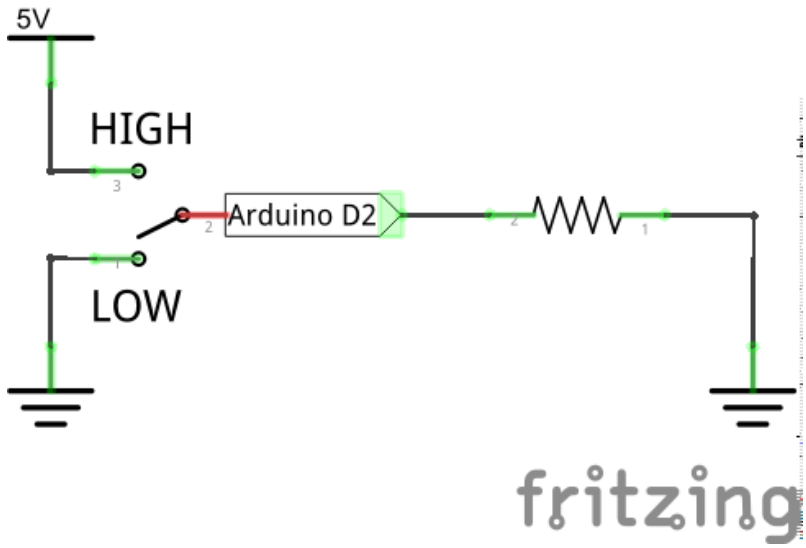
Тема 2. Работа с цифровыми сигналами

Урок 2.1. Выводы общего назначения (GPIO)

У контроллера Arduino Uno есть множество выводов, обозначенных цифрами от 0 до 13 и от A0 до A5. Каждым из этих выводом мы можем управлять при помощи программы.

В терминах цифровой электроники, управлять - значит менять на выводе уровень напряжения. Другими словами, все что мы можем сделать с помощью программы - это соединить желаемый вывод либо с контактом питания +5В, либо с землей Gnd. Изобразим это на принципиальной схеме.

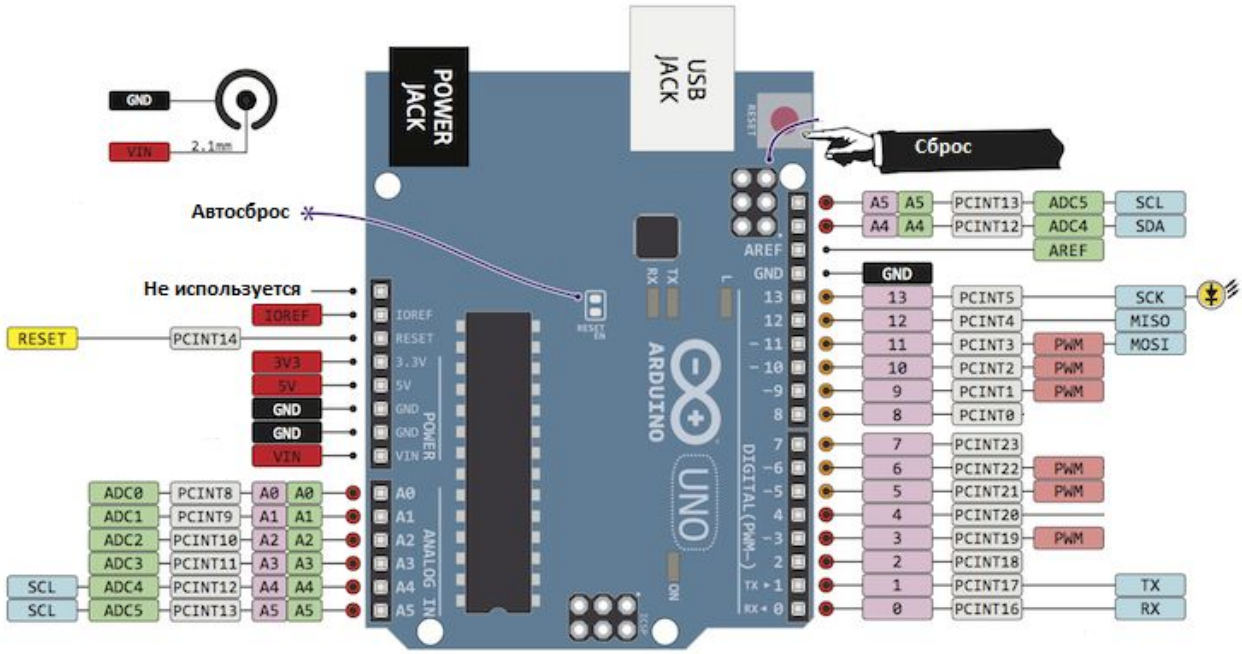


На схеме имеется резистор, соединенный справа с землей. Контакт Arduino D2 и переключатель прямо за ним символизируют внутреннее устройство контроллера Arduino Uno. Если мы в программе передадим на вывод №2 высокий уровень сигнала **HIGH**, то Arduino соединит этот вывод с питанием +5В, и через резистор побежит ток. Если же мы передадим сигнал **LOW**, то контроллер соединит вывод №2 с землей, и с обоих концов от резистора окажутся равные потенциалы - ток никуда не побежит.

Далее в уроках мы узнаем как передавать сигналы LOW и HIGH на выводы Arduino, и научимся управлять разными устройствами.

2.1.1. Схема выводов Arduino Uno (редакция 3)

Помимо базовой функции управления цифровым сигналом, Arduino имеет и другие интересные функции, помогающие соединять контроллер с другими устройствами. Ниже приведена схема выводов Arduino Uno, которую ещё часто называют “распиновкой”.



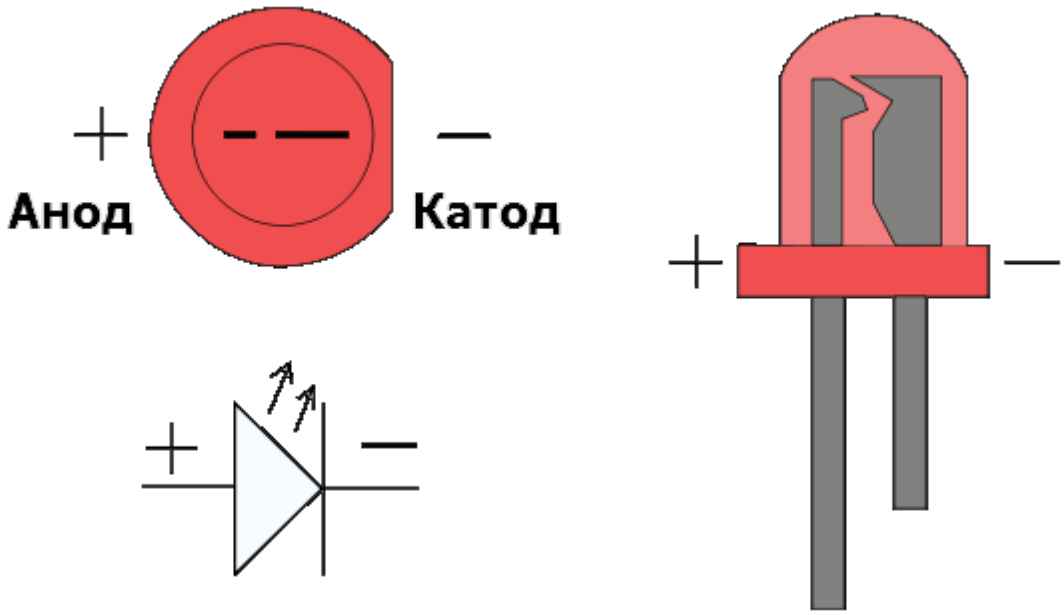
	Отрицательный вывод питания (земля, ground)
	Положительный вывод питания (+3.3В и +5В)
	Управляющие выводы (сброс)
	Цифровые выводы (включен/выключен)
	Аналоговые входы (снятие показаний с датчиков)
	ШИМ выходы (плавное управление двигателями)
	Интерфейсы (UART, I2C, SPI) (соединение с другими микроэлектронными устройствами)
	Обозначение в редакторе Arduino IDE

Урок 2.2. Светодиодный индикатор (LED)

2.2.2. Устройство светодиода

Устройства, называемые индикаторами, позволяют простому электронному прибору или сложному роботу общаться с человеком на языке примитивных сигналов. Существует множество различных по принципу действия индикаторов, которые преследуют одну цель - сообщить человеку о состоянии системы. Так, например, индикатор питания монитора дает понять подключен ли последний к источнику энергии.

Самый простой индикатор, который мы используем в первой лабораторной работе, называется светоизлучающим диодом. Это устройство, представляет собой полупроводниковый прибор, способный излучать свет при пропускании через него электрического тока в прямом направлении (от анода к катоду). Ниже приведена схема типичного светодиода с линзой.



Для того чтобы правильно включить светодиод в электрическую цепь, необходимо отличать катод от анода. Сделать это можно по двум признакам:

1) Анод светодиода имеет более длинный проводник
2) Со стороны катода, корпус светодиода немного срезан

В современной микроэлектронике применяются миниатюрные светодиоды для поверхностного монтажа. Такие индикаторы, например, имеются на Arduino Uno для информирования пользователя о состоянии системы.

2.2.3. Функция вывода цифровых данных

Для выводы высокого или низкого уровня сигнала на один из контактов общего назначения, в редакторе программ Arduino IDE используется функция **digitalWrite**. Вызов этой функции имеет вид:

digitalWrite(номер_контакта, уровень_сигнала);

где аргумент **уровень_сигнала** может принимать два значения: **HIGH** (высокий, +5В) или **LOW** (низкий).

Но чтобы контакт №2 именно выводил сигнал, а не принимал его, потребуется использовать специальную функцию для установки режима контактов - **pinMode**. Формат этой функции таков:

pinMode(номер_контакта, режим_контакта);

где аргумент режим_контакта может принимать значения: **OUTPUT** (вывод) и **INPUT** (ввод).

С учетом указанного выше, чтобы установить на выводе №2 высокий уровень сигнала достаточно запустить следующую программу.

```
void setup() {
  pinMode( 2, OUTPUT );   // установка 2-го контакта в режим вывода
  digitalWrite( 2, HIGH );   // перевод вывода №2 в активное состояние
}

void loop() {
}
```

Задания

Задание 1. Мигающий светодиод

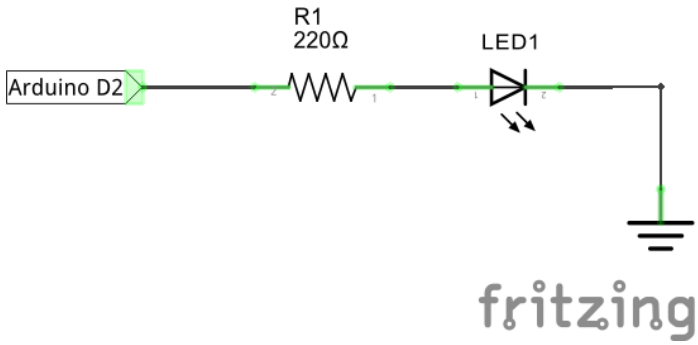
Результат работы
Сразу после запуска программы, светодиод начинает мигать один раз в секунду.

Используемые компоненты:

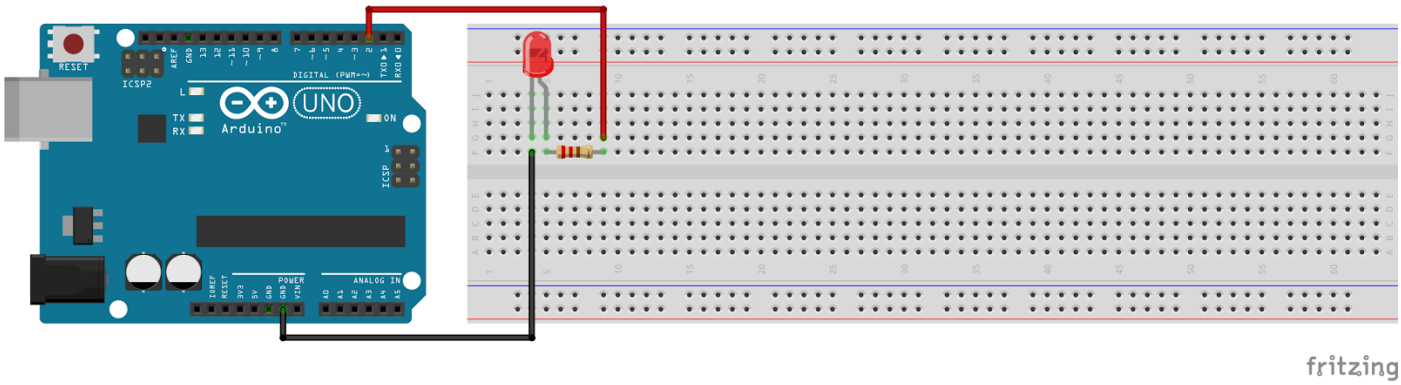
	
Светодиод, 1шт	Резистор 200Ом, 1шт

Важно отметить, что напряжение питания светодиода варьируется от 1.85 до 2.5 вольт, при рекомендуемой силе тока 20мА. Для правильной работы прибора, в цепь следует добавить ограничивающий резистор (от 200Ом до 500Ом). Ниже представлена электрическая схема подключения светодиода к Arduino Uno, а также макет собранного устройства.

Принципиальная схема



Внешний вид макета



Чтобы мигать светодиодом, мы будем последовательно зажигать его, передавая на ногу №2 сигнал **HIGH**, а затем гасить с помощью сигнала - **LOW**. Между включением и выключением светодиода обязательно нужно поставить задержку в несколько сотен миллисекунд, иначе мы не заметим как светодиод будет мигать. Вспомним, что контроллер Arduino штука очень быстрая, он может включать и выключать светодиод тысячи раз в секунду! Это во много раз чаще, чем мигает обычная лампа накаливания.

```
int led = 2;

void setup() {
  pinMode(led, OUTPUT);     // установка 2-го контакта в режим вывода
}

void loop() {
  digitalWrite(led, HIGH);   // перевод вывода №2 в активное состояние
  delay(1000);               // пауза 1-секунда
  digitalWrite(led, LOW);    // перевод вывода №2 в неактивное состояние
  delay(1000);               // пауза 1-секунда
}
```

Задание 2. Одновременная активация двух светодиодов

Результат работы

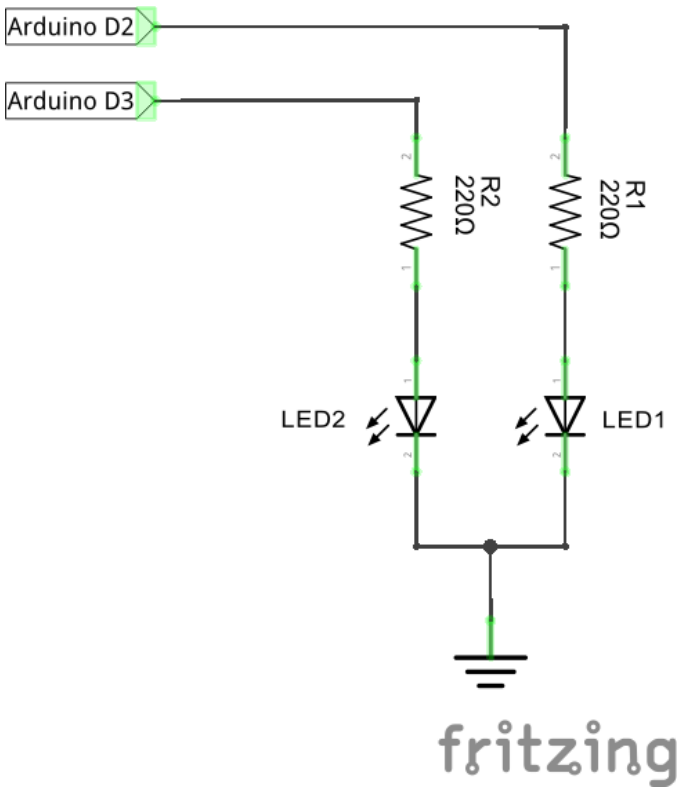
Сразу после запуска программы, оба светодиода начинают синхронно мигать.

Используемые компоненты:

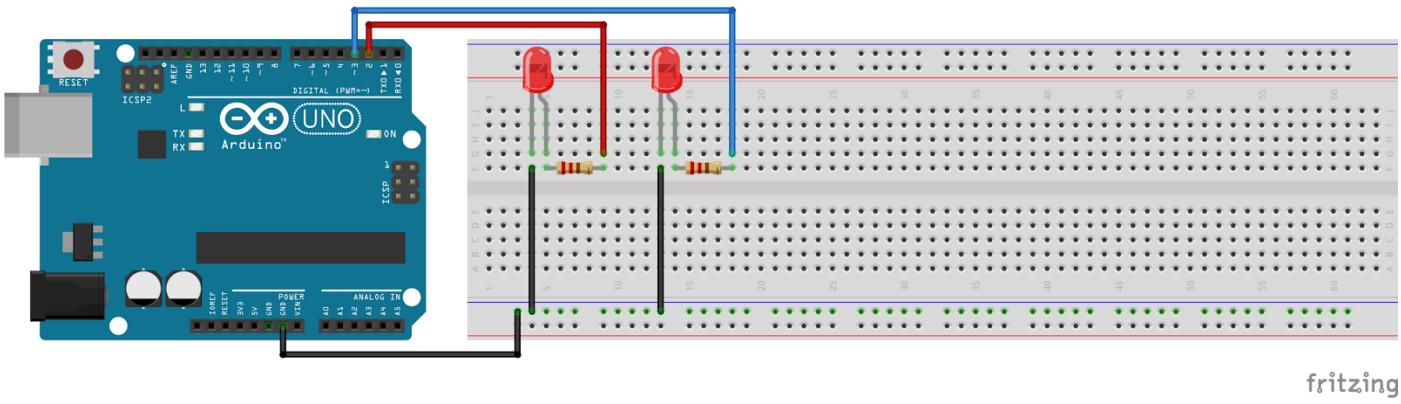
	
Светодиод, **2шт**	Резистор 200Ом, **2шт**

Схема включения второго светодиода аналогична схемам, представленным в предыдущем задании. Второй светодиод подключается к выводу **№3**.

Принципиальная схема



Внешний вид макета



Управляющая программа имеет вид:

```
int led_1 = 2;
int led_2 = 3;

void setup() {
   pinMode(led_1, OUTPUT);     // установка 2-го контакта в режим вывода
   pinMode(led_2, OUTPUT);     // установка 3-го контакта в режим вывода
}

void loop() {
   digitalWrite(led_1, HIGH);   // перевод вывода №2 в активное состояние
   digitalWrite(led_2, HIGH);   // перевод вывода №3 в активное состояние
   delay(1000);              // пауза 1-секунда

   digitalWrite(led_1, LOW);    // перевод вывода №2 в неактивное состояние
   digitalWrite(led_2, LOW);    // перевод вывода №3 в неактивное состояние
   delay(1000);              // пауза 1-секунда
}
```

Задание 3. Проблесковые маячки (самостоятельно)

Результат работы

После запуска программы, два светодиода начинают мигать с периодом 0.3 секунды. При этом, если один светодиод горит, второй в этот момент должен быть погашен.

Задание 4. Светофор (самостоятельно)

Результат работы

После запуска программы, три светодиода начинают зажигаться в следующей последовательности:

1) Все гаснут, зеленый зажигается на 3 секунды.
2) Зеленый мигает 3 секунды.
3) Все гаснут, желтый зажигается на 1 секунду
4) Все гаснут, красный зажигается на 3 секунды

Урок 2.3. Зуммер

2.3.1. Описание устройства зуммера (buzzer)

На предыдущем занятии мы познакомились с одним из самых распространенных электронных индикаторов - светоизлучающим диодом, или просто - светодиодом. Наряду со световой индикацией, часто используют и другой метод - звуковую индикацию. Иными словами, электронное устройство или робот, может привлечь внимание человека, и сообщить о своем состоянии не только светом, но и звуком.

Часто, для звуковой индикации используется специальный прибор, называемый зуммером. В отличие от динамика, в зуммере есть небольшой встроенный генератор звуковой волны определенной частоты, поэтому он может выдать только одну ноту. Это значительно упрощает работу с ним. Нам достаточно просто подать на него питание, и зуммер запищит.

В основе работы большинства зуммеров лежит пьезоэлектрический эффект, поэтому такие приборы называются пьезоэлектрическими звукоизлучателями.

Задания

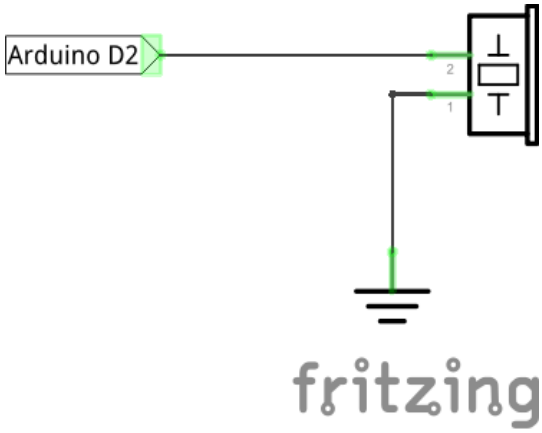
Задание 1. Звуковой код

Результат работы
После запуска программы, зуммер начинает выдавать по три быстрых звуковых импульса, каждые 2 секунды. Период каждого импульса - 0.6 сек.

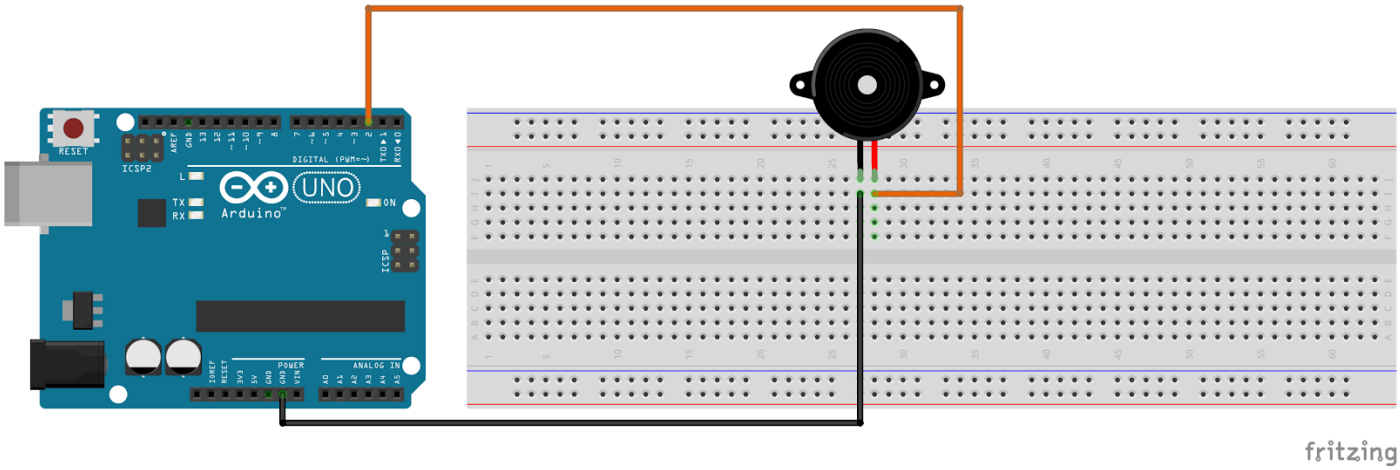
Используемые компоненты:


Зуммер - 1шт

Принципиальная схема



Внешний вид макета



Программа

```
int buzz = 2;
int k = 0;

void setup() {
   pinMode(buzz, OUTPUT);     // установка 2-го контакта в режим вывода
}

void loop() {
   for( k=0; k<3; k=k+1) {    // цикл из трех итераций
      digitalWrite(buzz, HIGH);   // перевод вывода №2 в активное состояние
      delay(300);              // пауза 1-секунда
      digitalWrite(buzz, LOW);    // перевод вывода №2 в неактивное состояние
      delay(300);              // пауза 1-секунда
   }
   delay(2000);
}
```

Задание 2. Сигнал SOS (самостоятельно)

Результат работы

После запуска программы, зуммер бесконечно выдает сигнал SOS: три коротких сигнала (0.5 сек), затем три длинных (1 сек), и, наконец, снова три коротких.

Урок 2.4. Механический выключатель - кнопка

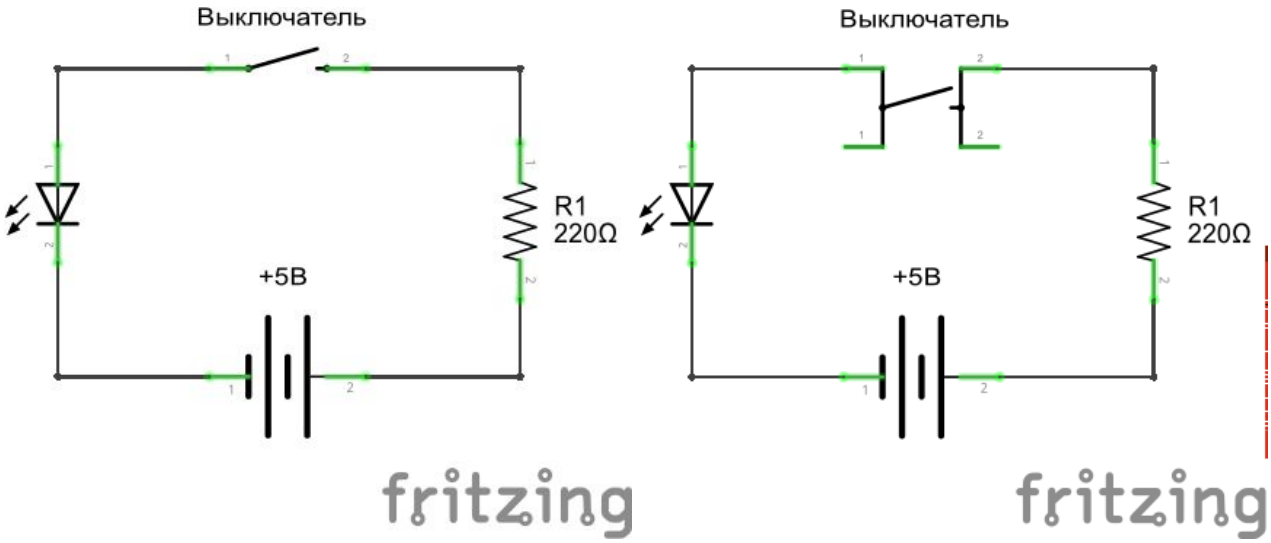
2.4.1. Описание устройства выключателя

Выключатель - это прибор, который позволяет замыкать и размыкать электрическую цепь. При этом, смена состояния выключателя может происходить разными способами, в зависимости от типа устройства. Механические выключатели наиболее распространены, и используются непосредственно человеком. К такому типу относятся различные тумблеры, кнопки, клавиши и рубильники. Электромагнитные и электронные, напротив, применяются в автоматических системах, и управляются при помощи электрических сигналов. Самым известным электромагнитным выключателем является реле (relay). Примером электронного выключателя может служить транзистор (transistor).

В нашей работы мы будем использовать простой механический выключатель (push-button), который встречается в двух вариантах:

а) справа - в виде кнопки с двумя выводами;

б) слева - с четырьмя выводами.



По сути, четыре вывода нужны лишь для удобства монтажа.

2.4.2. Функция ввода цифровых данных

Теперь попробуем запрограммировать Arduino для работы с кнопками. Для чтения цифрового сигнала с одного из контактов Arduino, необходимо воспользоваться функцией: **digitalRead**. Формат функции имеет вид:

результат = digitalRead(номер_контакта);

после вызова этой функции, переменная **результат** будет хранить уровень цифрового сигнала, детектируемый на соответствующем контакте.

Задания

Задание 1. Выключатель и зуммер

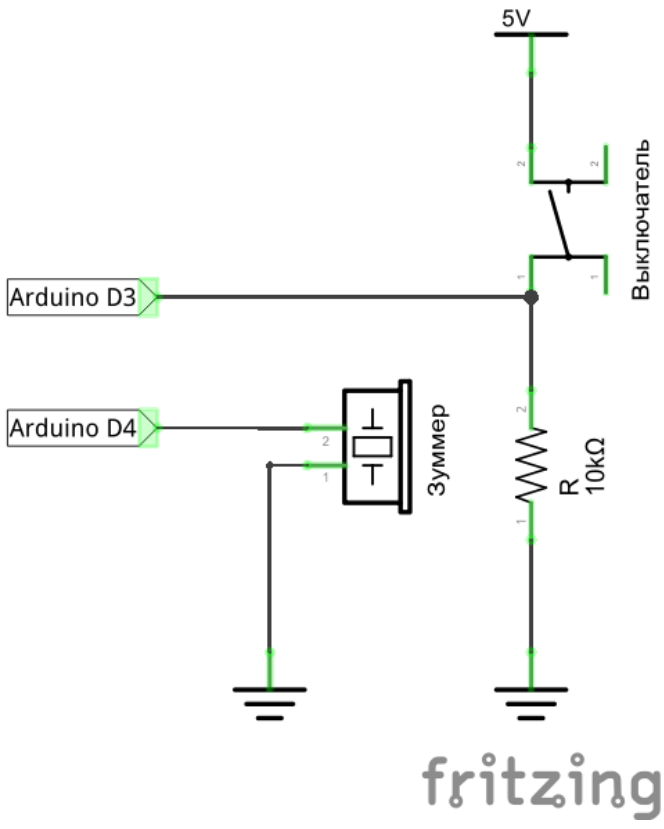
Результат работы

После запуска, программа начинает постоянно проверять состояние кнопки. По нажатию кнопки, зуммер включается, а при отпускании, напротив - выключается.

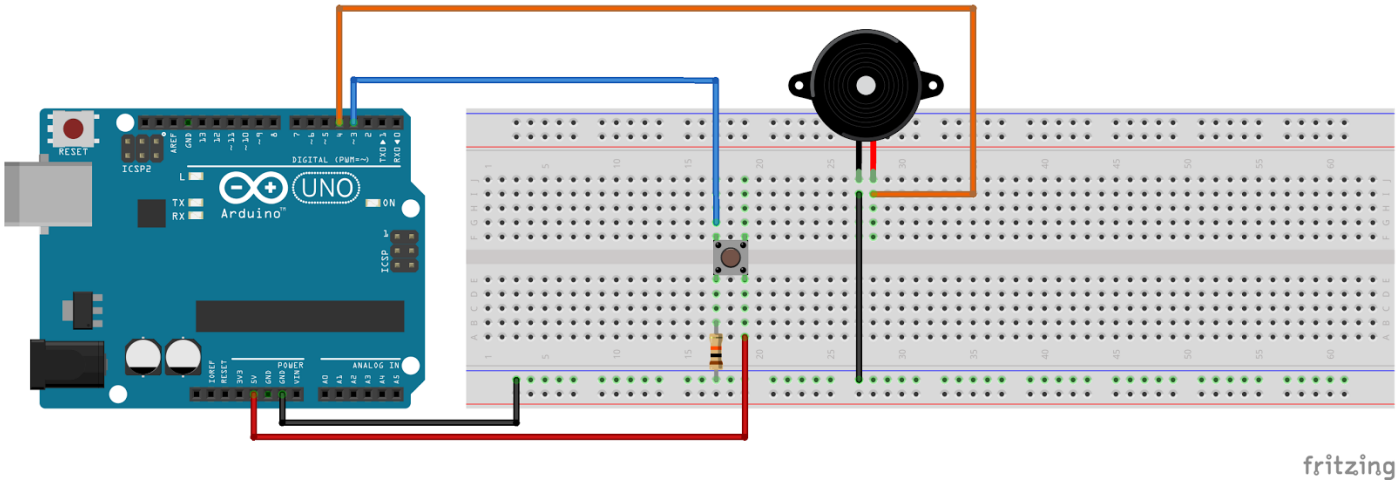
Используемые компоненты

		
Зуммер - 1шт	Кнопка - 1шт	Резистор 10КОм - 1шт

Принципиальная схема



Внешний вид макета



Программа

```
const int btn = 3;
const int buzz = 4;
byte val = 0;

void setup() {
   pinMode(btn, INPUT);     // установка 3-го контакта в режим ввода
   pinMode(buzz, OUTPUT);     // установка 4-го контакта в режим вывода
}

void loop() {
   val = digitalRead(btn);
   if( val == HIGH){    // условие
      digitalWrite(buzz, HIGH);   // перевод зуммера в активное состояние
   } else {
      digitalWrite(buzz, LOW);    // перевод зуммера в неактивное состояние
   }
}
```

Задание 2. Управление включением и выключением (важно)

Результат работы

В схеме присутствуют одна кнопка и зуммер. При нажатии на одну из кнопок, зуммер начинает издавать звук, до тех пор, пока кнопка не будет нажата второй раз.

В этом примере мы введем дополнительную переменную - **state**, в которой будем хранить текущее состояние зуммера. Нажимаем кнопку один раз: в переменную состояния записывается 1, нажимаем второй - 0, третий - снова 1, и т.д. И уже в зависимости от того, какое значение принимает эта переменная, мы будем активировать, или деактивировать зуммер. Вот как это будет выглядеть на языке си.

Программа

```
const int btn = 3;
const int buzz = 4;
byte val = 0;
bool state = 0;

void setup() {
   pinMode(btn, INPUT);     // установка 3-го контакта в режим ввода
   pinMode(buzz, OUTPUT);     // установка 4-го контакта в режим вывода
}

void loop() {
   val = digitalRead(btn);

   if( val == HIGH){    // условие смены состояния
      state = !state;
      delay(200);
   }

   if( state == true){    // условие активации зуммера
      digitalWrite(buzz, HIGH);   // перевод зуммера в активное состояние
   } else {
      digitalWrite(buzz, LOW);    // перевод зуммера в неактивное состояние
   }
}
```

Задание 3. Запуск и остановка проблескового маячка и сирены кнопкой (самостоятельно)

Результат работы

В схеме присутствуют две кнопки, светодиод и зуммер. После нажатии на одну кнопку, светодиод начинает мигать, а зуммер издавать периодический сигнал. Выключается маячок с помощью второй кнопки.

Задание 4. Генератор сигнала SOS (самостоятельно)

Результат работы

В схеме присутствует одна кнопка, светодиод и зуммер. После нажатия на кнопку, зуммер начинает передавать сигнал SOS. Параллельно с зуммером, светодиод отражает сигнал SOS световыми импульсами. Выключается генератор с помощью второго нажатия кнопки.

2.5. Итоговые задания

Полученные знания дают нам возможность сделать несколько интересных устройств.

Устройство 1. Игра “Ковбои”

Результат работы
В схеме присутствуют две кнопки, два светодиода и зуммер. После запуска программы зуммер начинает издавать короткие импульсы через неравные промежутки времени. Каждый игрок должен как можно быстрее нажать на кнопку сразу после сигнала зуммера. У игрока, нажавшего свою кнопку первым, на 2 секунда зажигается светодиод.

Устройство 2. Кодовый замок

Результат работы
В схеме присутствуют три кнопки, зеленый и красный светодиоды, а также зуммер. После запуска программы горит красный светодиод. Пользователю необходимо нажать три кнопки в правильной комбинации. Если кнопки нажаты правильно, загорается зеленый светодиод. Если кнопки нажаты неправильно - зуммер издает пять длинных импульсов.