

Синтезатор

В этом эксперименте мы создаем маленькую клавиатуру, на которой можно сыграть несколько нот.

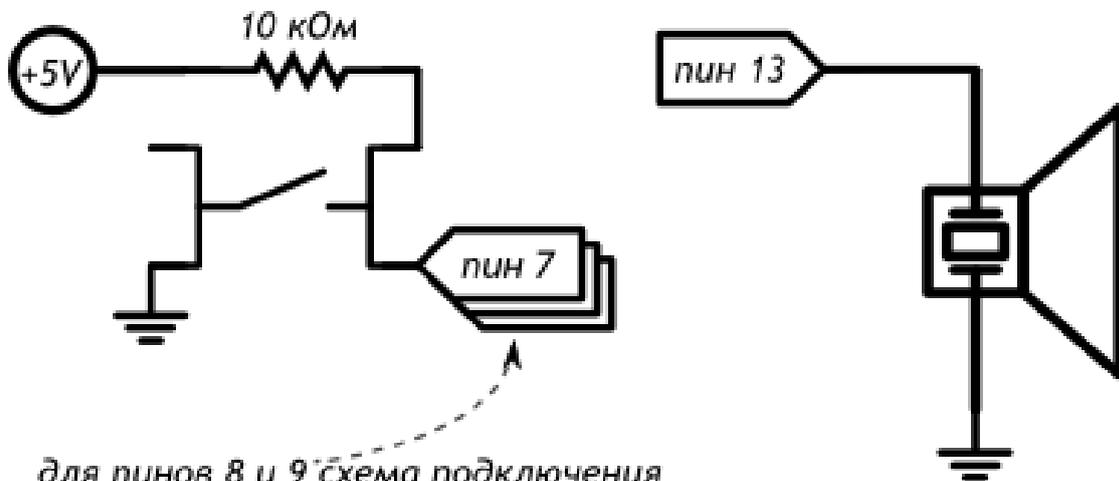
Список деталей для эксперимента

- 1 плата [Arduino Uno](#)
- 1 беспаячная [макетная плата](#)
- 1 [пьезопищалка](#)
- 3 тактовых [кнопки](#)
- 3 [резистора](#) номиналом 10 кОм
- 10 проводов [«папа-папа»](#)

Для дополнительного задания

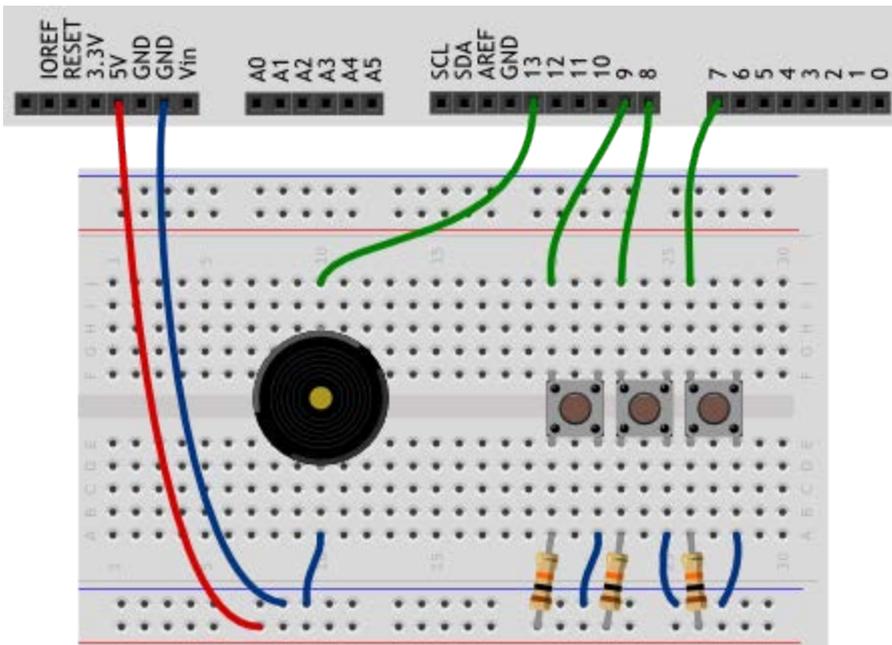
- еще 2 кнопки
- еще 2 резистора номиналом 10 кОм
- еще 2 провода

Принципиальная схема



для пинов 8 и 9 схема подключения кнопок точно такая же

Схема на макетке



Обратите внимание

- Ножки тактовой кнопки, расположенные с одной стороны, разомкнуты, когда кнопка не нажата. Ножки, расположенные друг напротив друга на противоположных сторонах макетки находятся на одной «рельсе». Воспользовавшись этим, мы можем расположить резистор с одной стороны макетки, а провод, подключаемый к порту Arduino, с другой стороны.
- В данном эксперименте мы подключаем кнопки по [схеме](#) с подтягивающим резистором.
- Для того, чтобы данный вариант программы работал, важно, чтобы кнопки были подключены к портам, находящимся рядом друг с другом, т.е. имеющим соседние номера.

Скетч

```
#define BUZZER_PIN 13 // пин с пищалкой (англ. «buzzer»)
#define FIRST_KEY_PIN 7 // первый пин с клавишей (англ. «key»)
#define KEY_COUNT 3 // общее количество клавиш

void setup()
{
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop()
{
  // в цикле бежим по всем номерам кнопок от 0-го по 2-й
  for (int i = 0; i < KEY_COUNT; ++i) {
    // на основе номера кнопки вычисляем номер её пина
    int keyPin = i + FIRST_KEY_PIN;
```

```

// считываем значение с кнопки. Возможны всего 2 варианта:
// * высокий сигнал, 5 вольт, истина — кнопка отпущена
// * низкий сигнал, земля, ложь — кнопка зажата
boolean keyUp = digitalRead(keyPin);

// проверяем условие «если не кнопка отпущена». Знак «!»
// перед булевой переменной означает отрицание, т.е. «не».
if (!keyUp) {
    // рассчитываем высоту ноты в герцах в зависимости от
    // клавиши, которую рассматриваем на данном этапе цикла.
    // Мы получим значение 3500, 4000 или 4500
    int frequency = 3500 + i * 500;

    // Заставляем пищалку пищать с нужной частотой в течение
    // 20 миллисекунд. Если клавиша останется зажатой, пищалка
    // вновь зазвучит при следующем проходе loop, а мы услышим
    // непрерывный звук
    tone(BUZZER_PIN, frequency, 20);
}
}
}

```

Пояснения к коду

- Благодаря тому, что в начале программы мы определили `FIRST_KEY_PIN` и `KEY_COUNT`, мы можем подключать произвольное количество кнопок к любым идущим друг за другом цифровым пинам, и для корректировки программы нам не придется менять параметры цикла `for`. Изменить понадобится лишь эти константы:
 - цикл в любом случае пробегает от 0 до `KEY_COUNT`;
 - перед считыванием порта мы задаем смещение на номер первого используемого порта — `FIRST_KEY_PIN`.
- Функция `digitalRead(pin)` возвращает состояние порта, номер которого передан ей параметром `pin`. Это может быть состояние `HIGH` или `LOW`. Или, выражаясь иначе: высокое напряжение или низкое, 1 или 0, `true` или `false`
- Поскольку мы получаем с порта одно из двух состояний, мы сохраняем его в переменную уже знакомого нам типа `boolean`, и можем работать с ней как с логическим значением.
- Мы используем логический оператор отрицания «не» `!`. Если `keyUp` имеет значение 0, выражение `!keyUp` будет иметь значение 1 и наоборот.
- Поскольку мы собрали схему с подтягивающим резистором, при нажатии кнопки мы будем получать на соответствующем порте 0.
- Действия, описанные в условном выражении `if`, выполняются, когда его условие имеет значение «истина» (единица). Поэтому для выполнения действия по нажатию, мы инвертируем сигнал с кнопки.

Вопросы для проверки себя

1. Почему мы не настраивали порты, к которым подключены кнопки, как INPUT, но устройство работает?
2. Каким образом мы избежали написания отдельного кода для чтения каждой кнопки?
3. Почему разные «ноты», издаваемые пицалкой, звучат с разной громкостью?
4. Для чего мы использовали оператор логического отрицания !?

Задания для самостоятельного решения

1. Сделайте так, чтобы наше пианино звучало в диапазоне от 2 кГц до 5 кГц.
2. Добавьте еще 2 кнопки и измените программу так, чтобы можно было извлечь 5 различных нот.