

Датчик цвета

Цифровой датчик цвета EV3 способен:

Различить семь различных цветов, обозначенных цифрами (1-черный, 2-синий, 3-зеленый, 4 – желтый, 5- красный, 6- белый, 7- коричневый);

Определить отсутствие цвета перед датчиком (код - 0);

Определить яркость отраженного от поверхности света по 100-балльной шкале;

Определить яркость внешнего освещения по 100-балльной шкале.

С помощью этого датчика можно решать задачи сортировки. Движения по линии и многие другие. Изображение датчика цвета и программный блок датчика приведены на (рис. 1.)

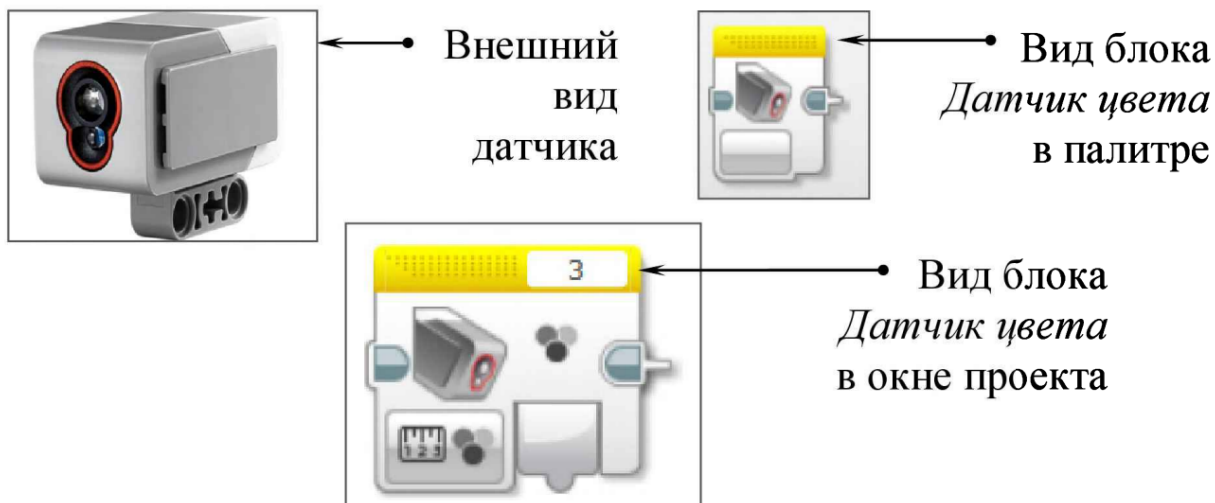
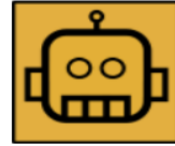


Рис. 1 Датчик цвета и программный блок датчика

Датчик цвета корректно определяет цвет или яркость отраженного света только на расстоянии 0,1 – 1 см!

Датчик может работать в трех режимах: **Измерение** (на выходе мы получаем число, соответствующее определенному цвету); **Сравнение**



(на выходе получаем логические значения Истина или Ложь);
Калибровка.

Вложение 1

Режим Измерение – Цвет

На рис. 2. Представлен выбор режима измерения цвета.

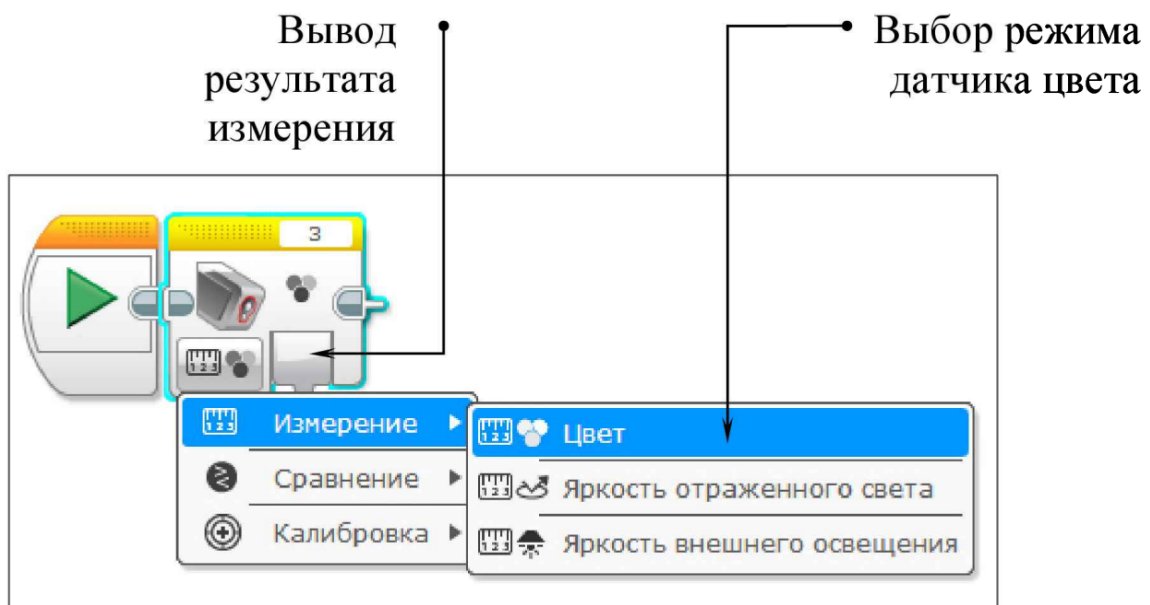


Рис 2. Выбор режима измерения цвета

Составьте программу (рис. 3.). Вывод результат измерения заведите на вход Текст блока Экран (режим Проводной). В результате работы программы на экран блока EV3 будет выводить номер определенного датчиком цвета.

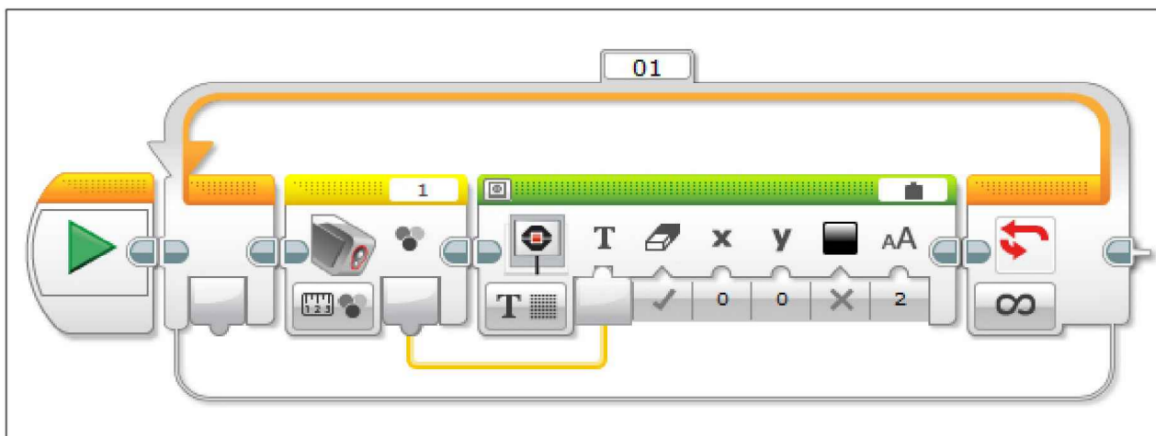
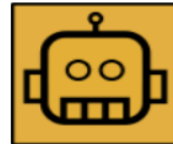


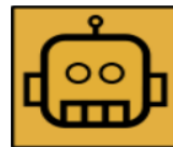
Рис 3 Вывод на экран EV3 номера определённого датчиком цвета

Если поверхность материала блестящая, то определение цвета может быть некорректным из-за бликов от подсветки датчика или яркого солнца. Поэтому если нет возможности изменить материал работа при проезде цветного определяемого участка в первом случае и закрывать поле от попадания прямых солнечных лучей во втором случае.

Ошибка при определении цвета может возникать и на матовых поверхностях при освещении помещения люминесцентными лампами. Белая поверхность может определяться красной, зелёной, синей, причем в одной и той же точке. Это связано с тем, что принцип работы ламп предполагает мерцание светового потока практически во всей области видимого спектра.

Человек может мерцание не замечать, но датчик, работая с частотой 1 кГц, вместо правильного цвета может фиксировать отражённые цветные лучи. Особенно часто это проявляется на белых поверхностях. Единственный способ выхода из этой ситуации – конструировать робота так, чтобы датчик цвета был закрыт сверху и с боков от прямого попадания света люминесцентных ламп.

Режим Измерение –Яркость отраженного света



На (рис. 4.) Представлен выбор режима измерения яркости отражённого света.

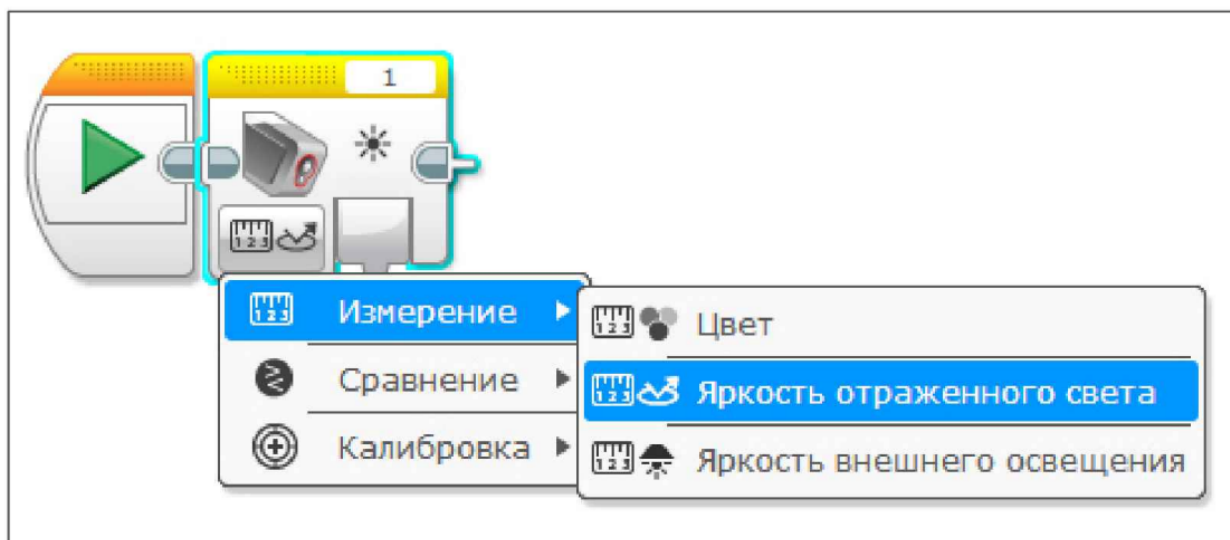


Рис 4 Выбор режима измерения яркости отраженного света

Датчик цвета, работая в режиме измерения яркости отраженного света, освещает светодиодом поверхность и фиксирует по 100-значной шкале количество отраженного от неё света. Если 100% свет, созданного датчиком, отражается, то датчик передаёт значение 100. Если от поверхности ничего не отражается, то значение датчика будет равно 0.(рис. 5).

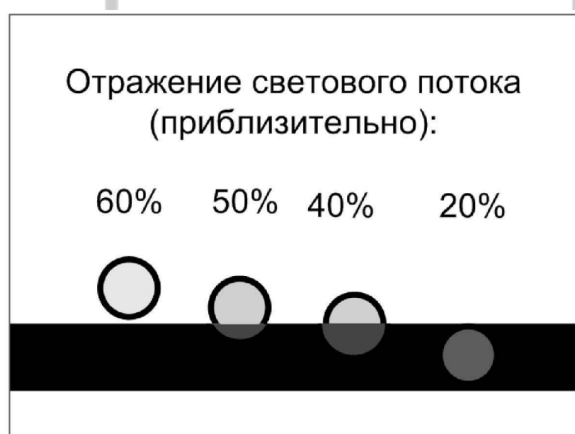
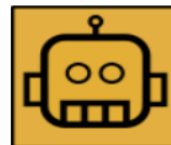


Рис. 5. Отражение светового потока при разном расположении датчика над поверхностью черной динии



Поскольку датчик цвета в указанном режиме фиксирует только процент отраженного света, то получаемые значения удобно сравнивать со шкалой градаций серого цвета. Можно сказать, что датчик различает 100 градаций серого цвета.

Число на выходе зависит не только от оттенка цвета. Но и от внешнего освещения и поверхности, от которой отражается свет.

Режим Измерение – Яркость внешнего освещения

На рис. 6. Представлен выбор режима измерения яркости внешнего освещения.

При работе датчиков включает синюю подсветку и определяет ее интенсивность в зависимости от окружающего освещения, но не реагирует на этот оттенок цвета в том случае, когда свет отражается от какой-либо поверхности.

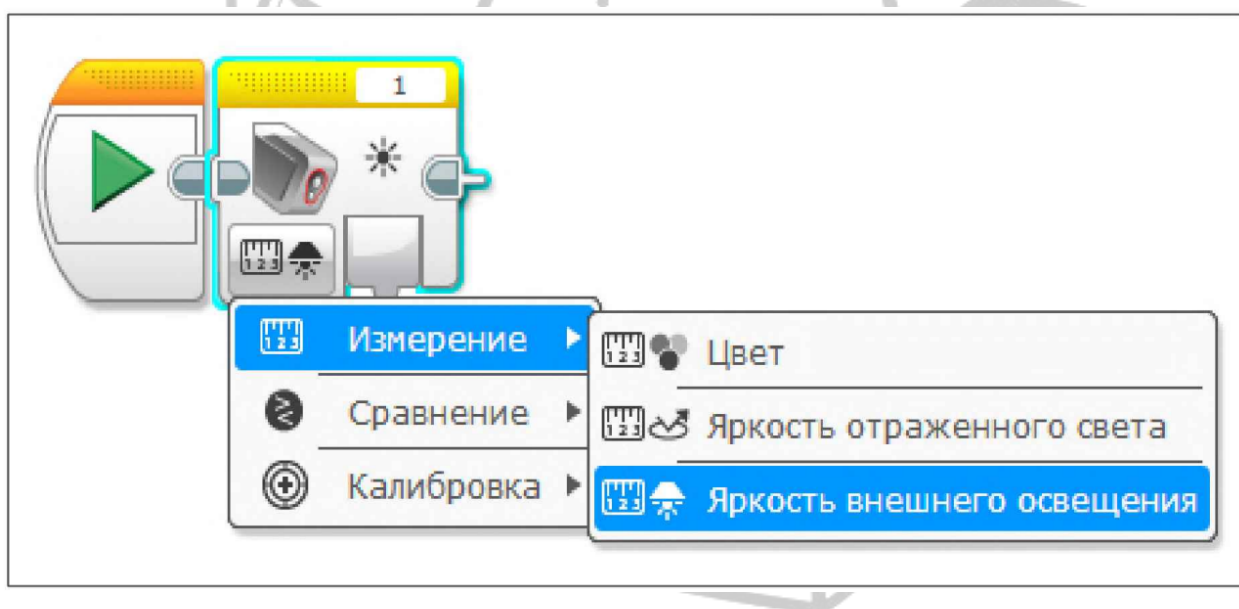
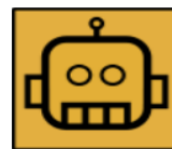


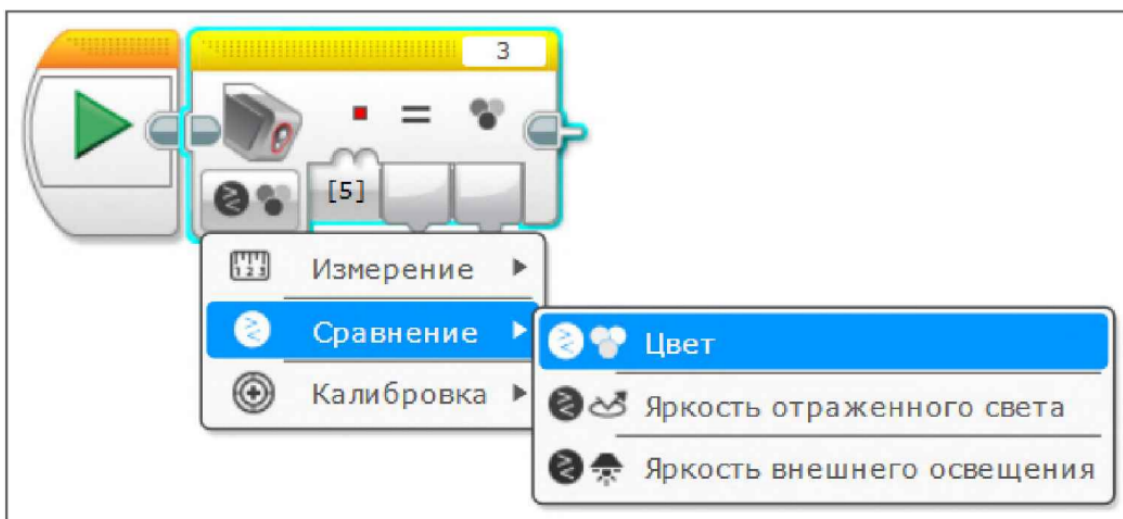
Рис. 6. Выбор режима измерения яркости внешнего освещения

Датчик можно использовать для управления какими-либо действиями в зависимости от освещения.

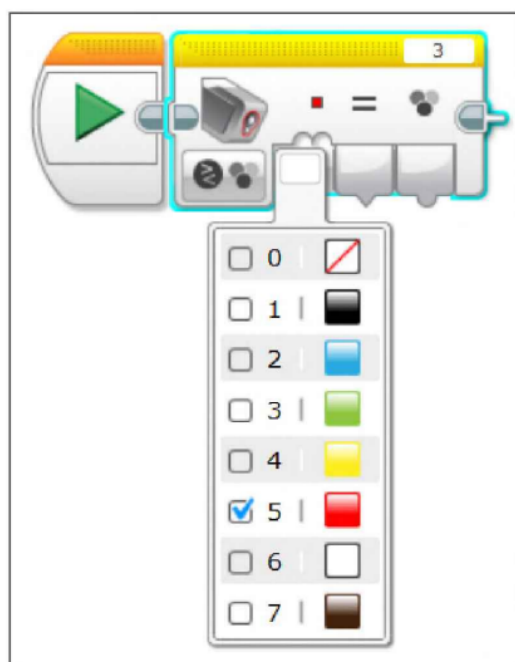
Режим Сравнение – Цвет



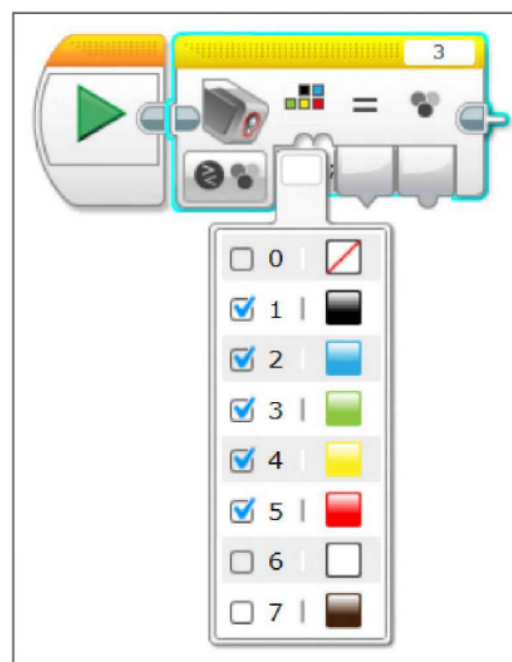
На рис. 7. Представлен выбор режима сравнения цвета: а) способ выбора режима сравнения цвета; б) реализация сравнения с несколькими цветами.



а)



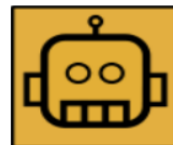
б)



в)

Рис 7 Режим сравнения цвета

В качестве результат измерения мы получим логическое значение Истина или Ложь в зависимости от того, соответствует ли



определённый цвет указанному нами одному цвету (вариант б) или одному из нескольких цветов (вариант в).

На рис. 8. приведён пример программы, в соответствии с которой мотор, подключенный в порт А, будет вращается до тех пор, пока датчик цвета не обнаружит жёлтый или зеленый цвет.

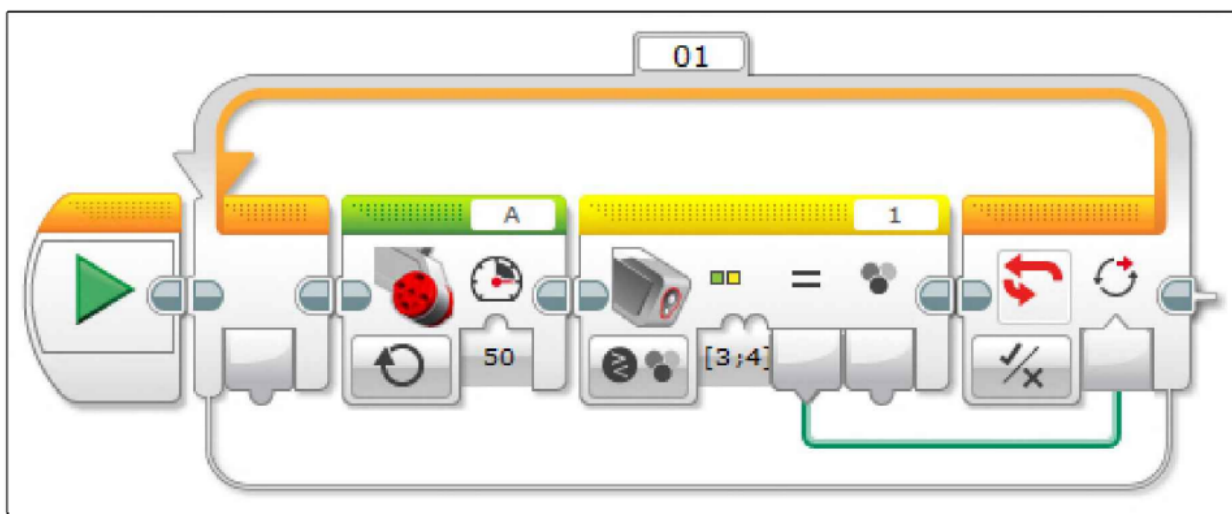


Рис. 8 Пример использования режима сравнения цвета

Режим Калибровки (нормализация)

На (рис. 9.) представлен выбор режима калибровки (нормализации) датчика цвета.

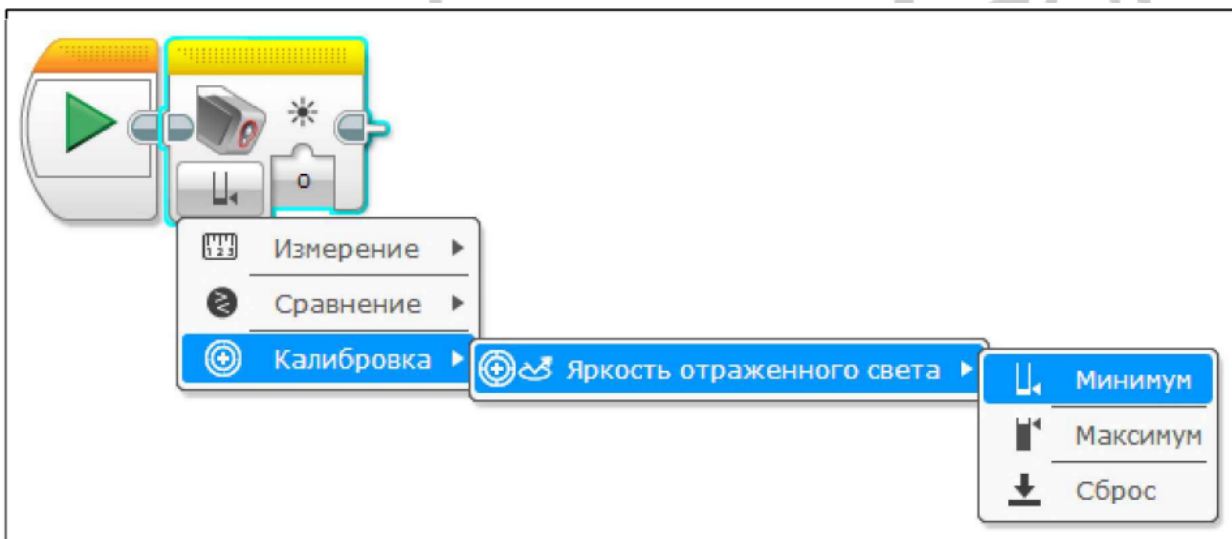




Рис. 9. Режим калибровки датчика цвета

Режим калибровки может быть применён при необходимости различать не очень контрастную линию: светло-зелёную, светло-голубую и др. на белом фоне.

Когда работа движется вдоль черной линии на белом поле, разница показания датчика достаточно велика. При качественном поле и хорошем освещении она может составлять 60-80 единиц. Такая разница позволяет реализовывать хорошее управление движением.

Если необходимо следовать вдоль линии светлого цвета, то разница в показаниях датчика: 5- 10 единиц. Подобная разница не способна обеспечить стабильное управление и является причиной съезда робота даже с достаточно плавной траектории.

Цель применения встроенной калибровки- расширение диапазона значений до 100 единиц. Иногда эту процедуру называют нормализацией значений.

Предположим, находясь над линией, датчик показывает значение 10, находясь над белым полем – 60. После осуществления встроенной калибровки (нормализации) произойдет конвертация минимального значения 10 в 0, а максимального значения 60 в 100. Так образом, диапазон разницы значений будет расширен с 50 до 100 единиц, обеспечив сильное управляющее воздействие на моторы!

На (рис. 10.) показан пример программы использования встроенной калибровки датчика.

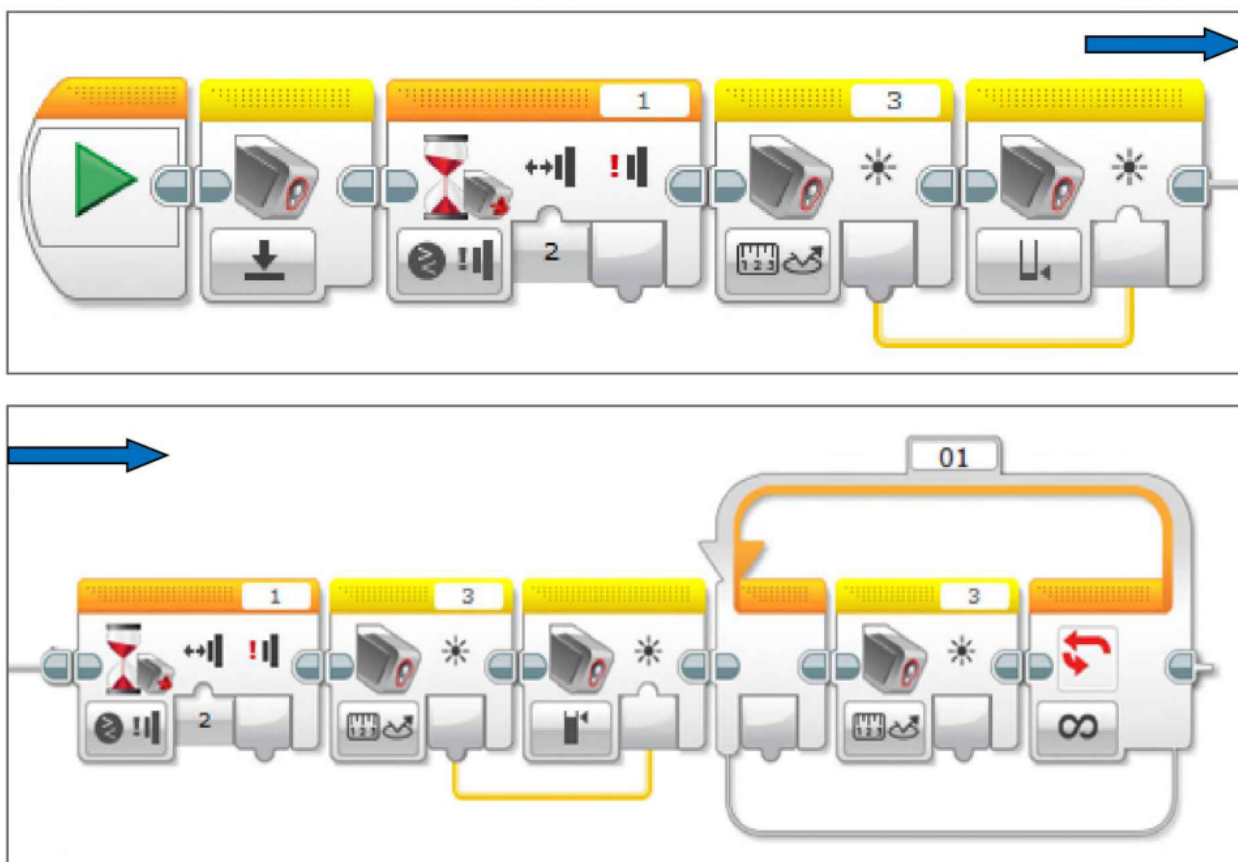
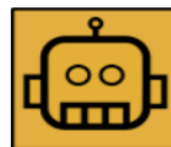


Рис.10. Пример выполнения режима калибровки

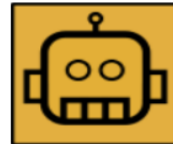
Технология использования функций встроенной калибровки следующая:

Запускаем программу;

Проводим сброс параметров датчика;

Выбираем самое тёмное на поле (или на линии, если создаете программу движения по линии) и запоминаем это значение как Минимум. Пусть это значение равно 10;

Выбираем самое светлое место на поле (или рядом с линией, если создаете создаете программу движения по линии) и запоминаем это как Максимум. Пусть это значение равно 60;



Дальнейшие показания датчиков конвертированы в новый расширенный диапазон.

Если мы не хотим проводить калибровку при каждом запуске робота, необходимо указать предварительно определенные значения максимального и минимального значения. На (рис 11.) представлен фрагмент программы с ручным заданием минимальных и максимальных значений.

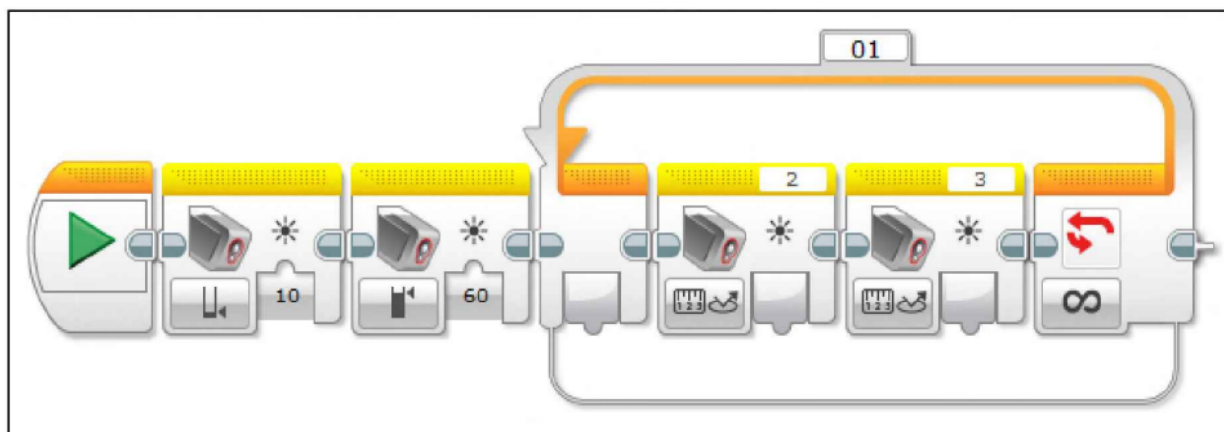


Рис. 11. Пример выполнения режима встроенной калибровки с ручных значений

Режим Ожидания

Выполнение программы происходит аналогично описанному режиму ожидания датчика касания: действия линейной программы выполняется до тех пор, пока не станет истинным условием, установления в блоке *Ожидание*. Датчик оценивает цвет, яркость отражённого и окружающего света. На рис. 12. представлен режим ожидания значения датчика цвета. Робот будет двигаться до тех пор, пока датчик цвета не обнаружит один или несколько заданных в блоке *Ожидания* цветов.

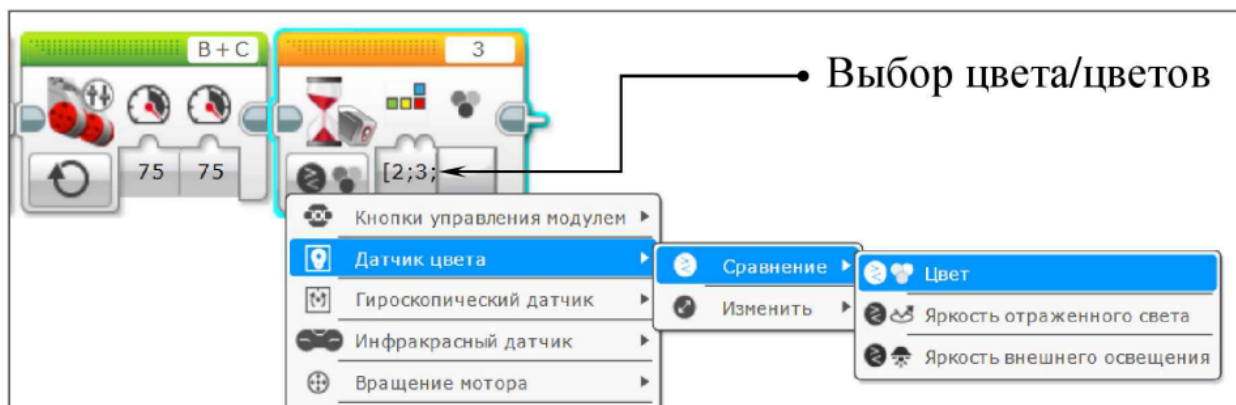
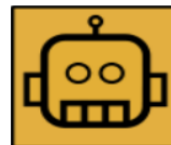


Рис. 12. Режим Ожидания датчика цвета

Существует ещё одна возможность реализации алгоритма ожидания с помощью структуры *Цикл*: в качестве условия завершения работы устанавливается результат измерения цвета (рис, 13). Программа будет работать аналогично программе, приведённой на (рис. 8)- мотор, подключенный в порт А, будет крутиться до тех пор, пока датчик цвета не обнаружит желтый или зеленый цвет.

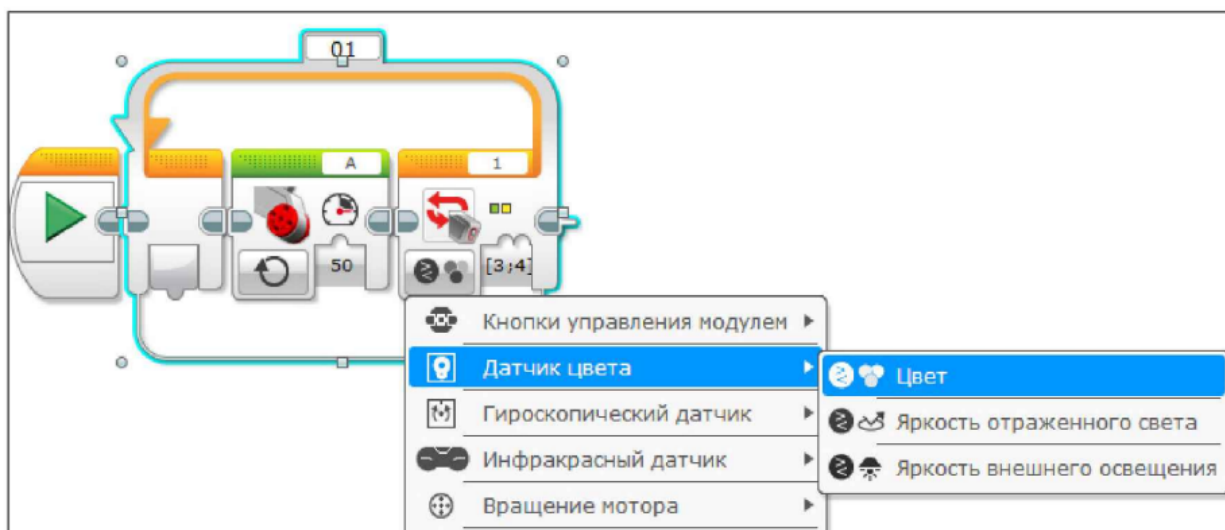
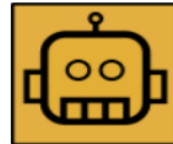


Рис 13. Использование значения датчика цвета в условии завершения цикла



Пример с датчиком цвета

📖 Пример 1: Движение до черной линии (Метод 1).



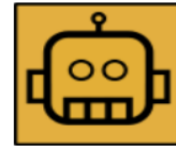
Эта программа заставляет робота двигаться до тех пор, пока датчик цвета не определит черный цвет, после этого робот останавливается. Программа использует блок «Ожидание» датчика цвета – «Сравнение» – режим «Цвет» для тестирования черного цвета.

Если вы используете эту программу и датчик цвета на вашем роботе направлен вниз и находится близко к светлой цветной поверхности с толстой черной линией на ней, то робот может двигаться до тех пор, пока не достигнет линии.

📖 Пример 2: Движение до черной линии (Метод 2).

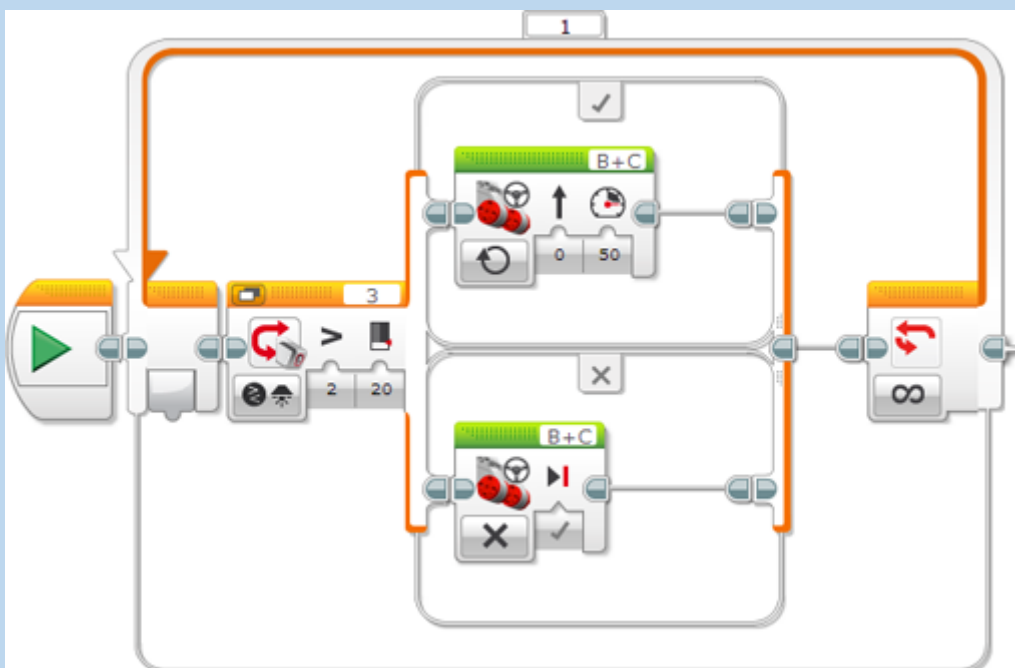


Эта программа заставляет робота двигаться до тех пор, пока датчик цвета не определит темный цвет, после этого робот останавливается. Программа использует блок «Ожидание» датчика цвета – «Сравнение» – режим «Яркость отраженного света» и ждет, пока яркость света не достигнет менее 50 %.

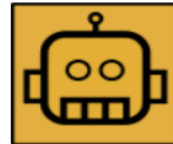


В сравнении с примером 1, эта программа позволяет вам регулировать темноту линии, изменив пороговое значение (здесь 50 %). Кроме того, робот будет останавливаться на любом темном цвете, не только на черном.

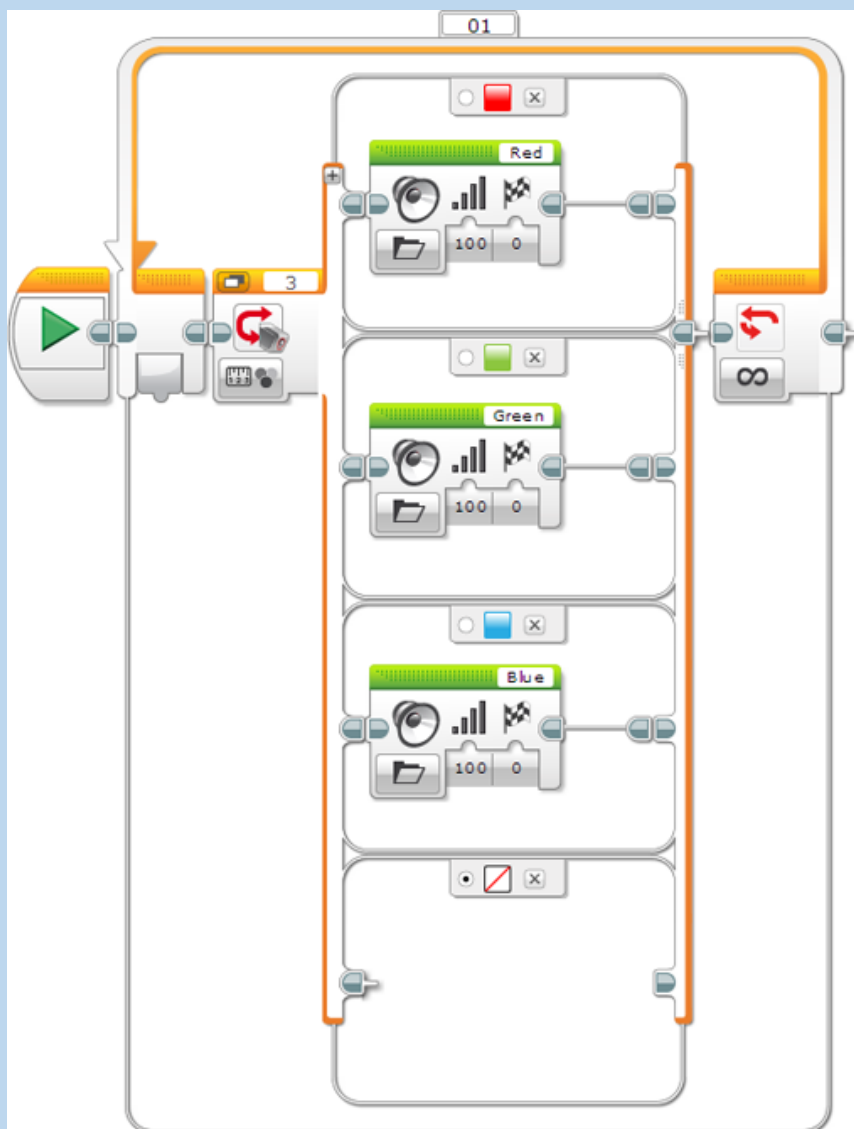
📖 Пример 3: Движение только при включенном свете в комнате



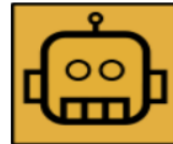
Эта программа заставляет робота двигаться при включенном свете в комнате и останавливает его при выключении света. Программа использует блок «Если ... то» датчика цвета – «Сравнение» – режим «Яркость внешнего освещения» для проверки того, что яркость света составляет более 20 %. Блок «Если ... то» включает или выключает мотор. «Если ... то» повторяется, таким образом, робот будет продолжать реагировать на изменения освещенности.



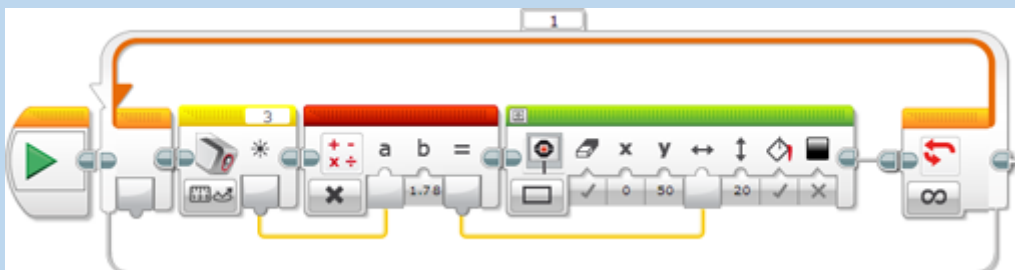
📖 Пример 4: Говорит «красный», «зеленый» и «синий» при определении



В этой программе EV3 говорит «красный», «зеленый» и «синий» при определении этих цветов датчиком цвета. Программа использует блок «Если ... то» датчика цвета – «Измерение» – режим «Цвет» для выбора между разными звуковыми блоками в зависимости от определенного цвета. По умолчанию добавлен и выбран случай «Без цвета», поэтому при обнаружении других цветов, не принадлежащих к этим трем, EV3 ничего не будет говорить.



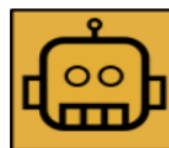
📖 Пример 5: Отображение счетчика отраженного света



Эта программа выводит графический счетчик света на экран EV3. Программа использует блок датчика цвета в режиме [«Измерение»](#) – «Яркость отраженного света» для измерения отраженного света (0-100) и получения данных через шину данных. Результат умножается на 1,78 для подгонки по ширине экрана EV3 (178 пикселей) и затем используется, как ширина заполненного прямоугольника. Процесс повторяется, и экран постоянно обновляется.

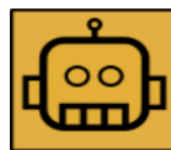
Во время работы этой программы попытайтесь передвигать датчик цвета по поверхностям разных цветов и форм. Вы увидите, какие цвета отражают больше света.





Блок	Режим	Использование
Ожидание	Датчик цвета – Сравнение – Цвет	Подождите, пока датчик определит один из выбранных цветов.
Ожидание	Датчик цвета – Сравнение – Яркость отраженного света	Подождите, пока яркость отраженного света не достигнет определенного значения.
Ожидание	Датчик цвета – Сравнение – Яркость внешнего освещения	Подождите, пока яркость внешнего освещения не достигнет определенного значения.
Ожидание	Датчик цвета – Изменение – Цвет	Подождите, пока не изменится определенный цвет.
Ожидание	Датчик цвета – Изменение – Яркость отраженного света	Подождите, пока яркость отраженного света не изменится на определенное значение.
Ожидание	Датчик цвета – Изменение – Яркость внешнего освещения	Подождите, пока яркость внешнего освещения не изменится на определенное значение.
Цикл	Датчик цвета – Цвет	Повторяйте последовательность блоков, пока не будет определен один из выбранных цветов.
Цикл	Датчик цвета – Яркость отраженного света	Повторяйте последовательность блоков, пока яркость отраженного света не достигнет определенного значения.
Цикл	Датчик цвета – Яркость внешнего освещения	Повторяйте последовательность блоков, пока яркость внешнего освещения не достигнет определенного значения.
Если ... то	Датчик цвета – Измерение – Цвет	Выберите между двумя или более последовательностями блоков в зависимости от определяемого цвета.
Если ... то	Датчик цвета – Сравнение – Цвет	Выберите между двумя последовательностями блоков в зависимости от того, определился ли или нет один из определенных цветов.





Если ... то	Датчик цвета – Сравнение – Яркость отраженного света	Выберите между двумя последовательностями блоков в зависимости от яркости отраженного света.
Если ... то	Датчик цвета – Сравнение – Яркость внешнего освещения	Выберите между двумя последовательностями блоков в зависимости от яркости внешнего освещения.
Датчик цвета	Измерение – Цвет	Измерьте определенный цвет (0-7) и получите результат через шину числовых данных.
Датчик цвета	Измерение – Яркость отраженного света	Измерьте яркость отраженного света (0-100) и получите результат через шину числовых данных.
Датчик цвета	Измерение – Яркость внешнего освещения	Измерьте яркость внешнего освещения (0-100) и получите результат через шину числовых данных.
Датчик цвета	Сравнение – Цвет	Сравните определенный цвет с одним или более выбранными цветами, и получите результат через шину логических данных (истина, если соответствует одному из выбранных цветов).
Датчик цвета	Сравнение – Яркость отраженного света	Сравните яркость отраженного света с пороговым значением и получите результат через шину логических данных.
Датчик цвета	Сравнение – Яркость внешнего освещения	Сравните яркость внешнего освещения с пороговым значением и получите результат через шину логических данных.
Датчик цвета	Калибровка – Яркость отражённого света – Минимум	Определить минимальную яркость света.
Датчик цвета	Калибровка – Яркость отражённого света – Максимум	Определить максимальную яркость света.
Датчик цвета	Калибровка – Яркость отражённого света – Сброс настроек	Возвращает настройки датчика цвета в состояние по умолчанию.



