

# Школа LXF

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

## Взаимодействуем с окружением, или И роботы чувствовать умеют

**Татьяна Казанцева** продолжает рассказывать о робототехнике, которая обходится без реальных роботов.



**Наш эксперт**

**Татьяна Казанцева** в свободное время от корпения над написанием методики скрещивания Scratch и Arduino оттачивает навыки работы со свободным ПО для использования в школе и дома.

Сегодня речь пойдет о датчиках, которые позволят нашему роботу ориентироваться в пространстве. В прошлом выпуске Школы (LXF156) мы начали разбираться, как можно изучить основы робототехники в условиях, когда реальные роботы недоступны. Мы научили робота двигаться, разобрались, как им можно управлять, и даже учли возможности внесения искажений в передаваемые команды.

Итак, мы научим виртуального робота взаимодействовать с окружением, оснастив его датчиками (конечно, тоже виртуальными), которые будут отражать реально существующие компоненты при построении реальных роботов.

### Коснись меня

Самый простой датчик, которым оснащают роботов, это датчик касания (или бампер). Такие датчики позволяют роботу определять наличие препятствий по ходу движения, прикоснувшись к ним. Самый простой датчик – это датчик касания, расположенный по ходу движения робота. Для демонстрации движения нам понадобится также виртуальный лабиринт, по которому наш робот будет разъезжать.

Робот в нашем случае будет представлять собой круглую платформу, спереди которой располагается датчик касания. Сразу

стоит отметить, что рисовать датчики надо на правой стороне робота, так как именно туда смотрит направление движения 0.

Также нам понадобится поле для его запуска, которое в нашем случае будет иметь вид пространства, окруженного сплошным забором.

Алгоритм робота будет следующим:

- 1 Робот едет вперед, пока не коснется забора
- 2 Если срабатывает датчик касания, то робот отъезжает назад и поворачивается на 90 градусов по часовой стрелке.
- 3 Цикл повторяется.

А программа будет такой:

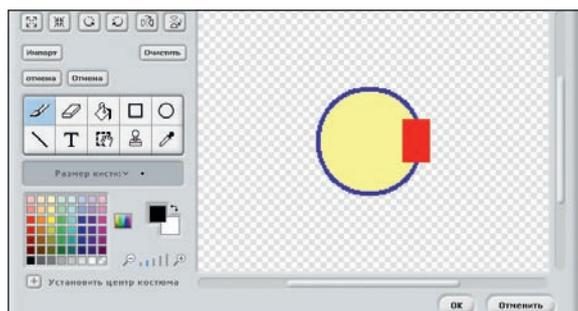


Запускаем мы робота щелчком по нему (это позволяет нам поставить его в любую точку поля).

Как видно, мы используем цвет датчика (красный) для определения соприкосновения с цветом стены (синий). Это позволит нам в дальнейшем использовать несколько датчиков.

Усложним задачу. Используем лабиринт, показанный на рис. 2.

Поставив робота в его начало и запустив, вы можете обнаружить, что он застрял в средней его части и не может выбраться. Что же делать? Попробовать изменить поворот в направлении против часовой стрелки? Тогда робот будет постоянно возвращаться назад.



► Рис. 1. Вот так будет выглядеть наш робот.

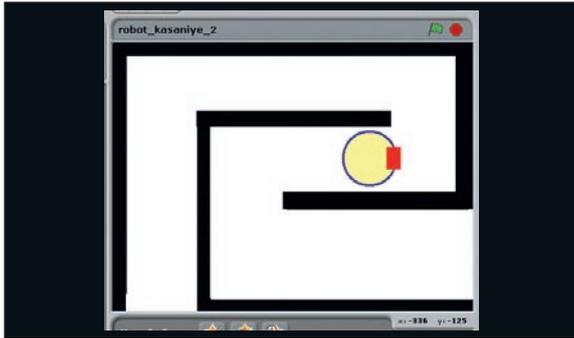


Рис. 2. Заведем нашего робота в лабиринт, и пусть он попробует добраться до упора.

Можно сделать следующее:

- 1 Столкнувшись со стенкой, робот поворачивается вправо и пробует поехать.
  - 2 Если он снова упирается в стенку за минимальное число движения (к примеру, на величину отъезда, умноженную на два), то он поворачивается на 180 градусов.
- Видоизмененный код программы будет выглядеть теперь так.



Мы добавили еще стоп-линию зеленого цвета (нарисуйте ее в конце лабиринта) и звуки при соударении со стенкой и успешном прохождении лабиринта (рис. 3):

Позэкспериментируйте с разными лабиринтами и алгоритмами.

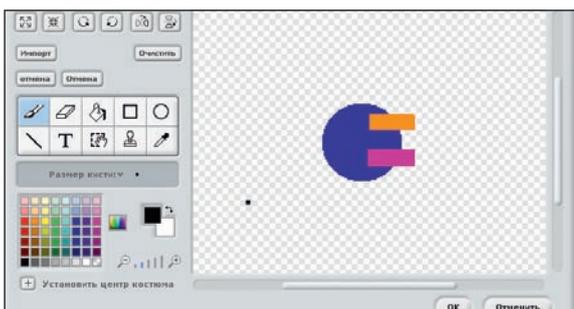


Рис. 4. Новый робот — чуть меньше, зато с двумя датчиками.

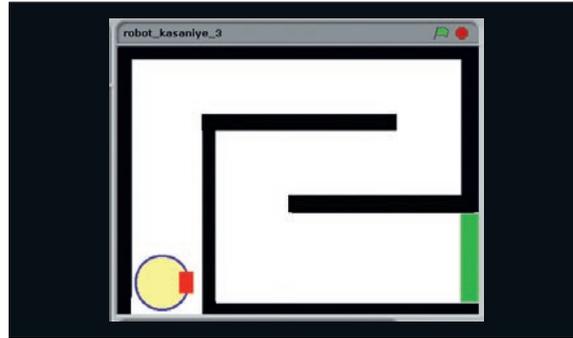


Рис. 3. Наш робот теперь сумеет остановиться при достижении конца лабиринта.

## Усложним задачу

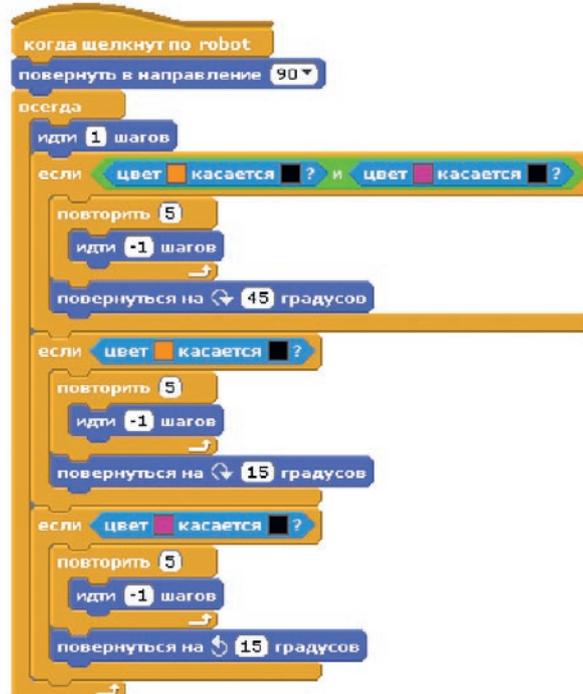
Теперь усложним нашего робота и поставим на него два датчика — справа и слева по движению, и пустим робота ездить по полю с разбросанными кубиками (рис. 4).

Так как у нас теперь два датчика касания, то мы сможем точнее позиционировать робота и определять нахождение препятствий.

Но и алгоритм работы также усложнится:

- 1 Робот движется вперед.
- 2 При срабатывании датчика касания робот отъезжает назад.
- 3 Если сработал правый датчик, то мы поворачиваемся на небольшой угол (10–20 градусов) влево.
- 4 Если сработал левый датчик, то мы поворачиваемся вправо.
- 5 Если сработали оба датчика, то мы поворачиваемся вправо на 45 градусов.

В блоках Scratch это будет выглядеть так:



Запустив программу, вы через некоторое время можете столкнуться с тем, что ваш робот упрется в угол и не сможет из него выйти. Это связано с ограничением алгоритма, который при переменном касании датчиков поворачивается на определенный угол то вправо, то влево.

Выйти из положения можно с помощью увеличения угла поворота при последовательных касаниях или просто задавая угол случайным числом (от 10 до 20 градусов будет достаточно).

Тогда программа примет следующий вид:

»

```

когда щелкнут по robot
  повернуть в направление 90
  всегда
    идти 1 шагов
    если цвет [касается ?] и цвет [касается ?]
      повторить 5
        идти -1 шагов
        повернуться на 45 градусов
    если цвет [касается ?]
      повторить 5
        идти -1 шагов
        повернуться на [выдать случайное от 10 до 20] градусов
    если цвет [касается ?]
      повторить 5
        идти -1 шагов
        повернуться на [выдать случайное от 10 до 20] градусов
  
```

### Да будет свет

Следующий датчик, который мы рассмотрим и попытаемся смоделировать, будет датчиком освещения. В реальности это обычно фоторезистор или фототранзистор, который, реагируя на уровень света, меняет свою характеристику (к примеру, сопротивление). За счет преобразования в Scratch будет передан сигнал, который будет изменяться от 0 до 100, где 0 – минимальный уровень освещения, 100 – максимальный.

Источник света у нас будет реализован в виде спрайта конуса освещенности, который можно будет передвигать. Работа может быть взята из первого опыта, разместив источник света посередине спереди.

Итоговое поле будет выглядеть, как показано на рис. 5.

Алгоритм работы будет такой:

- 1 После включения робот начинает крутиться на месте, ища источник света.
- 2 Если он его находит, то начинает ползти к нему.
- 3 Как только уровень сигнала падает, робот опять начинает крутиться на месте.

А программа получится такая:

```

когда щелкнут по Robot
  всегда
    если цвет [касается ?]
      идти 1 шагов
    или
      повернуться на 15 градусов
  
```

Вы можете самостоятельно доработать программу и использовать два датчика света, как и в случае датчиков касания. Также вы можете ввести переменную, чтобы смоделировать изменение уровня освещенности при перемещении от источника или к нему. Это позволит роботу двигаться в правильном направлении, так как в текущем алгоритме он движется только вперед и может уехать в обратном от источника света направлении.

### Что подо мной

Последний датчик, который мы рассмотрим в данной статье – датчик линии. По сути, это тот же датчик света, но работаю-

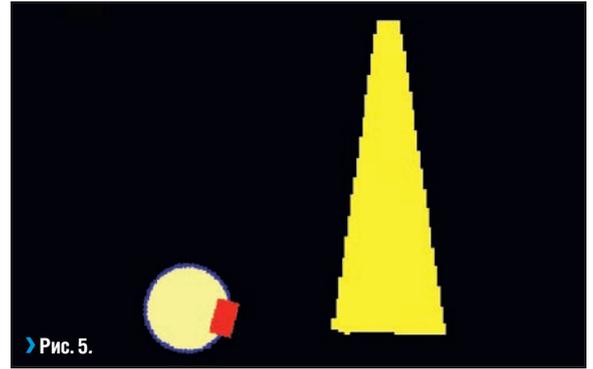


Рис. 5.

щий на отражение. Обычно датчик линии работает в невидимом инфракрасном диапазоне и позволяет определять, черная или белая поверхность находится под ним.

Алгоритм работы датчика линии аналогичен датчику касания, с той разницей, что сигналом будет не касание стенки, а нахождение черной линии, по которой должен двигаться робот.

Поэтому возьмите исходные данные от первого примера, замените поле, по которому будет двигаться робот, на поле с линией, и уменьшите размер датчика до небольшой точки (рис. 6).

Теперь модернизируйте алгоритм. В случае, когда робот не находит линию, он должен вращаться. То есть условие должно быть инверсным, и мы должны только повернуться на небольшой угол:

```

когда щелкнут по Robot
  повернуть в направление 0
  всегда
    идти 1 шагов
    если не цвет [касается ?]
      повернуться на 15 градусов
  
```

Увы, но наш робот может двигаться только по линии, которая имеет правые повороты. Для движения по линии произвольной формы нужно использовать два датчика, которые позволят также останавливать робота на горизонтальных черных линиях.

Видоизмените алгоритм самостоятельно, используя за основу рассмотренный ранее алгоритм движения робота с двумя датчиками касания, и заставьте робота ездить по произвольному пути.

На этом мы закончим наш мини-цикл. Вы можете узнать больше об робототехнике, использовании реальных микроэлектронных и робототехнических платформ Scratchduino, а также всю информацию, как самостоятельно построить их, методику обучения и многое другое на вики проекта Edumandriva по адресу <http://edumandriva.ru/wiki/index.php/ScratchDuino>. LXF

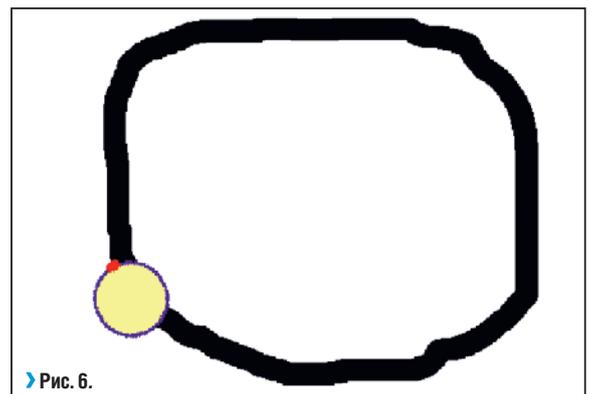


Рис. 6.

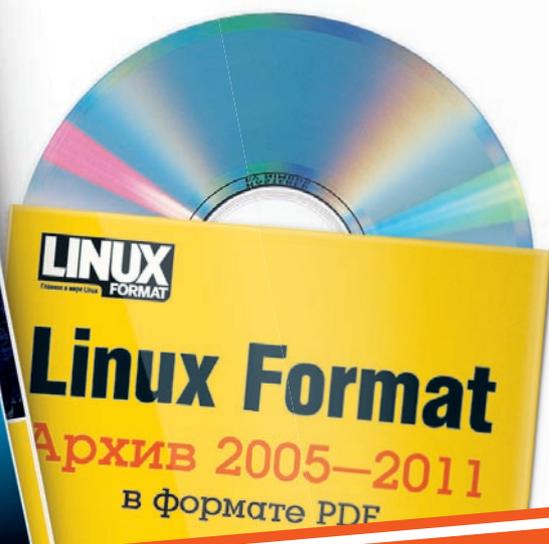
# ПОДПИШИСЬ НА ЖУРНАЛ LINUX FORMAT!

2012

**LINUX**  
FORMAT  
Главное в мире Linux



Оформи в редакции  
подписку на печатную  
версию журнала  
и получи в подарок  
диск с архивом номеров,  
а также подписку  
на электронную версию  
издания в формате PDF.



**Стоимость подписки** Годовая — 2160 руб., полугодовая — 1170 руб. без учета стоимости доставки.

**Адреса и телефоны редакции** Санкт-Петербург, Лиговский пр., 50, корп. 15, тел. (812) 309-06-86.  
Москва, Красноказарменная ул., 17, тел. (499) 271-49-54.

**Варианты доставки** Почтой по России простой бандеролью — журнал доставляют прямо в почтовый ящик  
» Почтой по России заказной бандеролью — в почтовый ящик приходит извещение, номера выдают на почте » Курьером  
«ГНУ/Линуксцентра» по Москве и Санкт-Петербургу » Курьерской службой СПСР по России » В виде PDF-файлов  
для подписчиков электронной версии.

[shop.linuxformat.ru](http://shop.linuxformat.ru)