

codey rocky + mblock 5 Краткое руководство



Версия документа: v1.1

Дата: 2018/2/2

Автор: Vreloom

Поддержка программного обеспечения и версии: mblock5 Alpha.8

Скачать

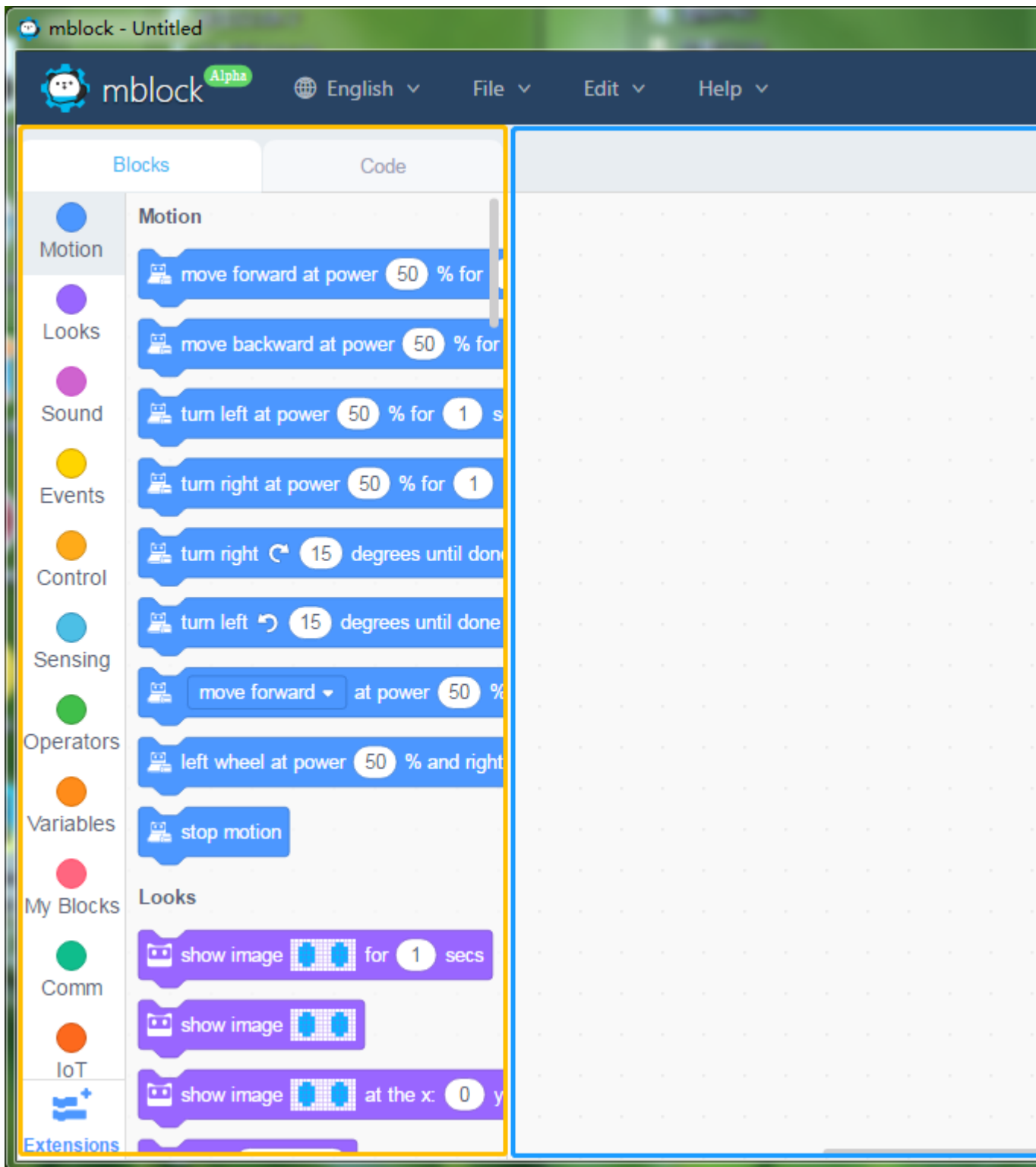


mblock 5 Alpha

Для последней версии перейдите на следующий веб-сайт:

<http://www.mblock.cc/software/mblock/mblock5/>

Интерфейс и глоссарий

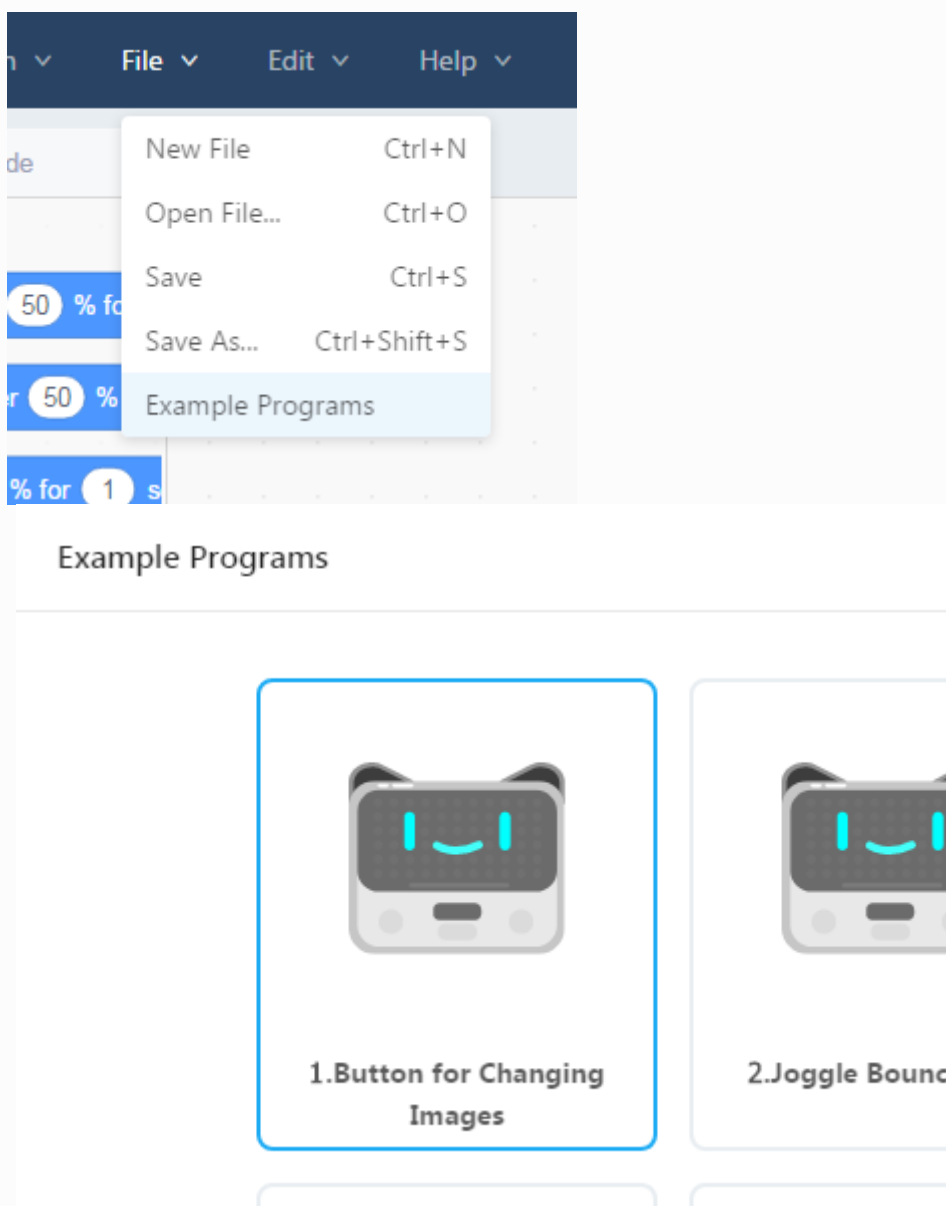


Область блоков, площадь скриптов, площадь сцены

- Область блоков: область, где предусмотрены блоки, доступные для программирования. Вы можете найти любые блоки, которые вам нужны по категориям и цветам.
- Scripts Area: область, где блоки можно перетаскивать здесь, чтобы писать программы.

- Область сцены: в этой области вы можете отображать свои работы, подключать устройство, загружать на устройство, добавлять спрайты и фон и т. Д.

Примеры программ

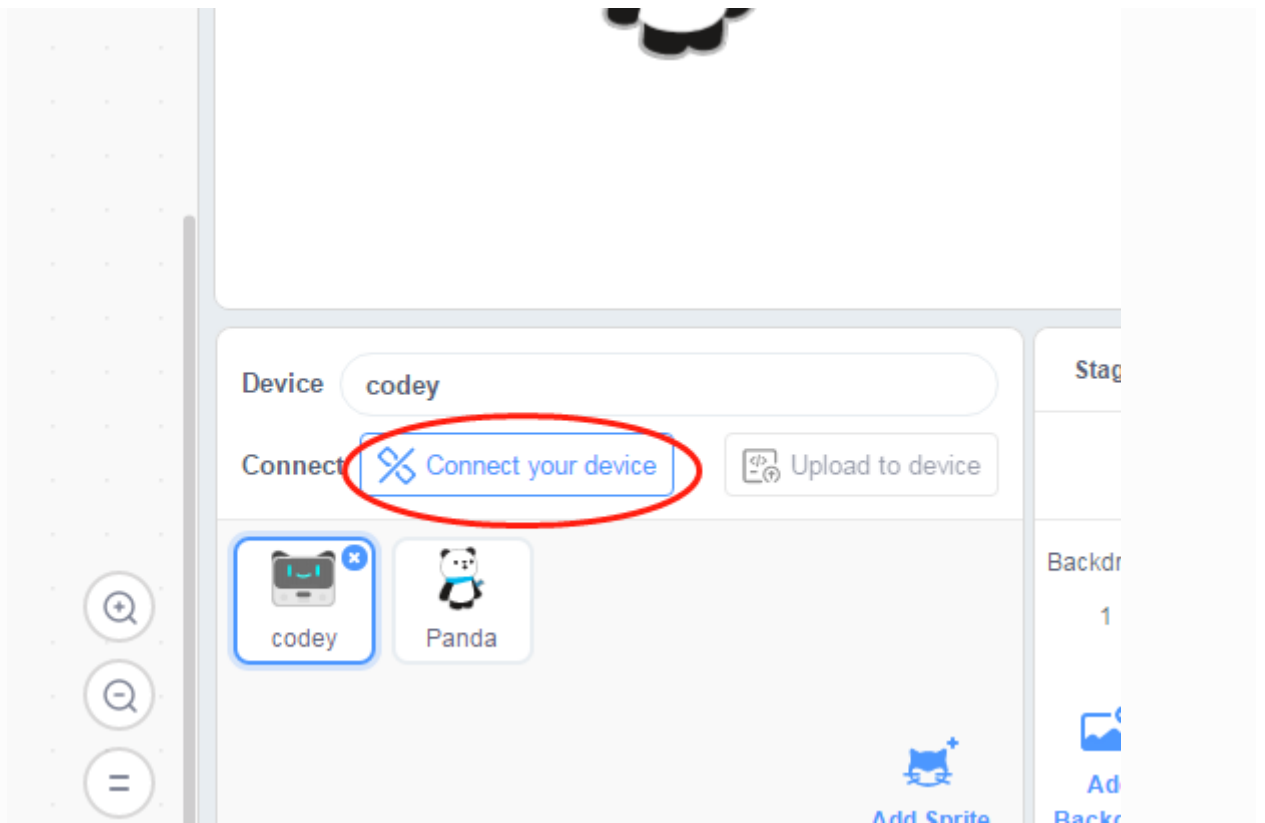


Alpha.8 имеет 12 новых примеров программ. Некоторые из проектов в этом документе можно найти в примерах программ.

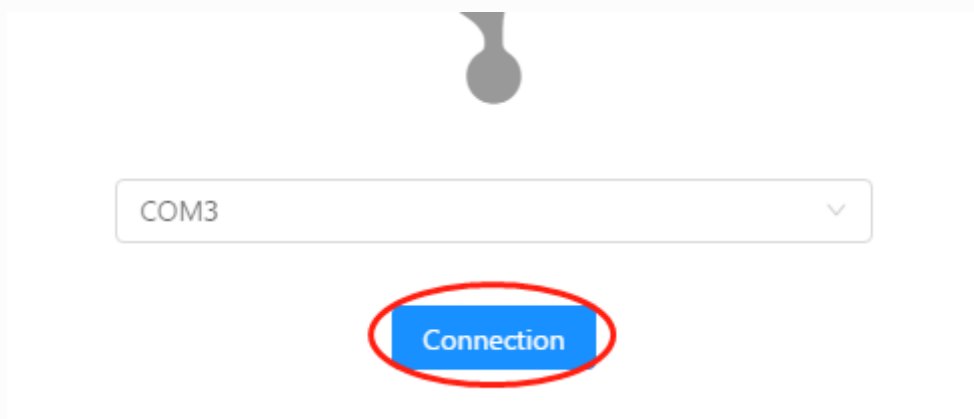
Подключение устройства

Во-первых, включите свой кодекс. Затем подключите код с компьютером через USB-кабель или адаптер Bluetooth.

1. Запустите программный mblock. Нажмите «Подключить устройство».



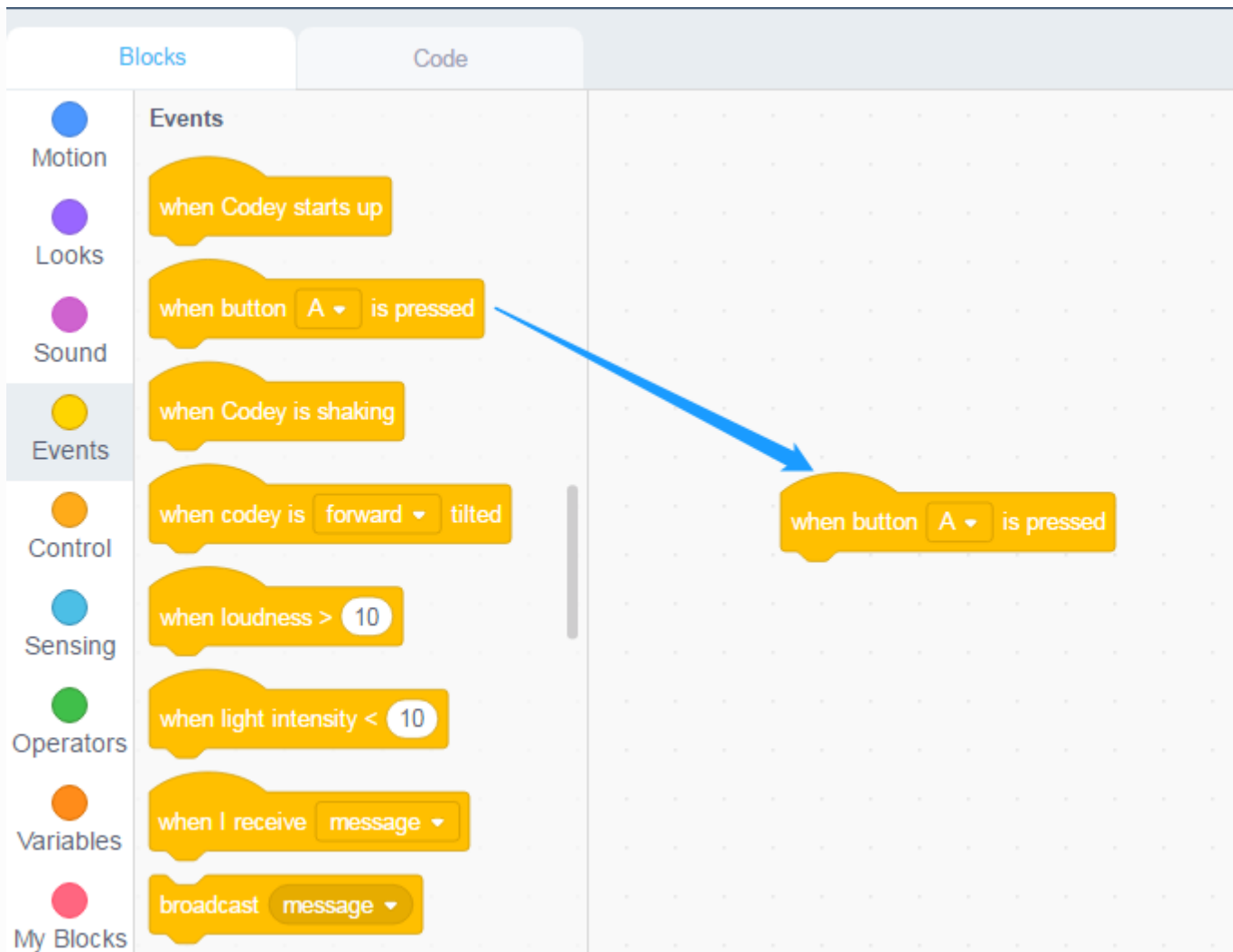
1. Нажмите Connect.



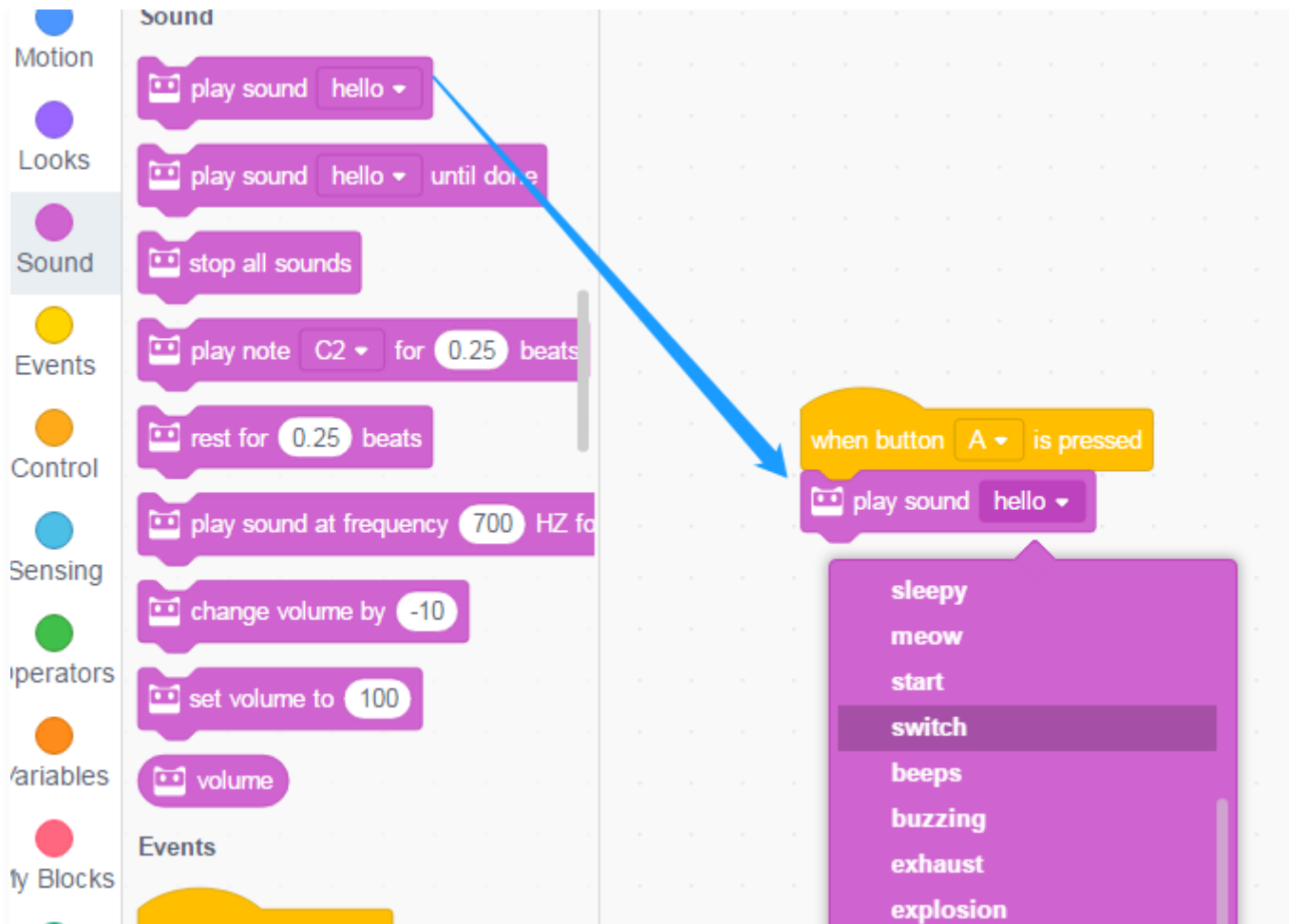
Примечание. COM3 здесь обозначает номер последовательного порта, и он может отличаться от разных систем и компьютеров. Во всяком случае, вы можете просто нажать «Подключиться».

1. Начните свой первый проект (начинающий)

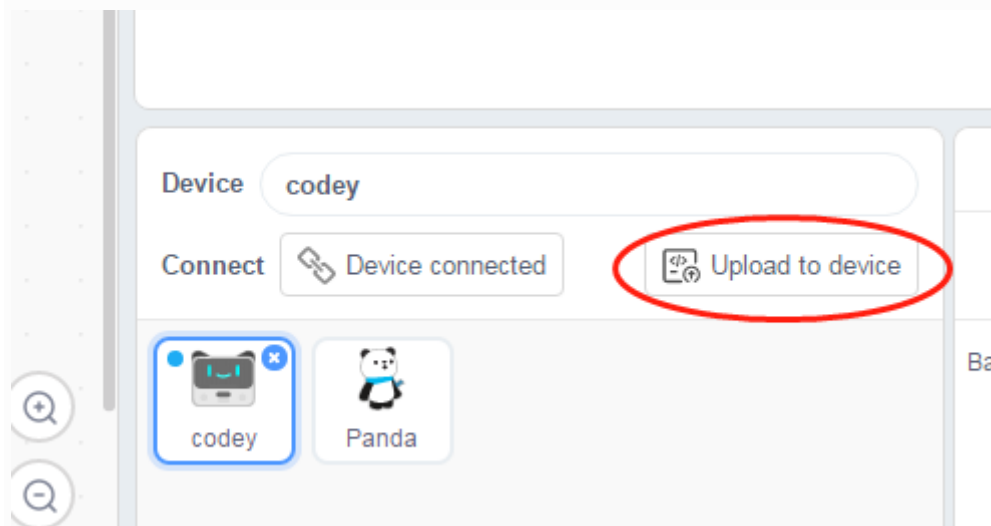
1. Перетащите блок событий в область скриптов.



1. Добавьте звуковой блок и выберите звуковой эффект.

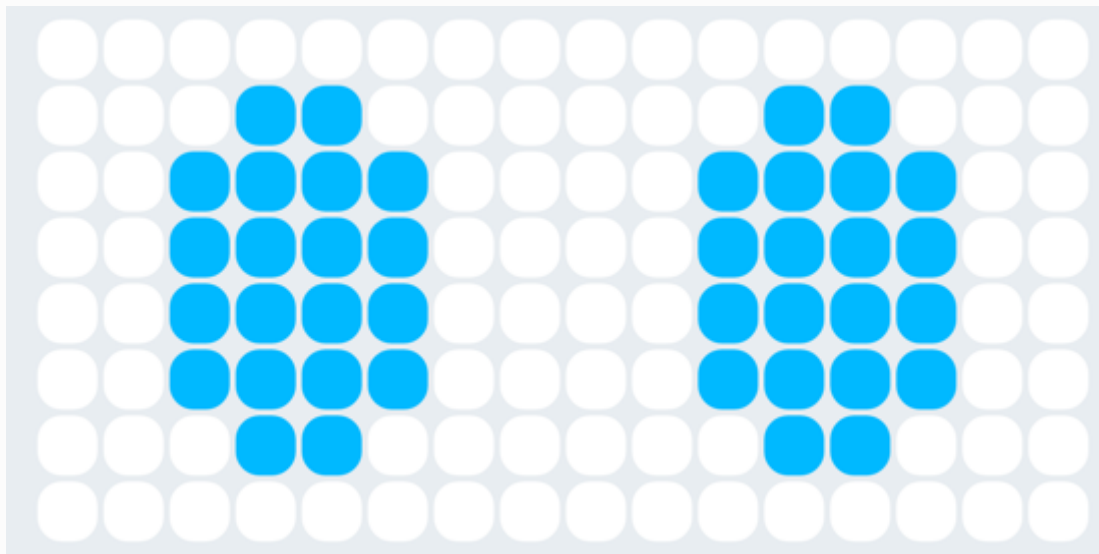
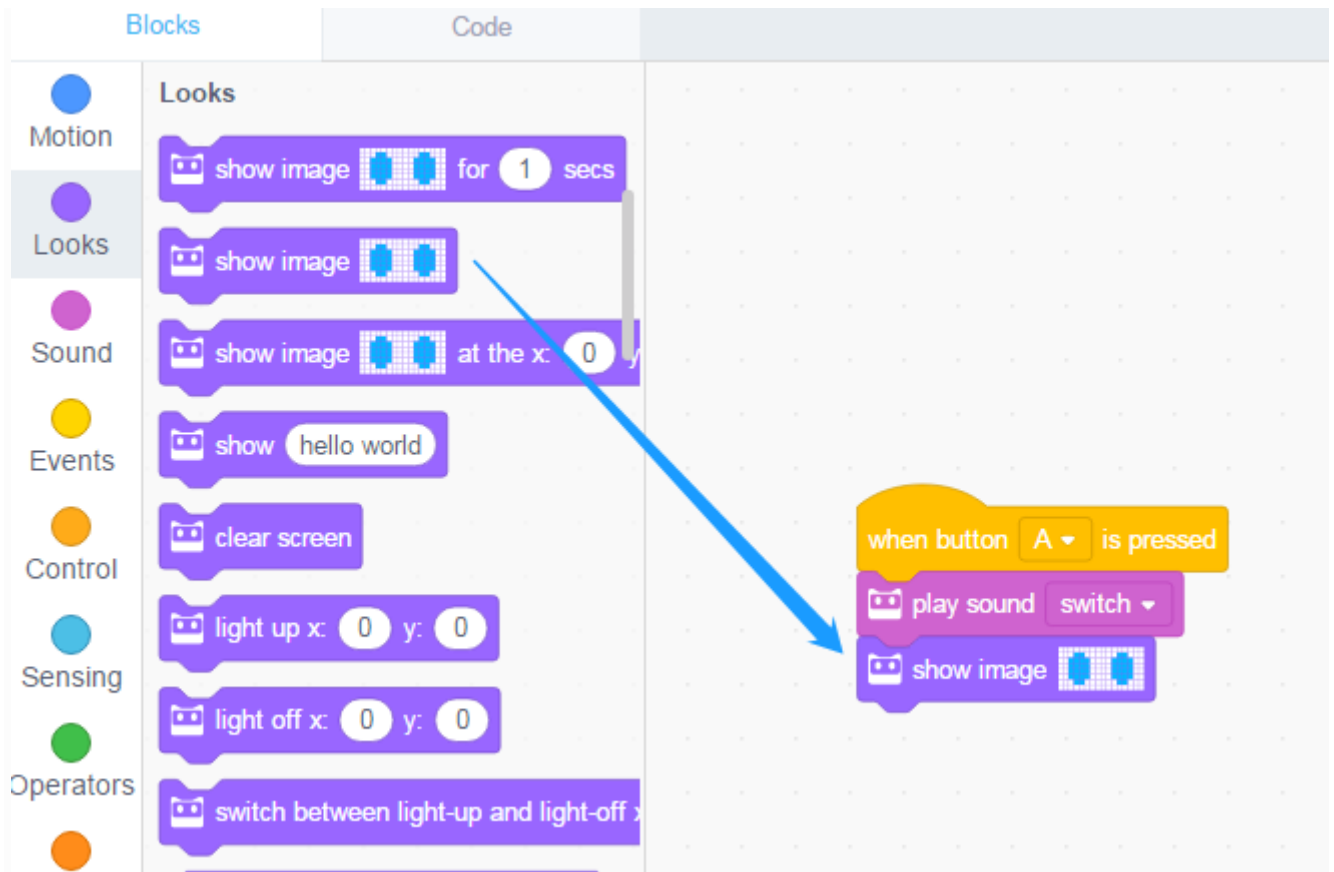


1. Нажмите «Загрузить на устройство».

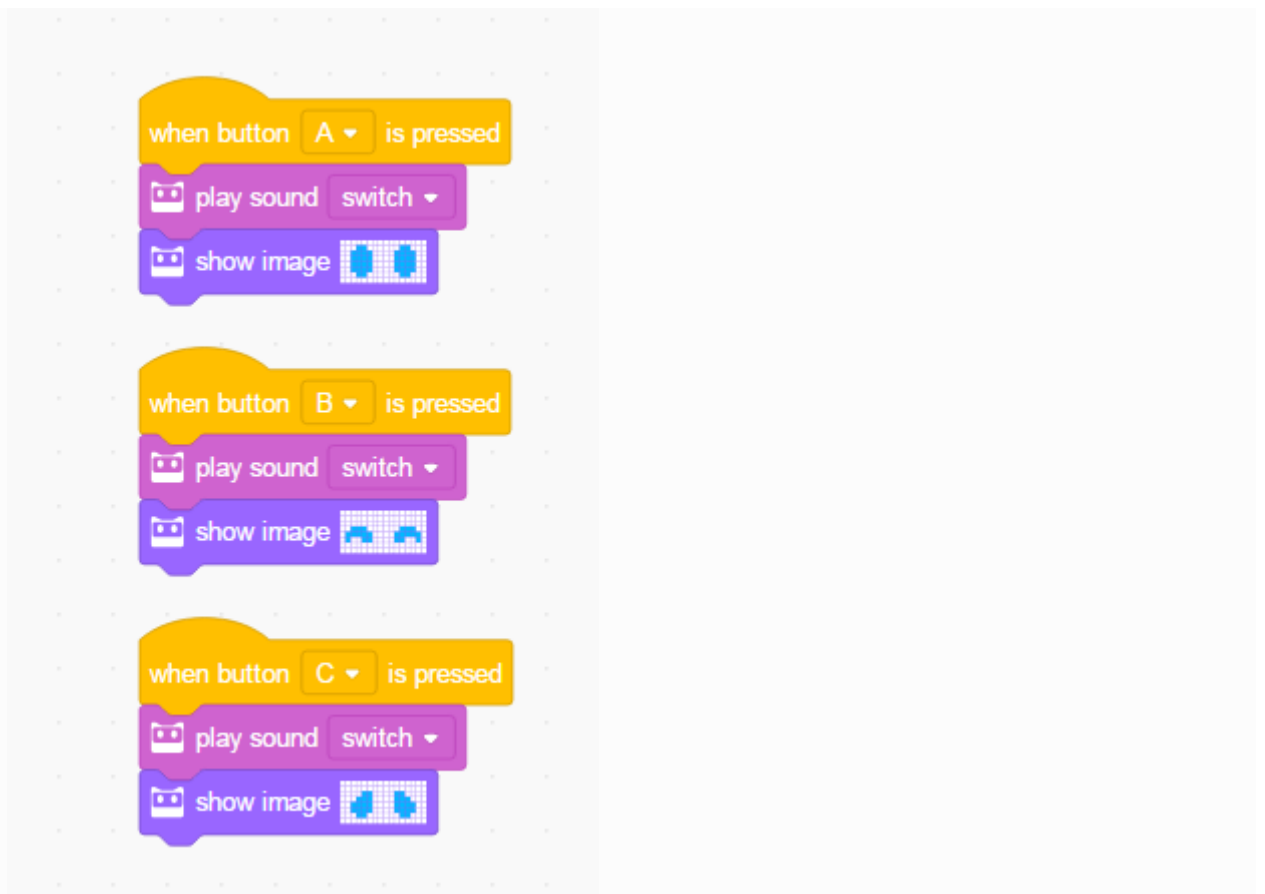


Нажмите кнопку A на коде, и вы услышите некоторые звуки от codey.

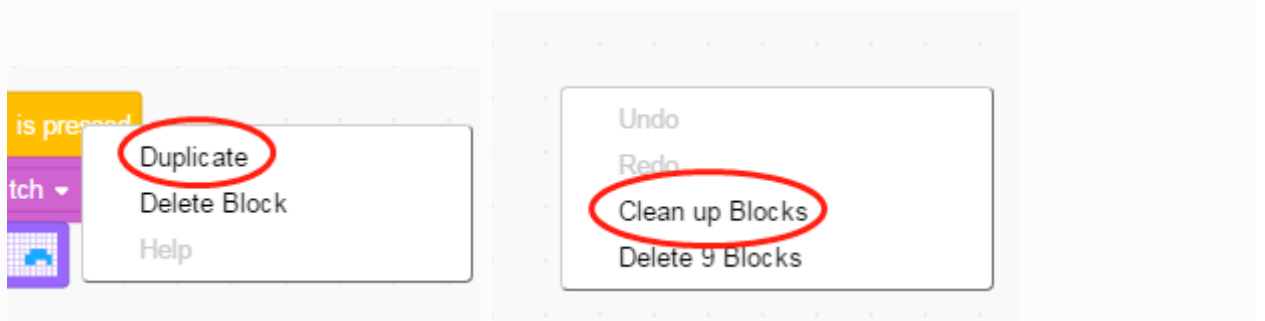
1. Добавьте блок изображения. Вы можете выбрать изображение по умолчанию, или вы можете создать изображение, которое вам нравится с помощью мыши.



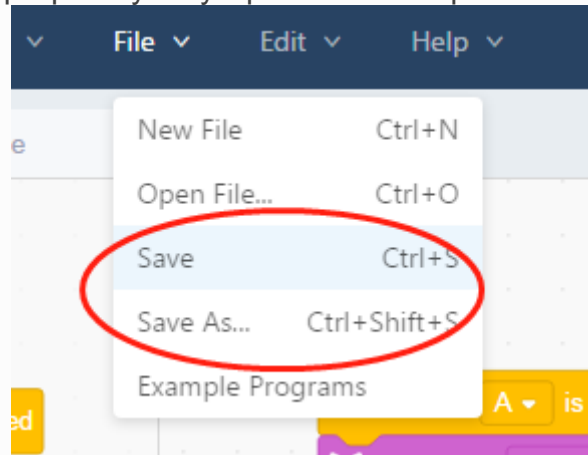
1. Давайте создадим больше изображений и будем управлять ими с помощью кнопок А, В, С.



Совет. Вы можете щелкнуть правой кнопкой мыши по блокам для дублирования сценариев и щелкнуть правой кнопкой мыши в пустой области сценариев, чтобы изменить все блоки.



1. Загрузите программу на устройство и сохраните



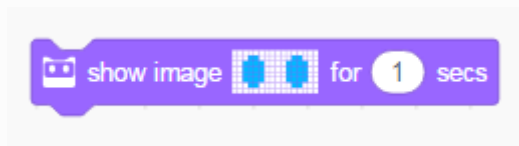
программу.

2. Последний шаг! Покажите свой проект друзьям и семьям!

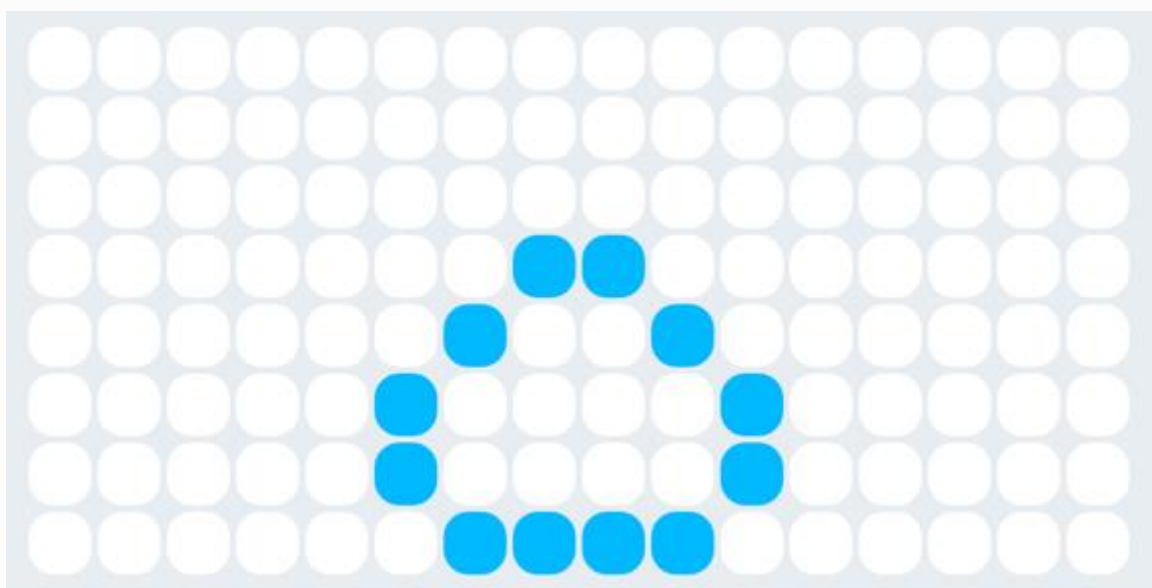
2. Сделайте анимацию: Joggle Bouncy Balls (начинающий)

В этом проекте вы узнаете, как сделать анимацию с codeu и как вызвать ее забавным способом:

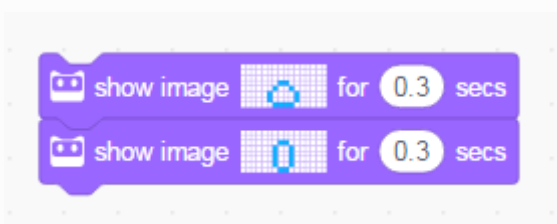
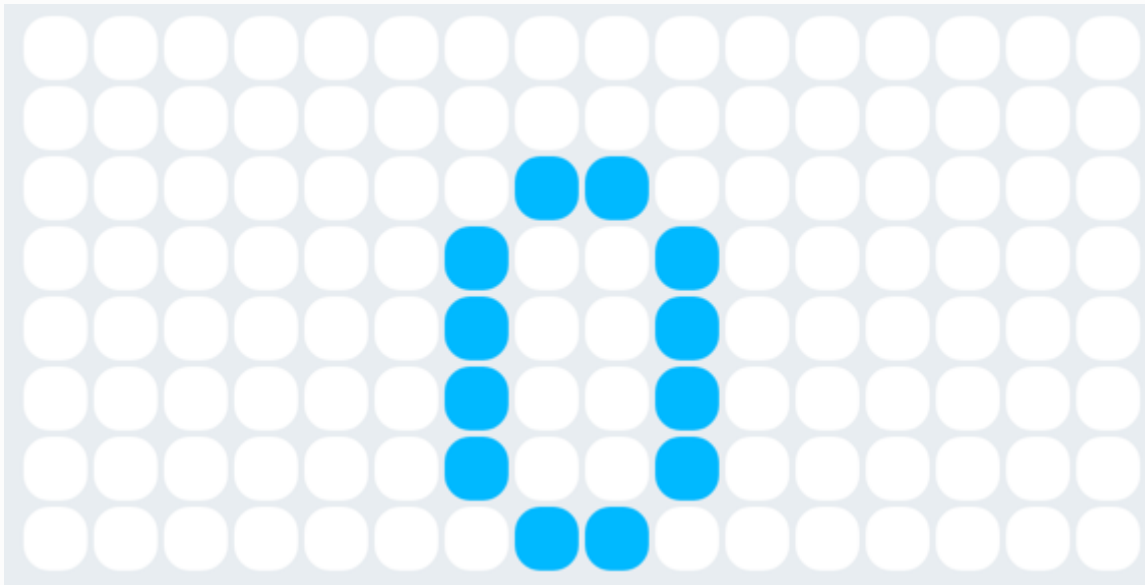
1. В этом проекте необходим следующий блок:



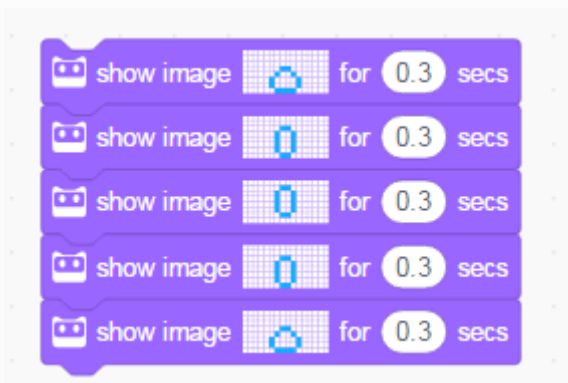
1. Откройте новый файл. Затем нарисуйте гибкий шар. Сделайте так, чтобы он выглядел как ложка на полу.



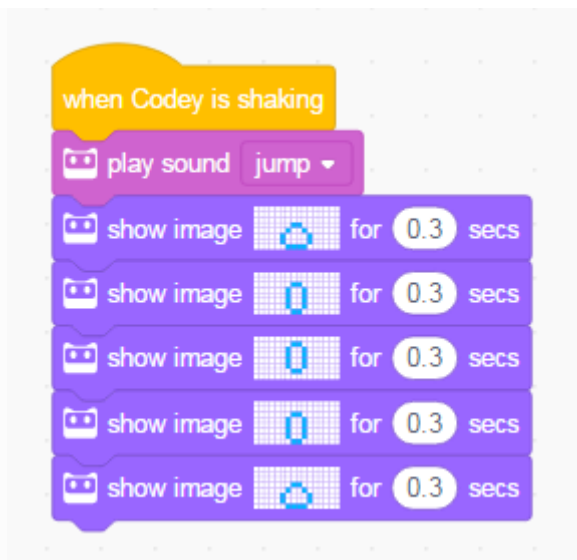
1. Продолжай. Теперь нарисуйте мяч, который подпрыгивает. Затем соедините этот блок и первый и установите время на 0,3 секунды.



1. Продолжайте заканчивать изображение. Составьте весь процесс прыгающего мяча от прыжка до посадки.



1. Добавьте звук и событие триггера.



1. Загрузите программу и встряхните код. Теперь, что происходит?

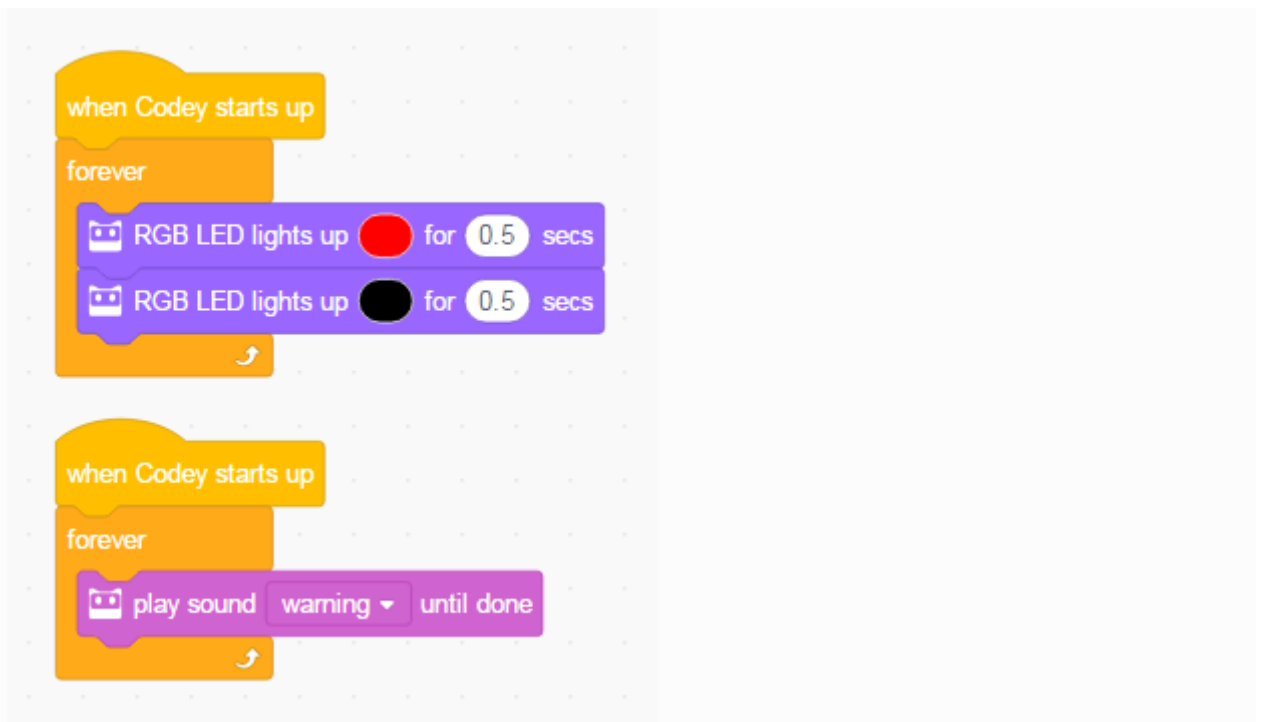
Связывая кодекс с вашим телом, вы можете обнаружить, что бодрый шар будет подпрыгивать, когда вы вскочите.

Советы: 1. Вы можете использовать разные события для запуска анимации. 2. Вы можете изменить скорость анимации, отрегулировав продолжительность времени.

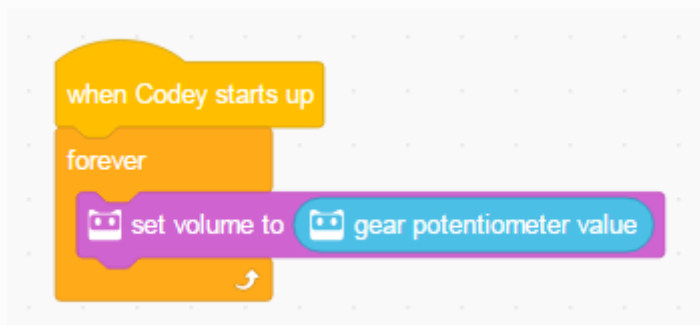
3. Выполнить одновременно: Тревога (начинающий)

Комбинация блоков может просто служить сценарием. ***Что касается codey, он может выполнять шесть сценариев не более одного раза.*** ,

Эта программа включает в себя два сценария. Один скрипт предназначен для того, чтобы заставить codey повторить звук, а другой скрипт - включить и выключить флеш-код.



Совет. Вы можете добавить еще один скрипт для управления объемом codey с помощью потенциометра.



Примечание. Вы можете найти потенциометр шестерни на стороне койки.

gear potentiometer

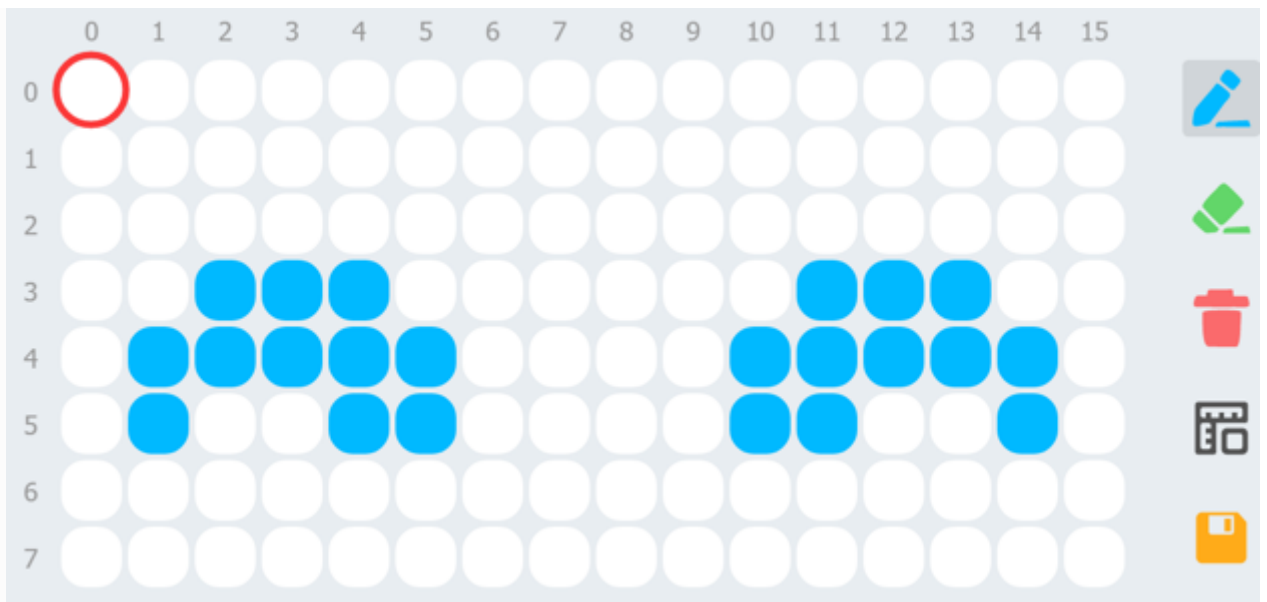



4. Images Координаты (промежуточные)

1. В этой демонстрации используются следующие блоки

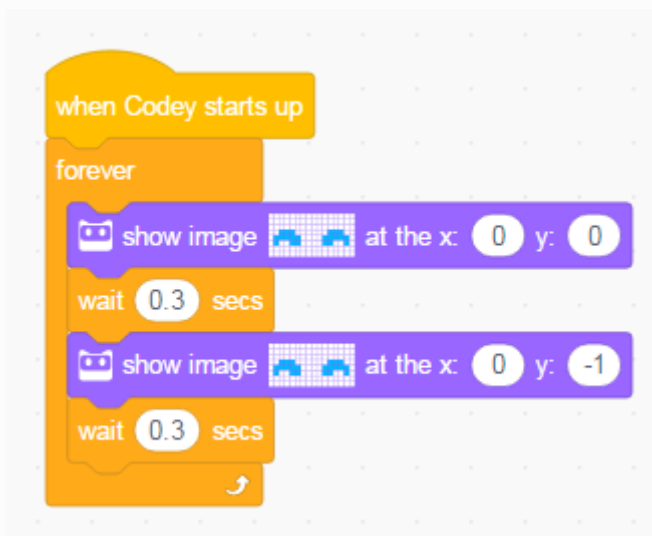


Вы можете использовать этот блок для отображения изображений в любых заданных координатах. Чтобы получить точную координату, обратитесь к левому верхнему углу изображения.



Совет. Откройте редактор изображений. Просто нажмите на значок «», чтобы показать координаты.

1. Попробуйте выполнить следующий скрипт и проверьте, как работает программа.

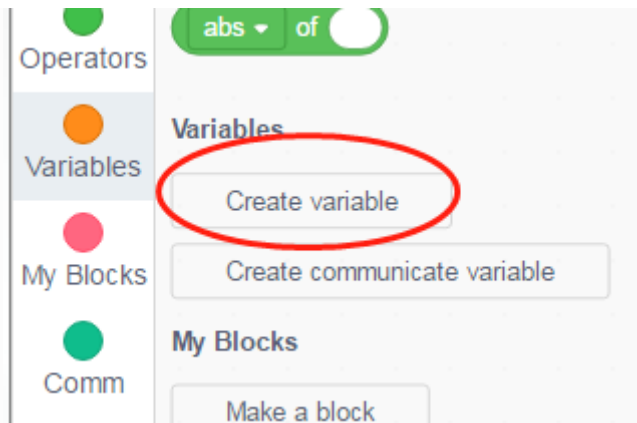


5. Вращающийся и управляемый координатами кролик(расширенный)

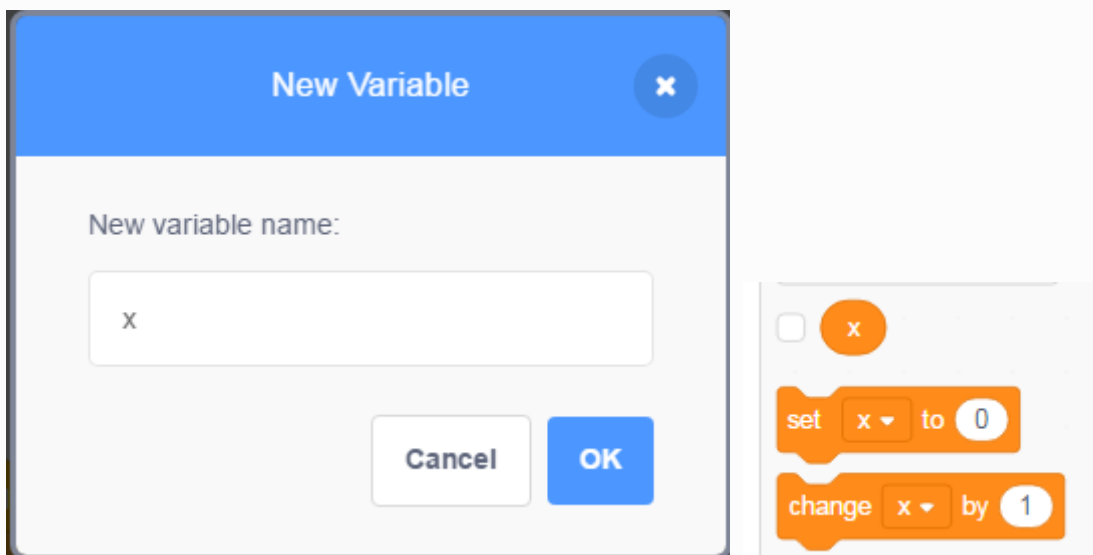
Спасибо Ekime за то, что вы создали для нас изображение кролика и помогли писателю завершить этот проект.

Функция Variables может позволить движению изображения:

1. Создать переменную

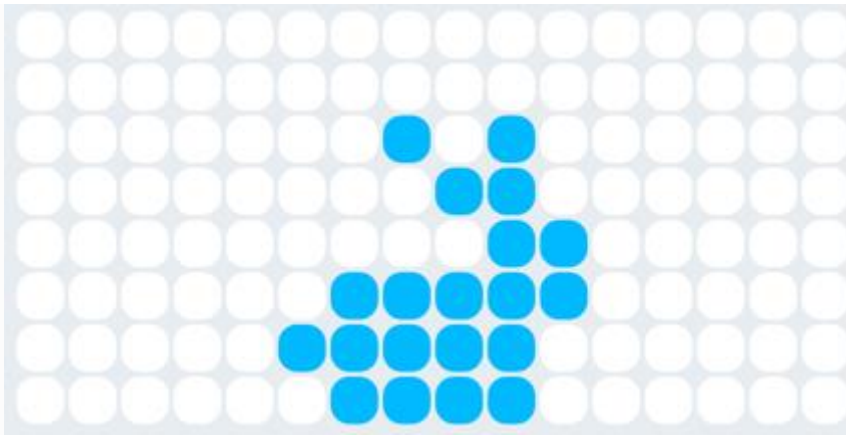


В mblock 5 вы можете назвать переменные только латинскими буквами. Китайские символы не поддерживаются, но вы можете назвать переменные английскими буквами или пиньинами.

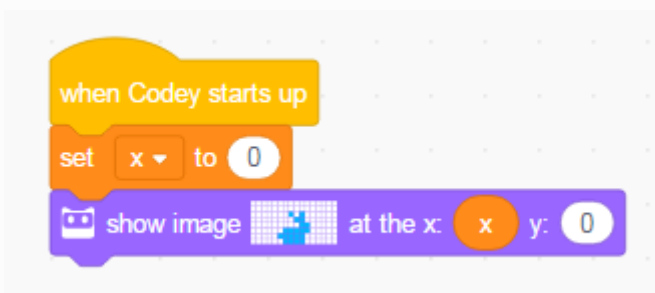


1. Используйте следующий блок, чтобы составить кролика.

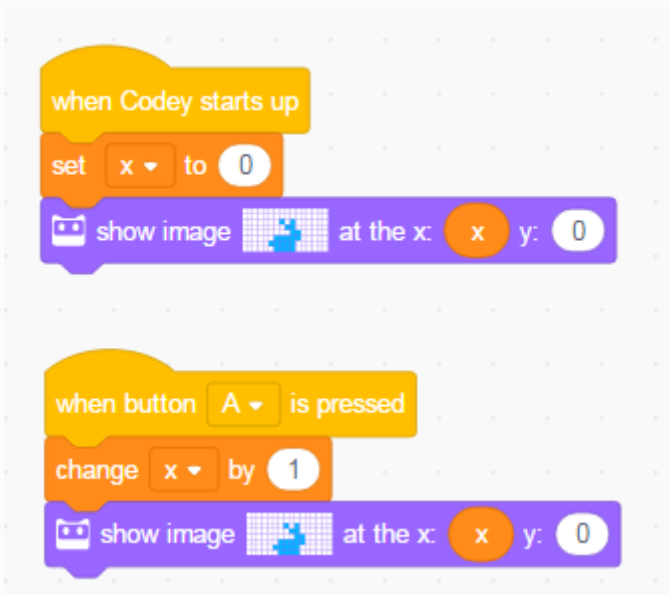




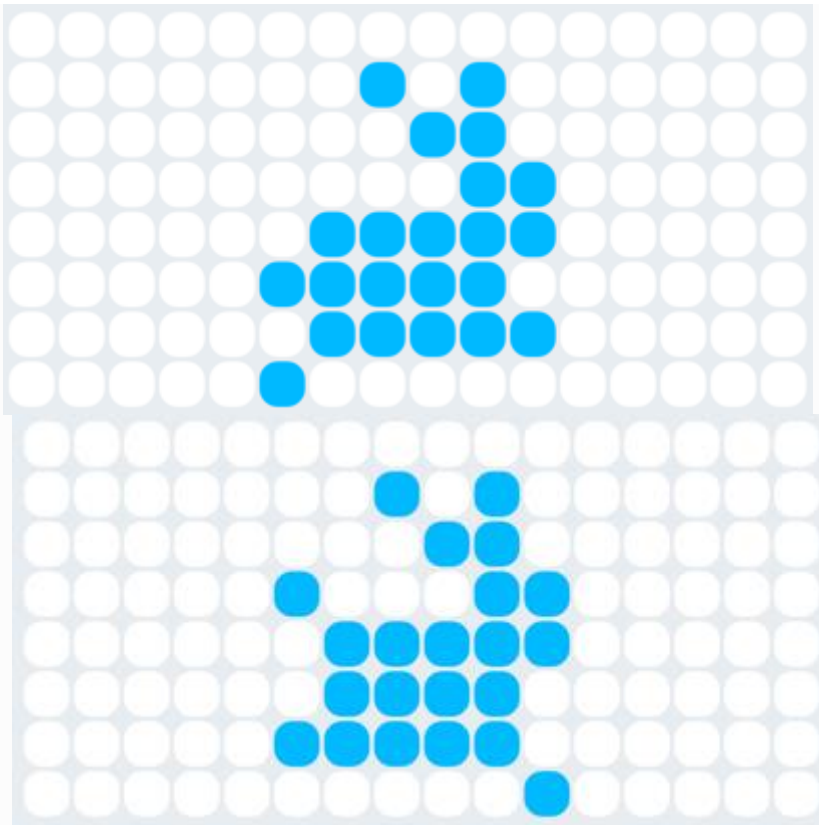
1. Установите переменную в координату x изображения.



1. Добавьте еще один скрипт, где переменная может быть изменена при нажатии кнопки. Вы можете наблюдать, как изменяется изображение, и вы можете обнаружить, что кролик выйдет из экрана, если переменная превышает определенное значение.



1. Теперь завершите сценарий. Поверните изображения и измените координаты. Смотри! Кролик бежит!



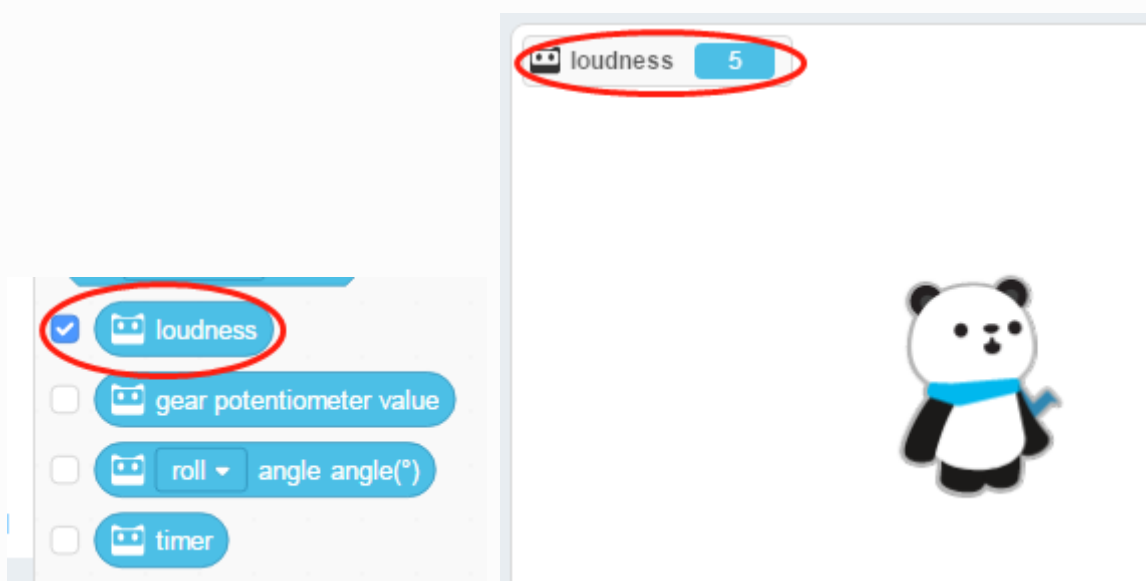
```
when Codey starts up
  set x to -11
  repeat 8
    show image [grid icon] at the x: x y: 0
    wait 0.2 secs
    change x by 1
    show image [grid icon] at the x: x y: 0
    wait 0.2 secs
    change x by 1
    show image [grid icon] at the x: x y: 0
    wait 0.2 secs
    change x by 1
```

Совет. Дублируйте блок изображения, а затем внесите некоторые изменения. Таким образом, вы можете сделать анимацию с минимальной светой.

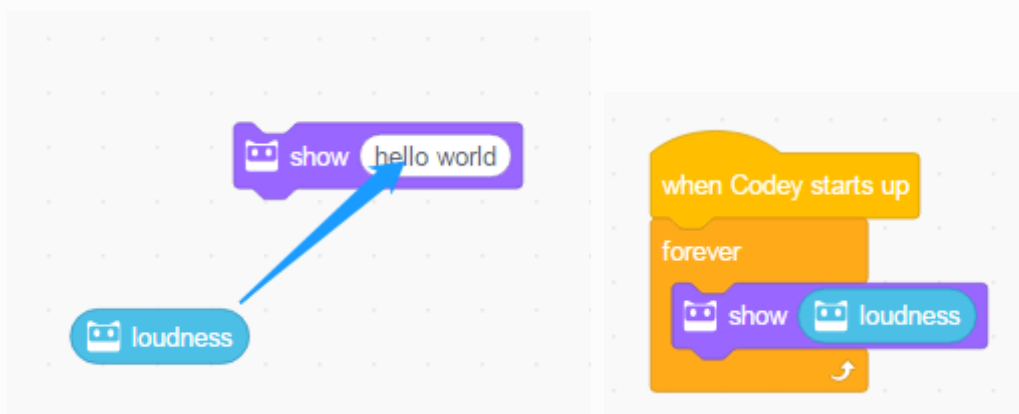
Значения датчика 6.Display (промежуточный)

codey имеет множество датчиков и может таким образом обнаруживать изменения окружающей среды вокруг, включая громкость, свет, жест и т. д. Вы можете получить доступ к значениям датчика двумя способами:

1. Убедитесь, что ваше устройство хорошо подключено. Затем выберите значение, которое вы хотите получить, и покажите его на сцене.

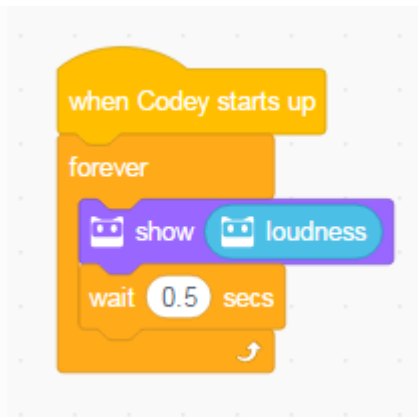


1. Вы также можете использовать скрипт, как показано ниже, и показать значения датчиков на экране codey.



1. Загрузите вышеупомянутую программу на устройство. Затем сделайте несколько звуков для коней. Посмотрите, значение на сцене и экране меняется.

Совет. Добавьте блок ожидания в блок повтора, чтобы замедлить изменение значения на экране.



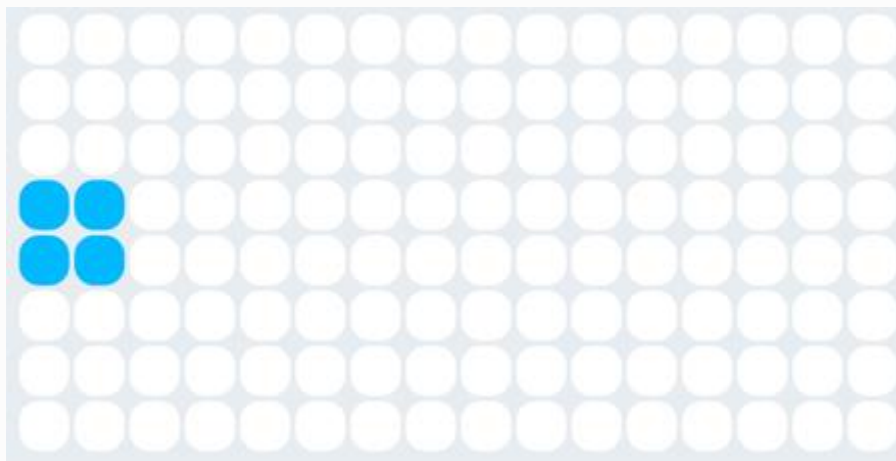
7. Индикатор объема (средний)

С координатой изображения и звуковым датчиком вы можете обеспечить динамическое отображение громкости на экране codey:

1. Подключите устройство. Выберите блок громкости и проверьте громкость в текущей среде (значения могут отличаться в других средах).



1. Используйте следующий блок для создания простого изображения.

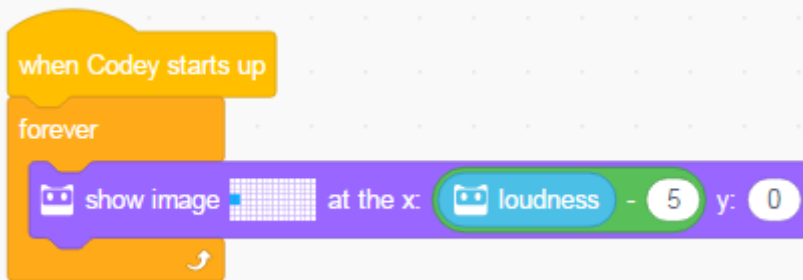


1. Для получения значений объема используйте арифметические блоки в категории «Операторы».

Operators



1. Завершите программу и загрузите ее. Теперь вы можете кричать на свой код и проверить, работает ли он.

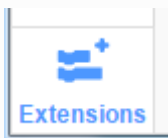


Совет. Вы можете регулировать значение с помощью нескольких блоков, одновременно наблюдая изменение громкости.



8. Избегайте препятствий (продвинутых)

Если вы хотите начать работу с каменистыми, сначала нужно добавить скальное расширение в mblock 5.



Add-on



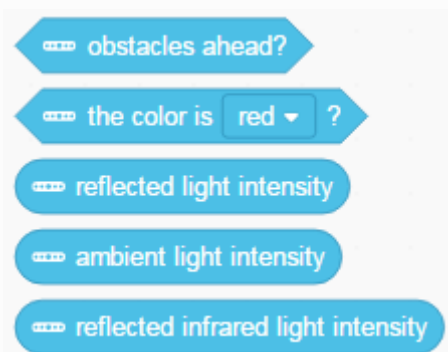
rocky

A combination with the rocky brings you a codey rocky! More fun is coming!

В этом проекте понадобится скалистый кодекс и его датчик. Каменистый скальный цвет имеет ИК-датчик цвета, который можно переключать вперед или вниз. В этом проекте вам нужно переключить ИК-датчик цвета вперед, чтобы обнаружить препятствия перед скалистым коди.



Все блоки для этого датчика отмечены значком датчика.



1. В этом проекте необходим следующий блок.

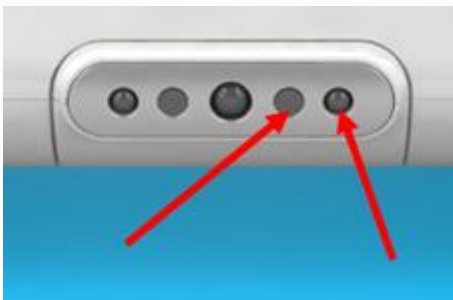
Объяснение: Если перед ИК-датчиком цвета обнаружено препятствие, результат для этого блока будет «истинным», иначе он будет «ложным».

obstacles ahead?

Получение ИК-отправления IR

Примечание. Основной принцип обнаружения препятствий лежит в ИК-отражении. Когда обнаружено обнаружение препятствий: чем светлее цвет объекта, тем длиннее расстояние; чем темнее цвет объекта, тем короче расстояние.

1. Завершите следующую программу. Вы можете помещать препятствия перед скалистым треском, чтобы наблюдать, как изменится его лицо.

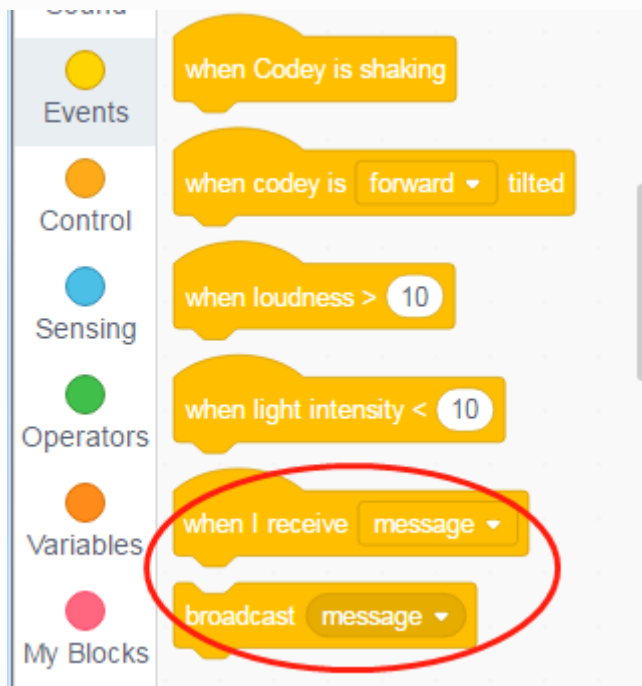


1. Завершите программу. Убедитесь, что скалистый кодекс будет двигаться дальше, когда препятствие не обнаружено, и поверните направо, когда обнаружены препятствия.

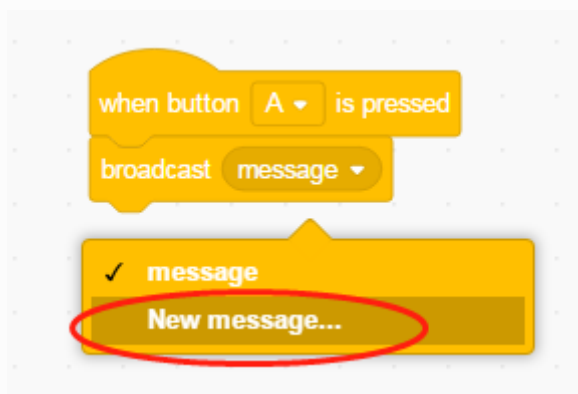
```
when Codey starts up
  forever
    if obstacles ahead? then
      show image [obstacles ahead]
      turn right at power 50 % for 1 secs
    else
      show image [clear path]
      move forward at power 50 %
```

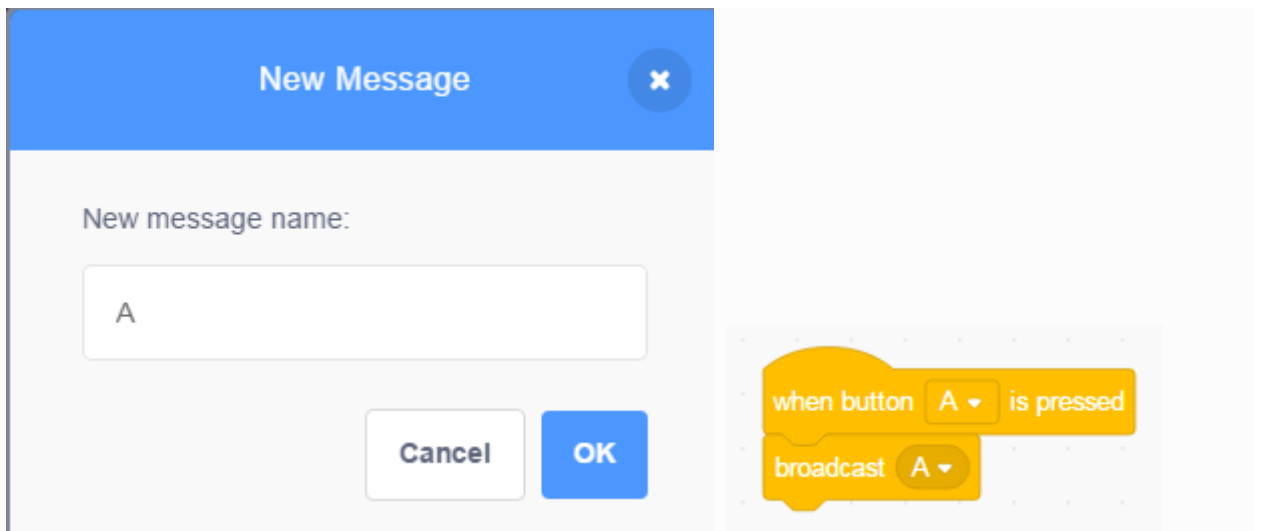
9. Интеракция с сценическими сценариями: контроль музыкальных инструментов с трансляцией (промежуточная)

Функция широковещательной передачи может расширять возможности использования кодекса для спрайтов с помощью спрайтов на сцене. Таким образом, вы можете выполнить свои интерактивные сценические проекты:

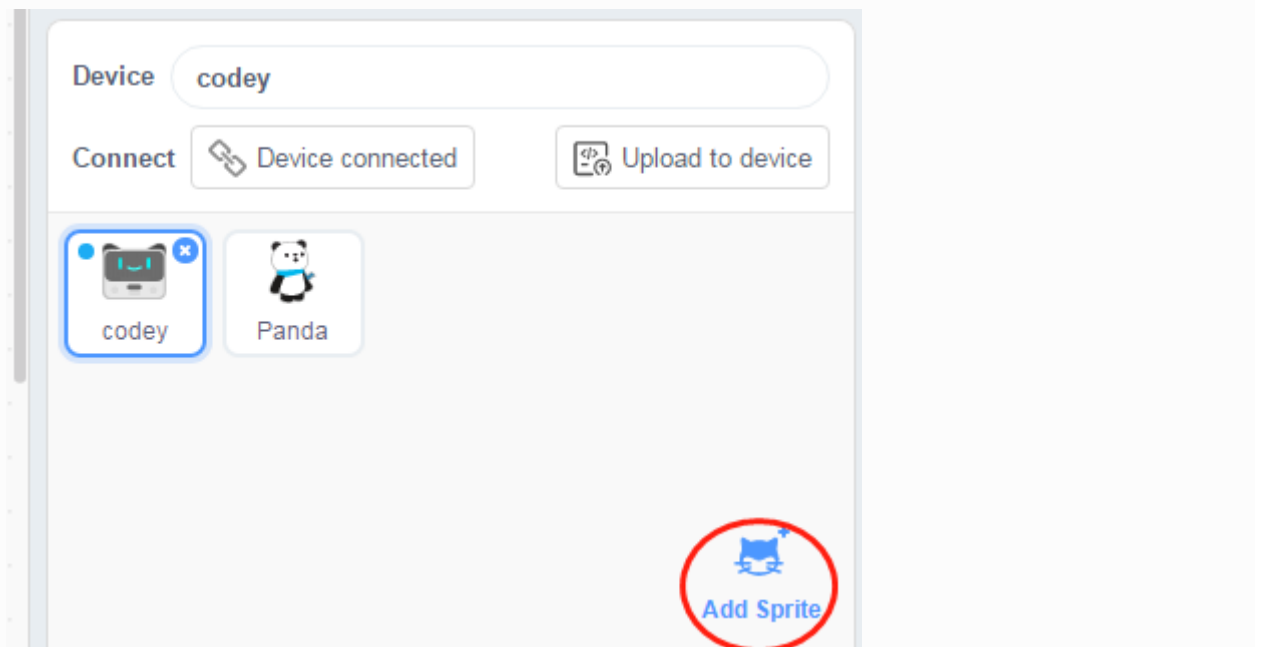


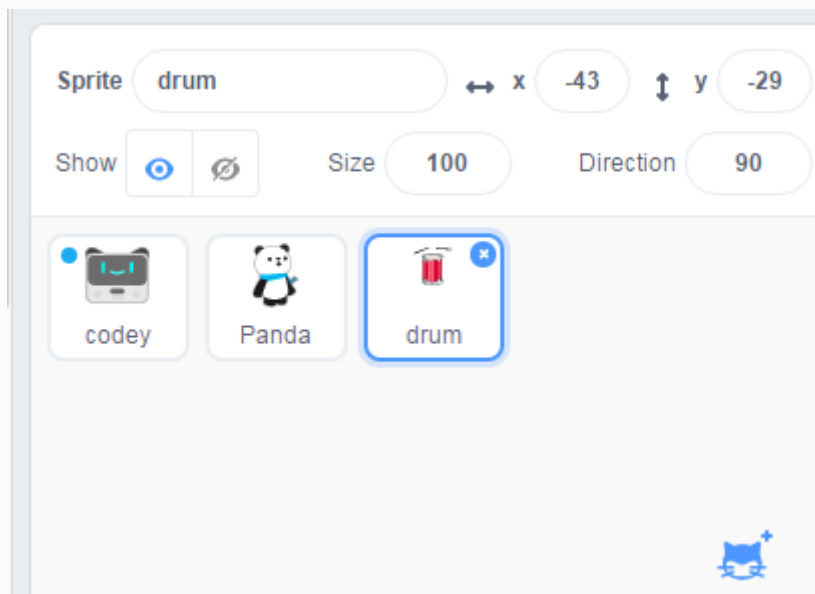
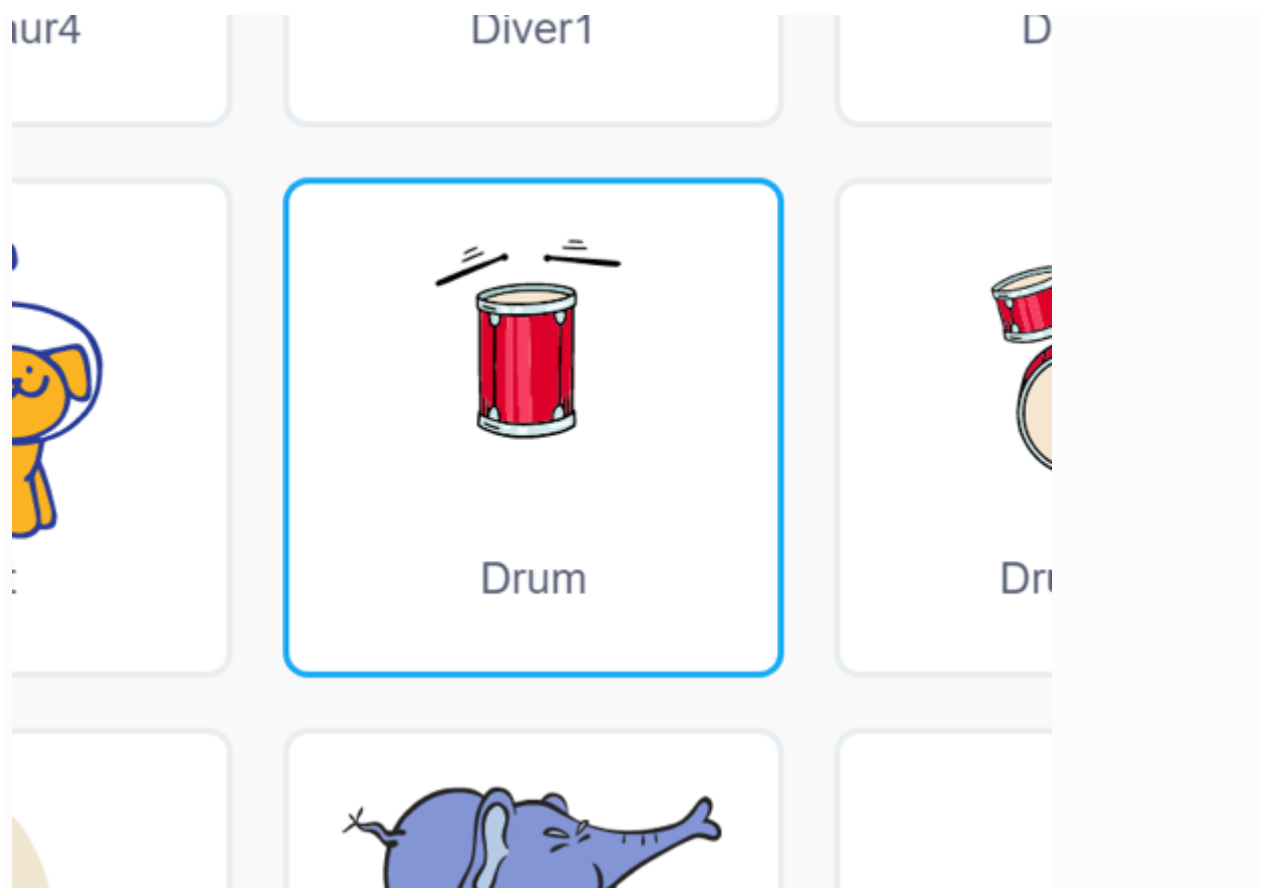
1. Создать новое сообщение: при нажатии кнопки А транслировать А.



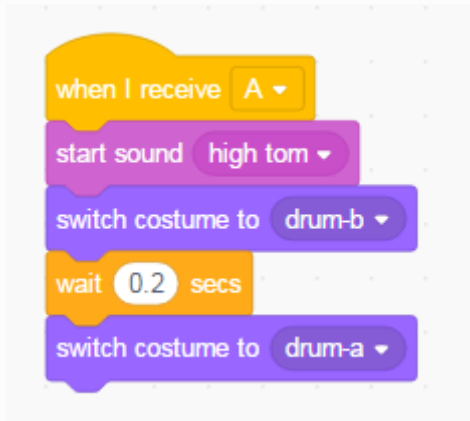
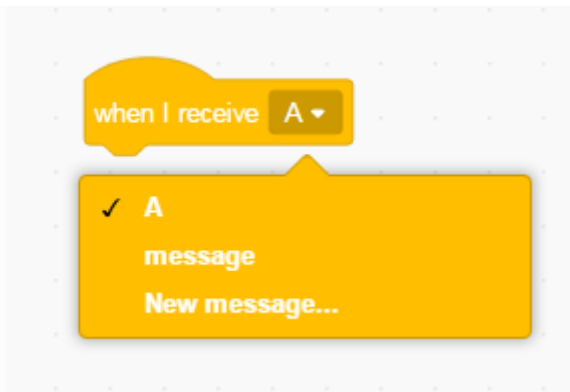


1. Добавьте спрайт. Новый спрайт появится в области сцены.

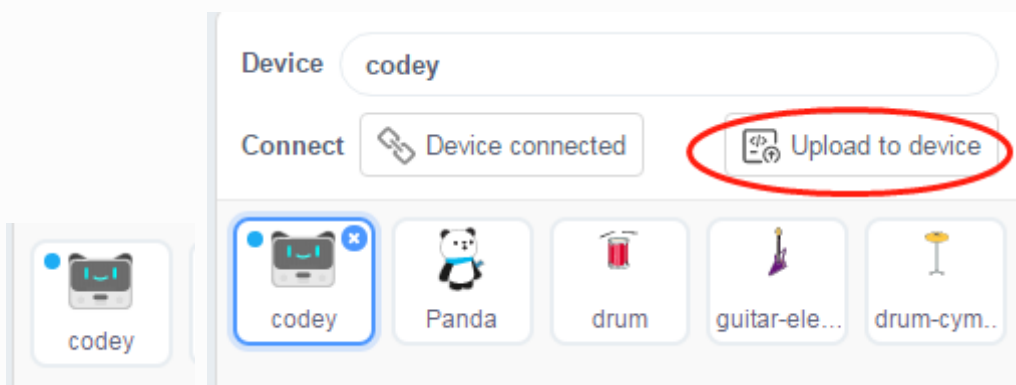




1. Нажмите на новый спрайт (барабан) и создайте для него программу.



1. Нажмите на codey и загрузите свою программу на устройство. Затем, если вы нажмете кнопку A, вы услышите барабанные удары из спрайт-барабана на компьютере.



1. Точно так же вы можете добавить спрайты на музыкальные инструменты. Давайте устройм концерт для Панды!

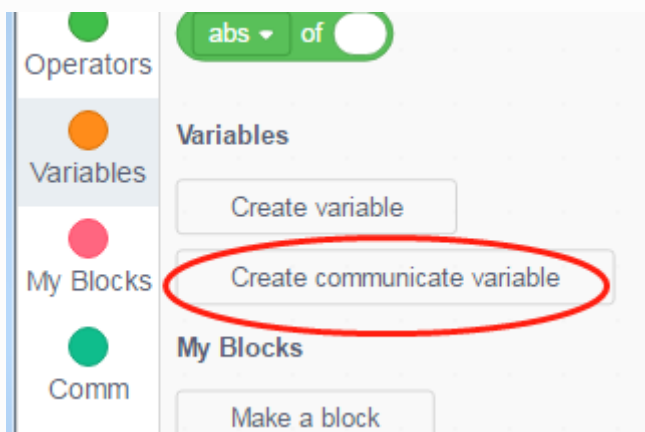


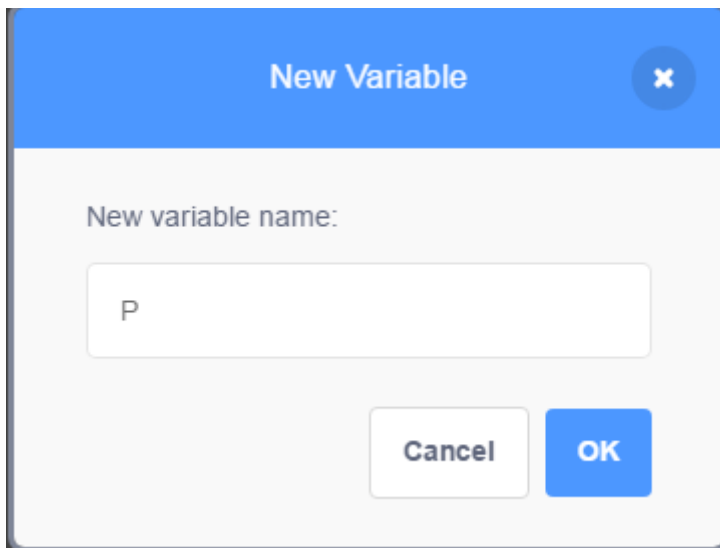
10. Узнать переменный и механический потенциометр(расширенный)

Функция Broadcast - это всего лишь один вариант. Вы можете заставить codey взаимодействовать со сценой, создав также переменную передачи.

В этом проекте вы можете использовать потенциометр механизма кодирования для изменения размера спрайта:

1. Создать коммуникационную переменную





1. Установите переменную Communicate Variable на значение потенциометра шестерни.



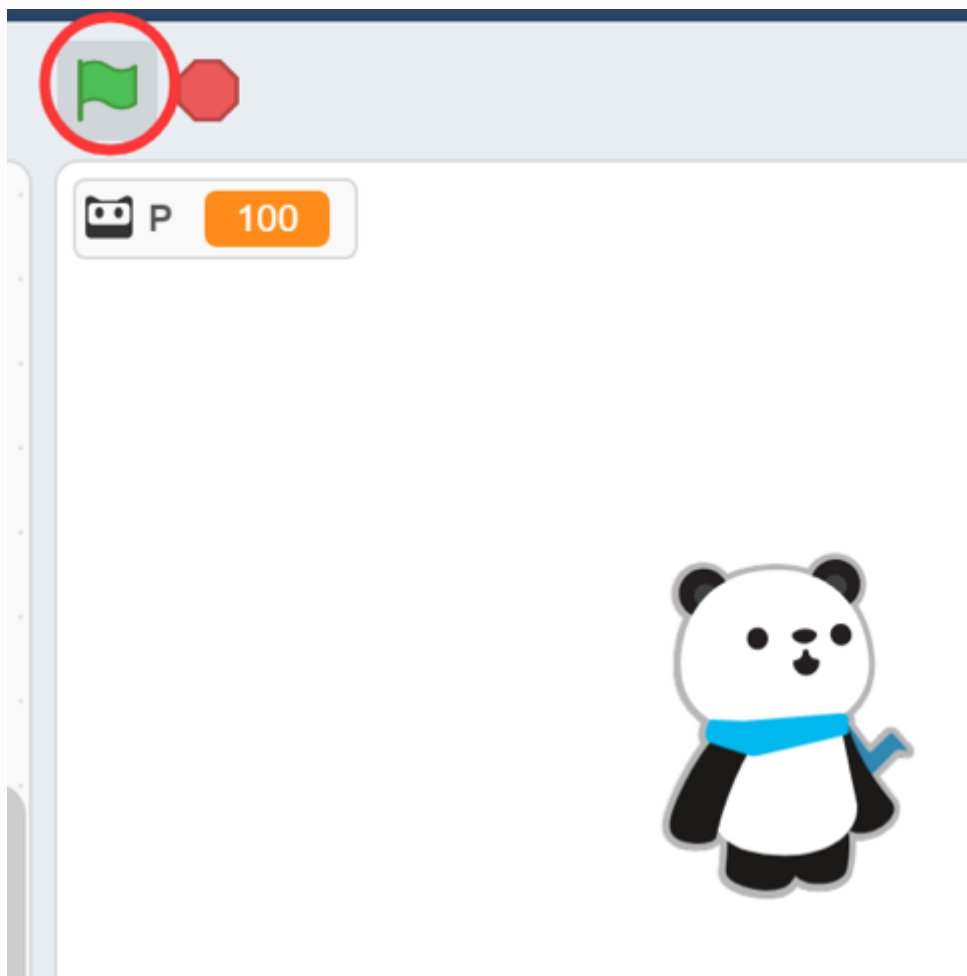
1. Нажмите на спрайт Panda, и вы обнаружите, что переменная связи отображается в области данных.



1. Задайте размер спрайта для значения переменной связи.



5. Загрузите программу codey на устройство и нажмите зеленый флаг. Потяните потенциометр шестерни и наблюдайте, как изменяется срит.



Обсудить и поделиться

Хотите поделиться с нами своими проектами и навыками? Пожалуйста, посетите нас на официальном форуме codey rocky:

<http://forum.makeblock.com/>

Когда вы изучаете язык программирования, первая строка кода должна быть: «Hello, World!»

codey довольно прост в использовании ::

```
import codey
codey.show("Hello, World!")
```

Codey

функция

когда запускается codey

```
import codey
при получении (сообщении)
```

```
def callback():
    pass
codey.on_message(msgstr, callback)
```

трансляция «привет»

```
codey.message('hello')
```

Example1

```
import codey
codey.show('hello world')
```

Example2

```
import codey
def on_start_callback():
    codey.message('hello')
codey.on_start(on_start_callback)
def on_message_callback():
    codey.show('hello world')
codey.on_message('hello', on_message_callback)
```

скалистый

rocky - это шасси, содержащее два колеса, ИК-излучатель, ИК-приемник, датчик цвета RGB, датчик освещенности и излучатель белого света

функция

двигаться вперед с мощностью 50 в течение 1 секунды

```
rocky.forward(50, 1)
```

двигаться назад при мощности 50 в течение 1 секунды

```
rocky.backward(50, 1)
```

поверните налево при мощности 50 в течение 1 секунды

```
rocky.left(50, 1)
```

поверните направо при мощности 50 в течение 1 секунды

```
rocky.right(50, 1)
```

пример

```
import rocky

rocky.forward(50, 1)
rocky.back(50, 1)
rocky.left(50, 1)
rocky.right(50, 1)
```

Лицевая панель

Лицевая панель представляет собой матрицу с 8 * 16 светодиодами. Каждый из светодиодов управляется с помощью ари. функция

показать изображение `【^ _ ^】` для (1) сек

```
codey.face(
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0'
    '0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0'
    '0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0', 1)
```

показать изображение `【^ _ ^】`

```
codey.face(
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0'
    '0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0'
    '0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0')
```

показать изображение `【^ _ ^】` при x [0], y [0]

```
codey.face_at(
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0'
    '0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0'
    '0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0', 0, 0)
```

показать "привет мир"

```
codey.show('hello world')
```

очистить экран

```
codey.clear()
```

загорать x: (0) y: (0)

```
codey.pixel(x, y)
```

свет выключен x: (0) y: (0)

```
codey.pixel_off(0,0)
```

пример



```
import codey

codey.face(
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0'
    '0 0 0 1 1 0 0 0 0 0 0 1 1 0 0'
    '0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0'
    '0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0'
    '0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0'
    '0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0'
    '0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0'
    '0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0'
    '0 0 0 0 0 0 0 0 0 0 0 0 0 0 0', 1)
codey.show('hello')
codey.clear()
codey.pixel(3, 3)
```

Светодиодный индикатор RGB

В скалистом ведется свет rgb, который поддерживает различные цвета.

функция

Светодиод горит **【красный】** , для (1) секунды

```
codey.color('#334455', 1)
```

Светодиод горит **【красный】**

```
codey.color('#334455')
```

Светодиод выключен

```
codey.color('#000')
```

установите значение RGB LED на красный (255) зеленый (0) синий (0)

```
codey.rgb(255, 0, 0)
```

пример

```
import codey
```

```
codey.color('#ff0000', 1)
```

```
codey.color('#00ff15', 1)
codey.color('#0021ff', 1)
```

Громкий динамик

функция

параметры звука доступны для выбора:

- `hello.wav` : Здравствуйте
- `hi.wav` : Здравствуй
- `bye.wav` : пока
- `yeah.wav` : Да
- `wow.wav` : Вау
- `laugh.wav` : смех
- `hum.wav` : гул
- `sad.wav` : грустный
- `sigh.wav` : вздох
- `annoyed.wav` : раздражен
- `angry.wav` : сердитый
- `surprised.wav` : испуганный
- `yummy.wav` : pettish
- `curious.wav` : любопытно
- `embarrassed.wav` : смущен
- `ready.wav` : готов
- `sprint.wav` : спринт
- `sleepy.wav` : храп
- `meow.wav` : мяу
- `start.wav` : Начало
- `switch.wav` : переключатель
- `beeps.wav` : звуковые сигналы
- `buzzing.wav` : шум
- `exhaust.wav` : проветривание
- `explosion.wav` : взрыв
- `gotcha.wav` : Попался
- `hurt.wav` : болезненный
- `jump.wav` : Прыгать
- `laser.wav` : лазерный
- `level up.wav` : уровень повышен
- `low energy.wav` : мало энергии
- `metal clash.wav` : металл-столкновение
- `prompt tone.wav` : подсказка

- `right.wav` : правильно
- `wrong.wav` : неправильно
- `ring.wav` : рингтон
- `score.wav` : Гол
- `shot.wav` : выстрел
- `step_1.wav` : шаг 1
- `step_2.wav` : шаг 2
- `wake.wav` : активировать
- `warning.wav` : предупреждение

играть звук **【привет】** . Звук может быть прерван. `codey.say ('hello.wav')`

играть звук **【привет】** до конца. Звук не будет прерван.

```
codey.say('hello.wav', True)
```

ОСТАНОВИТЬ ЗВУК

```
codey.mute()
```

play note (60) для (0.25) удара. 参照 表 待 定.

```
codey.play(435, 500)
```

остановки (0.25) бит

todo

воспроизводить звук на частоте (700) Гц для (1) секунды

todo

изменить объем на (-10)

```
codey.change_volume(-10)
```

установить громкость на (100)

```
codey.set_volume(100)
```

объем

```
codey.get_volume()
```

пример

```
import codey
```

```
codey.set_volume(100)
codey.say('hello.wav', True)
codey.say('hi.wav', True)
```

гироскоп

функция

когда кодекс дрожит

```
def callback():  
    pass  
codey.on_shake(callback)
```

когда Codey будет 【вперед, назад, влево, вправо】 tilted. The первый параметр доступен для выбора `forward` является: `backward`, `left`, `right`, .

```
def callback():  
    pass  
codey.on_tilt('forward', callback)
```

пример

```
import codey  
  
def on_shake_callback():  
    codey.color('#ff0000')  
  
codey.on_shake(on_shake_callback)
```

ИК-приемник

функция

** Пример>

кнопка

Codey имеет три кнопки, `A`, `B`, `C`, .

функция

при нажатии кнопки 【A, B, C pressed】

```
def callback():  
    pass  
codey.on_button('A', callback)
```

Первый параметр может быть: `B`, `C`, .

пример

```
import codey  
  
def on_button_callback():  
    codey.color('#ff0000')  
  
codey.on_button('A', on_button_callback)
```

Потенциометр передачи

функция

пример

Световой датчик

функция

когда интенсивность света <(10)

```
def callback():  
    pass  
  
codey.on_light_under(50, callback)
```

пример

```
import codey  
  
def on_light_under_callback():  
    codey.color('#ff0000')  
  
codey.on_light_under(10, on_light_under_callback)
```

Звуковой датчик

функция

когда [] громкость > (10)

```
def callback():  
    pass  
  
codey.on_sound_over(10, callback)
```

пример

```
import codey  
  
def on_sound_over_callback():  
    codey.color('#ff0000')  
  
codey.on_sound_over(10, on_sound_over_callback)
```

Общий синтаксис-контроль

функция

подождите 1 сек.

```
import time  
time.sleep(1)
```

повторить в 10 раз

```
for count in range(10):  
    pass
```

навсегда

```
while True:  
    pass
```

ЕСЛИ ..., ТО

```
if False:  
    pass
```

если ... then..else

```
if False:  
    pass
```

```
else:  
    pass
```

Подожди до

```
while not False:  
    pass
```

ПОВТОРЯТЬ ДО

```
while not False:  
    pass
```

Общие синтаксические операторы

функция

1 + 1

1 + 1

1 - 1

1 - 1

1 * 1

1 * 1

1/1

1 / 1

выбирать случайные от 1 до 10

```
import random  
random.randint(1, 10)
```

1 < 2

1 < 2

1 == 1

```
1 == 1
```

```
2 < 1
```

```
2 < 1
```

а также

```
False and False
```

ИЛИ

```
False or False
```

НЕ

```
not True
```

ПОДКЛЮЧИТЬСЯ `hello` и `world`

```
str('hello') + str('world')
```

1 строка `word`

```
'world'[0]
```

длина `word`

```
len('world')
```

`hello` содержит `world`?

```
str('hello').find(str('world')) > -1
```

1 mod 3

```
1 % 3
```

круглый `3.4`

```
round(3.4)
```

абс-1

```
import math  
math.fabs(-1)
```

ПОТОЛОК `3.4`

```
import math  
math.ceil(3.4)
```

SQRT `4`

```
import math  
math.sqrt(4)
```

грех `2`

```
import math  
math.sin(2 / 180.0 * math.pi)
```

CO3 `2`

```
import math
```

```
math.cos(2 / 180.0 * math.pi)
```

загар2

```
import math  
math.tan(2 / 180.0 * math.pi)
```

КАК В2

```
import math  
math.asin(2) / math.pi * 180
```

ЭКОСа2

```
import math  
math.acos(2) / math.pi * 180
```

загар2

```
import math  
math.atan(2) / math.pi * 180
```

ln2

```
import math  
math.log(2)
```

log2

```
import math  
math.log10(2)
```

e^2

```
import math  
math.exp(2)
```

10^2

```
import math  
math.pow(10,2)
```