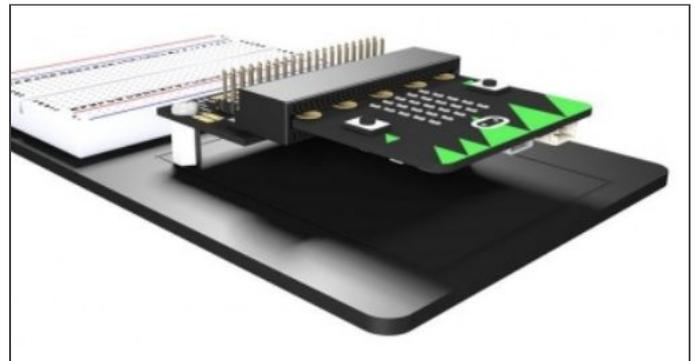
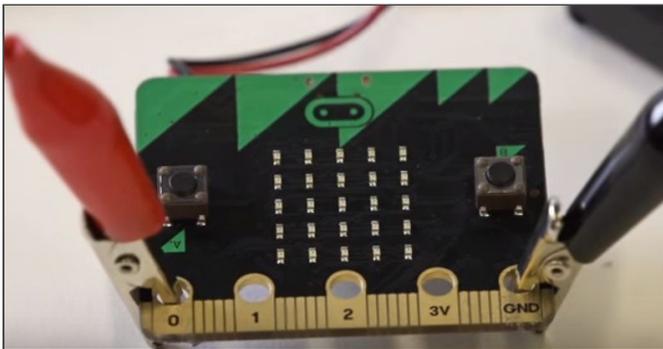


# Программируем BBC micro:bit

## Шаг 8. Пины ввода / вывода

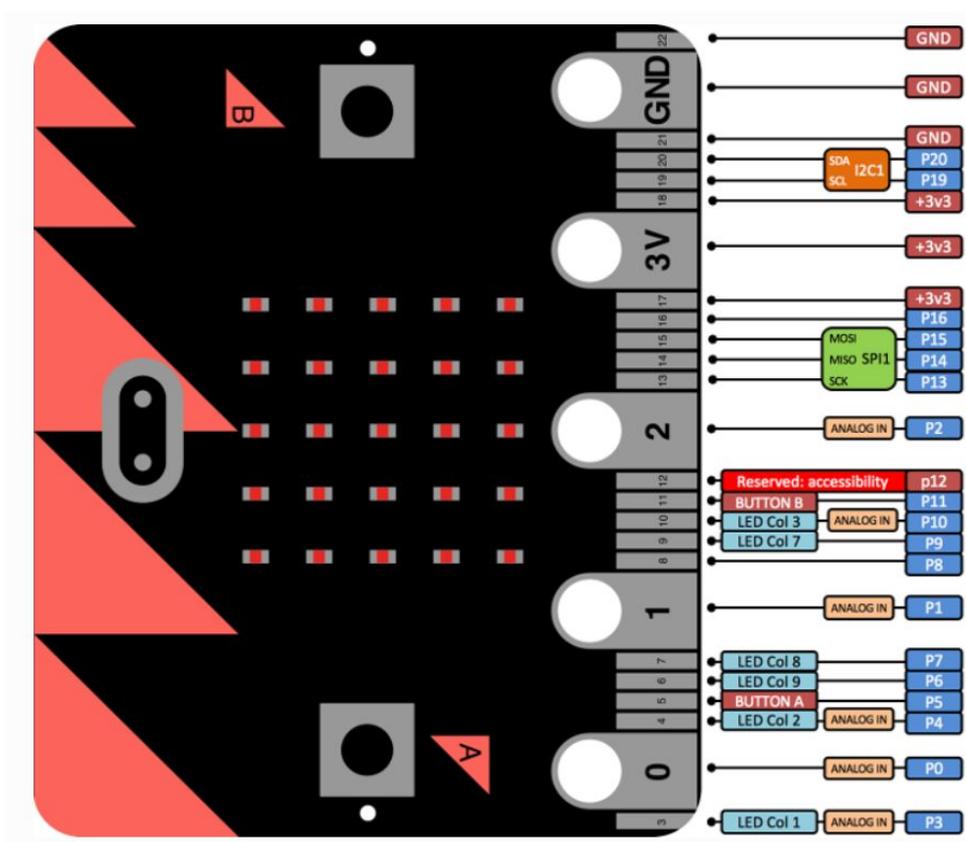
Сделав первые шаги в освоении BBC micro:bit, начинаешь ощущать ограниченность внутренних ресурсов микроконтроллера. Хочется подключить внешние датчики (например, [такой](#), [такой](#) или [такой](#)), моторчики, светодиоды, сделать микроконтроллер мозговым центром управляемой системы. И для этого у BBC micro:bit есть необходимый ресурс — пины ввода / вывода. *Pin* — в переводе с английского означает “штырь, булавка”. Тонкими торчащими вверх иголками выглядят пины на микроконтроллере Arduino. Здесь же они — плоские и гладкие, но традиционно называются пинами. В общей сложности их 25 штук — пять больших и 20 маленьких. К большим пинам подключение выполняется с помощью зажимов-крокодилов. Для подключения малых пинов удобно использовать специальную плату расширения, которую производит, например, фирма Kitronik.



[Подключение с помощью зажимов-крокодилов](#)

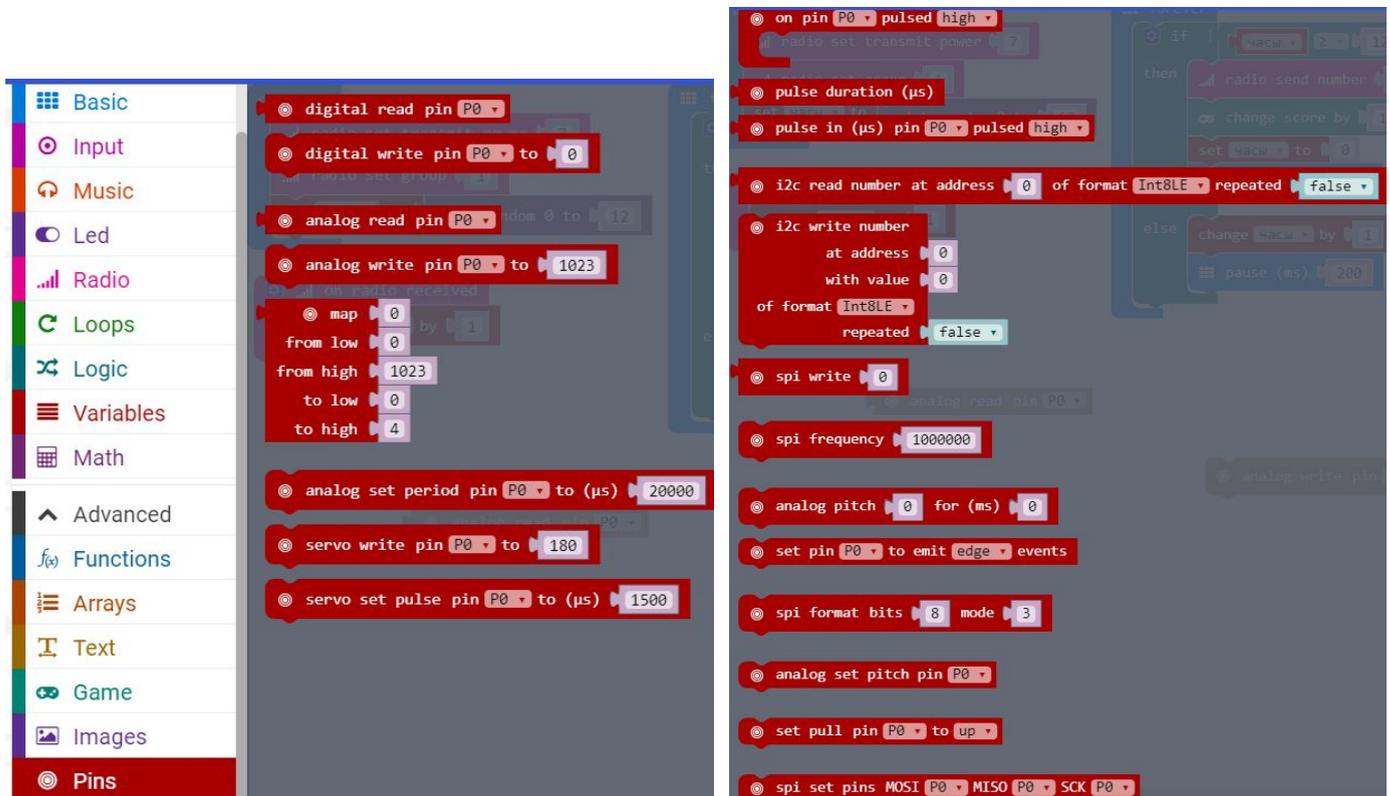
Использование платы расширения [Edge Connector Breakout Board for BBC micro:bit](#)

### Схема пинов микроконтроллера:



Некоторые пины допускают аналоговый ввод. Они помечены на схеме бежевой меткой. Для большинства возможны цифровой и аналоговый вывод. Аналоговый ввод и вывод осуществляется с помощью [ШИМ](#), при этом микроконтроллер обеспечивает 1024 уровня сигнала — от 0 до 1023.

### Команды для работы с пинами:



Рассмотрим примеры использования пинов в проектах, которые можно протестировать в эмуляторе.

### “Скорость реакции”

Это игровой проект, в котором игрок должен быстро реагировать на включение светодиода. Программа определяет время реакции как временной интервал между включением светодиода и касанием специально выделенной области на игровом поле. [Видеоролик](#) демонстрирует, как проходит игра для двух игроков.

Игровое поле делается из подручных материалов с использованием алюминиевой фольги, которая является хорошим и удобным в подобных ситуациях проводником. Подробную инструкцию можно посмотреть [здесь >>](#)

На [странице описания проекта](#) можно посмотреть [его код](#).

Но мы рассмотрим несколько другой вариант этой игры.

Игровое поле такое же, как описано выше, только отсутствует “кнопка” Старт.

При нажатии кнопки А на дисплее после обратного отсчёта времени и случайной временной паузы загорается столбик светодиодов — либо справа (x=4), либо слева (x=0). Если загорается левый столбик светодиодов, игрок должен коснуться фольги на левой стороне игрового поля, иначе — на правой стороне.

Если игрок всё сделал правильно, на дисплее появляется весёлая рожица (Happy) и проигрывается весёлая мелодия, иначе — грустная рожица (Sad) и печальная мелодия. Ошибкой также является

фальстарт, который фиксируется программой. При удачной игре на дисплее выводится время реакции — количество десятых долей секунды.

Тем, кто себя уверенно чувствует в программировании микроконтроллера, предлагается самостоятельно написать код для проекта.

Для тех, кто ещё не готов самостоятельно преодолеть трудности программирования, код программы приведён ниже.

The image shows two Scratch code blocks for a BBC micro:bit project. The left block is triggered by 'on button A pressed' and contains the following logic: show numbers 3, 2, and 1; clear the screen; set 'играем' (playing) to false and 'фальстарт' (false start) to false; pause for 1000ms + a random number between 0 and 1999; if 'фальстарт' is not true, then set 'сарт' (start) to 'running time (ms)', set 'играем' to true, stop animation, clear the screen, set 'случай' (chance) to a random number between 0 and 1; if 'случай' is 0, then a for loop from 0 to 4 plots points at (0, k) with a 100ms pause; else, a for loop from 0 to 4 plots points at (4, k) with a 100ms pause. The right block is triggered by 'on pin P1 pressed' and contains: if 'играем' is true, then set 'играем' to false, set 'end' to 'running time (ms)', if 'случай' is 0, then show a micro:bit icon, start a melody 'power down' once, pause for 1000ms, and show a number calculated as (end - сарт) / 100; else, show a micro:bit icon, start a melody 'dadadum' once; else, set 'фальстарт' to true, show a micro:bit icon, and start a melody 'dadadum' once.

Код для пина P2 такой же, как и для P1, только значение переменной “случай” надо изменить на 1.

Эту программу можно проверить в эмуляторе, щёлкая указателем мыши нужные пины.

На странице [“Проекты”](#) официального сайта BBC micro:bit можно познакомиться с простыми, но содержательными STEM-проектами, в которых используются пины микроконтроллера. Обратите внимание на кнопки, которые размещаются под окном с кодом программы:

The image shows three Scratch code blocks. The first block is triggered by 'on pin P0 pressed' and contains: set 't0' to 'event timestamp', show a 5x5 grid of LEDs. The second block is triggered by 'on pin P1 pressed' and contains: set 't1' to 'event timestamp', show a 5x5 grid of LEDs, set 'd' to 't1 - t0', and show the number 'd'. The third block is triggered by 'on start' and contains: show a 5x5 grid of LEDs. At the bottom right, there are three icons: a pencil (edit), a list (blocks), and a play button (run), with the play button circled in red.

1. Вызов редактора, в который загружен код программы
2. Код программы на языке Java Script загружается в текущее окно
3. Загрузка эмулятора
4. Скачивание hex-файла на компьютер

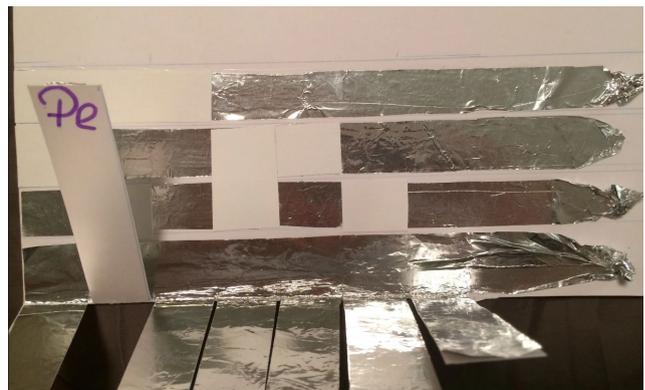
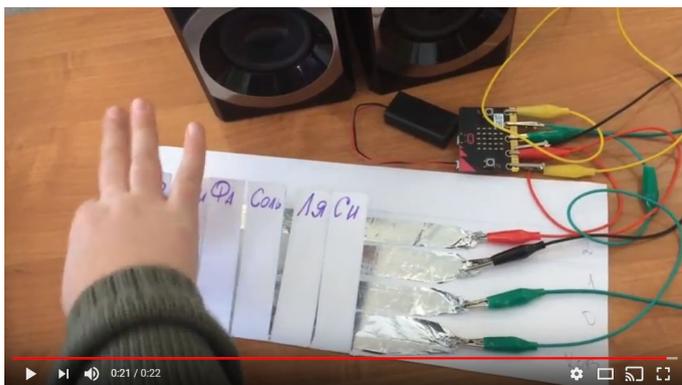
Таким образом вы можете просмотреть и протестировать программы в разных разделах страницы “Проекты”, в том числе, в разделе STEM.

Отдельно хочется остановиться на проекте, опубликованном Elena Astapenko в сообществе [STEM c BBC micro:bit](#).

### Электронное пианино от Elena Astapenko

Elena предложила конструкцию электронного пианино (точнее, одной октавы пианино), в которой семь “клавиш” подключены к пинам микроконтроллера, но при этом используются только три пина!

Остроумное решение заключается в том, что из фольги строится электрическая цепь, в которой задействованы все возможные сочетания значений цифровых сигналов на трёх пинах:



Слева — [видеоролик](#), в котором показана работа “пианино”. Справа — его внутренняя конструкция.

Таким остроумным способом можно обойтись без дополнительных устройств.

Код программы:

```
let line2 = 0
let line1 = 0
let line0 = 0
basic.forever(() => {
  line0 = pins.digitalReadPin(DigitalPin.P2)
  line1 = pins.digitalReadPin(DigitalPin.P8)
  line2 = pins.digitalReadPin(DigitalPin.P1)
  if (line0 || (line1 || line2)) {
    basic.pause(30)
    if (!(line2) && (!(line1) && line0)) {
      music.playTone(262, music.beat(BeatFraction.Half))
    }
    if (!(line2) && (line1 && !(line0))) {
      music.playTone(294, music.beat(BeatFraction.Half))
    }
    if (!(line2) && (line1 && line0)) {
      music.playTone(330, music.beat(BeatFraction.Half))
    }
    if (line2 && !(line1) && !(line0)) {
      music.playTone(349, music.beat(BeatFraction.Half))
    }
    if (line2 && !(line1) && line0) {
      music.playTone(392, music.beat(BeatFraction.Half))
    }
  }
})
```

```
    if (line2 && (line1 && !(line0))) {  
        music.playTone(440, music.beat(BeatFraction.Half))  
    }  
    if (line2 && (line1 && line0)) {  
        music.playTone(494, music.beat(BeatFraction.Half))  
    }  
}  
})
```

---

Это была последняя, восьмая остановка экспресс-курса. Много осталось неизведанного, но основа для дальнейших самостоятельных шагов, хочется надеяться, заложена.

Дерзайте, придумывайте, воплощайте. Делитесь своими идеями, достижениями, сомнениями и проблемами в сообществе единомышленников [STEM с BBC micro:bit](#).

**Удачи!**