правого моторов нашего робота. На какой угол повернется робот вокруг своей оси - зависит от размера (диаметра) колес и расстояния между ними. На **Рис. 7** значение параметра "**Градусы**" равно **385**. Данное значение позволяет роботу, собранному по инструкции **small-robot 45544** развернуться вокруг своей оси. Если у вас другой робот, то вам придется подобрать другое значение. Можно ли это значение найти математически? Можно, но об этом мы поговорим позднее.

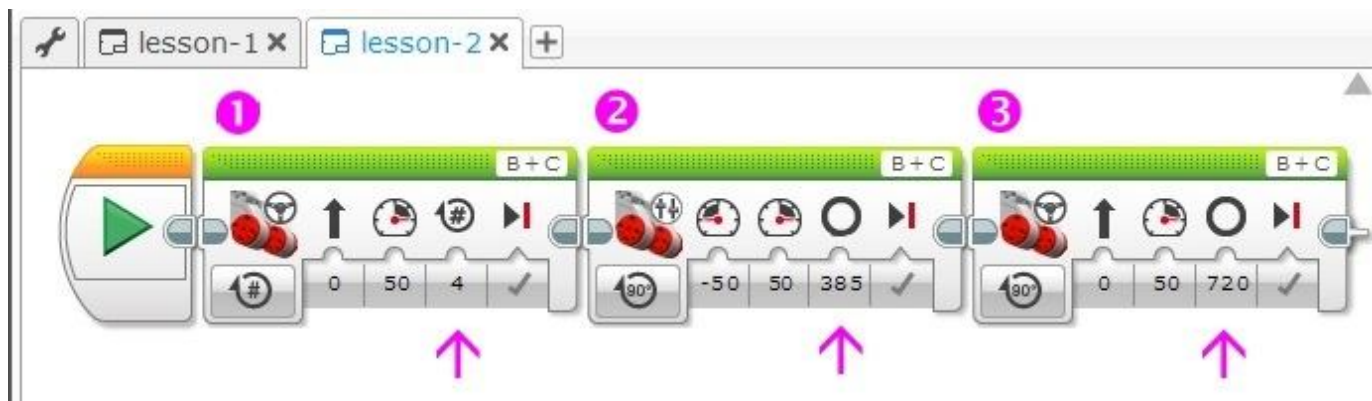


Рис. 7

Задача 2: Установите на ровной поверхности какое-либо препятствие (банку, кубик, небольшую коробку), отметьте место старта вашего робота. Создайте в проекте новую программу: lesson-2-2, позволяющую роботу объехать вокруг препятствия и вернуться к месту старта.

Сколько программных блоков вы использовали? Поделитесь своим успехом в комментарии к уроку...

2.4. Экран, звук, индикатор состояния модуля

Программный блок **"Экран"** позволяет выводить текстовую или графическую информацию на жидкокристаллический экран блока EV3. Какое это может иметь практическое применение? Во-первых, на этапе программирования и отладки программы можно выводить на экран текущие показания датчиков во время работы робота. Во-вторых, можно выводить на экран название промежуточных этапов выполнения программы. Ну а в-третьих, с помощью графических изображений можно "оживить" экран робота, например с помощью мультипликации.

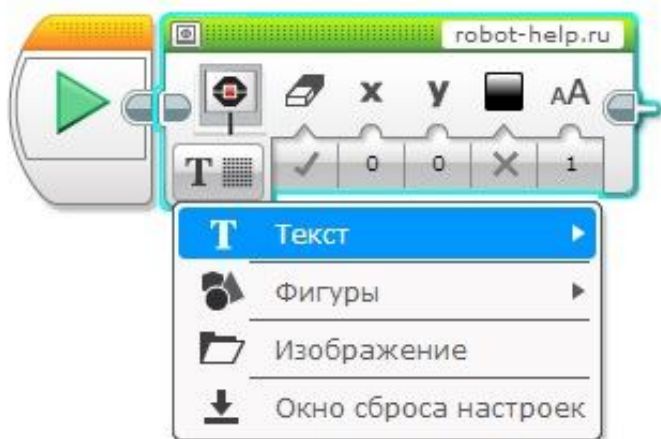


Рис. 8

Программный блок **"Экран"** имеет четыре режима работы: режим **"Текст"** позволяет выводить текстовую строку на экран, режим **"Фигуры"** позволяет отображать на экране одну из четырех геометрических фигур (прямоугольник, круг, квадрат, точка), режим **"Изображение"** может вывести на экран одно изображение. Изображение можно выбрать из богатой коллекции изображений или нарисовать свое, используя редактор изображений. Режим **"Окно сброса настроек"** сбрасывает экран модуля EV3 к стандартному информационному экрану, показываемому во время работы программы.

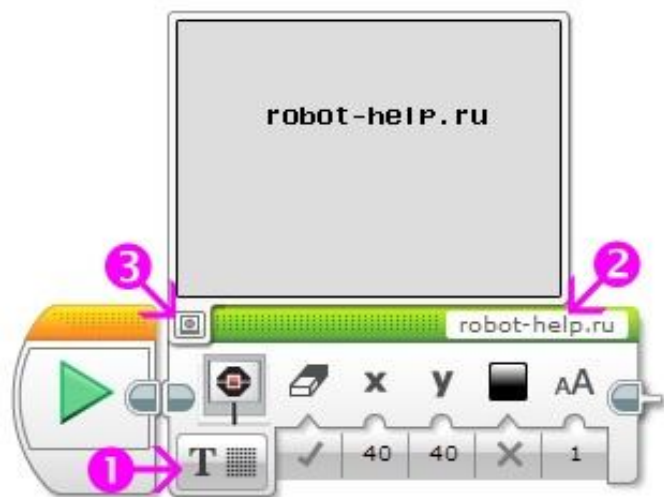


Рис. 9

Рассмотрим параметры программного блока **"Экран"** в режиме **"Текст"** (Рис. 9 поз.1). Строка, предназначенная для вывода на экран, вводится в специальное поле (Рис. 9 поз. 2). К сожалению, в поле ввода текста можно вводить только буквы латинского алфавита, цифры и знаки препинания. Если режим **"Очистить экран"** установлен в значение **"Истина"**, то экран перед выводом информации будет очищен. Поэтому, если вам требуется объединить текущий вывод с информацией уже находящейся на экране, то установите этот режим в значение **"Ложь"**. Режимы **"X"** и **"Y"** определяют точку на экране, с которой начинается вывод информации. Экран блока EV3 имеет 178 пикселей (точек) в ширину и 128 пикселей в высоту. Режим **"X"** может принимать значения от 0 до 177, режим **"Y"** может принимать значения от 0 до 127. Верхняя левая точка имеет координаты (0, 0), правая нижняя (177, 127)

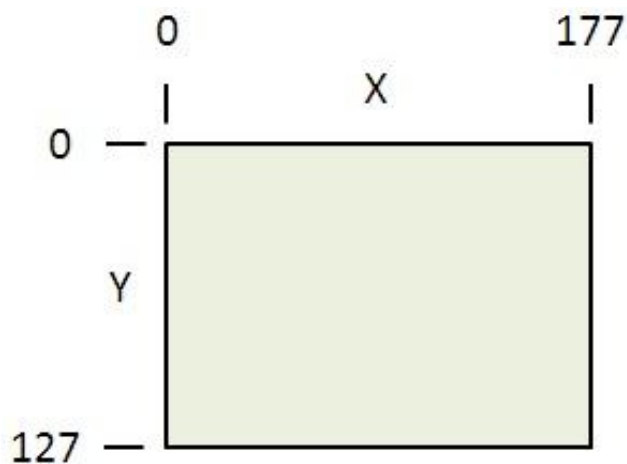


Рис. 10

Во время настройки программного блока **"Экран"** можно включить режим предварительного просмотра (Рис. 9 поз. 3) и визуально оценить результат настроек вывода информации.

В режиме **"Фигуры"** (Рис. 11 поз. 1) настройки программного блока меняются в зависимости от типа фигуры. Так при отображении круга необходимо будет задать координаты **"X"** и **"Y"** центра окружности, а также значение **"Радиуса"**. Параметр **"Заполнить"** (Рис. 11 поз. 2) отвечает за то, что будет отображен либо контур фигуры, либо внутренняя область фигуры будет заполнена цветом, заданным в параметре **"Цвет"** (Рис. 11 поз. 3).

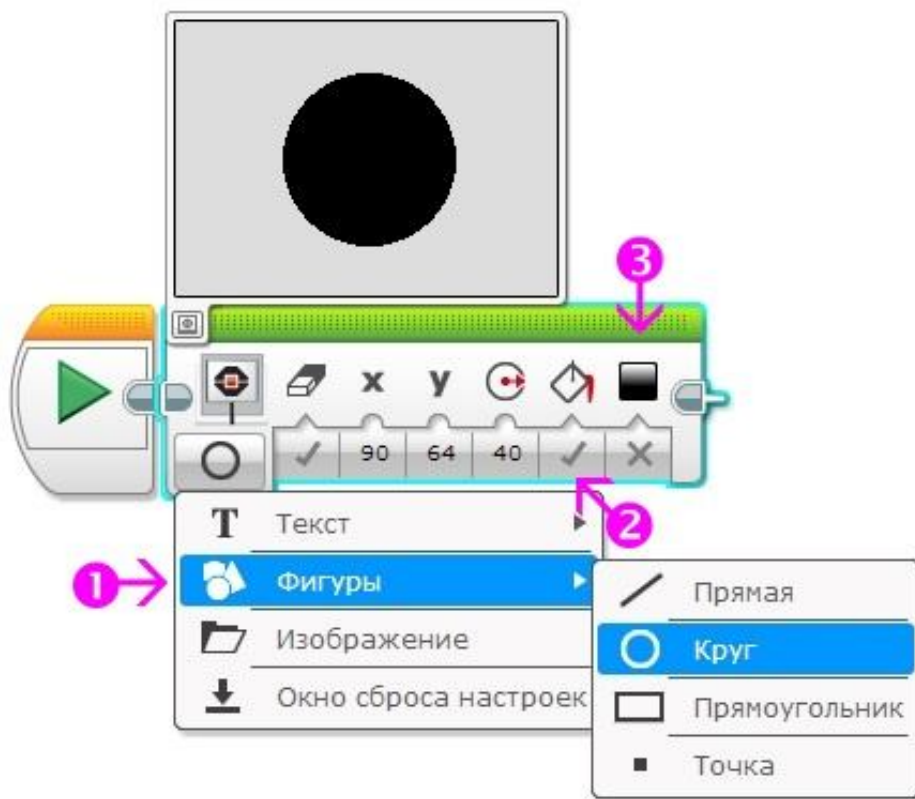


Рис. 11

Для отображения прямой необходимо задать координаты двух крайних точек, между которыми располагается прямая.

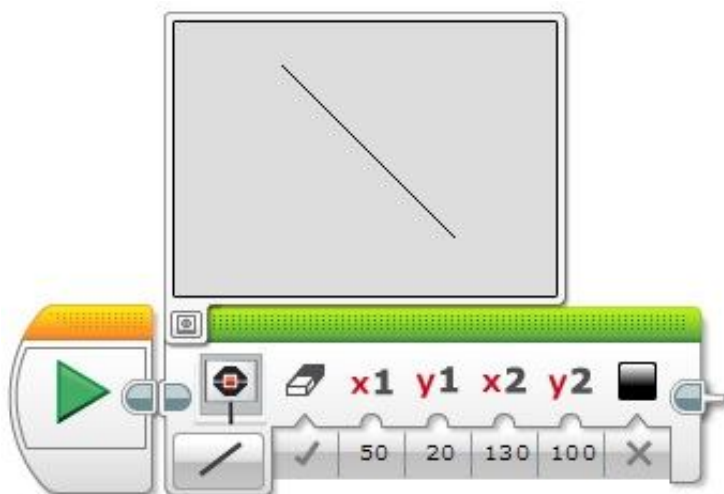


Рис. 12

Чтобы отобразить прямоугольник следует задать координаты "X" и "Y" левого верхнего угла прямоугольника, а также его "Ширину" и "Высоту".

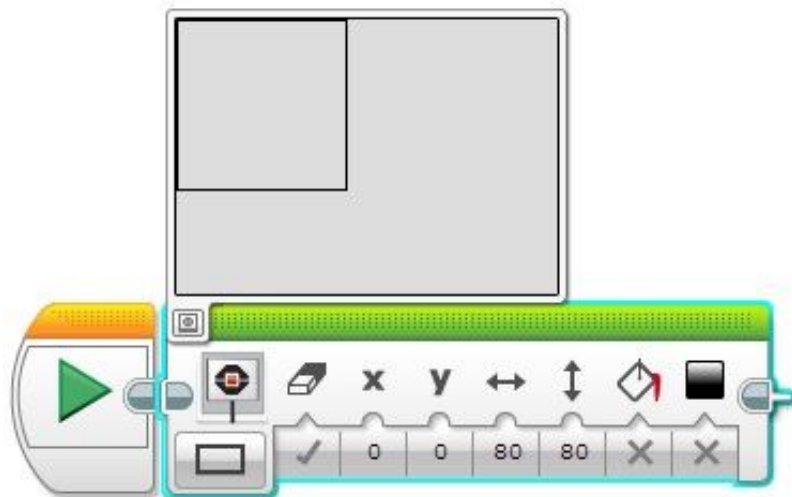


Рис. 13

Отобразить точку проще всего! Укажите лишь её координаты "X" и "Y".

Режим **"Изображение"**, наверное, самый интересный и самый используемый режим. Он позволяет выводить на экран изображения. Среда программирования содержит огромную библиотеку изображений, отсортированную по категориям. В дополнение к имеющимся изображениям вы всегда можете создать свой рисунок и, вставив его в проект, вывести на экран. (**"Главное меню среды программирования" - "Инструменты" - "Редактор изображения"**). Создавая своё изображение, вы можете также вывести на экран символы русского алфавита.

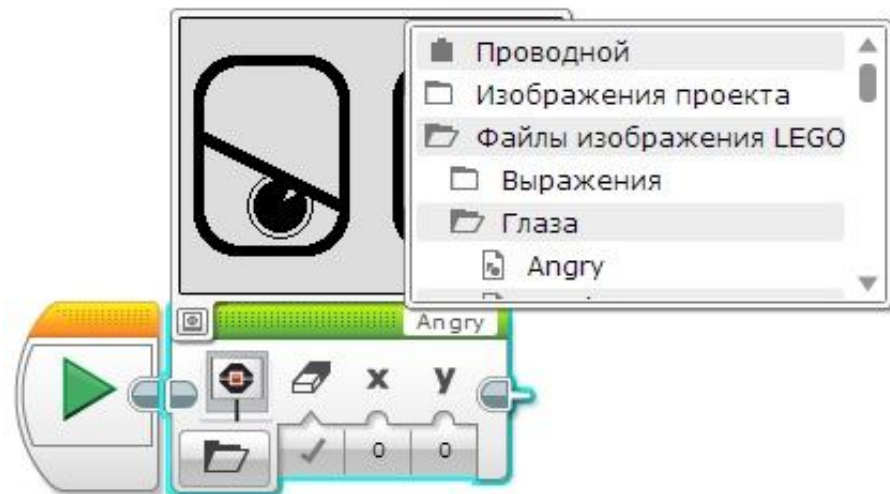


Рис. 14

Как вы видите - отображению информации на экране главного модуля EV3 среда программирования придает огромное значение. Давайте рассмотрим следующий важный программный блок **"Звук"**. С помощью этого блока мы можем выводить на встроенный динамик блока EV3 звуковые файлы, тона произвольной длительности и частоты, а также музыкальные ноты. Давайте рассмотрим настройки программного блока в режиме **"Воспроизвести тон"** (Рис. 15). В этом режиме необходимо задать **"Частоту"** тона (Рис. 15 поз. 1), **"Продолжительность"** звучания в секундах (Рис. 15 поз. 2), а также громкость звучания (Рис. 15 поз. 3).

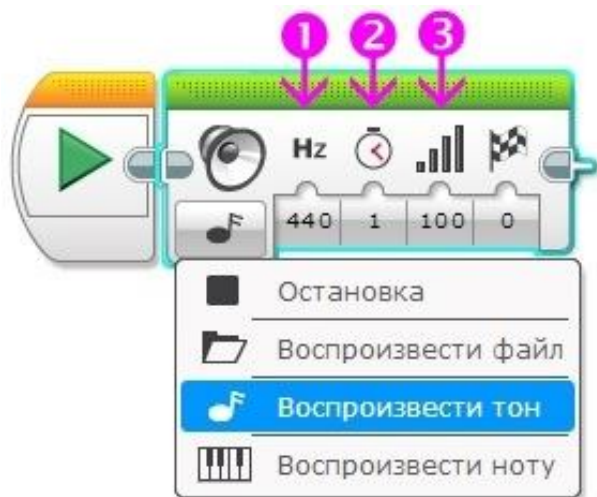


Рис. 15

В режиме **"Воспроизвести ноту"** вам вместо частоты тона необходимо выбрать ноту на виртуальной клавиатуре, а также установить длительность звучания и громкость (**Рис. 16**).



Рис. 16

В режиме **"Воспроизвести файл"** вы можете выбрать один из звуковых файлов из библиотеки (**Рис. 17 поз. 1**), либо, подключив к компьютеру микрофон, с помощью Редактора звука ("Главное меню среды программирования" - "Инструменты" - "Редактор звука") записать собственный звуковой файл и включить его в проект.

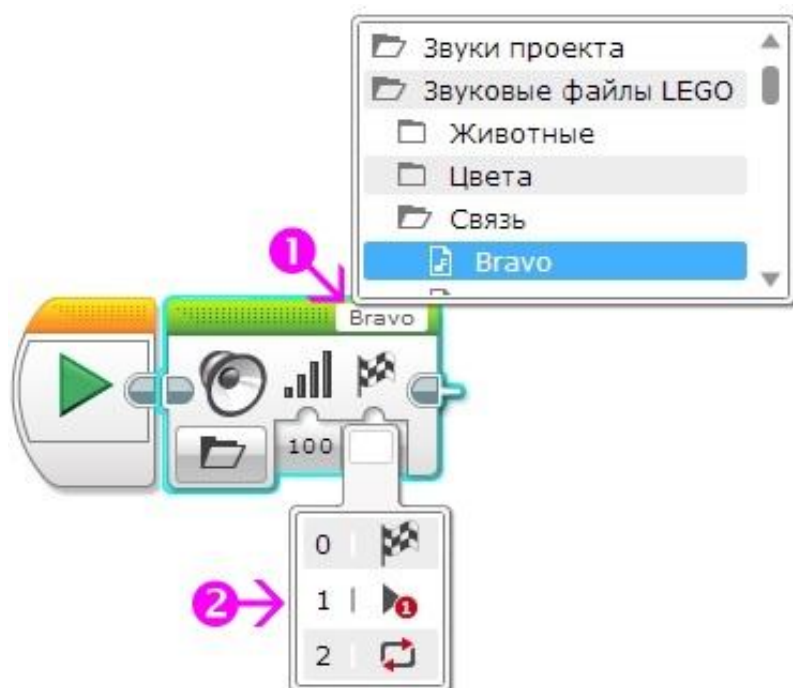


Рис. 17

Давайте отдельно рассмотрим параметр **"Тип воспроизведения"** (Рис. 17 поз. 2), общий для всех режимов программного блока **"Звук"**. Если данный параметр установлен в значение **"Ожидать завершения"**, то управление следующему программному блоку будет передано только после полного воспроизведения звука или звукового файла. В случае установки одного из двух следующих значений начнется воспроизведение звука и управление в программе перейдет к следующему программному блоку, только звук или звуковой файл будет воспроизведен один раз или будет повторяться, пока не его не остановит другой программный блок **"Звук"**.

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 (Рис. 18 поз. 1). Параметр **"Цвет"** задает цветовое оформление индикации (Рис. 18 поз. 2). Параметр **"Импульсный"** отвечает за включение - отключение режима мерцания цветовой индикации (Рис. 18 поз. 3). 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.

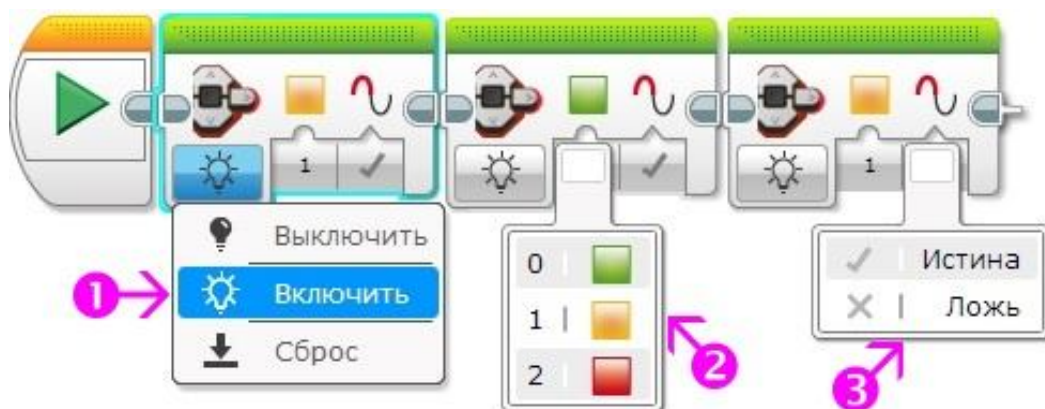


Рис. 18

Давайте используем полученные знания на практике и немного "раскрасим" нашу программу из Задачи 1.

Задача 3:

1. Воспроизвести сигнал **"Start"**
2. Включить зеленую немигающую цветовую индикацию
3. Отобразить на экране изображение **"Forward"**
4. Проехать прямолинейно вперед на 4 оборота двигателя.
5. Включить оранжевую мигающую цветовую индикацию
6. Развернуться
7. Включить зеленую мигающую цветовую индикацию
8. Отобразить на экране изображение **"Backward"**
9. Проехать на 720 градусов

10. Воспроизвести сигнал **"Stop"**

Попробуйте решить задачу 3 самостоятельно, не подглядывая в решение! Удачи!

Решение задачи 3