



Изучаем датчик цвета Lego Mindstorms EV3

Lego mindstorms EV3

Изучаем датчик цвета



Датчик цвета может работать в трех различных режимах:

- в режиме "**Цвет**" датчик может определить цвет поднесенного к нему предмета;
- в режиме "**Яркость отраженного света**" датчик направляет световой луч на близкорасположенный предмет и по отраженному пучку определяет яркость предмета;
- в режиме "**Яркость внешнего освещения**" датчик может определить - насколько ярко освещено пространство вокруг.

Режим "Цвет"

- В режиме **"Цвет"** датчик цвета достаточно точно умеет определять семь базовых цветов предметов, находящихся от него на расстоянии примерно в 1 см. Это следующие цвета: **"черный"**=1, **"синий"**=2, **"зеленый"**=3, **"желтый"**=4, **"красный"**=5, **"белый"**=6 и **"коричневый"**=7.
- Если предмет удален от датчика или некорректно определяется цвет предмета - датчик информирует об этом состоянием **"Без цвета"**=0.

Оранжевая палитра, программный блок "Переключатель"

- **Задача №1:** необходимо написать программу, называющую цвета предметов, подносимых к датчику цвета.
- В решении **Задачи №1** нам поможет программный блок **"Переключатель"** Оранжевой палитры. Этот блок в зависимости от настроек выбирает для выполнения программные блоки, расположенные в одном из своих контейнеров. Рассмотрим настройку этого блока в режиме работы с датчиком цвета.
- Создадим новую программу **"lesson-C"**, установим в программе блок **"Переключатель"**, выберем режим **"Датчик цвета" - "Измерение" - "Цвет"** (Рис. 1). В отличие от программного блока **"Ожидание"**, программный блок **"Переключатель"** не ждет, пока наступит определенное событие, а проверяет текущее состояние и выполняет программные блоки, находящиеся в контейнере, сопоставленном текущему состоянию.

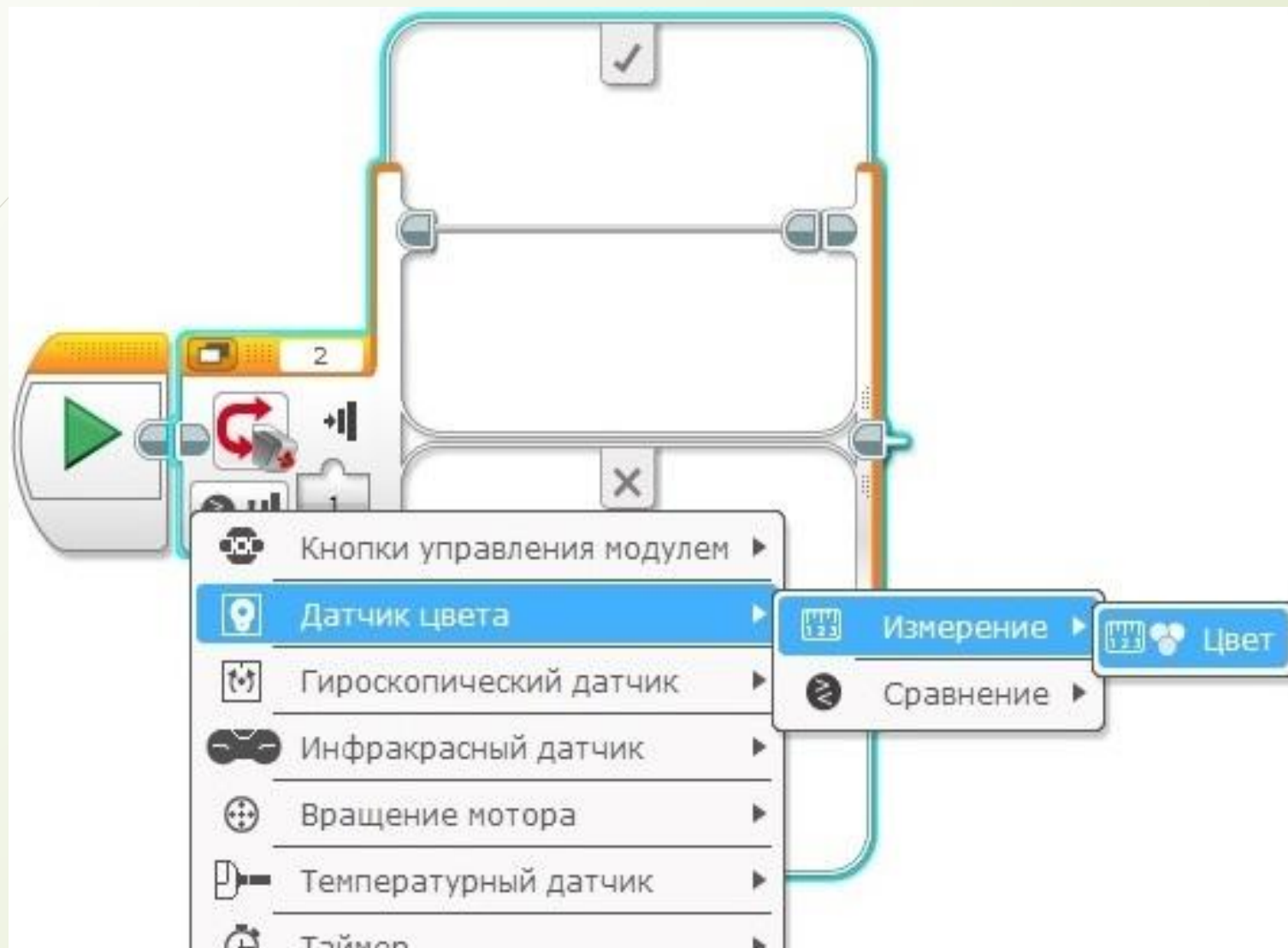


Рис. 1

Рассмотрим подробнее настройки программного блока **"Переключатель"**:

- выбранный режим устанавливает изображение датчика цвета в блоке (**Рис. 2 поз. 1**),
- порт, к которому подключен датчик, отображается в соответствующем поле блока (**Рис. 2 поз. 2**),
- в настройках каждого программного контейнера выбирается значение, в соответствии с которым будут выполняться программные блоки, вложенные в этот контейнер (**Рис. 2 поз. 3**),
- один из контейнеров должен быть объявленным **"Вариантом по умолчанию"** - в случае, если значению, полученному от датчика, не соответствует ни один контейнер, то выполняется контейнер, объявленный **"Вариантом по умолчанию"** (**Рис.2 поз. 4**),
- Кнопка "+" добавляет программный контейнер в блоке **"Переключатель"** (**Рис. 2 поз. 5**),
- Программный блок **"Переключатель"** может автоматически растягиваться, чтобы вместить все блоки, помещаемые внутрь. С помощью меток, помеченных красными стрелками, можно самому изменять размеры блока (**Рис.2**).

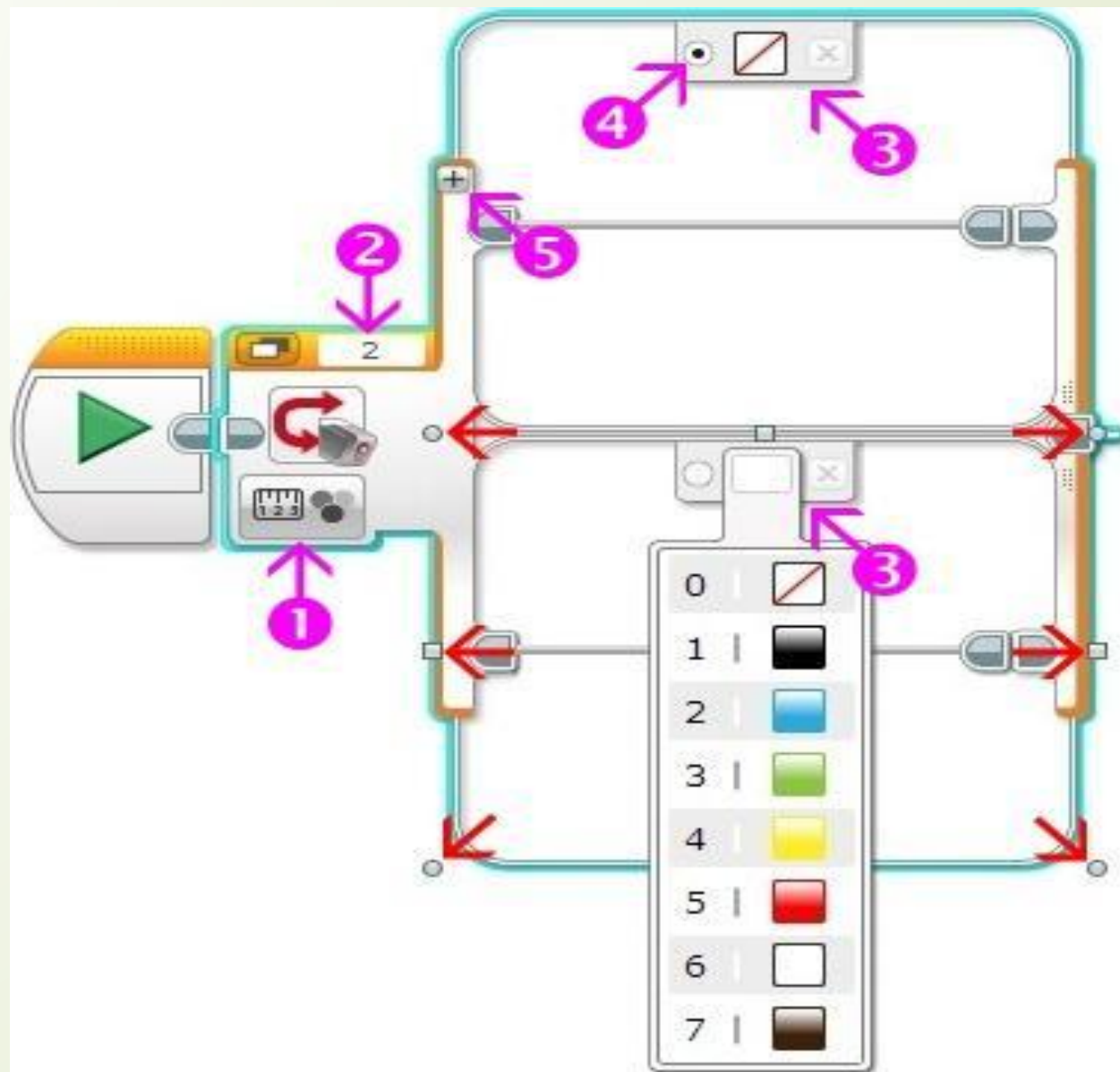


Рис. 2

Продолжим формирование программного блока **«Переключатель»**

- создадим необходимое количество контейнеров, соответствующее количеству цветов для распознавания + вариант **"Без цвета"**,
- в настройках контейнеров установим распознаваемые цвета,
- вариантом по умолчанию выберем вариант **"Без цвета"**,
- в каждый контейнер кроме варианта **"Без цвета"** (этот контейнер останется пустым) поместим программный блок **"Звук"** зеленой палитры.
- каждому цвету сопоставим соответствующий звуковой файл.

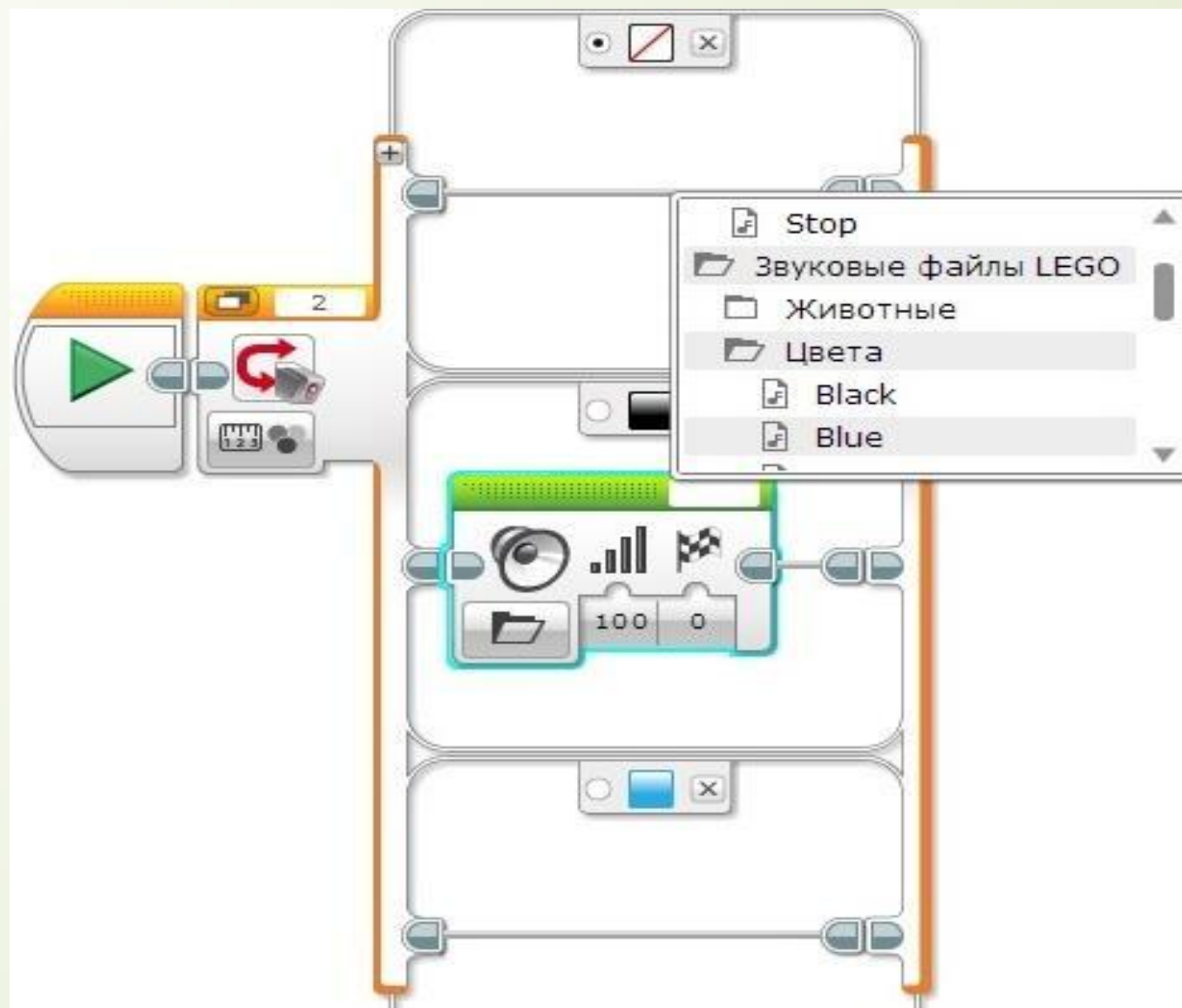


Рис. 3

- Наш программный блок **"Переключатель"** значительно увеличился в размерах. Специальная кнопка (Рис. 4 поз. 1) позволяет переключить режим отображения блока на экране на **"Вид с вкладками"**. Изменим размеры блока для комфортного визуального отображения.

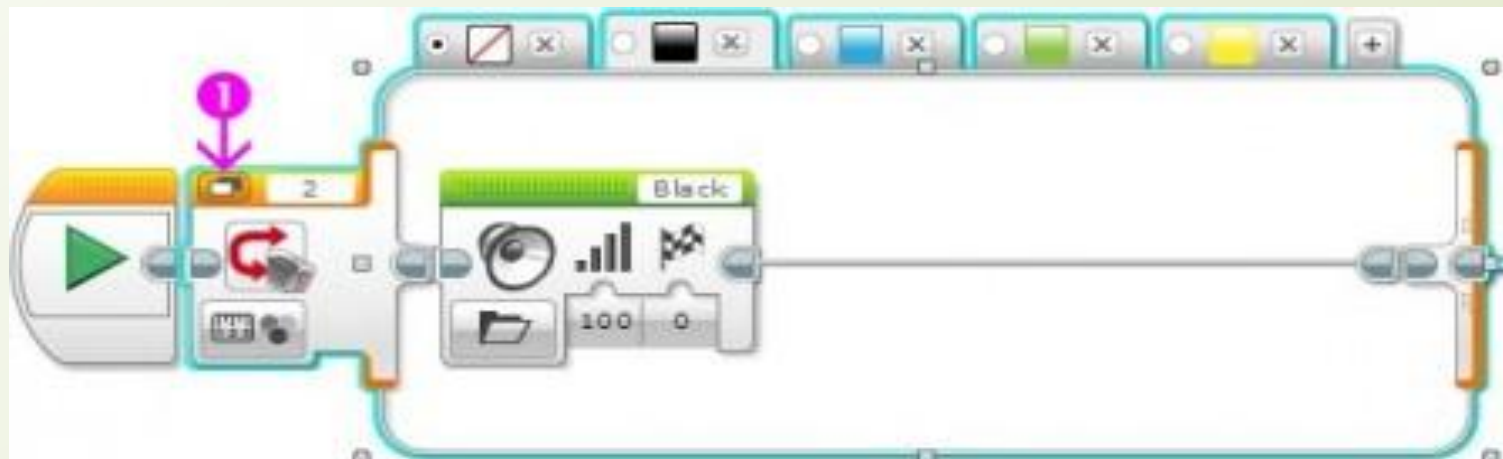


Рис. 4

- Осталось вставить наш настроенный программный блок "Переключатель" внутрь программного блока "Цикл" Оранжевой палитры. Программа готова! Загрузим её в робота и протестируем работу! (Рис. 5)

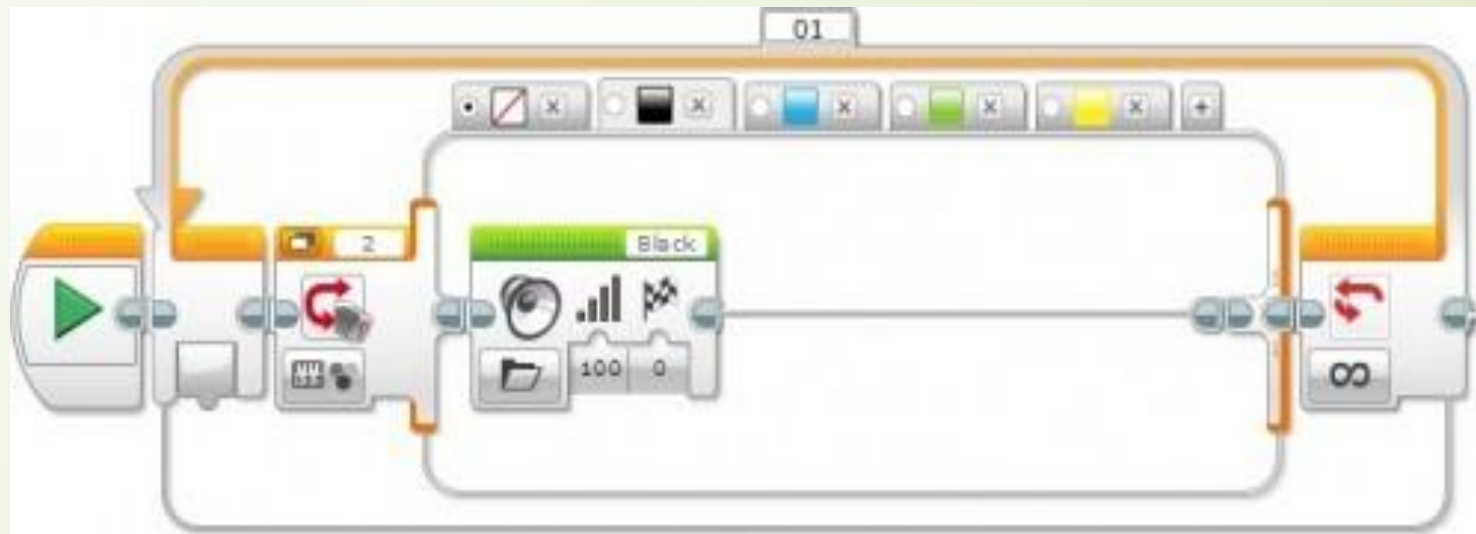


Рис. 5

Оранжевая палитра, программный блок "Прерывание цикла"

Добавим в нашу программу движение. Сделаем следующее поле для выполнения задания:

- Возьмем белый лист бумаги формата А4 или А3;
- Нанесем на него последовательно, на равном расстоянии несколько цветных полос. Полосы можно наклеить из цветной бумаги, цветной изоленты или нарисовать и закрасить;
- Последнюю полосу сделаем черного цвета (**Рис. 6**).

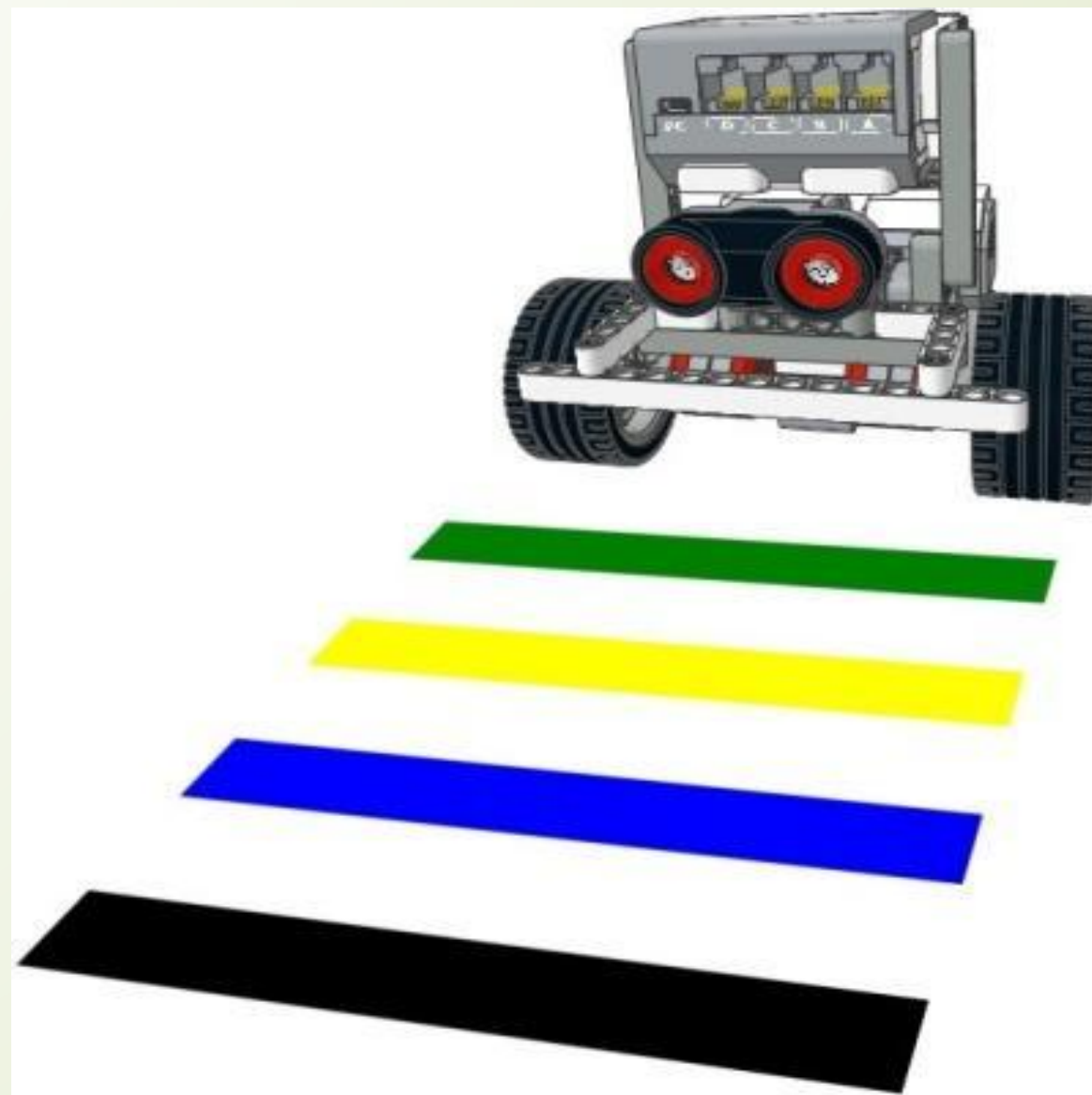


Рис. 6

- **Задача №2:** необходимо написать программу прямолинейного движения робота, называющего цвета полос, над которыми он проезжает. При достижении черной полосы робот проговаривает **"Stop"** и останавливается.
- За основу решения данной задачи возьмем программу, решающую **Задачу №1**. При решении **Задачи №2** нам потребуется прервать выполнение цикла. Этой цели служит программный блок **"Прерывание цикла"** Оранжевой палитры. С помощью данного блока можно организовать выход из цикла, заданного параметром **"Имя прерывания"** (Рис. 7 поз. 1).

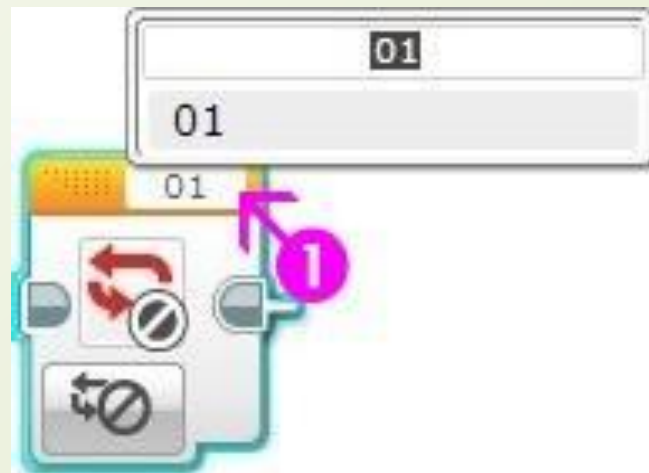


Рис. 7

Решение Задачи №2

- Внутри цикла перед программным блоком **"Переключатель"** добавим программный блок **"Рулевое управление"**, тем самым заставим нашего робота двигаться (**Рис. 8 поз. 1**). Во время движения робот будет проверять текущее состояние датчика цвета и произносить название цвета. Если полосы будут широкими, а робот будет двигаться медленно, то, возможно, он станет произносить название цвета более одного раза, так как проверка цвета будет происходить неоднократно. Если такое положение дел вас не устроит - увеличьте скорость робота, чтобы он быстрее проезжал цветные полосы.
- В соответствии с условием задачи нам надо изменить поведение контейнера программного блока **"Переключатель"** для черного цвета.
- В программном блоке **"Звук"** изменим звуковой файл **"Black"** на **"Stop"** (**Рис. 8 поз. 2**).
- Добавим в контейнер программный блок, выключающий моторы (**Рис. 8 поз. 3**).
- Нам требуется прервать выполнение программного блока **"Цикл"**, чтобы завершить выполнение программы. Для этого поместим в контейнер программный блок **"Превышение цикла"** Оранжевой палитры (**Рис. 8 поз. 4**). У данного программного блока существует только одна настройка - название прерываемого цикла. В сложной программе со множеством циклов важно правильно устанавливать эту настройку, чтобы остановить выполнение нужного цикла (**Рис. 8 поз. 5**). В нашей программе за программным блоком **"Цикл"** отсутствуют другие программные блоки, поэтому программа завершится.

Датчик цвета – режим "Яркость отраженного света"

- Итак, мы приступаем к изучению следующего режима работы датчика цвета, который называется "**Яркость отраженного света**". В этом режиме датчик цвета направляет поток красного света на близкорасположенный предмет или поверхность и измеряет количество отраженного света. Более темные предметы будут поглощать световой поток, поэтому датчик будет показывать меньшее значение, по сравнению с более светлыми поверхностями. Диапазон значений датчика измеряется от **0** (очень темный) до **100** (очень яркий). Данный режим работы датчика цвета используется во множестве задач по робототехнике, например, для организации движения робота по заданному маршруту вдоль черной линии, нанесенной на белое покрытие. При использовании этого режима рекомендуется располагать датчик таким образом, чтобы расстояние от него до исследуемой поверхности составляло примерно **1 см (Рис. 9)**.

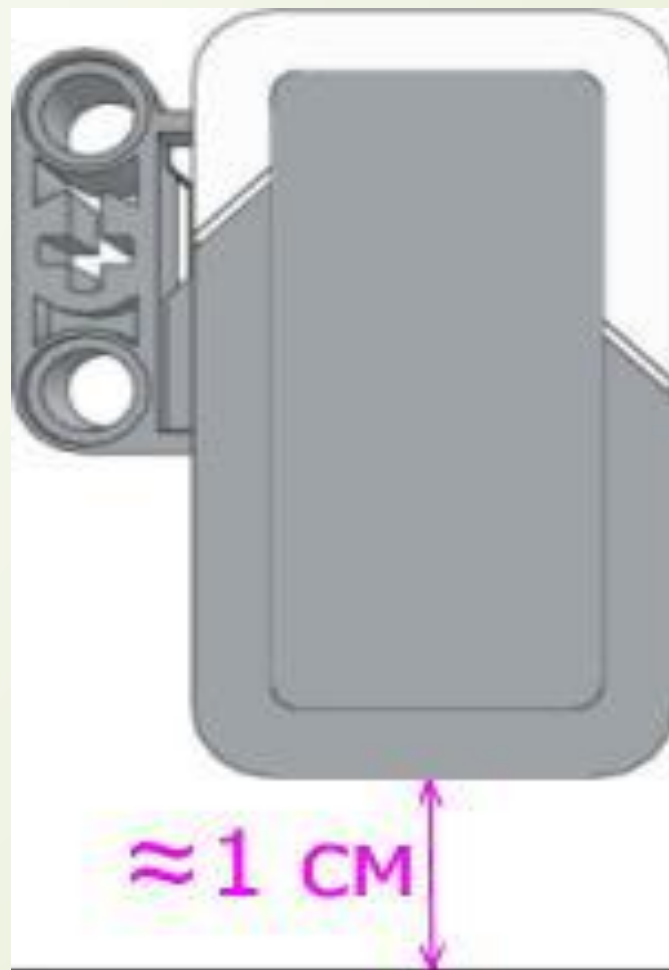


Рис. 9

- Датчик цвета уже подключен к порту "2" модуля EV3.
- Установим робота, таким образом, чтобы датчик цвета расположился над белой поверхностью. "**Страницу аппаратных средств**" среды программирования переключим в режим "**Просмотр портов**" (Рис. 10 поз. 1). В этом режиме мы можем наблюдать все выполненные нами подключения. На Рис. 10 отображено подключение к портам "B" и "C" двух больших моторов, а к порту "2" - датчика цвета.

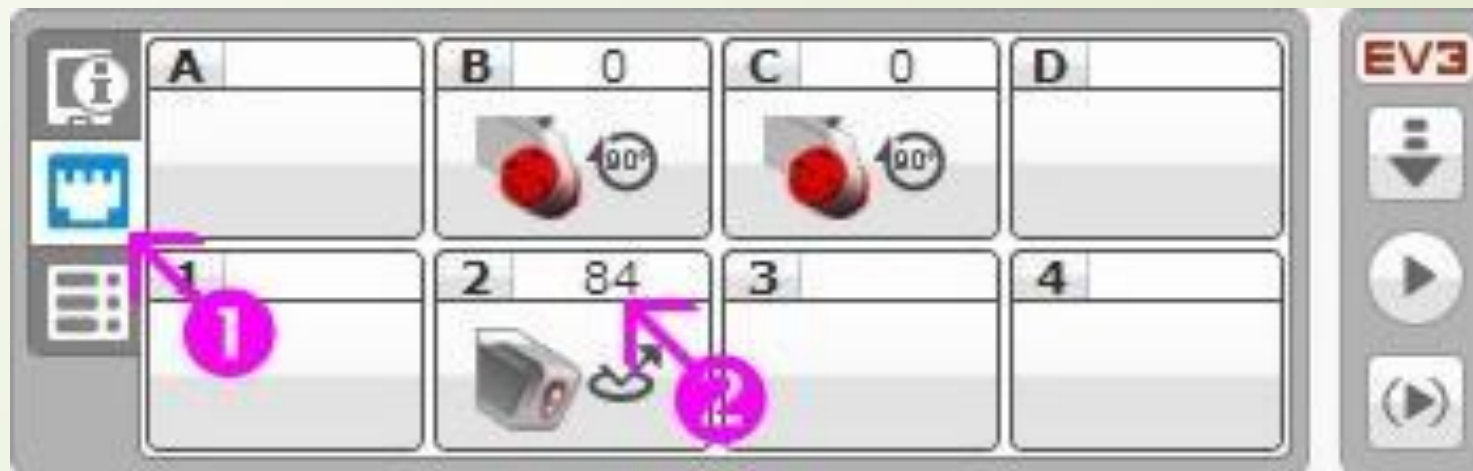


Рис. 10

- Для выбора варианта отображения показаний датчиков необходимо нажать на изображение датчика и выбрать нужный режим (**Рис. 11**)

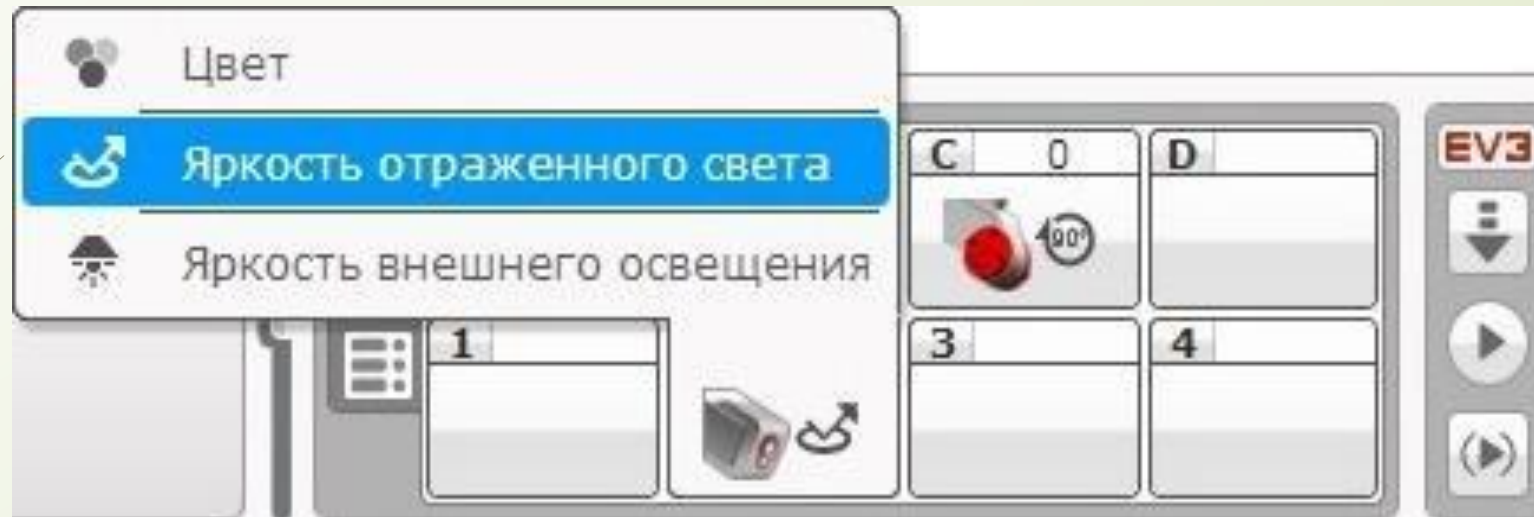


Рис. 11

- На **Рис. 10 поз. 2** мы видим, что значение показания датчика цвета над белой поверхностью равно **84**. В вашем случае может получиться другое значение, ведь оно зависит от материала поверхности и освещения внутри помещения: часть освещения, отражаясь от поверхности, попадает на датчик и влияет на его показания. Установив робота таким образом, чтобы датчик цвета расположился над черной полосой, зафиксируем его показания (**Рис. 12**).

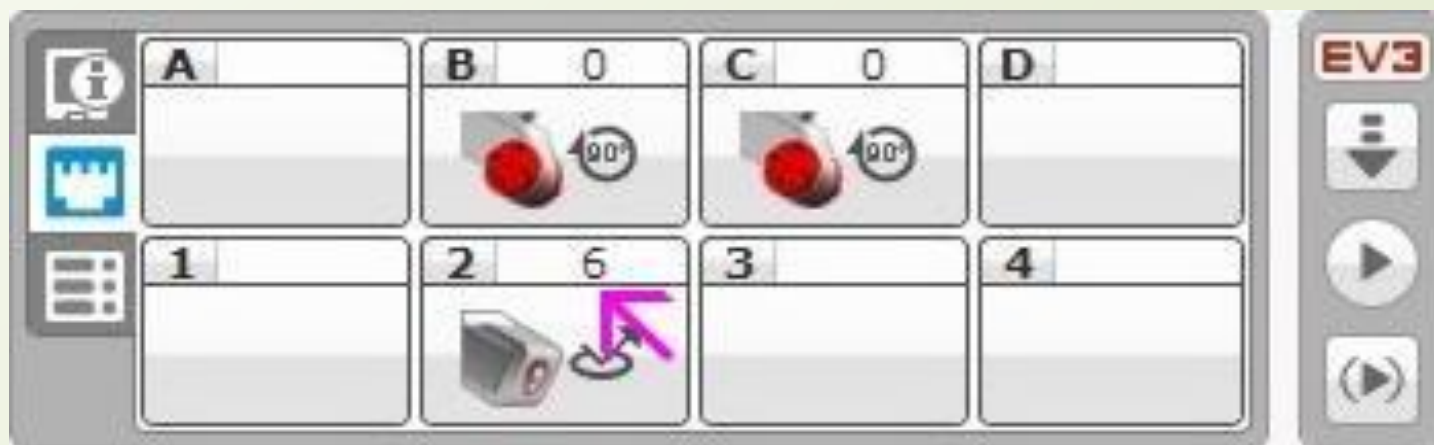


Рис. 12

- **Задача №3:** необходимо написать программу движения робота, останавливающегося при достижении черной линии.

Решение:

- Проведенный эксперимент показал нам, что при пересечении черной линии, значение датчика цвета в режиме **"Яркость отраженного света"** равняется **6**. Значит, для выполнения **Задачи №3** наш робот должен двигаться прямолинейно, пока искомое значение датчика цвета не станет меньше **7**. Воспользуемся уже знакомым нам программным блоком **"Ожидание"** Оранжевой палитры. Выберем требуемый условию задачи режим работы программного блока **"Ожидание"** (Рис. 13).

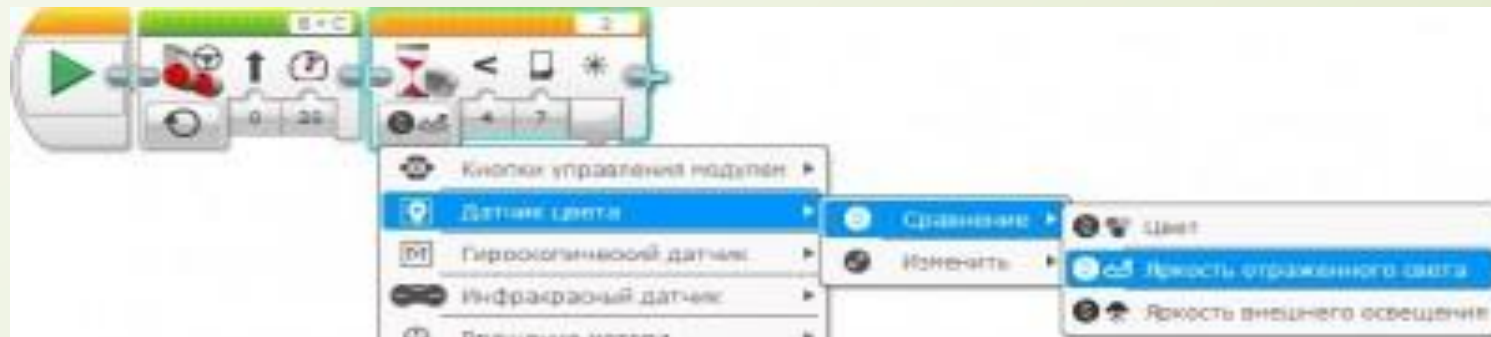


Рис. 13

- Необходимо также настроить параметры программного блока "Ожидание". Параметр "Тип сравнения" (Рис. 14 поз. 1) может принимать следующие значения: "Равно"=0, "Не равно"=1, "Больше"=2, "Больше или равно"=3, "Меньше"=4, "Меньше или равно"=5. В нашем случае установим "Тип сравнения" в значение "Меньше". Параметр "Пороговое значение" установим равным 7 (Рис.14 поз. 2).

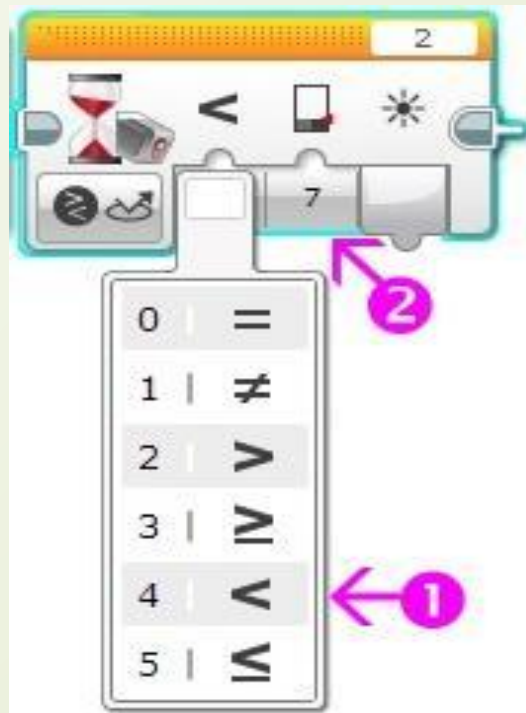


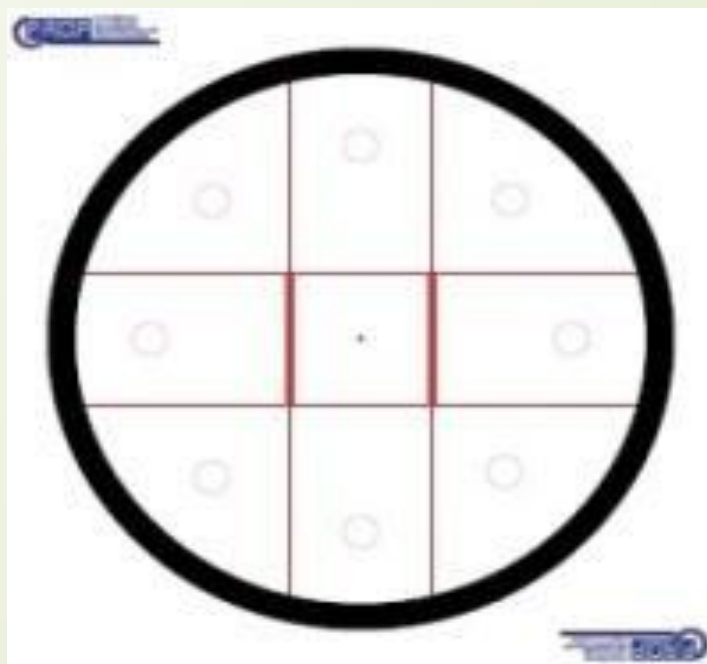
Рис. 14


- Как только установится значение датчика цвета меньше **7**, что случится, когда датчик цвета окажется расположенным над черной линией, нам необходимо будет выключить моторы, остановив робота. Задача решена (**Рис. 15**).



Рис. 15

- Для продолжения занятий нам понадобится изготовить новое поле, представляющее собой черную окружность диаметром примерно 1 метр, нанесенную на белое поле. Толщина линии окружности равняется 2 - 2,5 см. Для основы поля можно взять один лист бумаги размером А0 (841x1189 мм), склеить вместе два листа бумаги размером А1 (594x841 мм). На этом поле разметить линию окружности и закрасить её черной тушью. Можете также скачать макет поля, выполненный в формате Adobe Illustrator, а затем заказать его печать на баннерной ткани в типографии. Размер макета равен 1250x1250 мм.



- 
- **Задача №4:** необходимо написать программу для робота, передвигающегося внутри круга, окантованного черной окружностью по следующему правилу:
 - робот движется вперед прямолинейно;
 - достигнув черной линии, робот останавливается;
 - робот отъезжает назад на два оборота моторов;
 - робот поворачивает вправо на 90 градусов;
 - движение робота повторяется.

Решение задачи №4

- Начать прямолинейное движение вперед (**Рис. 16 поз. 1**);
- Ожидать пересечения черной линии датчиком цвета (**Рис. 16 поз. 2**);
- Двигаться назад на 2 оборота (**Рис. 16 поз. 3**);
- Повернуть направо на 90 градусов (**Рис. 16 поз. 4**); значение угла поворота рассчитано для робота, собранного по инструкции small-robot-45544 (**Рис. 16 поз. 5**);
- Повторять команды 1 - 4 в бесконечном цикле (**Рис. 16 поз. 6**).

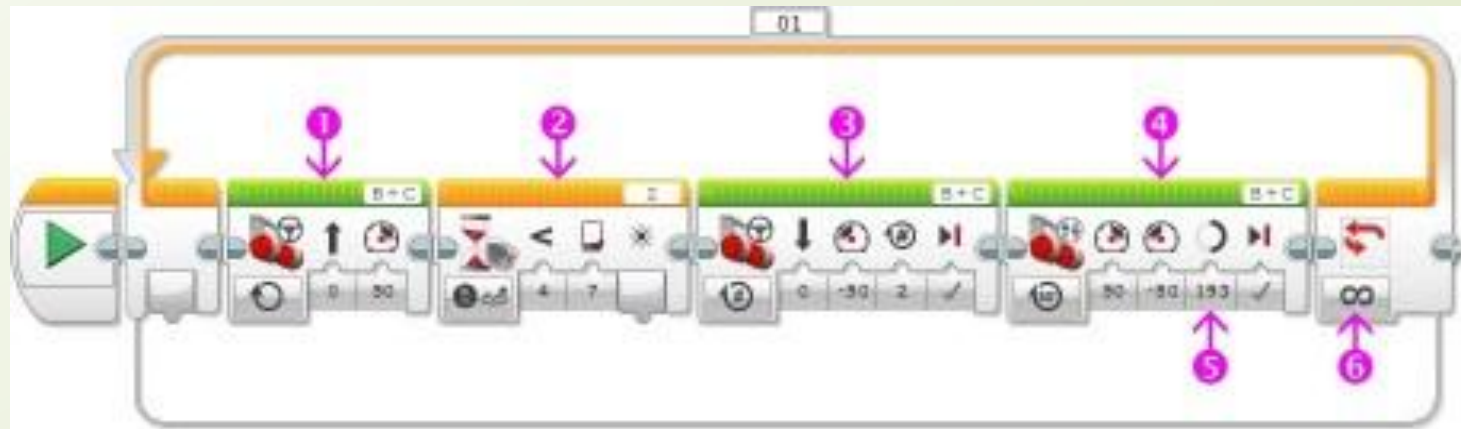


Рис. 16

Датчик цвета – режим "Яркость внешнего освещения"

- Режим работы датчика цвета **"Яркость внешнего освещения"** очень похож на режим **"Яркость отраженного света"**, только в этом случае датчик не излучает освещение, а измеряет естественное световое освещение окружающей среды. Визуально данный режим работы датчика можно определить по слабо светящемуся синему светодиоду. Показания датчика изменяются от **0** (отсутствие света) до **100** (самый яркий свет). При решении практических задач, требующих измерения внешнего освещения, рекомендуется располагать датчик, так, чтобы датчик оставался максимально открытым и не загороживался другими деталями и конструкциями.

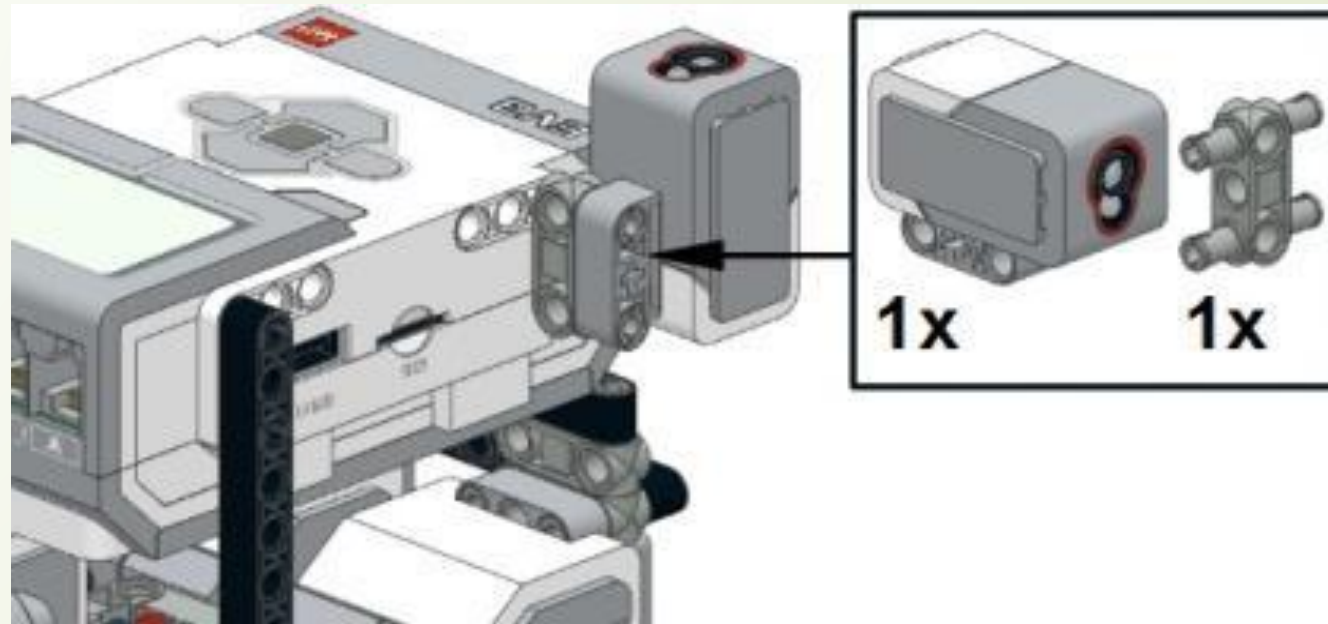


Рис. 17

- **Задача №5:** необходимо написать программу, изменяющую скорость движения нашего робота в зависимости от интенсивности внешнего освещения.
- Чтобы решить эту задачу, нам надо узнать, как получать текущее значение датчика. А поможет нам в этом Желтая палитра программных блоков, которая называется "**Датчики**".

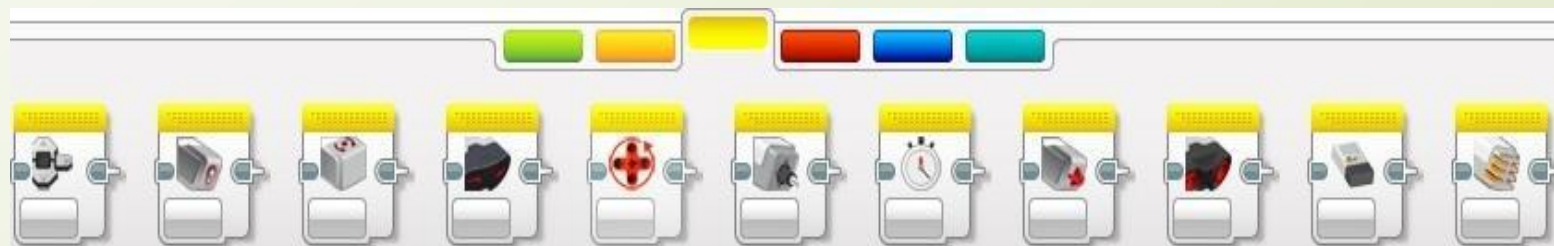
Желтая палитра - "Датчики"

- Желтая палитра среды программирования Lego mindstorms EV3 содержит программные блоки, позволяющие получать текущие показания датчиков для дальнейшей обработки в программе. В отличие, например, от программного блока **"Ожидание"** Оранжевой палитры, программные блоки Желтой палитры сразу же передают управление к следующим за ними программным блокам.
- Количество программных блоков Желтой палитры отличается в домашней и образовательной версии среды программирования. В домашней версии среды программирования отсутствуют программные блоки для датчиков, не входящих в домашнюю версию конструктора. Но, при необходимости, их можно самостоятельно ПОДКЛЮЧИТЬ.



Желтая палитра. Домашняя версия среды программирования.

- Образовательная версия среды программирования содержит программные блоки для всех датчиков, которые можно использовать с конструктором Lego mindstorms EV3.

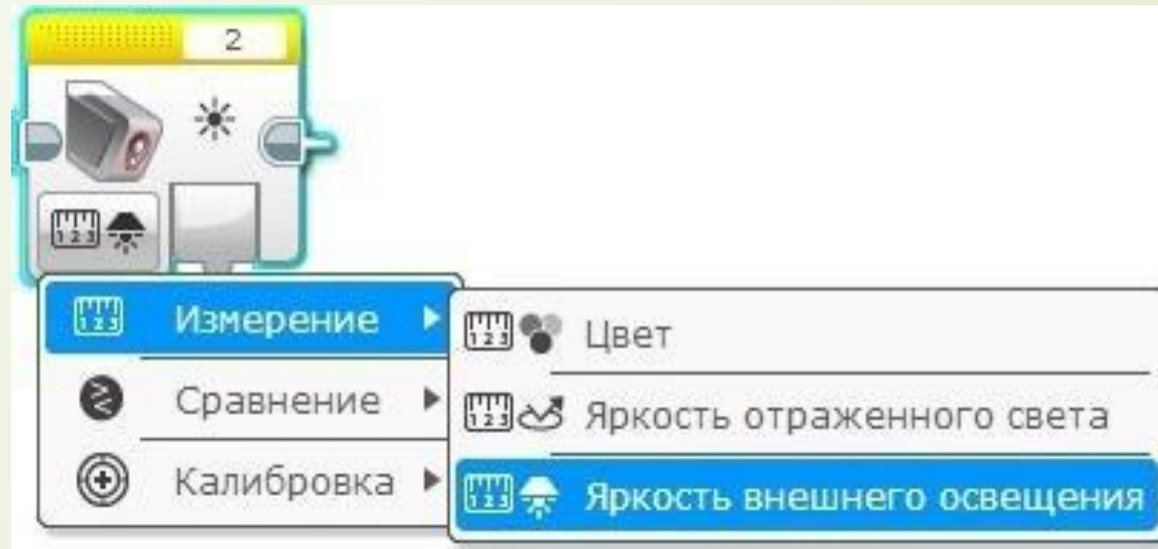


Желтая палитра. Образовательная версия среды программирования.

- Вернемся же к решению **Задачи №5** и посмотрим, как можно получать и обрабатывать показания датчика цвета. Как мы уже знаем: диапазон значений датчика цвета в режиме **"Яркость внешнего освещения"** находится в пределах от **0** до **100**. Такой же диапазон у параметра, регулирующего мощность моторов. Попробуем показанием датчика цвета регулировать мощность моторов в программном блоке **"Рулевое управление"**.

Решение:

- Считать текущее показание датчика цвета в режиме **"Яркость внешнего освещения"**;



- Подать полученное с датчика цвета значение на вход параметра "**Мощность**" программного блока "**Рулевое управление**";
- Повторять команды **1 - 2** в бесконечном цикле.

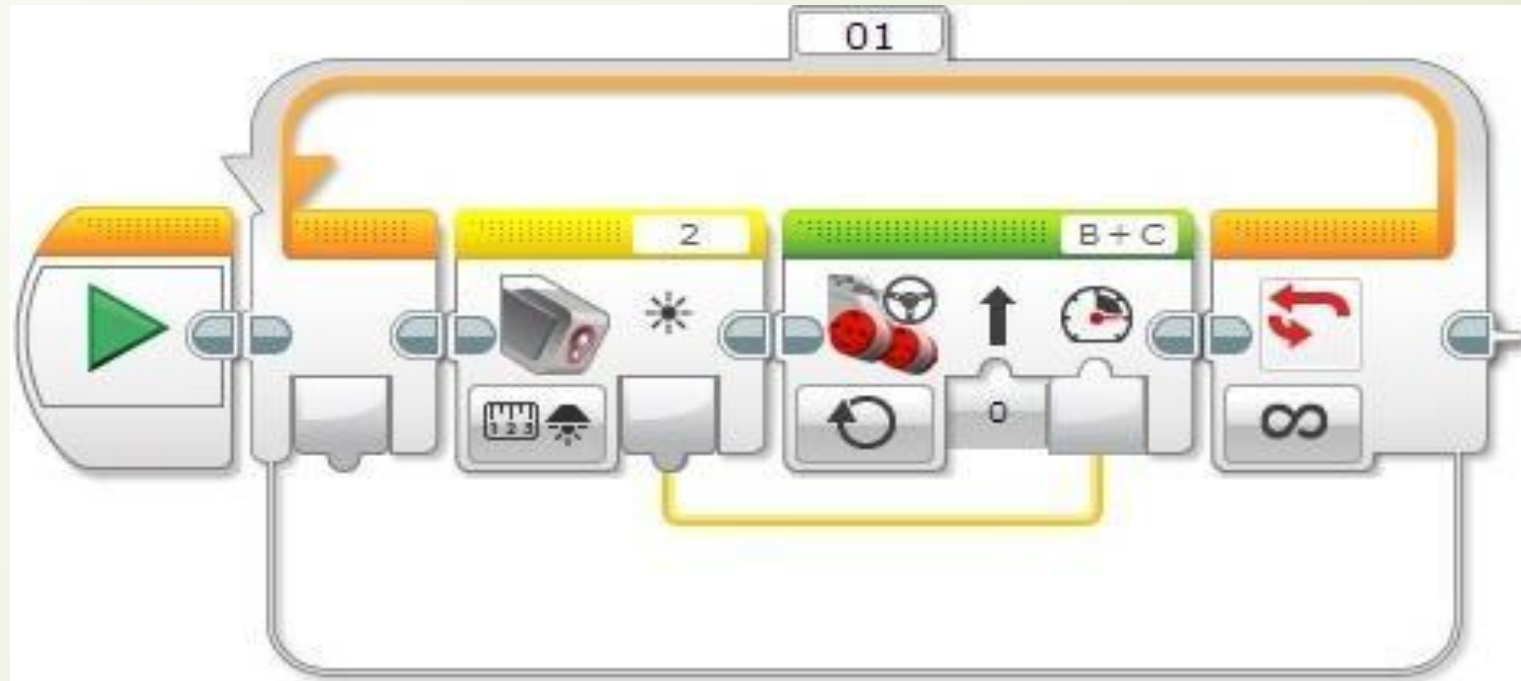


Рис. 18

Задача - Bonus

- Загрузите в робота и запустите на выполнение задачу, изображенную на рисунке ниже. Повторите эксперименты со светодиодным фонариком.

