**Тема Датчик цвета.**

**Введение:**

На этом уроке мы продолжаем знакомство с датчиками набора Lego mindstorms EV3. На очереди - датчик цвета, очень важный и полезный датчик! В большинстве конструкций он является, тем, чем у человека являются глаза. Поэтому изучению датчика цвета мы посвятим два последовательных урока, но в дальнейшем курсе еще вернемся к его изучению и использованию.

**5.1. Изучаем второй датчик - 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.



**Рис. 1**

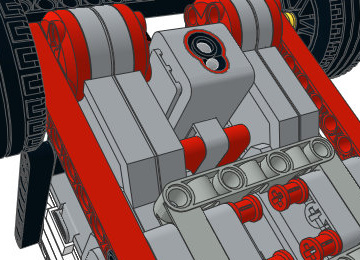
**5.2. Датчик цвета. Режим "Цвет"**

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.

Давайте, не затягивая, перейдем к практическому занятию!

**Задача №9:** необходимо написать программу, называющую цвета предметов, подносимых к датчику цвета.

Если вы собирали своего тренировочного робота по инструкции этого курса, то у вас датчик цвета уже размещен внутри робота и направлен вниз. Потребуется приложить некоторые усилия, может быть даже слегка разобрать-собрать нашу конструкцию, чтобы подключить кабелем датчик цвета, например к порту **"2"** модуля EV3. Для отладки программы нам также понадобится несколько цветных предметов: это могут быть кирпичики конструктора Lego, полоски цветной бумаги или цветные кубики. Для лучшего результата следует взять цвета, максимально приближенные к основным, но датчик довольно неплохо справляется с распознаванием подходящих оттенков. Чтобы не снимать датчик цвета и не крепить его в другом месте, во время выполнения программы можно держать робота перевернутым вверх колесами.

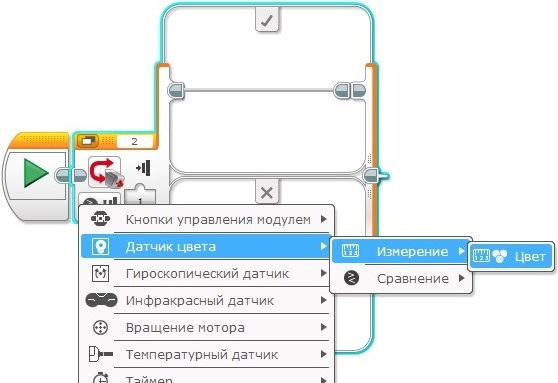


**Рис. 2**

**5.3. Оранжевая палитра, программный блок "Переключатель"**

В решении **Задачи №9** 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.

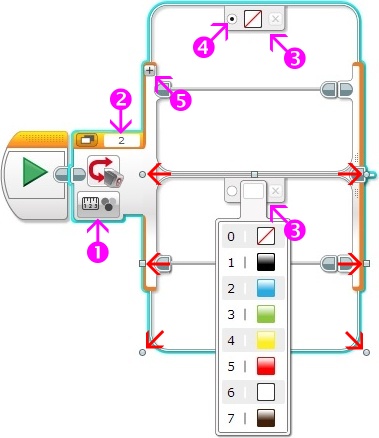
Создадим новую программу **"lesson-5-9"**, установим в программе блок **"Переключатель"**, выберем режим **"Датчик цвета"** - **"Измерение"** - **"Цвет" (Рис. 3)**. В отличие от программного блока **"Ожидание"**, программный блок **"Переключатель"** не ждет, пока наступит определенное событие, а проверяет текущее состояние и выполняет программные блоки, находящиеся в контейнере, сопоставленном текущему состоянию.



**Рис. 3**

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

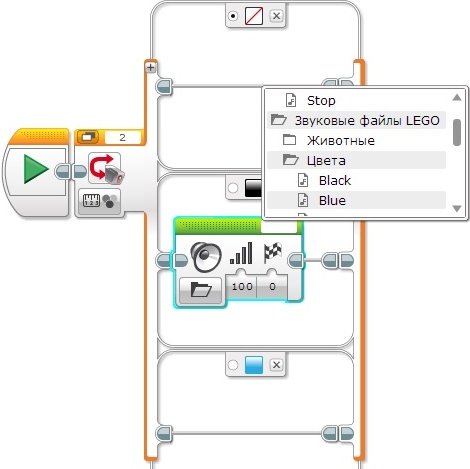
* выбранный режим устанавливает изображение датчика цвета в блоке **(Рис. 4 поз. 1)**,
* порт, к которому подключен датчик, отображается в соответствующем поле блока **(Рис. 4 поз. 2)**,
* в настройках каждого программного контейнера выбирается значение, в соответствии с которым будут выполняться программные блоки, вложенные в этот контейнер **(Рис. 4 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.4 поз. 4)**,
* Кнопка **"+"** добавляет программный контейнер в блоке **"Переключатель" (Рис. 4 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.4)**.



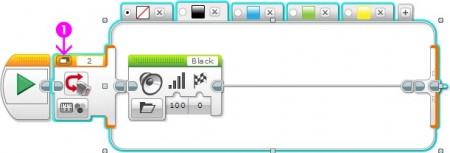
**Рис. 4**

 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.

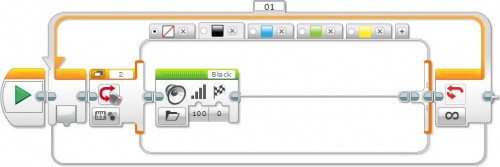
**Рис. 5**

Наш программный блок **"Переключатель"** значительно увеличился в размерах. Специальная кнопка **(Рис. 6 поз. 1)** позволяет переключить режим отображения блока на экране на **"Вид с вкладками"**. Изменим размеры блока для комфортного визуального отображения.

[](https://robot-help.ru/images/lego-mindstorms-ev3/lessons/lesson-5/006.jpg)

**Рис. 6**

Осталось вставить наш настроенный программный блок **"Переключатель"** внутрь программного блока **"Цикл"** Оранжевой палитры. Программа готова! Загрузим её в робота и протестируем работу! **(Рис. 7)**

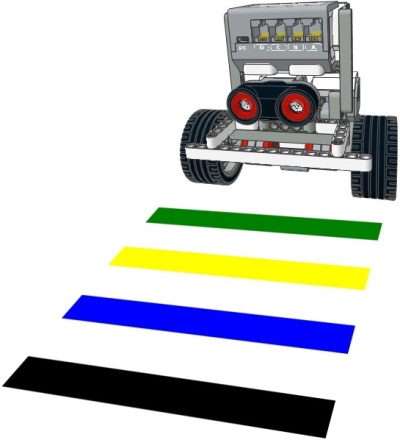
[](https://robot-help.ru/images/lego-mindstorms-ev3/lessons/lesson-5/007.jpg)

**Рис. 7**

**5.4. Оранжевая палитра, программный блок "Прерывание цикла"**

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

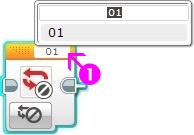
* Возьмем белый лист бумаги формата A4 или A3;
* Нанесем на него последовательно, на равном расстоянии несколько цветных полос. Полосы можно наклеить из цветной бумаги, цветной изоленты или нарисовать и закрасить;
* можете также [загрузить подготовленное изображение](https://robot-help.ru/images/lego-mindstorms-ev3/fields/lesson-5-10.pdf) и распечатать его на цветном принтере;
* Последнюю полосу сделаем черного цвета **(Рис. 8)**.



**Рис. 8**

**Задача №10:** 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 №9**. При решении **Задачи №10** 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. 9 поз. 1)**.



**Рис. 9**

Попробуйте решить Задачу №10 самостоятельно, не подглядывая в решение.