

Знакомство с вычислительными возможностями робота Lego mindstorms EV3

Содержание урока

Введение:

Наше третье занятие мы посвятим изучению вычислительных возможностей модуля EV3 и разберем примеры практических решений задач на вычисление траектории движения. Снова запускаем среду программирования Lego mindstorms EV3, загружаем наш проект lessons.ev3 и добавляем в проект новую программу - lesson-3-4. Добавлять новую программу в проект мы научились с вами на предыдущем уроке.

3.1. Красная палитра – операции с данными

Программные блоки, необходимые для выполнения различных операций над числовыми, логическими или текстовыми данными, сосредоточены в красной палитре среды программирования Lego mindstorms EV3. Красная палитра содержит 10 программных блоков. В отличие от зеленой палитры - с программными блоками красной палитры мы будем знакомиться постепенно, по мере продвижения по курсу программирования и возникновения необходимости в новых программных конструкциях.

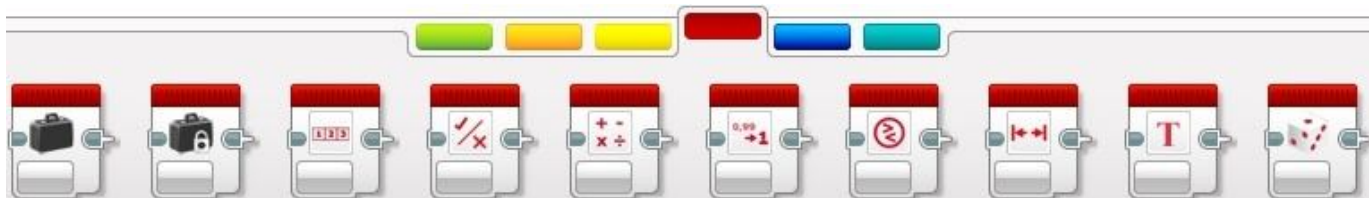


Рис.1

3.2. Числовые значения. Блок "Константа", блок "Переменная"

Среда программирования Lego mindstorms EV3 позволяет нам обрабатывать в своих программах пять различных типов данных: "Текст", "Числовое значение", "Логическое значение", "Числовой массив", "Логический массив". В сегодняшнем уроке мы научимся оперировать с числовыми данными. 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. Примеры: **-15; 3,145; 8; -247,34.**

Перед тем, как начать обрабатывать различные типы данных в наших программах, нам надо научиться их создавать и хранить. Для этих целей среда программирования Lego mindstorms EV3 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"** (Рис. 2) позволяет создавать ячейку памяти для хранения одного из пяти типов данных (Рис. 2 поз. 1). Требуемое значение записывается в ячейку на этапе создания программы (Рис. 2 поз. 2) и остается неизменным во время выполнения всей программы. Для получения значения, записанного в блок **"Константа"** используется **"Вывод"** (Рис. 2 поз. 3). Подробнее с извлечением данных из программных блоков мы познакомимся ниже при решении практической задачи Урока №3.

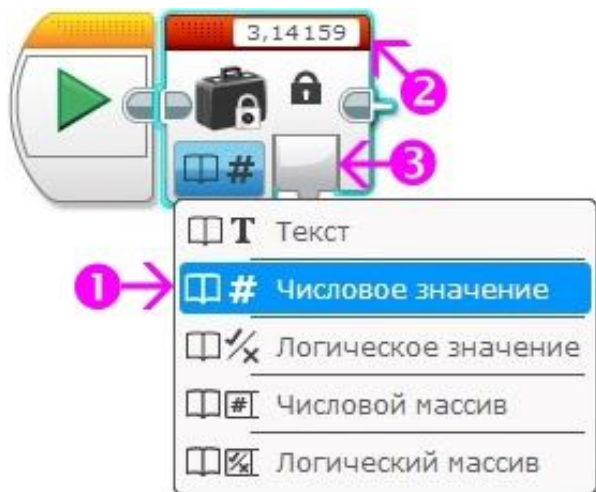


Рис. 2

В отличие от программного блока **"Константа"** - в блоке **"Переменная"** присутствуют два режима **"Считывание"** и **"Записать"** (Рис. 3 поз. 1). Перед первым использованием необходимо задать имя переменной, выбрав параметр блока **"Добавить переменную"** (Рис. 3 поз. 2). 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"** (Рис. 3 поз. 3).

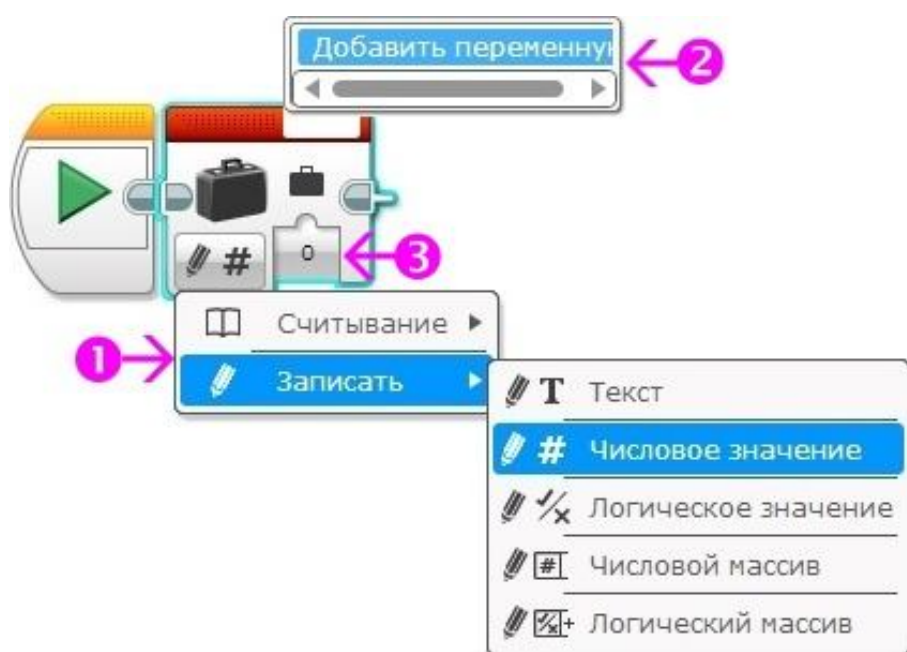


Рис. 33.3. Блок математика, блок округление

Для выполнения математических вычислений служит программный блок **"Математика"**. Он

позволяет выполнить выбранную математическую операцию (Рис. 4 поз. 1) над двумя числами, заданными параметрами "a" и "b". В режимах "Абсолютная величина" и "Квадратный корень" для вычисления доступен только один параметр "a".

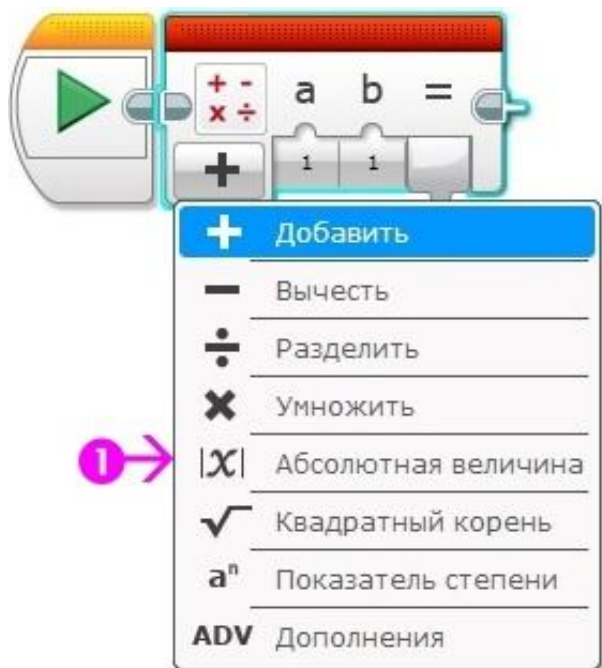


Рис. 4

Отдельно следует остановиться на режиме "Дополнения". В этом режиме количество параметров для расчета увеличивается до четырех: "a", "b", "c" и "d". В параметр "Уравнение" (Рис. 5 поз. 1) можно вписать любую произвольную формулу, производящую вычисления с этими параметрами.

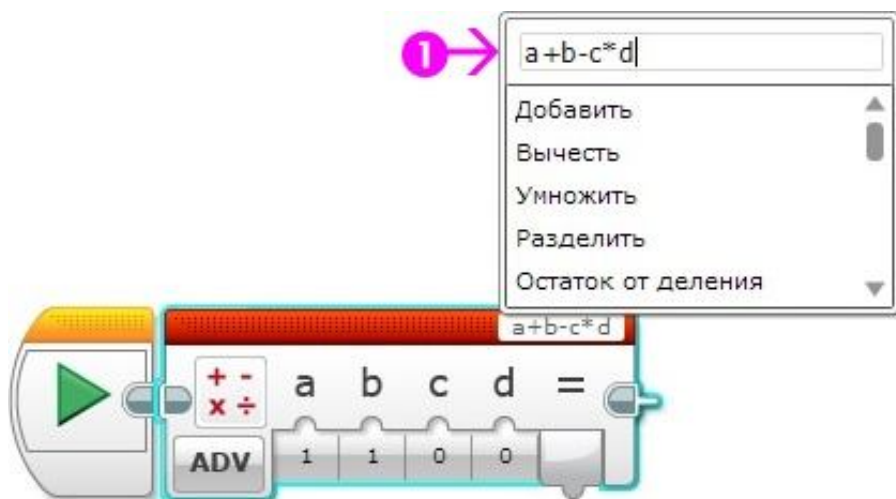


Рис. 5

Иногда возникает необходимость произвести округление результата вычисления. Например: при отладке программы, можно выводить на экран модуля EV3 округленные промежуточные расчеты, чтобы легче было визуально контролировать ход выполнения программы. Для этого предназначен программный блок "Округление" (Рис. 6). 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.



Рис. 6

3.4. Примеры выполнения вычислений в программе

Настало время применить полученные знания на практике.

Задача №4: необходимо написать программу прямолинейного движения для проезда роботом расстояния в 1 метр.

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**, а - из домашнего набора Lego mindstorms EV3 равен **43,2 мм**. Если переведем расстояние в 1 метр в миллиметры (**1000 мм**) и разделим на расстояние, которое робот проходит за один оборот мотора, то узнаем: сколько оборотов мотора необходимо для проезда всего заданного расстояния.



Рис. 7

Приступим к созданию программы:

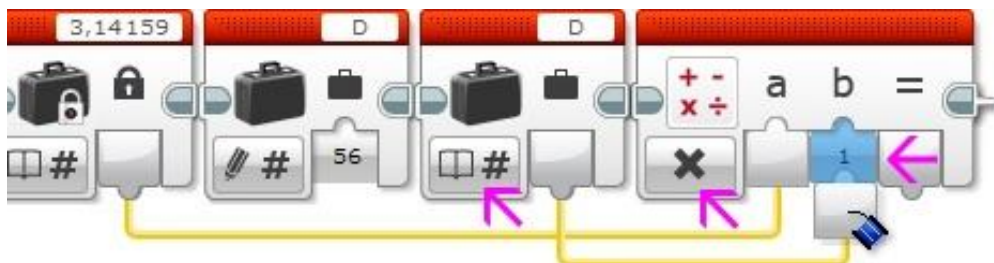
1. Используя программный блок "**Константа**", заведем в программу постоянное число Пи, равное примерно **3,14159**.



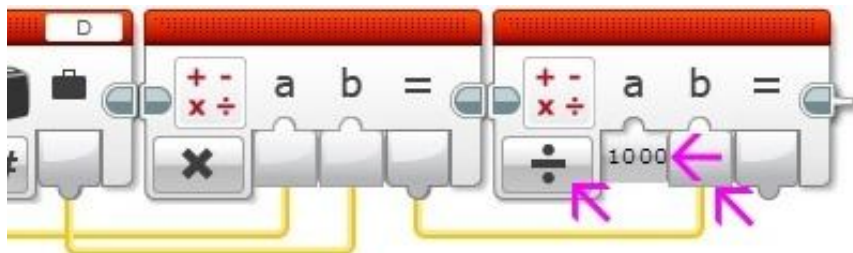
2. Используя программный блок **"Переменная"**, создадим в программе переменную **D** и занесем в нее значение диаметра колеса в зависимости от используемого конструктора (если вы использовали другие колеса, то самостоятельно измерьте диаметр и внесите значение в программный блок).



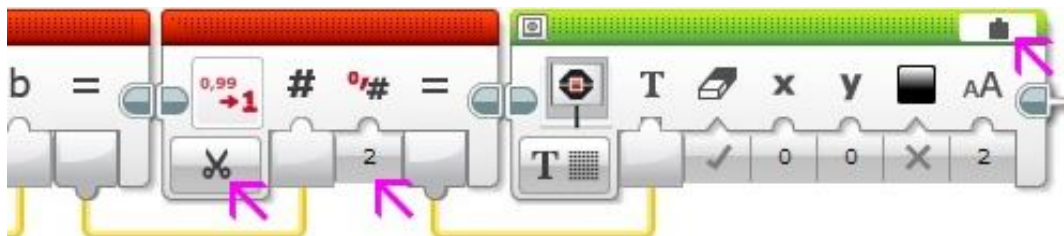
3. Используя программный блок **"Математика"**, умножим значение блока **"Константа"** на значение переменной **D**. Для передачи значения из переменной **D** в программный блок **"Математика"** используем второй программный блок **"Переменная"** в режиме **"Считывание"**! (Для передачи значений между 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)



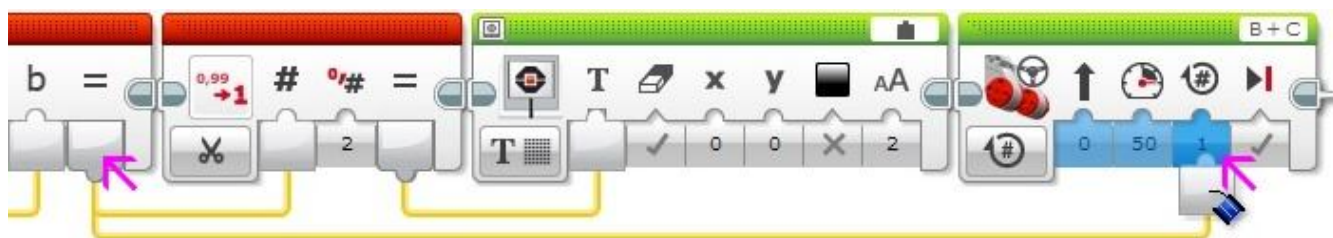
4. Используя программный блок **"Математика"**, разделим значение пути (1000 мм) на значение, полученное в шаге 3.



5. Полученное в шаге 4 значение, округлив до двух знаков после запятой, выведем на экран модуля EV3.



6. 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!



Задача №5: необходимо написать программу, рассчитывающую значение параметра "Градусы" для разворота нашего робота ([Урок №2. Задача №1](#))

Данная задача имеет сходство с предыдущей - нам только требуется найти расстояние, которое должны проехать колеса нашего робота. Для того, чтобы наш робот развернулся на **180 градусов** - необходимо, чтобы правое и левое колеса, проехав определенный путь по окружности, поменялись местами. Как видим из **Рис. 8** - каждое колесо при этом проедет ровно половину окружности с диаметром, равным расстоянию между центрами колес (**красная линия на Рис. 8**). Подходящей линейкой померяем расстояние между центрами колес. Для робота, собранного по инструкции **small-robot-45544**, это расстояние равно **120 мм**. Следовательно, умножив это значение на число **Пи (3,14159)** и разделив на **2**, мы найдем расстояние, которое должно проехать каждое из колес нашего робота. Как найти соответствующее этому расстоянию число оборотов мотора - мы разобрали в **Задаче 4** данного урока. Для того, чтобы перевести полученное число оборотов в градусы - вспомним соотношение: **1 оборот мотора = 360 градусов**. Следовательно, если мы, воспользовавшись программным блоком "Математика", умножим полученное значение оборотов на **360** и подадим результат в параметр "Градусы" программного блока "Независимое управление моторами" ([Урок №2 Рис.7 поз. 2](#)), то решим требуемую задачу.

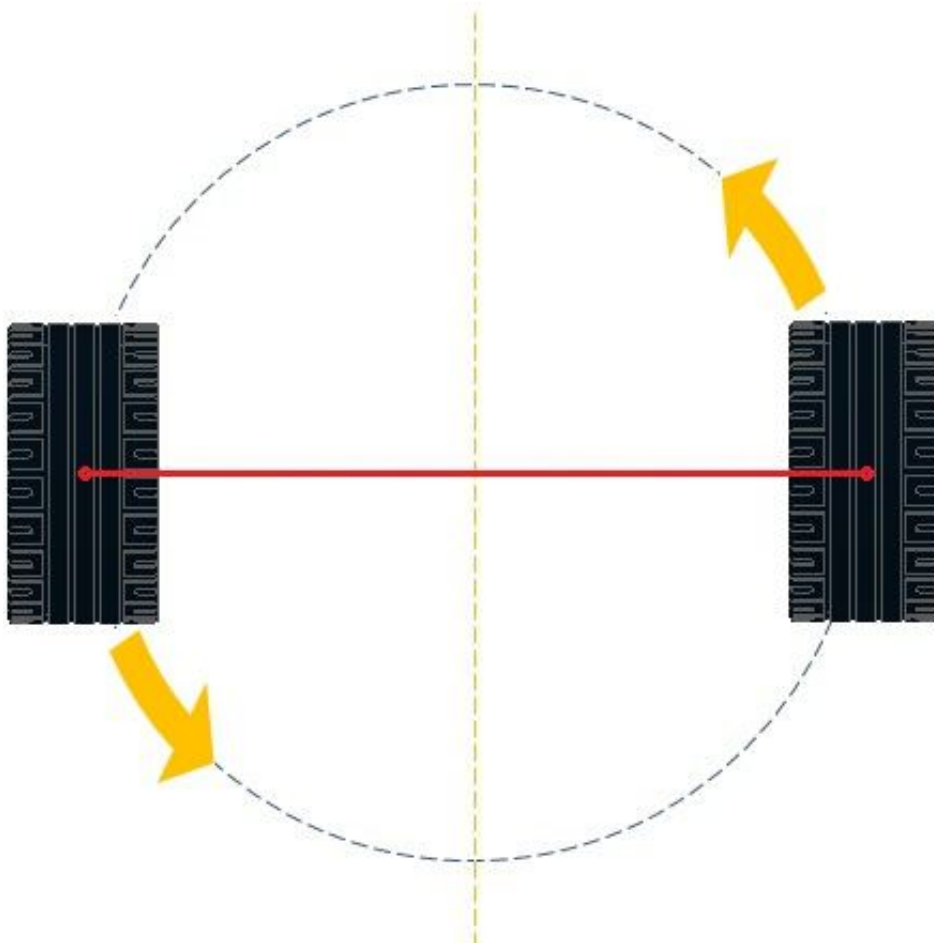


Рис. 8

Попробуйте написать программу для решения задачи №5 самостоятельно, не подглядывая в решение!

Решение Задачи №5 Оптимизация решения Задачи №5