

РУКОВОДСТВО по работе с набором **Smart Camera Add-on Pack**



Оглавление

Обзор набора Smart Camera Add-on pack	3
Внешний вид камеры и характеристики.....	4
Подключение камеры к роботу	5
Подключение к mBot.....	5
Подключение к Halocode и CyberPi.....	5
Программное обеспечение для работы с камерой.....	7
Интерфейс программы PixyMon.....	8
Меню пиктограмм.....	8
Верхнее меню.....	9
Окно настроек программы.....	9
Программирование в среде mBlock5	10
Режимы работы камеры и их настройка	11
Распознавание цветовых сигнатур (блоков).....	11
Использование кнопки «Обучение».....	12
Использование приложения PixyMon для изучения цветовых блоков	12
Блоки управления в среде mBlock5	22
Определение расстояние до цветовых объектов	27
Распознавание линий	29
Отслеживание линии через PixyMon	29
Обнаружение пересечений и «ветвлений»	31
Алгоритм следования по линии	34
Распознавание штрих-кодов	35
Сочетание следования по линии с распознаванием штрих-кодов.....	37

Обзор набора Smart Camera Add-on pack

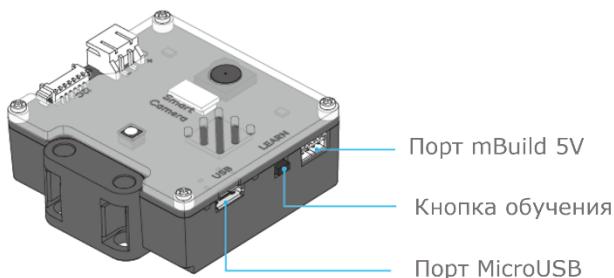
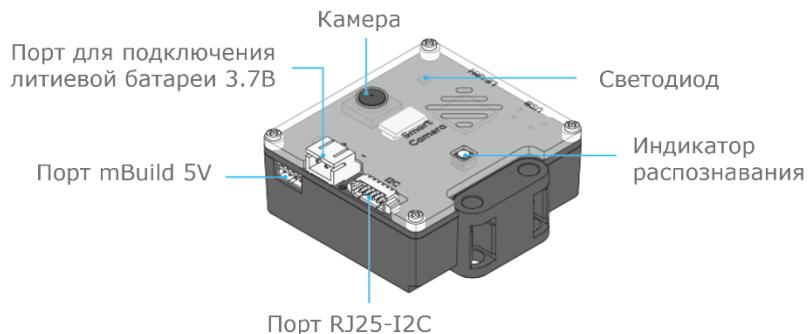


Набор Smart Camera Add-on pack является дополнением, которое может быть использовано совместно с основными робототехническими наборами от компании Makeblock. Камера совместима как с наборами на базе контроллеров Arduino-серии (mBot, mBot Ranger, Ultimate), так и с новыми решениями на базе чипа ESP32 (HaloCode, CyberPi, mBot2). Возможно подключение к роботам на платформе Arduino UNO. В частности, к российским роботам серии КЛИК.

Особенностью камеры является наличие в ней собственного процессора для обработки графической информации и памяти для хранения настроек. То есть, видеопоток в данном случае обрабатывается самой камерой, а не роботом, что сильно снижает требования к применяемым робототехническим контроллерам и упрощает работу.

Интеллектуальная камера может изучать и распознавать цветовые блоки, а также обнаруживать штрих-коды и линии, что позволяет использовать ее в различных сценариях, таких как сортировка, интеллектуальная логистика, отслеживание и следование за линией/объектами.

Внешний вид камеры и характеристики



- **Размеры:** 48 × 48 мм
- **Разрешение:** 640 * 480
- **Поле зрения:** 65,0 градусов
- **Фокусное расстояние:** 4,65±5% мм
- **Скорость распознавания:** 60 к/с
- **Расстояние:** 0,25–1,2 м
- **Питание:** литиевая батарея 3,7 В или модуль питания 5 В mBuild
- **Энергопотребление:** 0,9-1,3 Вт
- **Рабочая температура:** -10 °C –55 °C

Режим: Обнаружение цветовых сигнатур

Камера может распознавать одноцветные яркие объекты, и сообщать координаты цветового блока и его размер.

Режим: Обнаружение линий

В этом режиме камера может обнаруживать линии и сообщает информацию о координатах перекрестка, направлении и количестве ответвлений.

Режим: Распознавание штрих-кодов

Камера также может распознавать этикетки со штрих-кодом, входящие в комплект поставки.

Подключение камеры к роботу

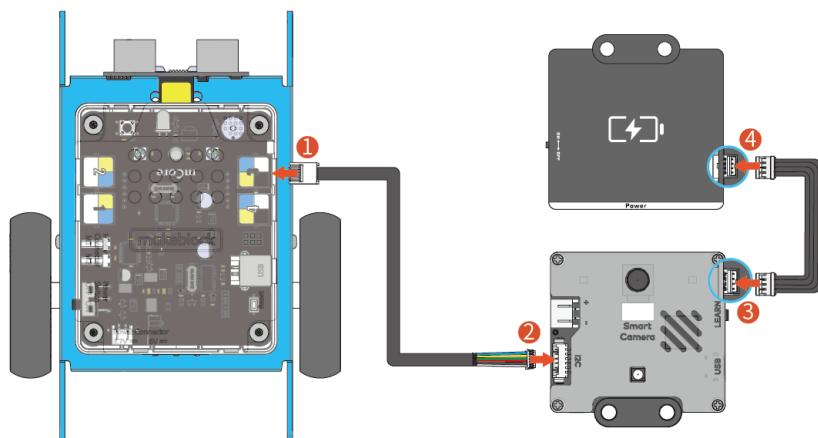
Умная камера может подключаться к различным устройствам от компании makeblock. Алгоритм подключения зависит от устройства и способа подключения питания.

Подключение к mBot

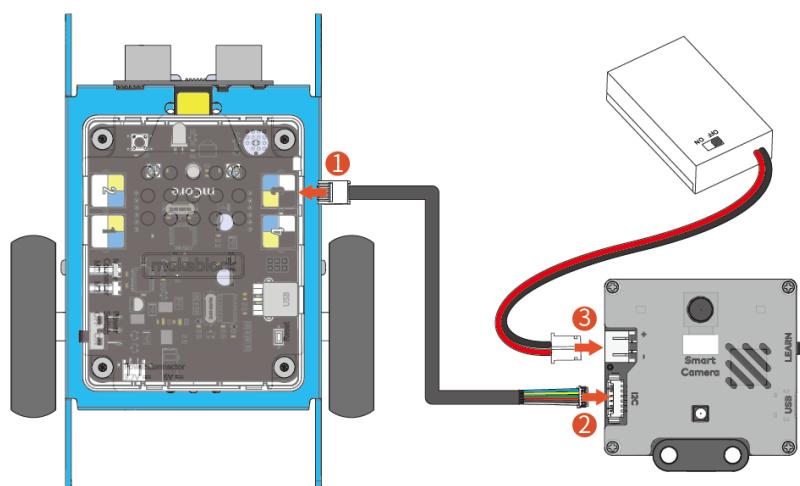
Для mBot и других роботов на базе Arduino-платформ от Makeblock подключение камеры осуществляется через разъем RJ-25 контроллера робота.

После подключения камеры к mBot вы можете питать её от литиевой батареи на 3.7 В или от модуля питания mBuild.

Способ 1. Питание через модуль питания mBuild (рекомендуется)

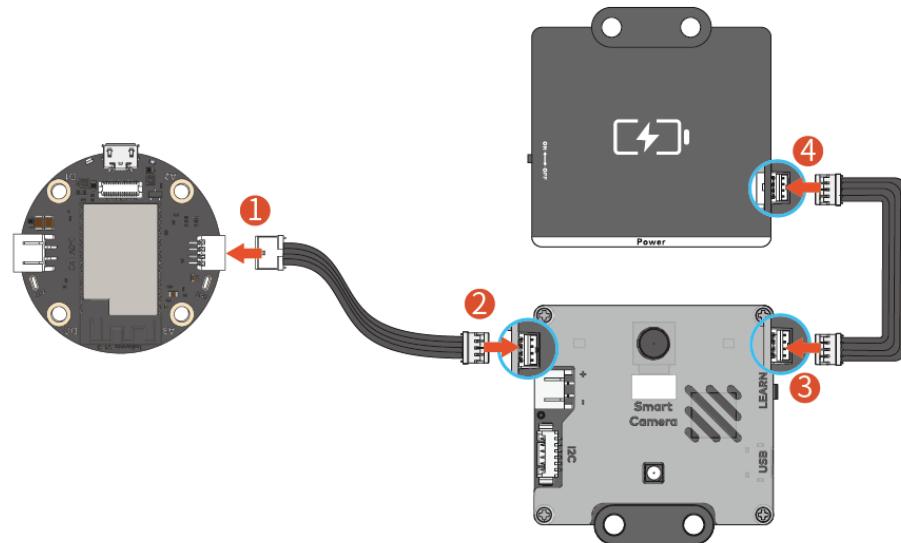


Способ 2. Питание через литиевую батарею 3.7В



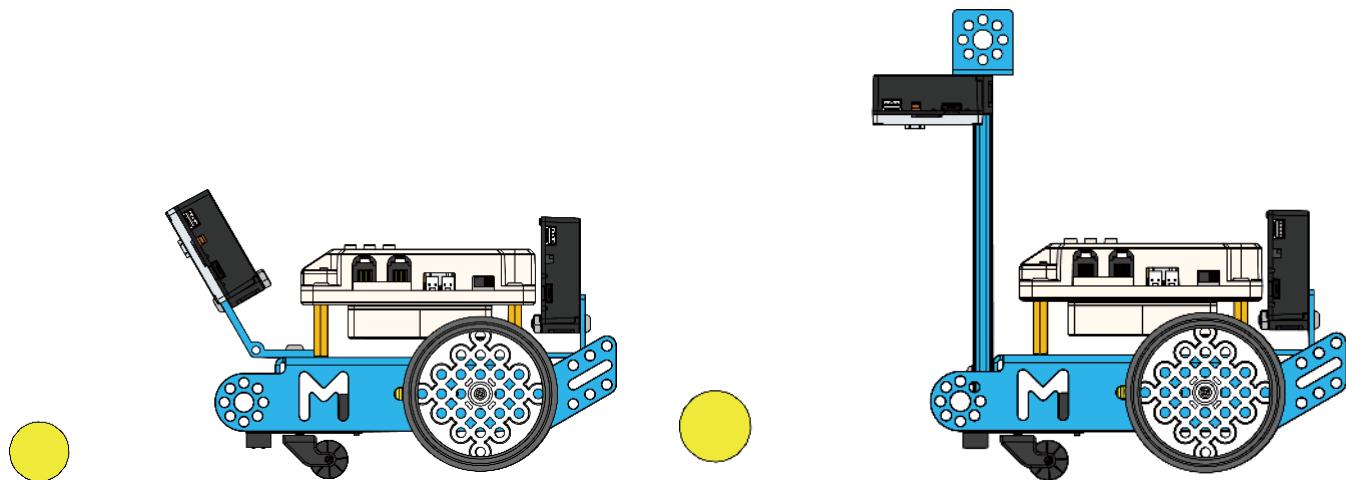
Подключение к Halocode и CyberPi

Подключение и дополнительное питание смарт-камеры рекомендуется осуществлять через порт mBuild.



Угол и высота, на которых установлена камера, влияют на результат обнаружения. Рекомендуется настроить положение камеры в зависимости от объекта, который вы хотите отслеживать.

Ниже приведены две ситуации, в которых положение мяча одинаковое, но углы установки интеллектуальной камеры различаются. Мы видим, что во второй ситуации камера расположена параллельно земле, что ограничивает её поле зрения, поэтому она не может «увидеть» шар. В результате угол зрения камеры разный, а значит и объект будет обнаружен в разное время.



Программное обеспечение для работы с камерой

Для настройки камеры применяется бесплатное свободно распространяемое программное обеспечение PixyMon v2.



Иконка ПО *PixyMon*

PixyMon v2 — это инструмент, который позволяет настроить камеру и посмотреть, что она видит. Приложение работает на различных платформах, включая Windows, MacOS и Linux, а также на других небольших встроенных системах, таких как Raspberry Pi и BeagleBone Black.

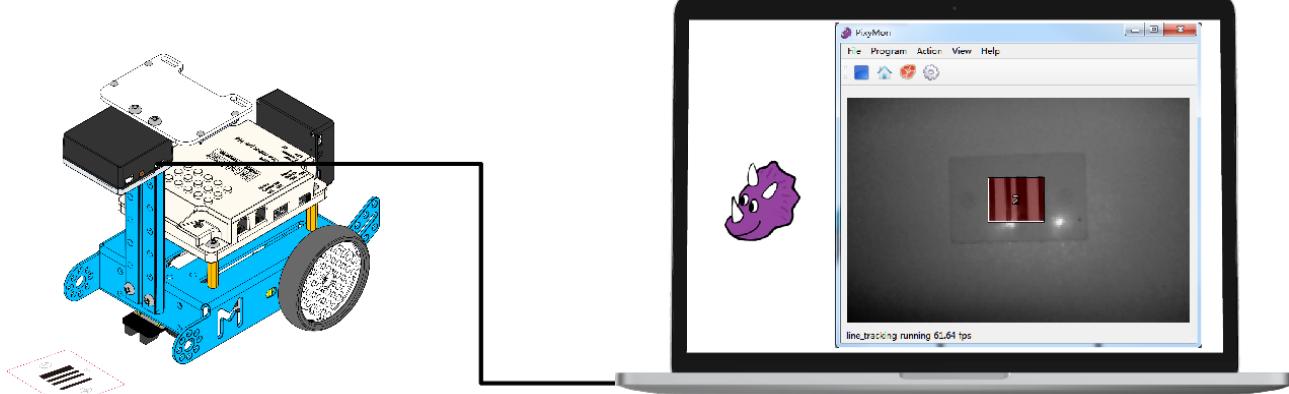
Ссылки для скачивания и документацию по установке можно найти на официальном сайте производителя ПО: <https://pixycam.com/downloads-pixy2/>



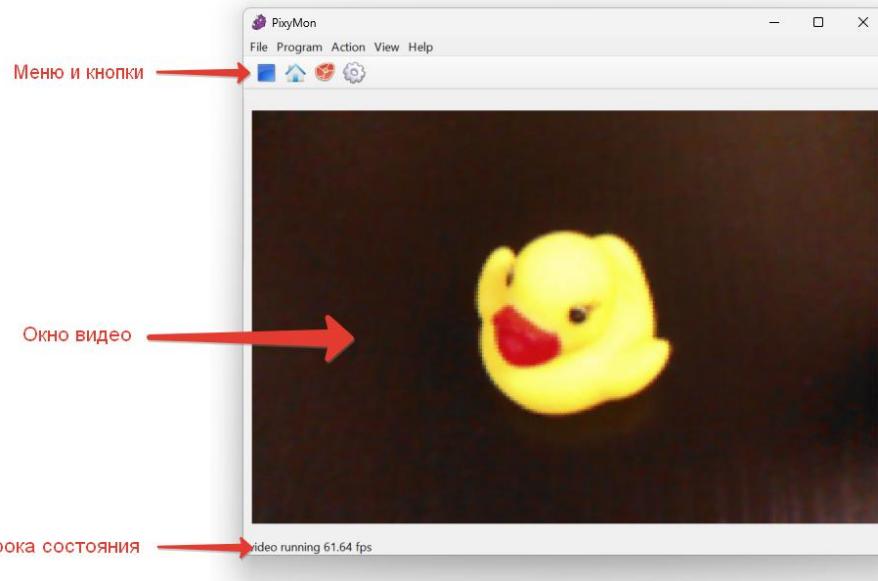
Внимание

Прошивки, опубликованные на данном сайте предназначены для камер *Pixy* и не подходят для оборудования *Makeblock*. Скачивайте только программное обеспечение *PixyMon V2*.

Для отображения информации с камеры в *PixyMon* необходимо подключить ее к компьютеру с помощью кабеля Micro USB. Когда подключение будет завершено, изображение появится в окне *PixyMon*.



Интерфейс программы PixyMon



Меню и кнопки: здесь размещено меню для настройки камеры и отображения, а также вынесены кнопки для наиболее частых действий в PixyMon.

Окно видео: здесь отображается видео поток в формате обработанного и необработанного видео.

Строка состояния: здесь отображается состояние видео потока, а также текущее положение курсора.

Меню пиктограмм



[1] Стоп/возобновление: нажатие этой кнопки останавливает воспроизведение видео в окне видео. Это удобно, когда нужно сделать снимок или ввести команды в окне команд/статуса. Нажмите эту кнопку еще раз, чтобы возобновить воспроизведение видео.

[2] Программа по умолчанию: нажатие этой кнопки запускает программу по умолчанию, которая выполняется при включении. Обычно это программа обнаружения цветовых блоков.

[3] Необработанное видео: при нажатии на эту кнопку отображается необработанное видео. Это полезно для настройки фокуса, яркости камеры и т. д.

[4] Настройка: при нажатии на эту кнопку открывается диалоговое окно «Настройка», в котором содержатся различные настраиваемые параметры для PixyMon.

Верхнее меню



File: содержит элементы для сохранения, загрузки и восстановления параметров конфигурации.

Program: содержит список доступных режимов работы камеры и позволяет переключаться между ними

Action: меню меняется в зависимости от того, какой режим работы камеры выбран. Общим для всех программ является действие по включению подсветки «**Toggle lamp**».

View: меню меняется в зависимости от того, какой режим работы камеры выбран, и позволяет настроить какие данные отображать в PixyMon.

Help: это меню содержит ссылку на справочную документацию (на английском языке) и информацию о программе «**About**».

Окно настроек программы



Если щелкнуть значок шестеренки на панели инструментов или выбрать **Конфигурация «Configure»** в меню **Файл «File»**, откроется диалоговое окно настроек, которое меняется в зависимости от режима работы камеры и в общем случае содержит две вкладки **Настройка «Tuning»** и **Эксперт «Expert»**.

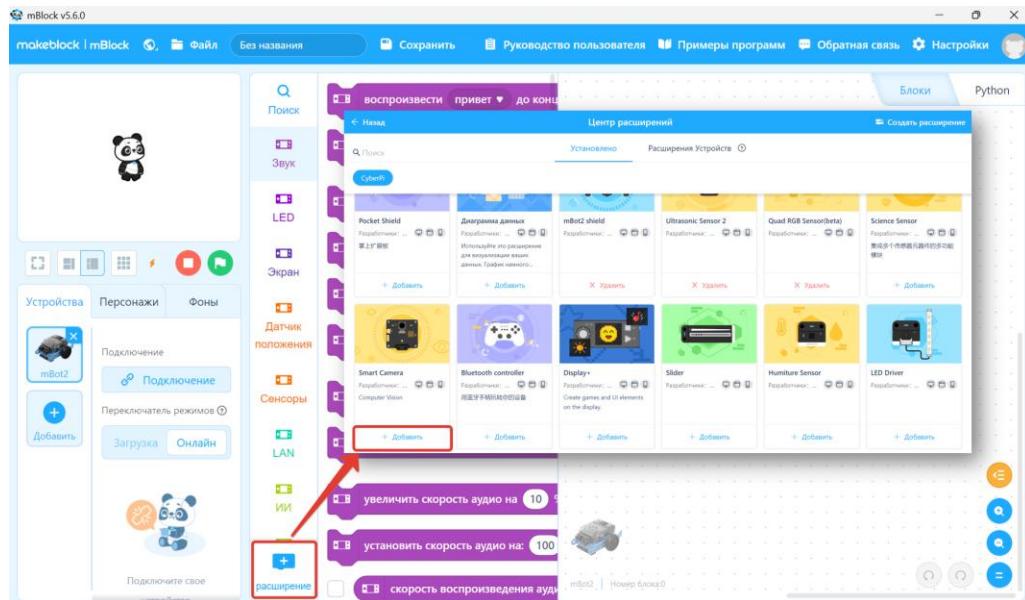
Вкладка **Настройка «Tuning»** содержит параметры, которые часто используются для подстройки сенсоров при запуске конкретной программы. Например, при запуске программы обнаружения цветных блоков здесь настраиваются диапазоны сигнатур для повышения точности обнаружения объектов.

Вкладка **Эксперт «Expert»**, содержит параметры, которые настраиваются реже, но все же могут быть важны для оптимальной работы конкретной программы. Например, при запуске программы обнаружения цветных блоков максимальное допустимое количество блоков, которое необходимо отслеживать, находится на вкладке «Expert», поскольку его обычно меняют один или два раза.

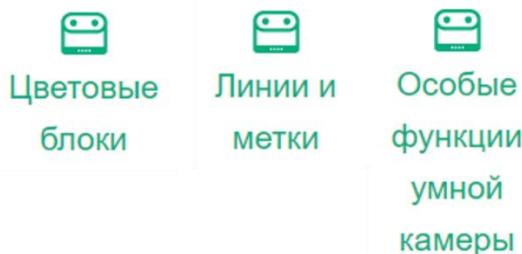
Программирование в среде mBlock5

Для программирования робота с использованием камеры в среде mBlock5 необходимо добавить расширение Smart Camera. Для этого:

- Перейдите во вкладку **Устройства**.
- Нажмите на кнопку **+расширение**.
- Выберите расширение Smart Camera и нажмите кнопку **+ Добавить**.



Будут добавлены три новых палитры блоков:

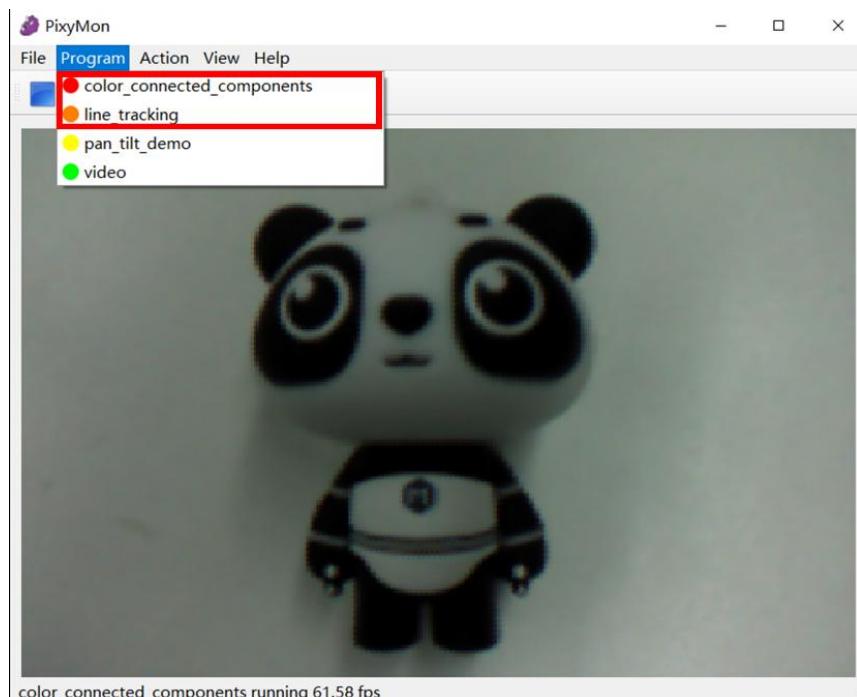


Принцип работы программных блоков, не зависит от выбранного устройства. В данном пособии разбор программных блоков будет приводиться на примере mBot2.



Режимы работы камеры и их настройка

PixyMon позволяет переключаться между режимами работы Smart Camera. На рисунке ниже отмечены две опции, которые отвечают за обнаружение цветовых блоков и линий соответственно. Важно отметить, что обнаружение линий и цветовых блоков не может выполняться одновременно, однако распознавание линий и штрихкодов осуществляются в одном и том же режиме — отслеживание линий.



Распознавание цветовых сигнатур (блоков)

Этот режим работы представляет собой алгоритм обнаружения объектов на основе оттенка — **Алгоритм Color Connected Components (CCC)**

Большинству из нас знакомо представление цветов RGB (красный, зеленый и синий). Камера использует алгоритм цветовой фильтрации для обнаружения объектов на основе цветового оттенка (hue) и насыщенности (saturation) каждого пикселя RGB. Методы цветовой фильтрации популярны, поскольку они быстрые, эффективные и относительно надежные.

Оттенок объекта практически не меняется при изменении освещения и экспозиции.

Алгоритм CCC камеры запоминает до 7 различных цветовых подписей, а это означает, что, если у вас есть 7 разных объектов с уникальными цветами, алгоритм цветовой фильтрации без проблем их идентифицирует.

Настраивать работу с цветовыми сигнатурами, можно как используя приложение PixyMon, так и кнопку обучения непосредственно на камере makeblock.

Использование кнопки «Обучение»

- Нажмите и удерживайте кнопку «Обучение», расположенную рядом с портом MicroUSB. Индикатор на задней панели сначала загорится белым, а затем начнет мигать красным, оранжевым, желтым, зеленым, голубым, синим и фиолетовым. Чтобы научить камеру распознавать первый цветовой блок, отпустите кнопку, когда индикатор загорится красным.



Внимание

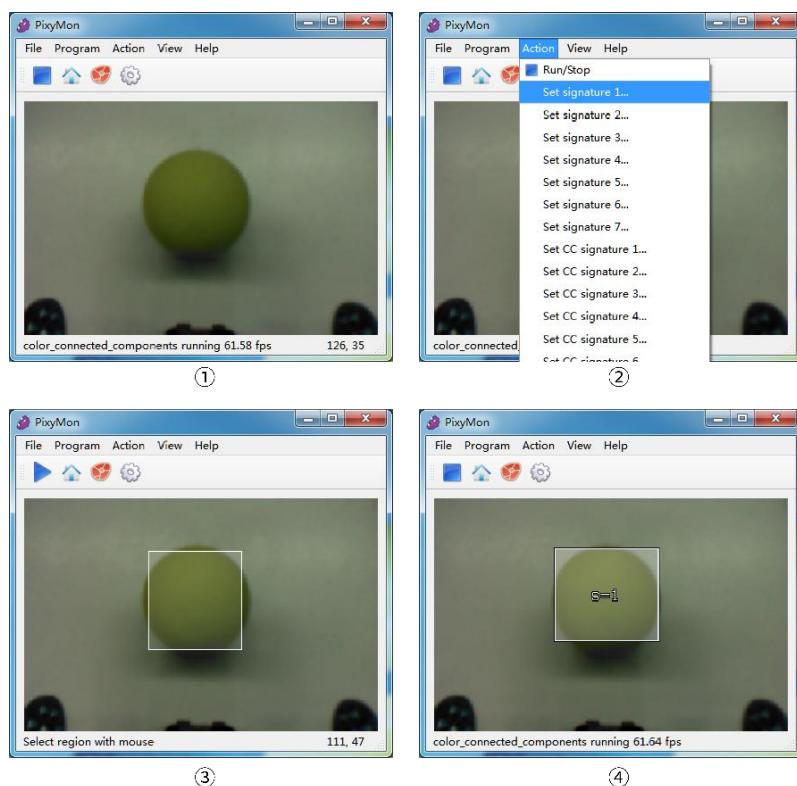
На данном этапе разные цвета светодиода НЕ связаны с цветом объекта и используются только для обозначения номера запоминаемой цветовой сигнатуры (блока). Например, цветовой блок 1 может быть синим шаром, даже если цветовой блок 1 отображается красным светом, а цветовой блок 6 может быть оранжевым объектом, хотя цветовой блок 6 обозначается синим светом.)

Красный	Оранжевый	Желтый	Зеленый	Голубой	Синий	Фиолетовый
Цветовой блок 1	Цветовой блок 2	Цветовой блок 3	Цветовой блок 4	Цветовой блок 5	Цветовой блок 6	Цветовой блок 7

- Поместите цветной объект, который нужно изучить, перед камерой.
- Наблюдайте за индикатором на передней или задней панели камеры и медленно перемещайте изучаемый объект, пока цвет индикатора не совпадет с цветом объекта.
- Используйте короткое нажатие на кнопку «Обучение», чтобы записать текущий изученный объект. После успешного обучения, когда камера будет распознавать изученный объект, цвет индикатора будет становиться таким же, как у изученного объекта. В этом режиме можно изучить и записать максимум семь объектов.

Использование приложения PixyMon для изучения цветовых блоков

- Перейдите в меню Программа (**Program**) и выберите режим распознавания цветовых блоков **color_connected_components**;
- Расположите объект для изучения в поле зрения камеры (рис. 1);
- Перейдите в пункт меню **Action > Set signature 1**. В этот момент окно видеозахвата PixyMon будет заморожено (рис. 2);
- Нажмите левую кнопку мыши и растяните прямоугольную область, чтобы выбрать тот цвет, который нужно запомнить (рис. 3);
- PixyMon вернется к динамическому отображению видео, после того как камера завершит обучение. Обнаруженный цветной блок будет заключён в прямоугольную рамку с его номером (рис. 3).



Внимание

Если вы захотите сбросить все запомненные цветовые сигнатуры используйте пункт меню **Action > Clear all signatures**.

Рекомендации по работе

1. Чем ярче цвет изучаемого объекта, тем точнее распознавание. Не стоит использовать для изучения черно-белые и серые объекты.



2. Умная камера не сможет различать типы объекты схожие по цвету. Например: отличить желтый мяч от желтой шляпы.

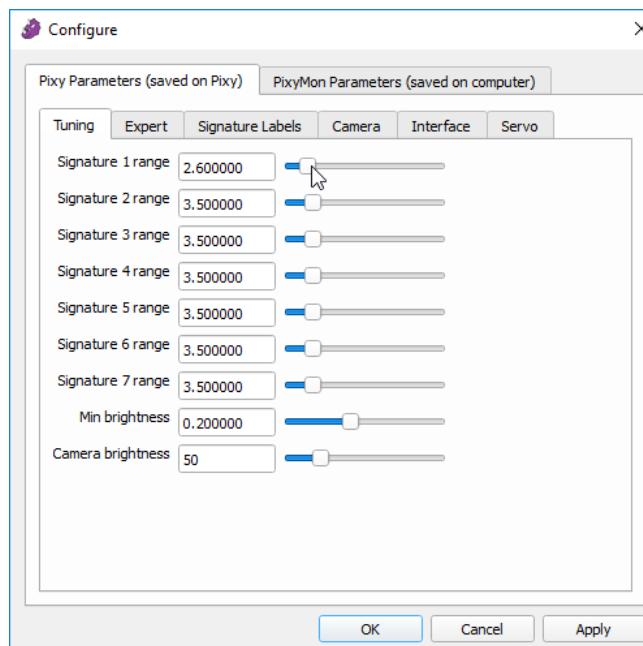


Тонкая настройка

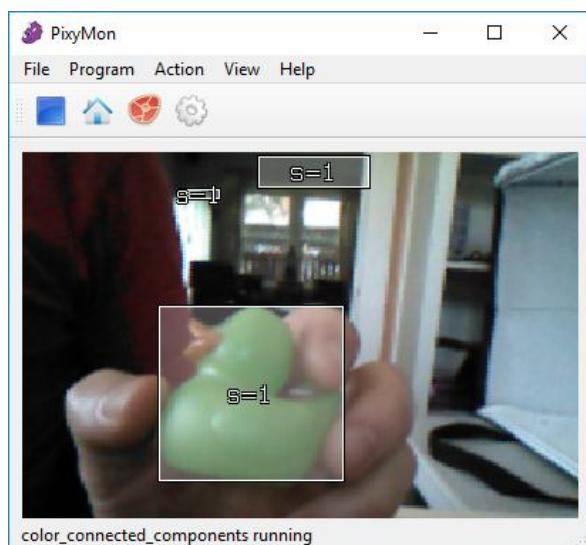
Иногда цветовые сигнатуры, которым вы обучаете камеру, необходимо «подправить», для повышения надежности распознавания объектов. Например, вы получаете:

- ложноположительные результаты (камера обнаруживает объекты, которые не являются тем, что вы предполагали). Часто это объект похожего оттенка. Например, камера считает, что ваша светло-зеленая рубашка соответствует зеленому мячу.
- ложноотрицательные результаты (камера не обнаруживает объект, которому вы её обучили, или обнаруживает объект не всегда).

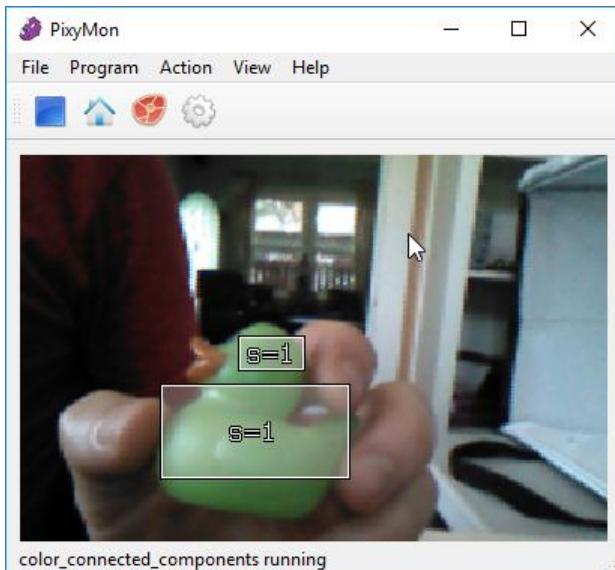
Тогда вы можете дополнительно настроить параметры цветовых сигнатур, открыв диалоговое окно настроек (значок шестеренки).



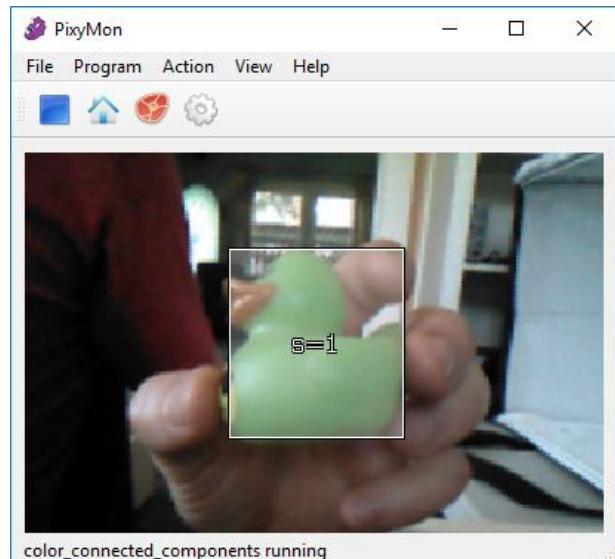
Используйте ползунок **Signature 1 range**, чтобы настроить параметры сигнатуры 1. Сдвиньте его влево, если вы хотите сделать камеру менее чувствительной (т. е. если вы видите ложные срабатывания, как показано на рисунке ниже).



Или сдвиньте ползунок вправо, если вы хотите сделать камеру более чувствительной (т. е. вы видите ложноотрицательные результаты) или обнаружение прерывистое, не постоянное (как на рисунке ниже):



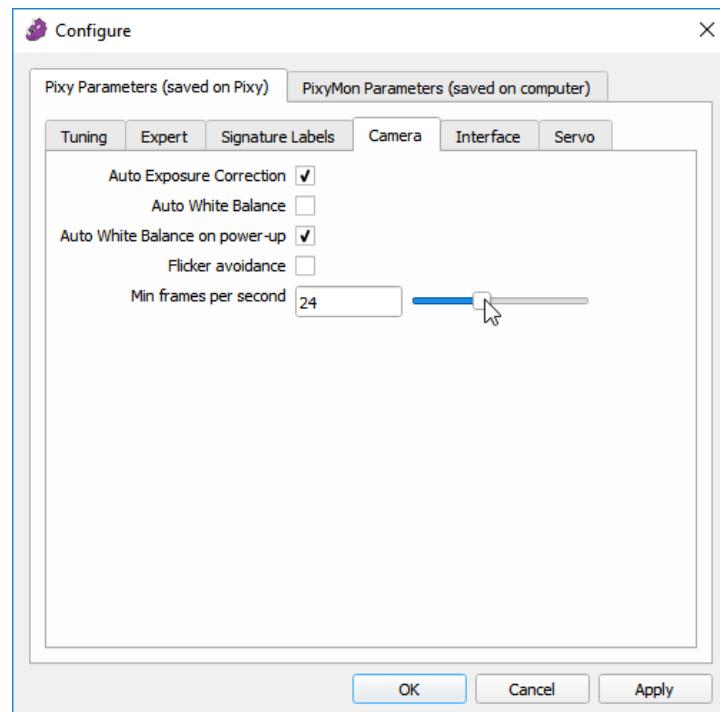
Добейтесь устойчивого постоянного обнаружения изученного объекта:



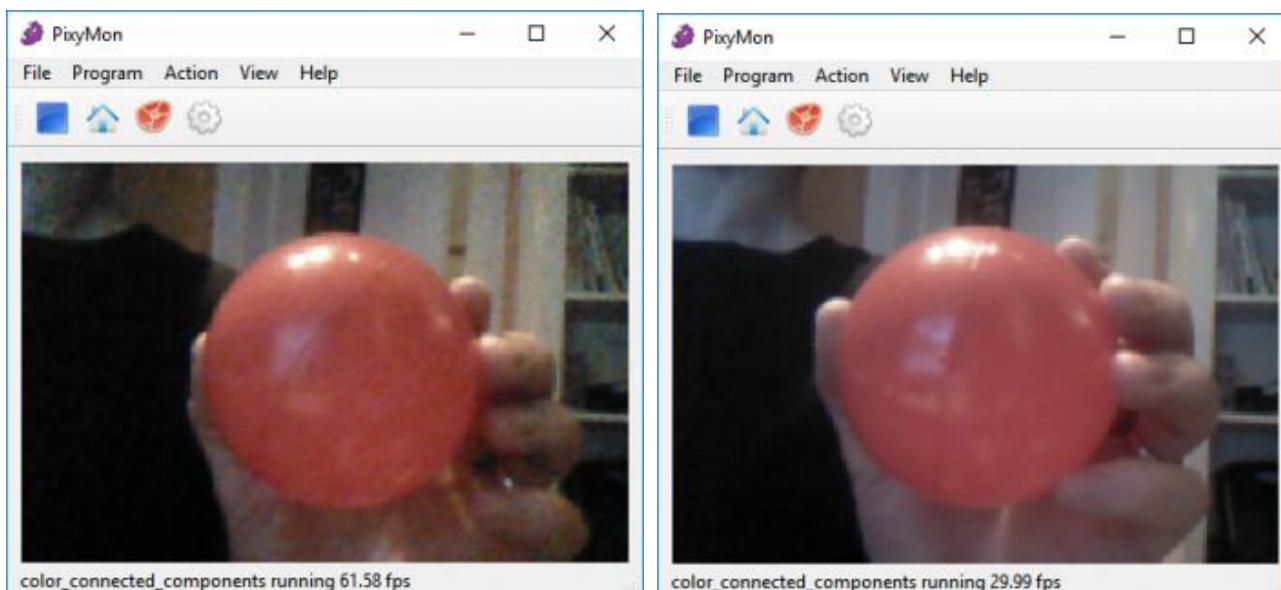
Таким образом можно настроить все семь цветовых сигнатур, чтобы максимизировать точность обнаружения. Обязательно нажмите «**Apply**» (Применить) или «**OK**», чтобы сохранить диапазоны ползунков. Откорректированные значения не будут сохранены, если вы нажмете «**Cancel**» (Отмена) или закроете диалоговое окно.

Так же, для улучшения качества обнаружения объекта можно использовать регулировку количества воспринимаемых камерой кадров в секунду.

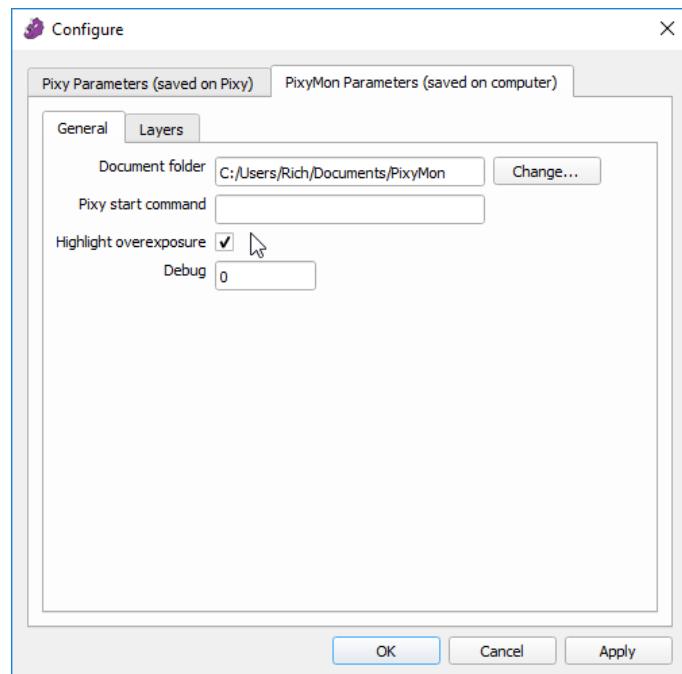
Установка ползунка «**Min frames per second**» (**Минимальное количество кадров в секунду**) на вкладке «**Camera**» на меньшие значения может улучшить качество изображения за счет уменьшения шума и/или улучшения экспозиции в условиях низкой освещенности. Недостаток данного метода (уменьшения количества кадров в секунду) заключается в том, что это снижает частоту обновления камеры.



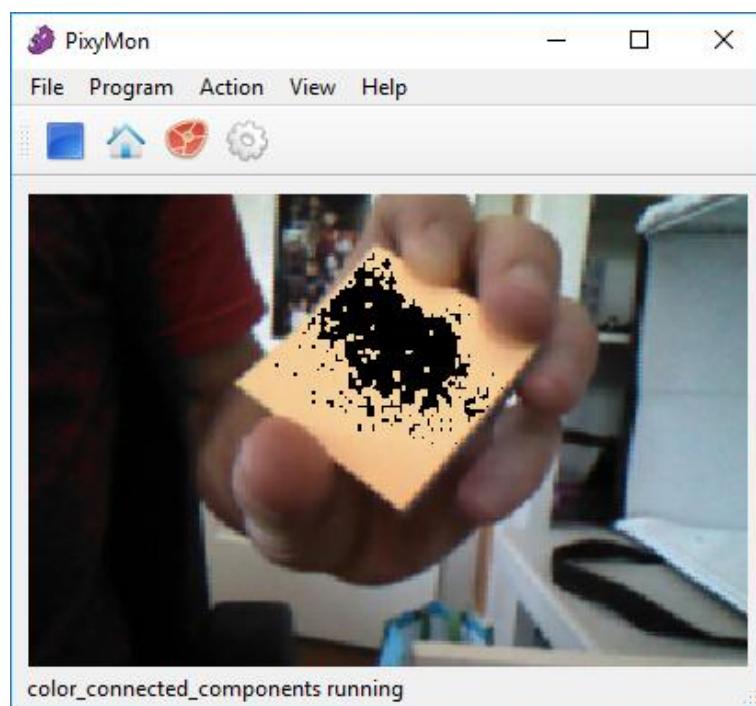
На двух изображениях ниже показана сцена при слабом освещении и снижение шума за счет уменьшения количества кадров в секунду. (Левый кадр — выше частота кадров, больше шума.)



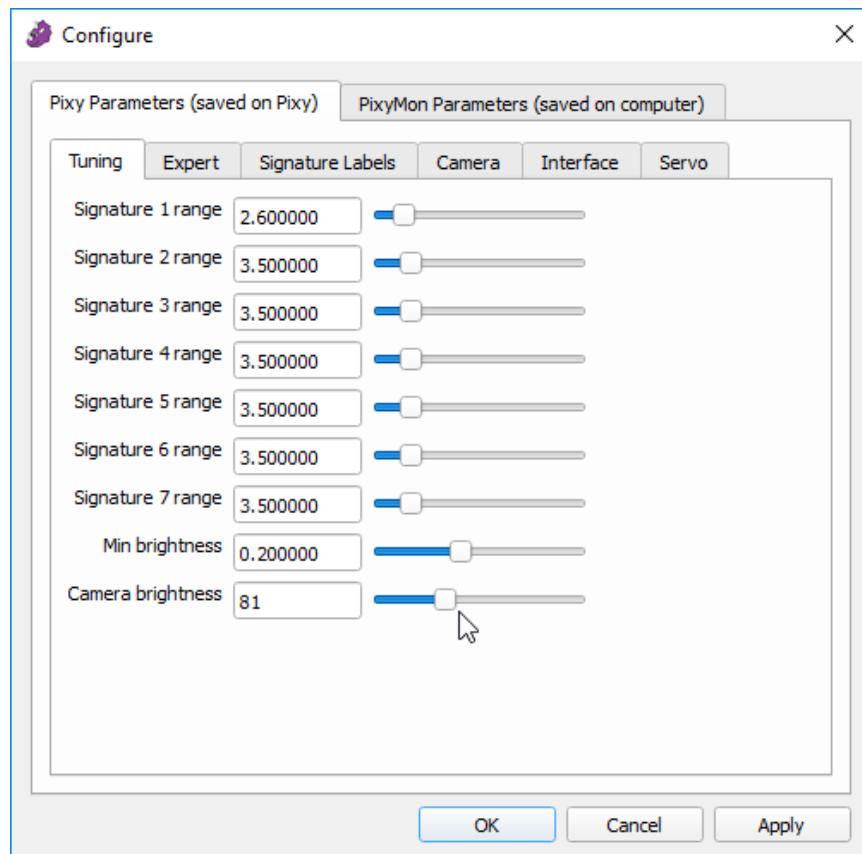
Камера должна получить правильную настройку экспозиции, иначе ее изображения не будут содержать достаточный динамический диапазон, и точность обнаружения пострадает. В PixyMon V2 есть функция, которая выделяет переэкспонированные области изображения. Вы можете включить её, установив флажок рядом с «**Highlight overexposure**» (Выделить передержку) на вкладке «**PixyMon parameters**» (Параметры PixyMon).



Когда эта опция включена, передержанные части изображения будут выделены черным цветом, как показано ниже:



Мы настоятельно рекомендуем включить подсветку передержки, поскольку если объект, который вы хотите обнаружить, переэкспонирован, точность обнаружения пострадает. Итак, после включения подсветки передержки перейдите на вкладку **«Tuning»** в **Pixy parameters** и выберите подходящую настройку **«Camera brightness»** яркости камеры, регулируя ползунок.



Хорошая настройка экспозиции оставляет почти все пиксели вашего объекта правильно экспонированными, но не слишком темными (недоэкспонированными). Небольшие части вашего объекта могут быть переэкспонированы, особенно если ваш объект блестящий. См. картинку ниже – обратите внимание, что небольшая часть объекта (красный шарик) имеет черную область, указывающую на передержку. Это допустимо.

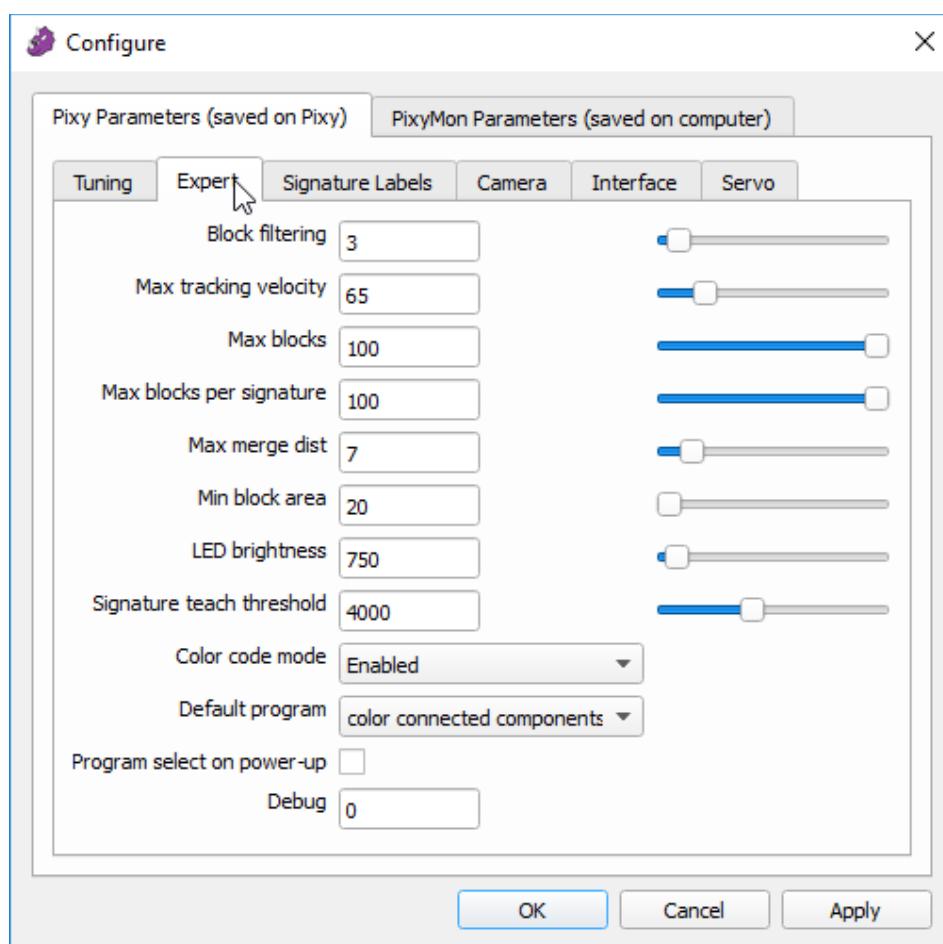


Отслеживание объекта

Как только камера обнаружит новый объект, она добавит его в таблицу объектов, которые в данный момент отслеживает, и присвоит ему индекс отслеживания. Затем она попытается найти объект (и каждый объект в таблице) в следующем кадре, найдя его наилучшее соответствие. Каждый отслеживаемый объект получает индекс от 0 до 255, который он будет сохранять до тех пор, пока он не покинет поле зрения камеры или пока она больше не сможет найти объект в последующих кадрах (из-за окклюзии, отсутствия освещения и т. д.).

Отслеживание полезно, когда вы хотите, чтобы ваша программа следила за определенным экземпляром объекта, даже если в кадре может быть несколько других подобных объектов.

Вкладка настроек Expert (Эксперт)



Block filtering (Фильтрация блоков): Увеличение этого параметра увеличивает объем фильтрации, применяемой камерой к обнаружению каждого блока. Увеличение этого параметра может уменьшить количество ложных срабатываний, но также увеличит время на обнаружение новых объектов. Камере потребуется больше времени, чтобы сообщить об обнаружении нового объекта и добавить его в список отслеживаемых.

Max tracking velocity (Максимальная скорость отслеживания): этот параметр устанавливает максимальное расстояние, которое обнаруженный объект может пройти между кадрами. Другими словами, объекты, скорость которых превышает эту скорость, будут считаться недействительными. Уменьшение этого параметра может уменьшить количество ложных срабатываний, особенно если вы не ожидаете, что обнаруженные объекты будут существенно перемещаться от кадра к кадру. Значение указывается в пикселях за период кадра (16,7 миллисекунды).

Max blocks (Максимальное количество блоков): этот параметр устанавливает максимальное общее количество блоков (объектов), которые камера будет сообщать за кадр. Например, если вы установите для этого параметра значение 2, то камера сообщит максимум о 2 объектах (2 крупнейших объекта) независимо от того, к какой сигнатуре принадлежат блоки, а обо всех остальных обнаруженных объектах не будет сообщено.

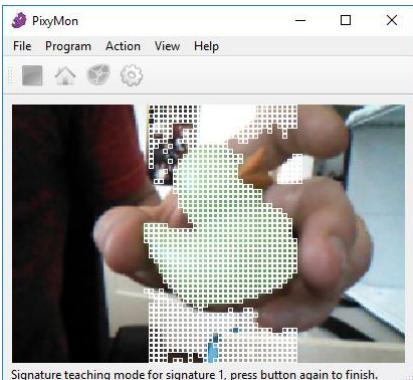
Max blocks per signature (Максимальное количество блоков на одну подпись): этот параметр устанавливает максимальное количество блоков (объектов), которые камера будет сообщать на каждую сигнатуру в каждом кадре. Например, если вы установите для этого параметра значение 2, камера будет сообщать максимум о 2 объектах для каждой сигнатурой (самые большие объекты в кадре), а обо всех других обнаруженных объектах с этой сигнатурой сообщаться не будет.

Max merge dist (Максимальное расстояние слияния): камера будет объединять объекты, имеющие одинаковую подпись и расположенные достаточно близко друг к другу в кадре. Этот параметр регулирует это расстояние. Увеличение этого параметра приведет к тому, что камера объединит больше обнаруженных блоков в один блок.

Min block area (Минимальная площадь блока): устанавливает минимальную площадь распознаваемых блоков. Если обнаруженный блок имеет меньшую площадь, он не будет отслеживаться. Площадь блока рассчитывается как ширина × высота.

LED brightness (Яркость светодиода): устанавливает яркость светодиода RGB камеры. Данная регулировка может быть полезна, т.к. светодиод камеры достаточно яркий.

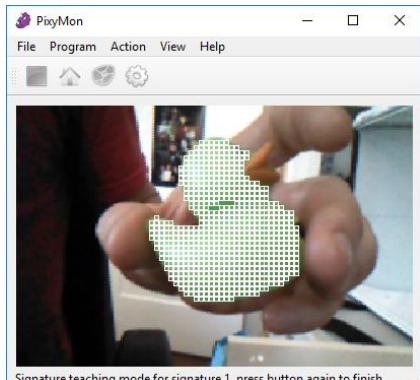
Signature teach threshold (Порог обучения подписи): этот параметр применяется, когда вы обучаете камеру подписи с помощью метода нажатия кнопки. Это позволяет настроить степень инклюзивности камеры при определении того, какие пиксели являются частью объекта, которому вы пытаетесь ее обучить.



Слишком низкий порог



Слишком высокий порог



Оптимальный порог

Color code mode (Режим цветового кода): этот параметр определяет поведение цветовых кодов, включая включение/отключение.

Цветовой код (СС) — это две или более цветные метки, расположенные близко друг к другу. Камера может обнаруживать и декодировать СС и представлять их как специальные объекты. СС полезны, если у вас есть много объектов, которые вы хотите обнаружить и идентифицировать (т. е. больше, чем можно обнаружить с помощью только семи отдельных цветовых сигнатур).

Схема цветового кода с двумя тегами и четырьмя разными цветами позволяет различать до 12 уникальных объектов. Возможны СС с 3, 4 и 5 тегами и/или более разными цветами, которые позволяют создавать гораздо больше уникальных объектов. (На самом деле, при использовании СС с 5 тегами и 6 цветами возможны тысячи уникальных кодов.)

СС также повышают точность обнаружения за счет уменьшения количества ложных обнаружений. То есть вероятность того, что определенные цвета будут встречаться как в определенном порядке, так и близко друг к другу, мала. Недостаток заключается в том, что вам нужно разместить СС на каждом объекте, который вы хотите обнаружить. Часто интересующий вас объект (желтый шар, фиолетовая игрушка) имеет уникальную цветовую подпись, и СС не нужны. Объекты с СС и объекты без СС можно без проблем использовать совместно.



СС могут быть особенно полезны для помощи работе в навигации. Например, внутренняя среда с СС, однозначно идентифицирующими каждый дверной проем и коридор, будет одновременно недорогой и надежной.

Опции включают в себя:

- **Enabled (Включено):** Цветовые коды включены, обычный режим.
- **Disabled (Отключено):** Цветовые коды отключены.
- **Color code only (Только цветовые коды):** Сообщаются только цветовые коды. Об объектах, без цветового кода, не будет сообщено.
- **Mixed (Смешанный):** это то же самое, что и обычный режим цветового кода, но камера также будет сообщать об обнаруженных цветовых блоках, которые не являются частью цветового кода (2 или более сингнатур цветового кода, расположенных близко друг к другу.)

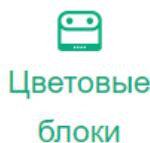
Default program (Программа по умолчанию): этот параметр отображается на всех вкладках Expert (Эксперт), независимо от программы, которая запущена в данный момент. Он устанавливает программу, которая запускается при включении питания. Доступные программы перечислены в раскрывающемся меню.

Переключатель Program select on power-up (выбор программы при включении питания): этот параметр отображается на всех вкладках Expert (Эксперт), независимо от программы, которая запущена в данный момент. Включение этого параметра позволит установить программу по умолчанию, которая запустится при включении питания камеры.

Debug (Отладка): этот параметр отображается на всех вкладках Expert (Эксперт), независимо от программы, которая запущена в данный момент. Он устанавливает уровень отображения информации об отладке в камере. Установка значения 0 означает, что отладочная информация не отображается.

Блоки управления в среде mBlock5

Программные блоки для работы в режиме распознавания цветовых сигнатур располагаются в палитре Цветовые блоки



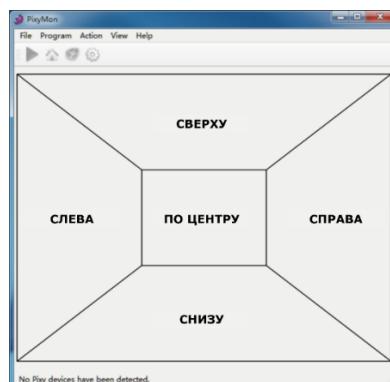
Для того, чтобы перейти в режим работы с цветовыми блоками, необходимо использовать команду:

умная камера 1 ▾ переключиться на режим определения цветовых блоков

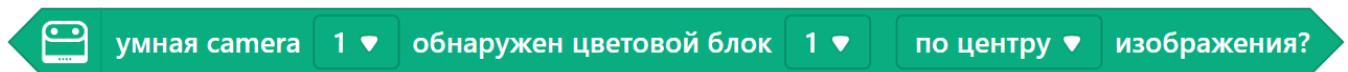
Есть два способа определить положение цветового блока. Первый — определение области, в которой находится цветовой блок, второй — определение координат центра цветного блока.

Области обзора

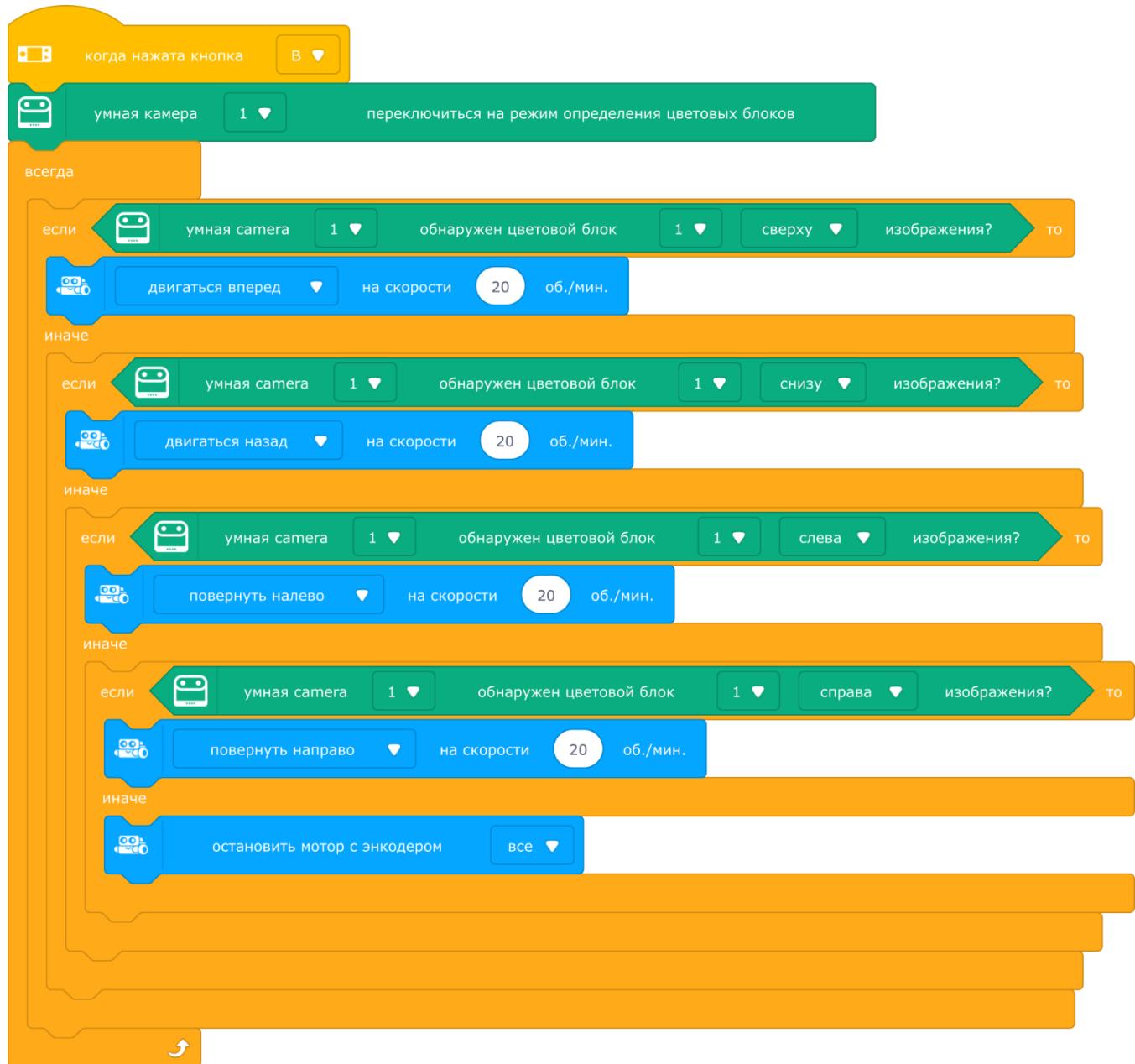
Изображение, полученное с помощью Smart Camera, можно разделить на пять областей (см. рисунок ниже). Знание того, в какой области находится цветной блок, может помочь роботу точно отслеживать его.



Для этого используется программный блок:

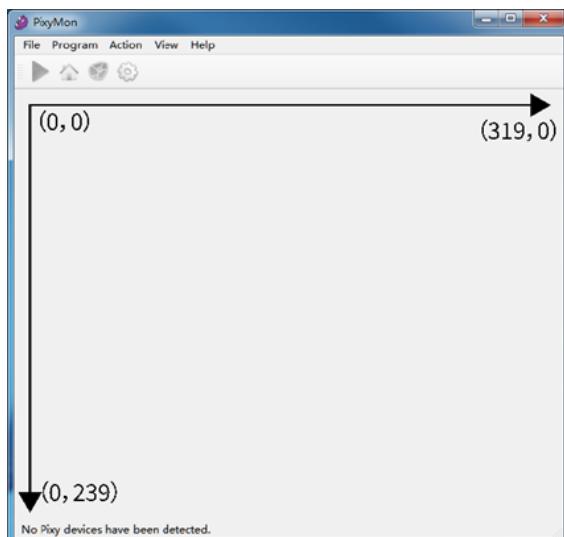


Пример программы:

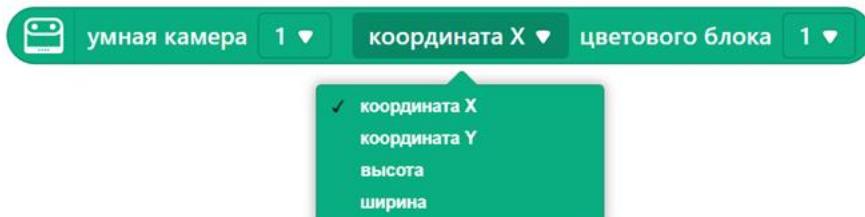


Определение координат цветовых блоков

Изображение, полученное с помощью Smart Camera, может быть представлено в ортогональной системе координат. Значения по оси X находятся в диапазоне от 0 до 319 и увеличиваются слева направо. Значения по оси Y увеличиваются сверху вниз и находятся в диапазоне от 0 до 239. Координаты X и Y однозначно определяют положение цветового блока в кадре.

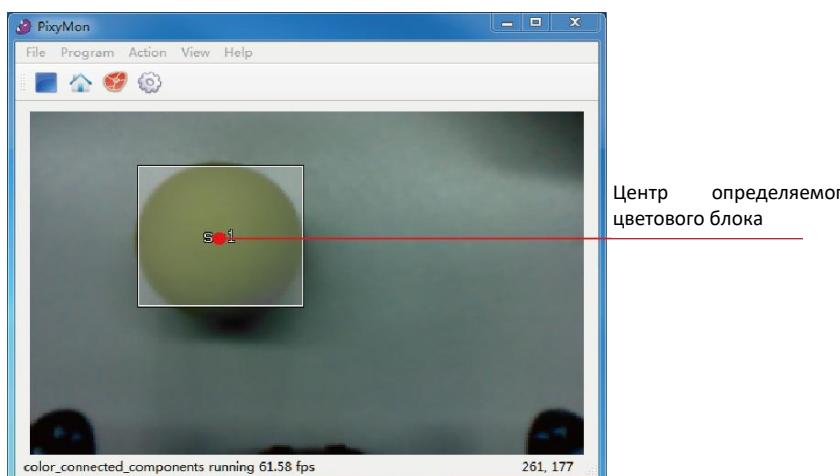


Для получения координат x и у цветового блока используется команда



Данный блок может определять не только координаты, но и высоту и ширину цветового блока.

Высота и ширина здесь — это не фактическая высота и ширина цветового блока, а высота и ширина прямоугольной рамки вокруг него. Если камера находится близко к цветовому блоку, то значения высоты и ширины будут больше.



Вы можете использовать координаты для определения положения объекта и построить алгоритм следования за объектом следующим образом: когда координата **x** больше или меньше определённого значения, mBot поворачивает влево или вправо; когда координата **y** больше или меньше определённого значения, mBot движется вперёд или назад.

Такой алгоритм не позволит организовать плавного следования за объектом, поэтому в mBlock есть еще один блок, расположенный в палитре «Особые функции умной камеры», который может помочь нам решить проблему:



Данный блок позволяет автоматически рассчитать скорость, необходимую для того, чтобы цветовой объект оставался в обозначенной области обзора камеры. Кроме того, если мы хотим, чтобы mBot более плавно отслеживал объект и следовал за ним, нам нужно установить значение коэффициента пропорциональности **Kр** для скорости двигателя.



Значение **Kр** позволяет регулировать скорость возврата к желаемому положению, иными словами, если цветовой объект сильно отклонился от необходимого положения, то скорость будет выше. Если отклонение не велико, то скорость возврата будет меньше. Диапазон значений **Kр** — от 0 до 1. Как правило, для резкого поворота требуется более высокое значение **Kр** и более высокая скорость. Однако при такой настройке mBot сильнее «заносится» на повороте.

Попробуйте экспериментировать со значением коэффициента пропорциональности и понаблюдайте за выполнением программы:

Значение Kр	Координаты	Плавность и положение
0.1	(160, 170)	Не плавно, по центру
0.3	(160, 170)	
0.4	(160, 170)	
0.5	(160, 170)	
()	(160, 170)	

Подбор значения коэффициента **Kр** осуществляется для каждого случая в отдельности и зависит от конструкции робота, типа трассы, динамики перемещения цветового блока и других параметров.



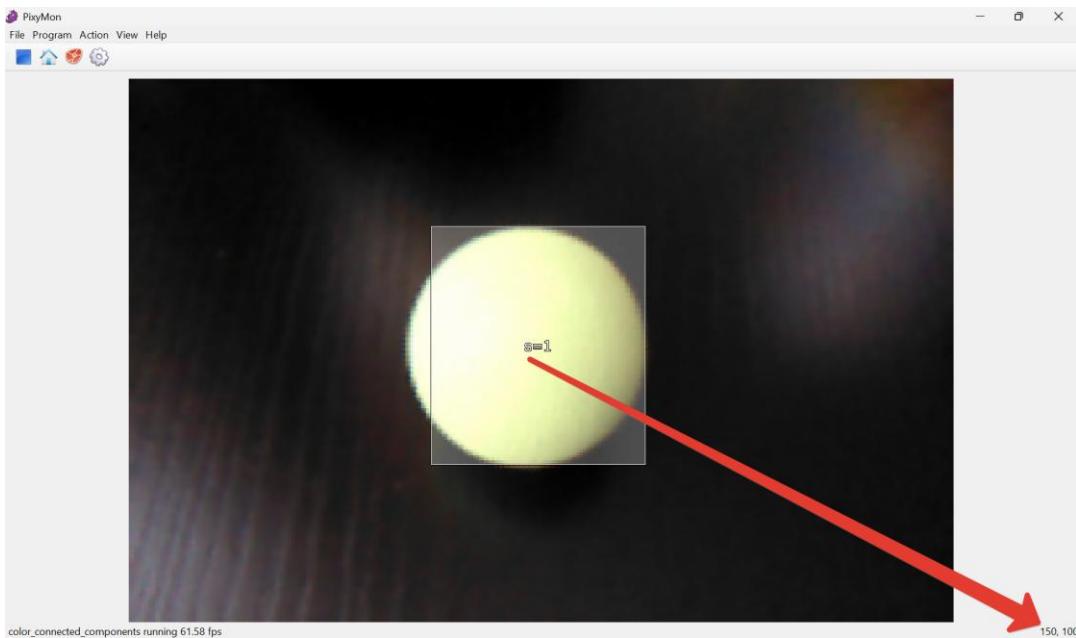
Пример



В данном примере скорость моторов будет рассчитываться автоматически в зависимости от заданных координат.

Для определения значений координат удобно использовать PixyMon:

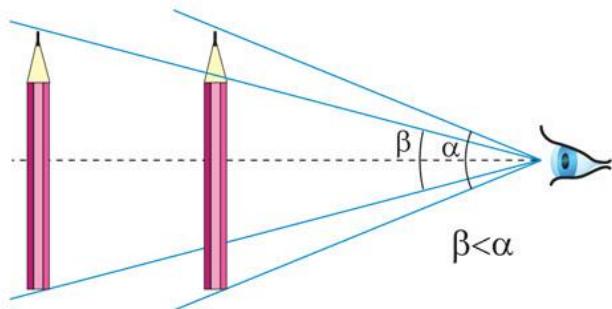
1. Разместить объект в центре окна видео.
2. Наведите курс на центр объекта.
3. Посмотрите координаты курса в строке состояния.



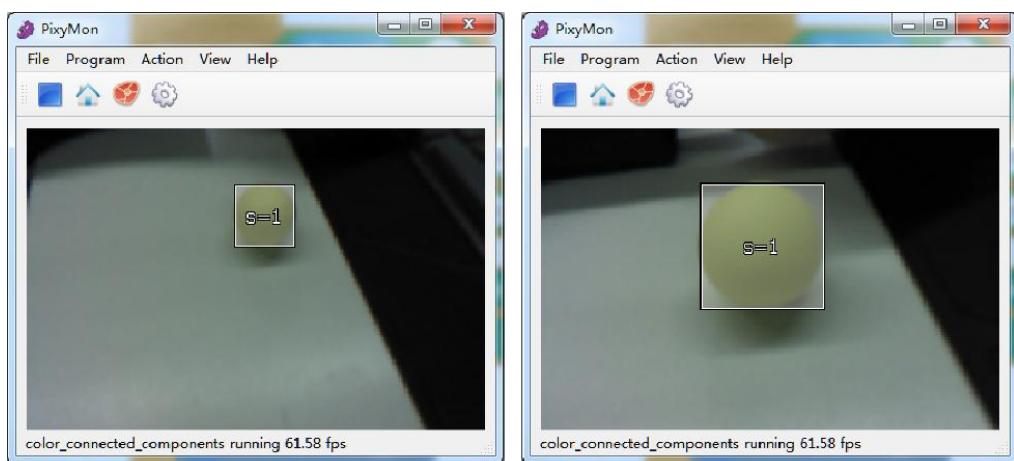
Определение расстояние до цветовых объектов

Изображение, получаемое с помощью камеры, является двумерным или плоским, поэтому модуль не может измерить расстояние до объектов. Для измерения расстояния до объектов можно осуществить на основе факта: то, что находится далеко, кажется меньше, а то, что близко, — больше.

То, что близкие объекты кажутся больше, а удалённые — меньше, — это зрительное восприятие. Когда мы подходим к чему-то, оно кажется больше из-за изменения угла зрения; когда объект находится далеко от нас, он кажется меньше. Наши глаза и два конца объекта образуют угол, называемый углом зрения. Чем дальше от нас находится объект, тем меньше будет угол зрения.

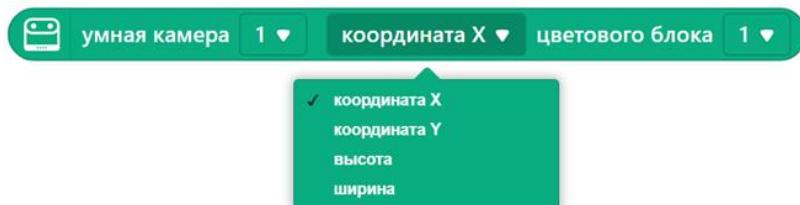


Аналогичный подход применим и к Smart камера. Поместите шар перед камерой и посмотрите, что происходит с изображениями, которые отображаются на экране PixyMon.



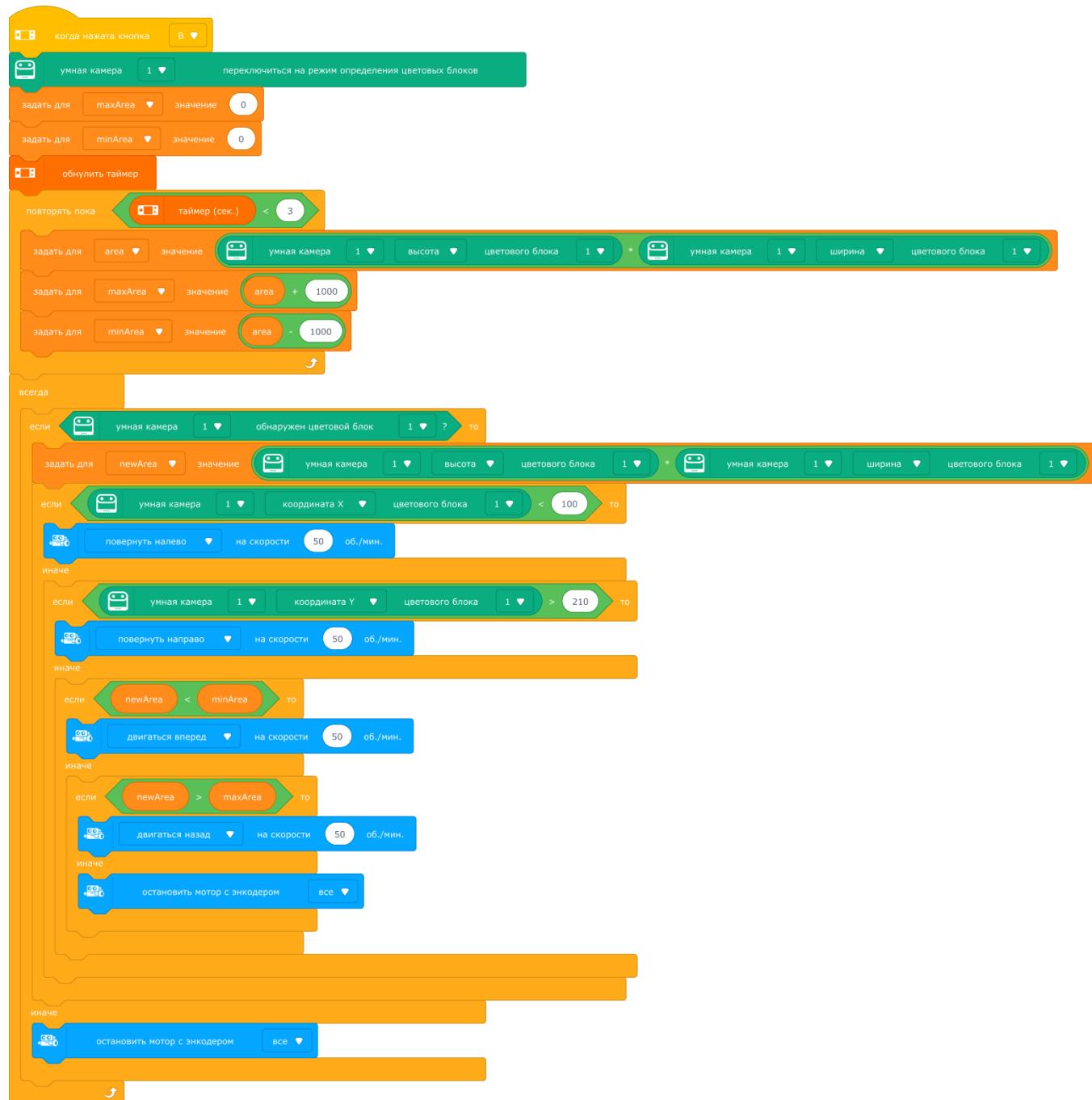
Один и тот же цветной блок кажется больше, когда вы приближаетесь к Smart Camera, и меньше, когда отдаляйтесь от модуля. Чтобы вычислить площадь прямоугольника, мы можем использовать формулу: **Площадь = Ширина × Высота**.

Для определения ширины и высоты используется блок:



Пример

Запрограммируем mBot так, чтобы он двигался вперёд или назад в зависимости от размера объекта, а также поворачивал влево или вправо в зависимости от координат.



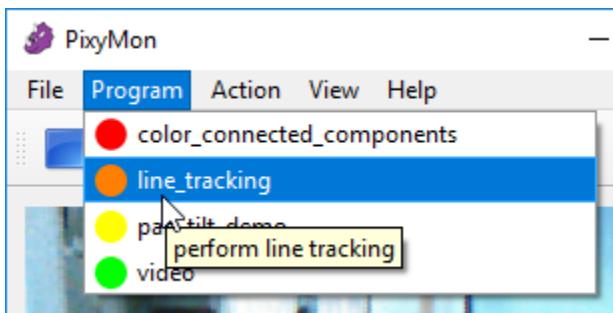
Распознавание линий

Следование по линии — одна из наиболее популярных задач в робототехнике. Большинство роботов, следующих по линии, используют дискретные фотодатчики, чтобы различать линию и фон. Этот метод лучше работает с толстыми линиями, а распознавание локализовано, что затрудняет прогнозирование направления линии или работу с пересечениями.

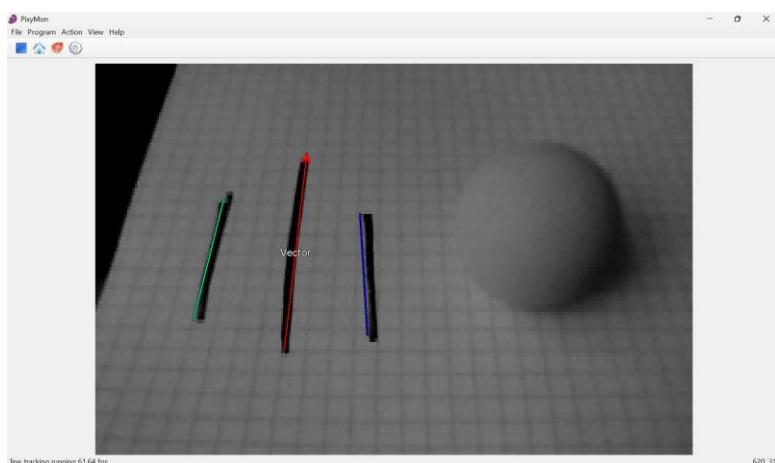
С помощью камеры можно рассмотреть общую проблему следования по линии. Во время вождения автомобиля ваши глаза воспринимают много информации о дороге, ее направлении (не приближается ли крутой поворот?) и есть ли впереди перекресток. Аналогичным образом, каждый из кадров умной камеры принимает информацию об отслеживаемой линии, ее направлении, других линиях и любых пересечениях, которые эти линии могут образовывать. Умная камера также может считывать простые штрих-коды, которые сообщают вашему роботу, что ему следует делать – повернуть налево, повернуть направо, замедлиться и т. д. Камера делает все это со скоростью 60 кадров в секунду.

Отслеживание линии через PixyMon

PixyMon отображает графически то, что обнаруживает камера, и это хороший способ ознакомиться с тем, на что способен алгоритм отслеживания линий. Начните с выбора алгоритма отслеживания линии в меню **Program**.

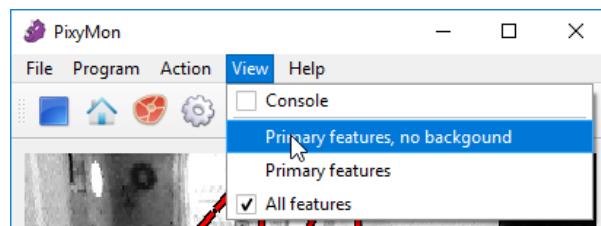


В этом режиме Smart Camera возвращает чёрно-белые изображения, поэтому она не может распознавать цветные блоки, как раньше. Нарисуйте линию на листе бумаги и поместите ее перед камерой. Карандаш подойдет, но лучше использовать черный маркер. Вы должны увидеть что-то вроде этого:

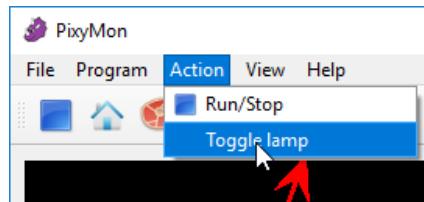


Камера нашла лучшую линию-кандидата для отслеживания и присвоила ей вектору. Вектор — это линия, которую камера в данный момент отслеживает. Если в кадре появляются новые линии, робот не переключается и не начинает следовать по этим линиям. Вы можете переместить бумагу и увидеть, что камера может довольно быстро отслеживать исходную линию (вектор). Камера отслеживает каждую линию, определяя, куда она движется в последующих кадрах, поэтому текущая линия следования останется таковой, пока не будет указано иное.

Стоит отметить, что ПО PixyMon замедляет алгоритм отслеживания обычно до 20-30 кадров в секунду, поскольку отправка необработанных данных кадра через USB занимает некоторое время, но, если вы измените вид на «без фона» (см. рисунок ниже), вы можете почувствовать, как выглядит алгоритм отслеживания при исходной частоте 60 кадров в секунду.

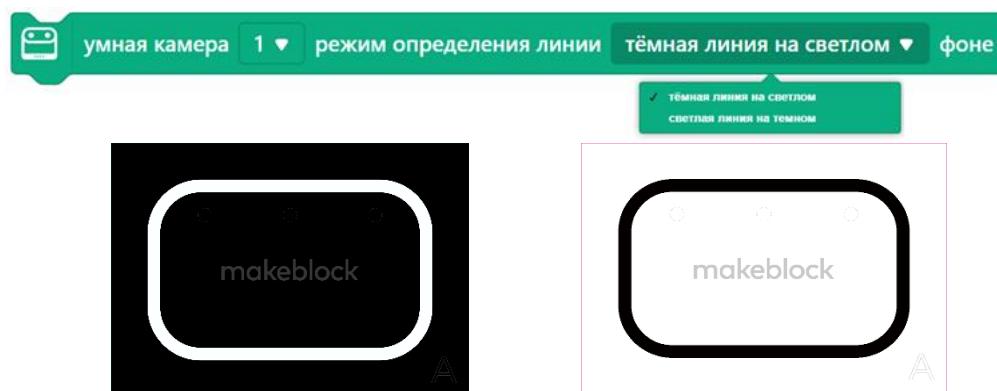


Вы также можете увидеть, что происходит, когда вы включаете подсветку камеры (**Toggle lamp**).



При включенной подсветке камера уменьшает время экспозиции (выдержки) изображения, что уменьшает размытость изображения. В общем, размытие в движении происходит потому, что камера снимает объект (в данном случае линию), пока он движется. Это может затруднить обнаружение и отслеживание линий. Включение подсветки камеры может снизить размытость изображения и помочь роботу двигаться быстрее.

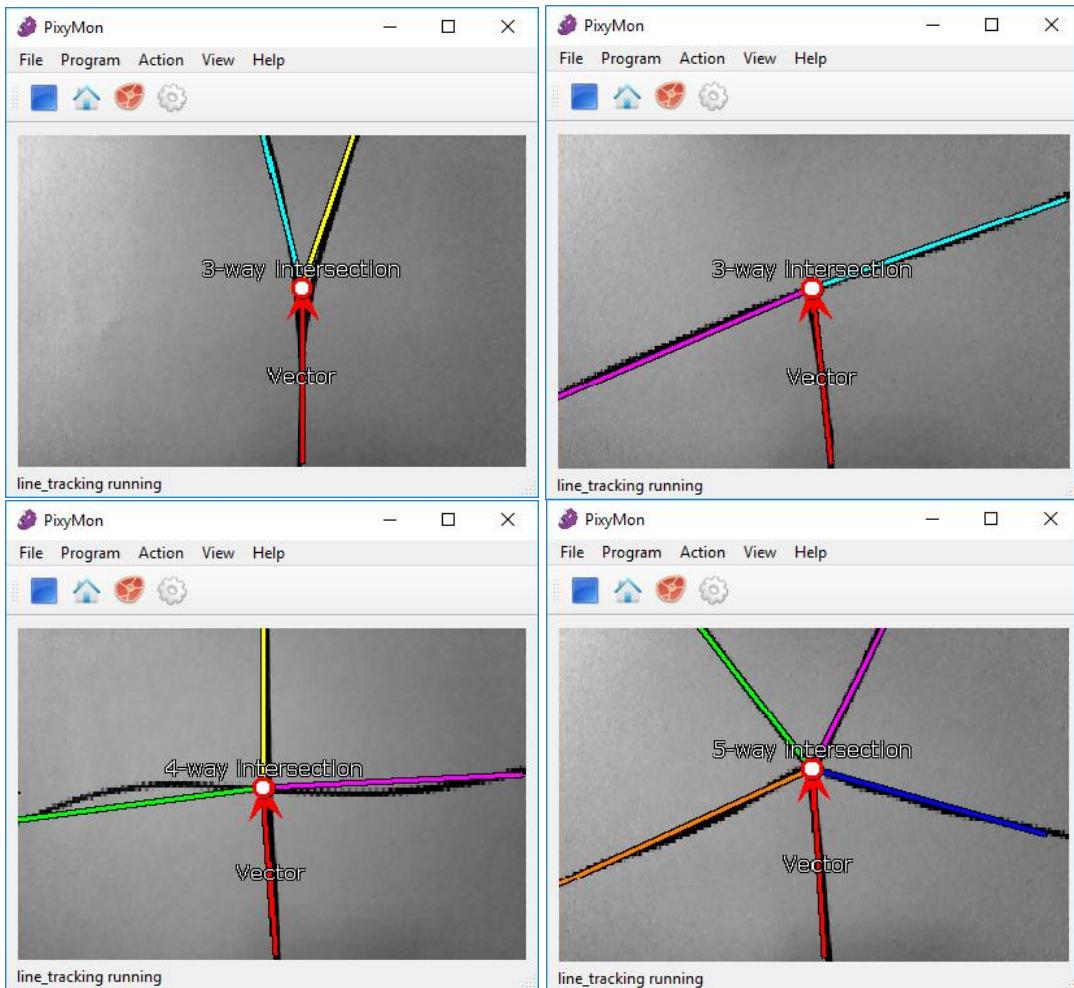
Алгоритмы умной камеры могут обнаруживать темные линии на светлом фоне и наоборот. Это можно настроить как в конфигурации программы PixyMon, так и используя блоки в среде mBlock5.



Вы также можете сообщить камере, что вас интересуют линии только в определенном диапазоне ширины. **Настройки → Tuning →Maximum | Minimum line width.**

Обнаружение пересечений и «ветвлений»

Перекрестки могут оказаться сложной задачей для робота, следующего по линии, поскольку они могут принимать самые разные формы. Будет ли ваш робот хорошо справляться с Т-образными перекрестками, или потерпит неудачу, когда увидит У-образные перекрестки? А как насчет четырех- или даже пятисторонних перекрестков? Что, если пересечения и линии нарисованы от руки или «извилистые»?



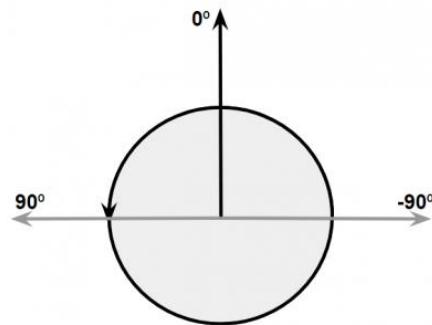
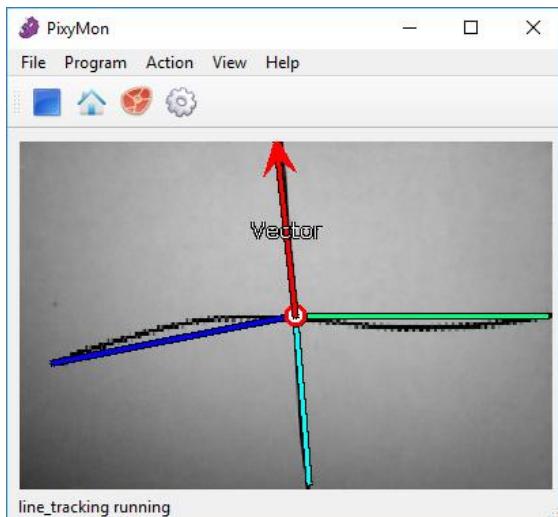
Камера поможет справиться и с этими случаями. Например, умная камера сообщает вашей программе, что она обнаружила впереди пересечение, сколько ветвей находится на перекрестке и углы каждой ветви. Затем ваша программа может решить, какую ветвь она хочет использовать, а затем сообщить об этом камере. После этого камера сделает ветку своим новым вектором и продолжит его отслеживание.



Внимание

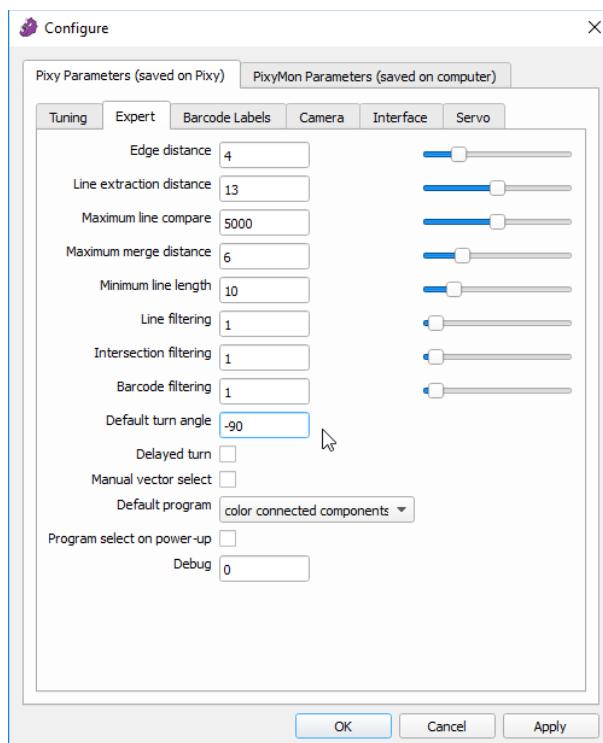
Чтобы программа PixyMon помечала такие перекрестки, обязательно установите параметр «Отложенный поворот» на вкладке конфигурации Expert (Эксперт), иначе камера немедленно выберет поворот/ветвь, и пересечение больше не будет актуальным и фактически исчезнет в программе PixyMon.

Нарисуйте четырехстороннее пересечение, проведя еще одну линию через исходную линию и поместив ее перед камерой. Камера определит перекресток и быстро выберет ветвь перекрестка по умолчанию, которая является «самой прямой» относительно предыдущего вектора движения.

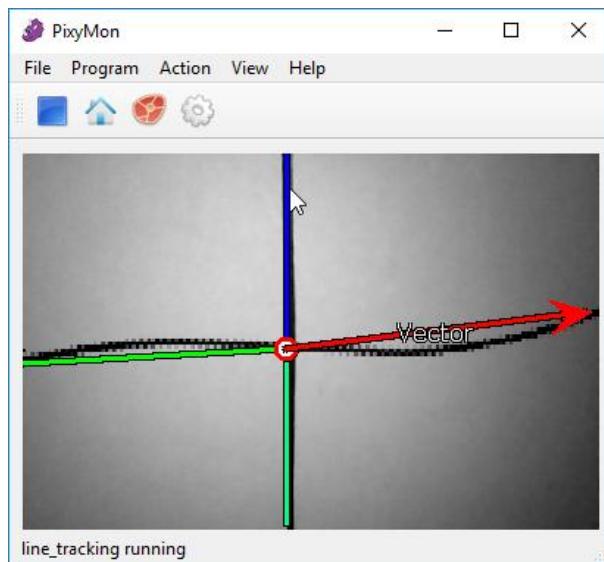


Вы можете изменить «угол поворота по умолчанию» на вкладке «Экспертная настройка». Вы увидите, что угол поворота по умолчанию равен 0 (по умолчанию), то есть прямой. Угол основан на схеме размещенной на рисунке выше.

Попробуйте изменить угол поворота по умолчанию на -90 (крайний правый поворот/ветвь) и нажмите «Применить».



Теперь, когда вы размещаете перекресток перед камерой, он выберет самую правую ветвь перекрестка.



По умолчанию камера автоматически выберет ветку на основе угла поворота по умолчанию. Вы можете изменить это поведение, включив «отложенный поворот» на вкладке «Экспертная настройка». Включите задержку поворота (**Delayed turn**) и нажмите «Применить». Теперь, когда вы размещаете пересечение перед камерой, она не будет выбирать ветку автоматически. Вместо этого камера пометит перекресток, потому что она не знает, какое действие предпринять.

Configure Dialog (Left):

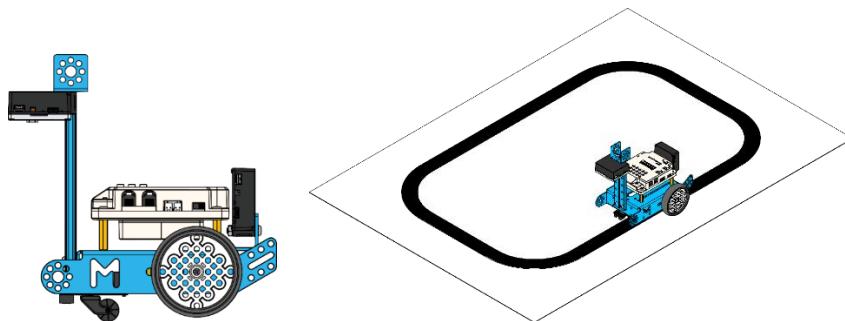
- Pixy Parameters (saved on Pixy) tab is selected.
- Tuning tab is active.
- Delayed turn checkbox is checked (indicated by a checked box).
- Other parameters include Edge distance (4), Line extraction distance (13), Maximum line compare (5000), Maximum merge distance (6), Minimum line length (10), Line filtering (1), Intersection filtering (1), Barcode filtering (1), Default turn angle (0), and Program select on power-up (unchecked).

PixyMon Application (Right):

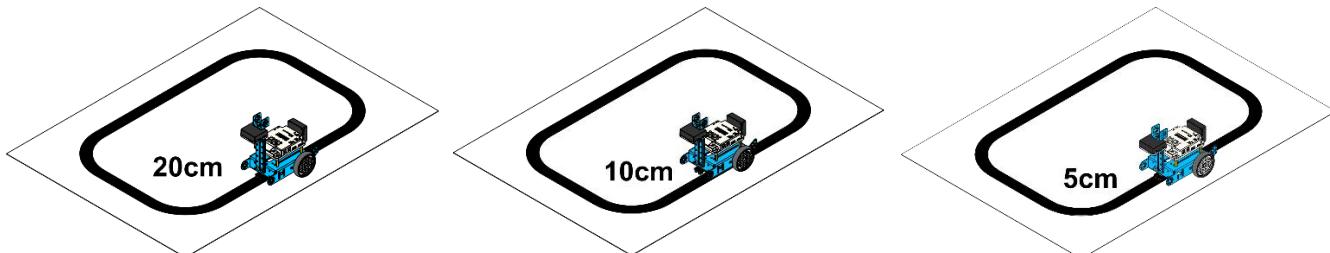
- File, Program, Action, View, Help menu bar is present.
- Tool icons: File, Home, Stop, Settings.
- Camera feed showing a 4-way intersection with a red circle highlighting the center.
- A red vector line points from the center of the intersection towards the top-right.
- The text "line_tracking running" is at the bottom.

Алгоритм следования по линии

В среде mBlock5 есть специальные блоки для следования по линии, которые автоматически рассчитывают необходимую скорость для моторов. Возьмите любую карту с черной линией на белом фоне или нарисуйте ее самостоятельно.

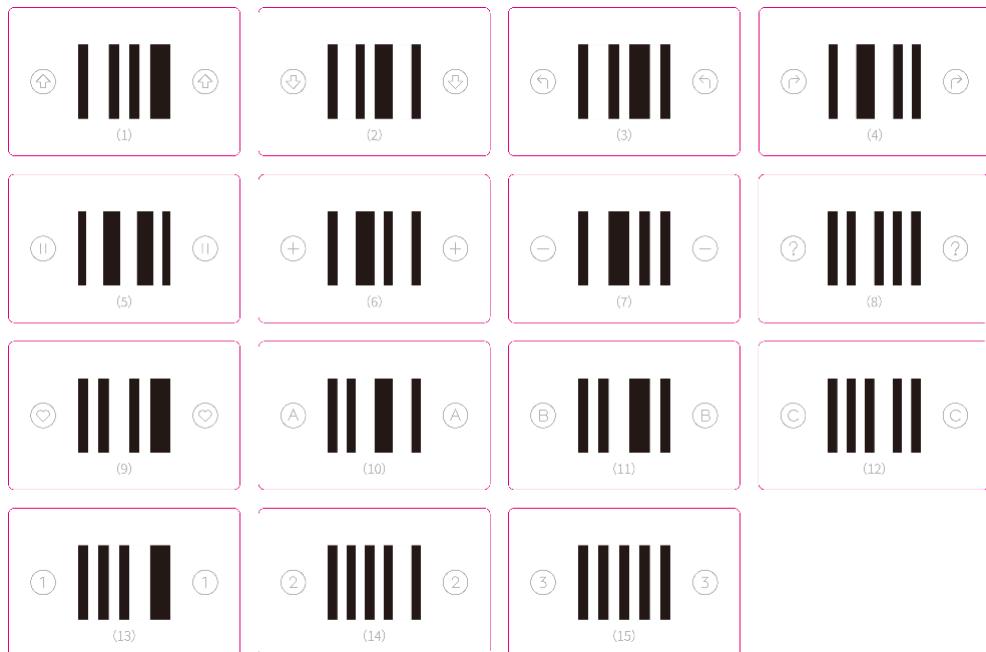


Высота, на котором установленна камера, напрямую влияет на качество отслеживания линии. Поэкспериментируйте с различной высотой расположения камеры, и выберите вариант, который позволяет достичь максимальной достоверности обнаружения.



Распознавание штрих-кодов

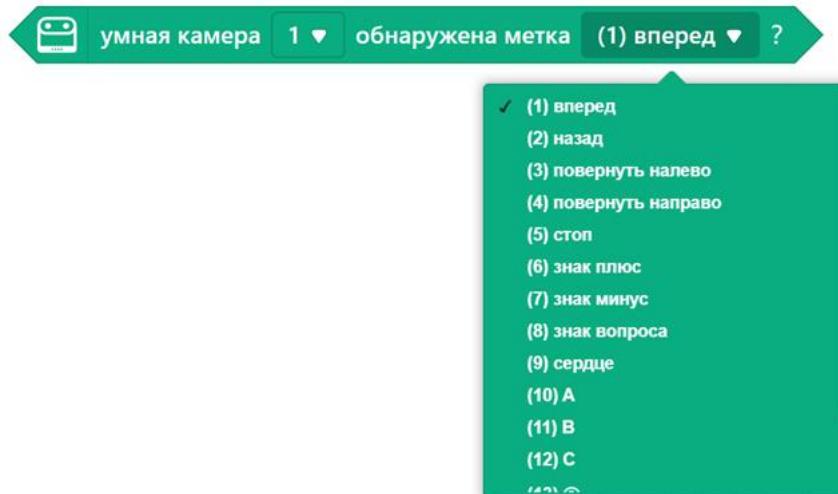
Камера может получать простые команды или информацию во время отслеживания линии, поэтому в ее функционал добавлена функция распознавания штрих-кода. Штрих-коды призваны действовать как «дорожные знаки». Вы можете добавлять к штрих-кодам графику, удобочитаемую для человека, при условии, что штрих-код не загораживается. В комплект Smart Camera Add-on Pack входят 15 наклеек, пронумерованных от 1 до 15.



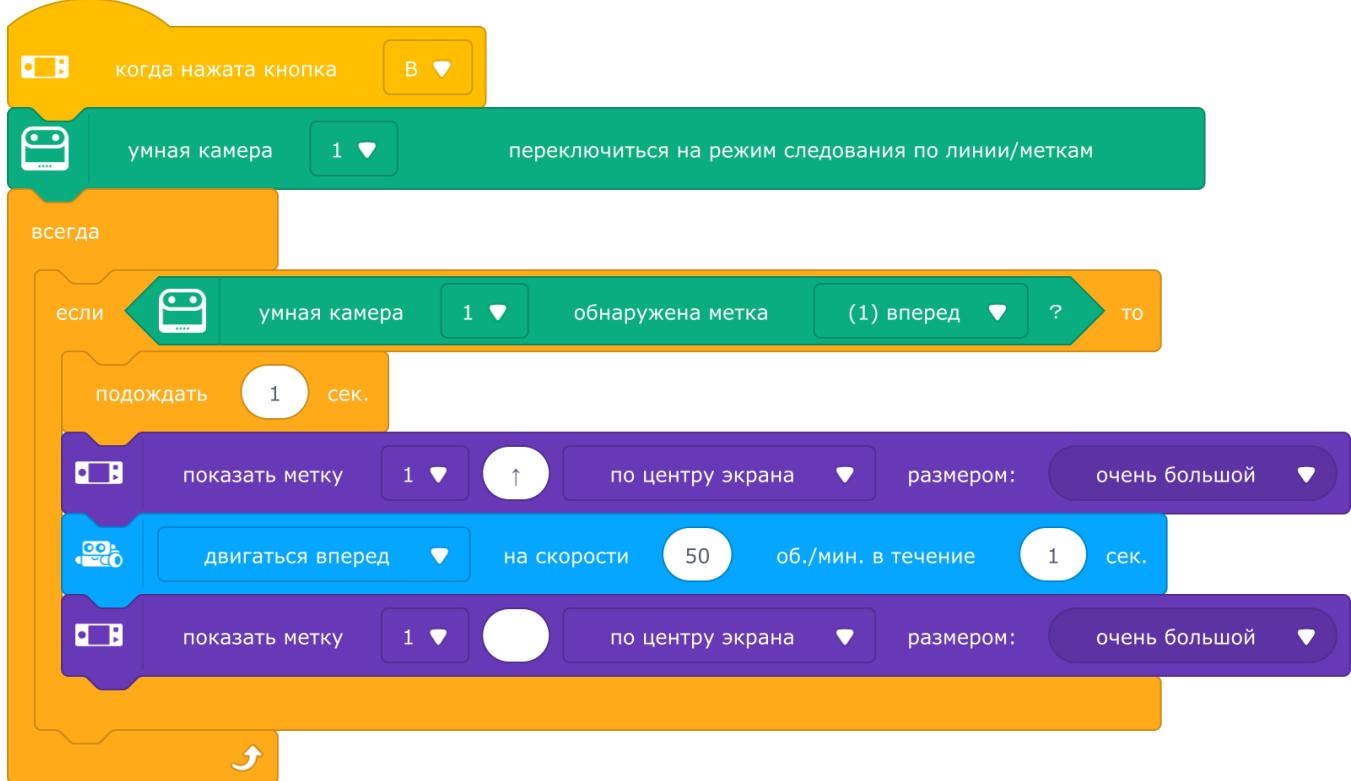
Прежде всего необходимо перевести камеру в режим распознавания линий и штрихкодов, используя блок:



Вы можете запрограммировать робота на совершение действий в зависимости от того, какой штрихкод распознает камера, используя блок:

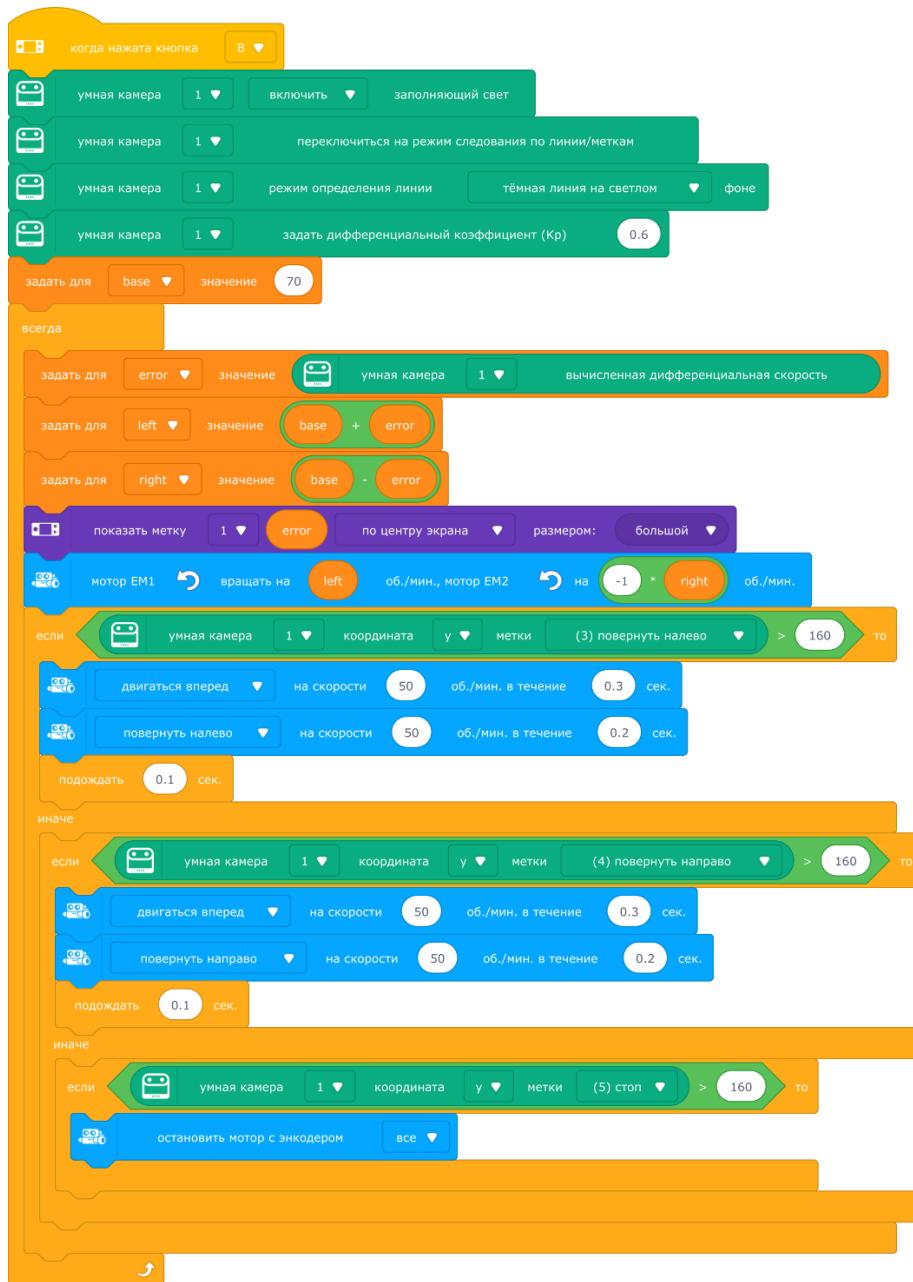
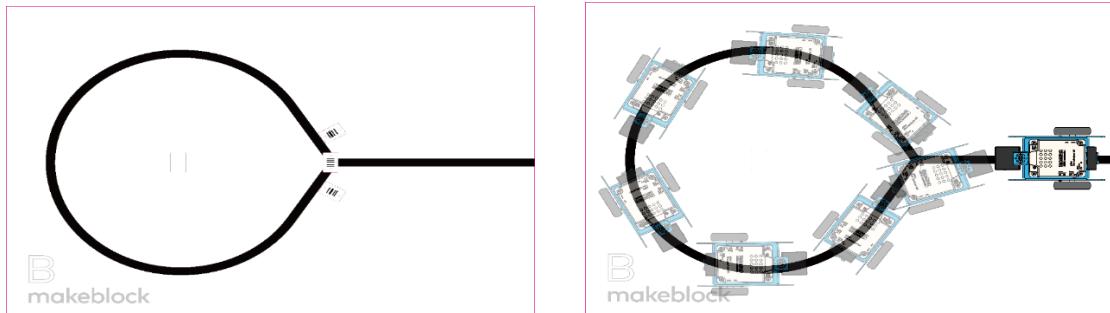


Пример



Сочетание следования по линии с распознаванием штрих-кодов

При движении по линии с перекрестками камера сама выбирает линию, которой будет следовать, однако вы можете настроить выбор ответвления, используя штрих-коды из комплекта.



```
when green flag clicked
  умная камера [1] включить заполняющий свет
  умная камера [1] переключиться на режим следования по линии/меткам
  умная камера [1] режим определения линии [тёмная линия на светлом фоне]
  умная камера [1] задать дифференциальный коэффициент (Kр) [0.6]
  задать для [base v] значение [70]
  всегда
    задать для [error v] значение [умная камера [1] вычислена дифференциальная скорость]
    задать для [left v] значение [base + error]
    задать для [right v] значение [base - error]
    показать метку [1] [error] по центру экрана размером [большой]
    мотор EM1 [v] вращать на [left] об./мин., мотор EM2 [v] на [-1 * right] об./мин.
    если [умная камера [1] координата [у] метки [(3) повернуть налево > 160] то
      двигаться вперед [50] об./мин. в течение [0.3] сек.
      повернуть налево [50] об./мин. в течение [0.2] сек.
      подождать [0.1] сек.
    иначе
      если [умная камера [1] координата [у] метки [(4) повернуть направо > 160] то
        двигаться вперед [50] об./мин. в течение [0.3] сек.
        повернуть направо [50] об./мин. в течение [0.2] сек.
        подождать [0.1] сек.
      иначе
        если [умная камера [1] координата [у] метки [(5) остановка > 160] то
          остановить мотор с энкодером
    конец
  конец
```