

Практическая работа 6

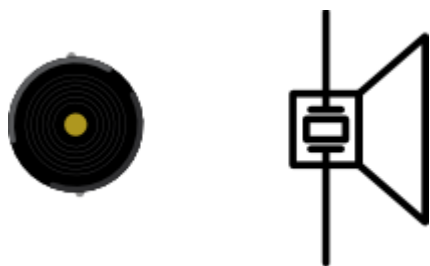
Проект Терменвокс

В этом эксперименте мы имитируем действие музыкального инструмента терменвокс: изменяем высоту звучания бесконтактным путем, больше или меньше закрывая от света фоторезистор.

Оригинальный инструмент был изобретён ещё в 1920 году, Львом Сергеевичем Терменом, человеком с непростой и насыщенной судьбой. А сейчас мы имеем возможность воспроизвести изобретение с помощью нехитрой электроники.

Прочитайте теоретические сведения

Пьезоизлучатель звука (англ. buzzer) переводит переменное напряжение в колебание мембраны, которая в свою очередь создаёт звуковую волну.



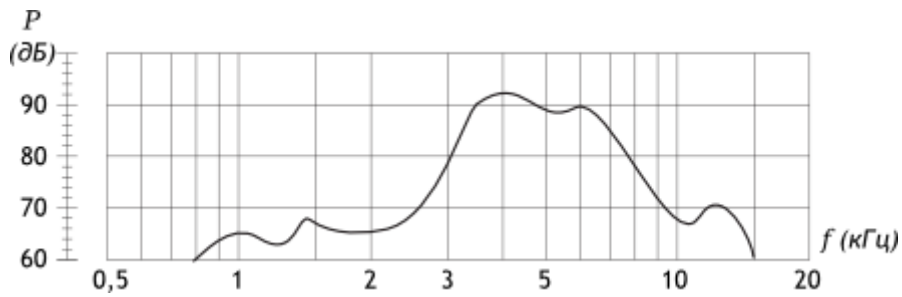
Иначе говоря, пьезодинамик — это конденсатор, который звучит при зарядке и разрядке.

Основные характеристики

Рекомендуемое (номинальное) напряжение	V	Вольт
Громкость (на заданном расстоянии)	P	Децибелл
Пиковая частота	f_p	Герц
Ёмкость	C	Фарад

Амплитудно-частотная характеристика

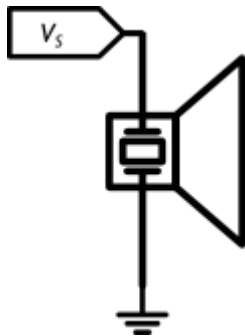
Амплитудно-частотная характеристика (АЧХ) определяет громкость звука в зависимости от частоты управляющего сигнала, который и определяет высоту звучащей ноты.



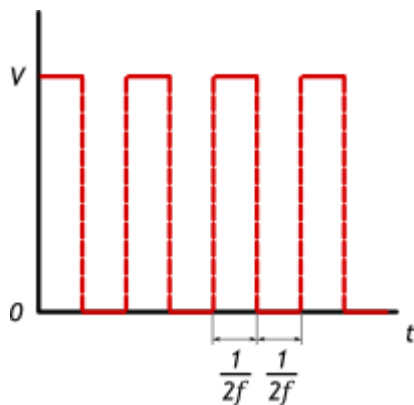
Идеальная АЧХ — это прямая, т.е. одинаковая громкость вне зависимости от частоты. Но мир не идеален и разные виды излучателей имеют разные отклонения от идеала.

Подключение напрямую

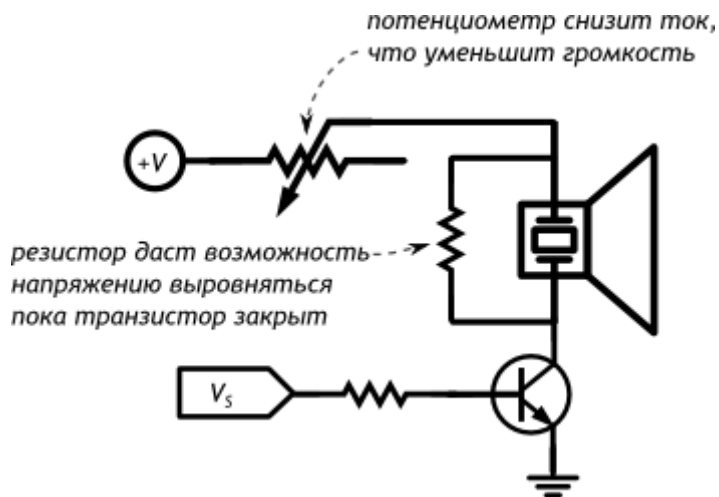
Пьезодинамик потребляет всего пару мА, поэтому можно смело подключать его прямо к микроконтроллеру



Для звучания нужно подавать на динамик квадратную волну. Какой частоты будет волна, такой частоты будет и звук



Подключение с регулировкой громкости



Фоторезистор (или LDR) — компонент, меняющий сопротивление в зависимости от количества света падающего на него. В полной темноте он имеет максимальное сопротивление в сотни килоОм, а по мере роста освещённости сопротивление уменьшается до десятков килоОм.

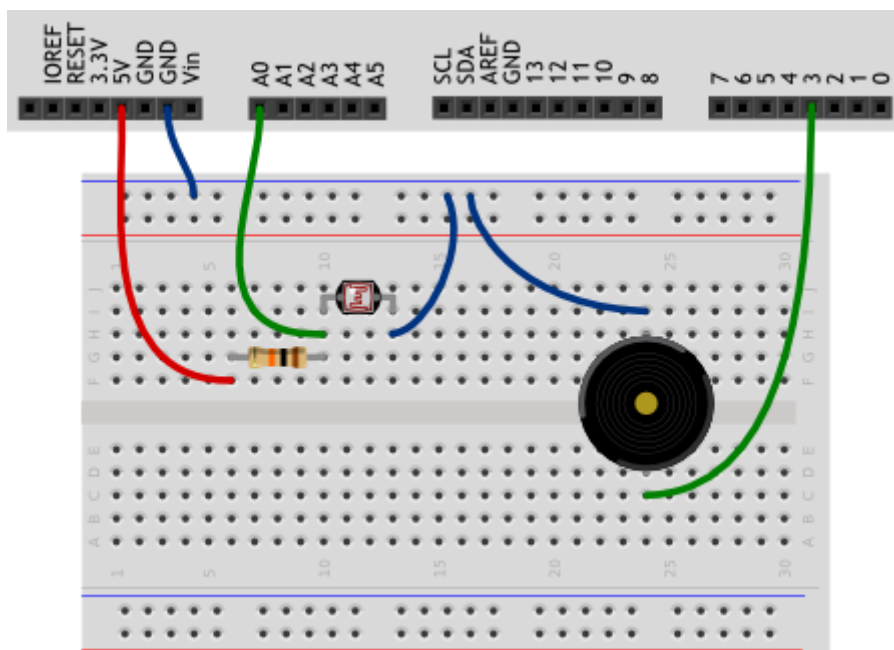
Задание 1. Ответить на вопросы

1. Опишите принцип работы пьезодинамика.
2. Как пьезодинамик обозначается на схеме?
3. Как подключить пьезодинамик так, чтобы можно было регулировать громкость?
4. Что такое фоторезистор?

Задание 2. Список деталей для эксперимента

1. 1 плата Arduino Uno
2. 1 беспаячная макетная плата
3. 1 пьезодинамик
4. 1 резистор номиналом 10 КОм
5. 6 проводов «папа-папа»
6. Фоторезистор

Схема на макетной плате:



1. Зарисуйте принципиальную схему установки.

Обратите внимание

✓ В данной схеме мы используем резистор нового номинала, посмотрите таблицу маркировки, чтобы найти резистор на 10 кОм или воспользуйтесь мультиметром

✓ Полярность фоторезистора, как и обычного резистора, не играет роли. Его можно устанавливать любой стороной

✓ В данном упражнении мы собираем простой вариант схемы включения пьезодинамика

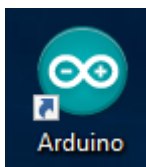
✓ Полярность пьезопищалки роли не играет: вы можете подключать любую из ее ножек к земле, любую к порту микроконтроллера (если у нее 2 ножки) Следите за записями на плате: VCC – подключить к порту VCC, GND – к порту GND.

✓ На Arduino Uno использование функции tone мешает использованию ШИМ на 3-м и 11-м портах. Зато можно подключить ее к одному из них

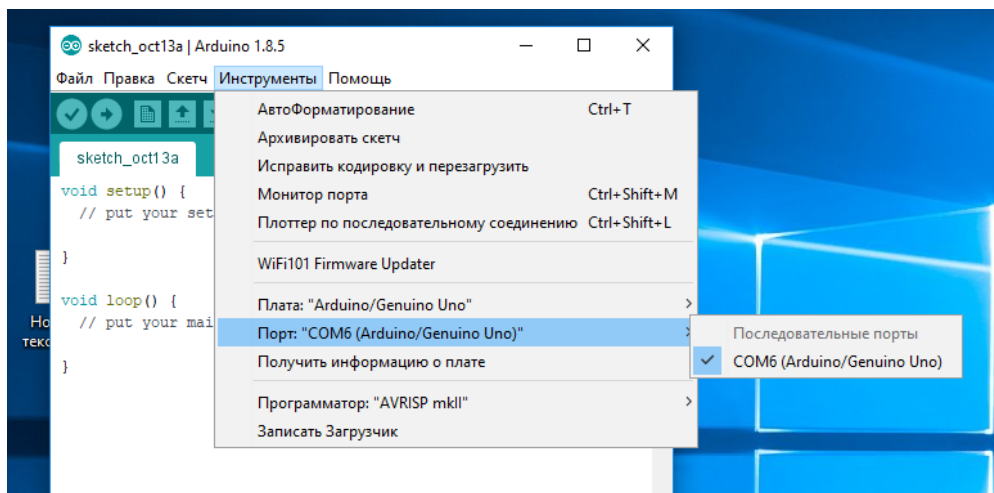
✓ Вспомните как устроен делитель напряжения: фоторезистор помещается в позицию R2 — между аналоговым входом и землей. Так мы получаем резистивный фотосенсор.

Задание 3. Программирование микроконтроллера

1. Запустите приложение



2. Убедитесь, что выбран нужный порт



Функции с возвращаемыми значениями

До сих пор мы использовали только переменные, которые так или иначе зависели только от номера вызова функции `loop`, т.е. так или иначе зависящие от времени, прошедшего с момента старта Arduino.

Это интересно, но не даёт таких возможностей к написанию программ, как получение значений из-вне. Допустим, к Arduino подключён какой-то сенсор: датчик освещённости, датчик газа, простой потенциометр или что-то ещё. Как получить его показания и использовать их в программе для чего-то полезного? Если говорить о сенсорах с аналоговым сигналом, для получения показаний с них существует встроенная функция `analogRead`. Давайте воспользуемся ей, чтобы сделать программу для устройства, которое изменяет яркость свечения светодиода, подключённого к 5-му пину в зависимости от поворота ручки потенциометра, подключённого к пину `A0`.

```

#define LED_PIN 5

#define POT_PIN A0

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

void loop()
{
    int value = analogRead(POT_PIN);

    analogWrite(LED_PIN, value / 4);
}

```

Первое, что мы видим — это макроопределение пина с потенциометром:

```
#define POT_PIN A0
```

В качестве значения используется не просто число, а нечто с именем A0. Оно написано большими буквами, поэтому можно предположить, что это встроенное макроопределение. И это действительно так! A0 для Arduino Uno, например, определено как 14. Для других плат значение может быть другим. Мы не хотим помнить об этих тонкостях, поэтому просто использовали A0. Это всегда означает «нулевой аналоговый вход».

Таким образом в качестве значения макроопределения мы использовали другое макроопределение. Так делать можно и это довольно распространённая практика.

Теперь рассмотрим loop. В первой строке определяется переменная value, а в качестве значения ей присваивается значения выражения analogRead(POT_PIN). **Но ведь это вызов функции, а не арифметическое выражение!**

Совершенно верно. Некоторые функции помимо того, что делают что-то полезное умеют так же возвращать значение обратно, в вызывающий код.

Функции вроде `pinMode` или `analogWrite` не возвращают ничего, по задумке их автора, а вот `analogRead` возвращает некоторое целочисленное значение.

Чтобы понять какие аргументы функция принимает, возвращает ли она что-нибудь и если возвращает, то что, следует обращаться к документации на эту функцию.

Что касается `analogRead`, она принимает один аргумент: номер пина, с которого необходимо считать значение аналогового сигнала. А возвращает эта функция значение от 0 до 1023, где:

- ✓ Входное напряжение в 0 В возвращается как 0
- ✓ 2 В возвращается как 409
- ✓ 2,5 В возвращается как 512
- ✓ И так далее, до напряжения в 5 В, которому ставится в соответствие 1023

Таким образом в первой строке `loop` мы просто считываем сигнал с потенциометра, получая угол поворота его ручки в виде целого числа в пределах от 0 до 1023.

Как мы помним, функция `analogWrite`, которой мы пользуемся для управления яркостью светодиода ожидает целое число от 0 до 255 в качестве второго аргумента. Но у нас оказалась переменная с другим диапазоном. Что делать? Просто поделить значение на 4. Так наш диапазон будет смасштабирован до того, который подходит для `analogWrite`. Максимуму из одного диапазона равному 1023 станет соответствовать максимум из другого диапазона: $1023 / 4 = 255$.

Вспоминая о компактной записи, мы можем сделать наш `loop` чуть лаконичнее:

```
void loop()
{
    analogWrite(LED_PIN, analogRead(POT_PIN) / 4);
}
```

3. Наберите в редакторе кода следующий код программы:

```
sketch_oct16a §
// даём имена для пинов с пьезопищалкой (англ. buzzer) и фото-
// резистором (англ. Light Dependent Resistor или просто LDR)
#define BUZZER_PIN 3
#define LDR_PIN A0

void setup()
{
  // пин с пьезопищалкой — выход...
  pinMode(BUZZER_PIN, OUTPUT);

  // ...а все остальные пины являются входами изначально,
  // всякий раз при подаче питания или сбросе микроконтроллера.
  // Поэтому, на самом деле, нам совершенно необязательно
  // настраивать LDR_PIN в режим входа: он и так им является
}

void loop()
{
  int val, frequency;

  // считываем уровень освещённости так же, как для
  // потенциометра: в виде значения от 0 до 1023.
  val = analogRead(LDR_PIN);

  // рассчитываем частоту звучания пищалки в герцах (ноту),
  // используя функцию проекции (англ. map). Она отображает
  // значение из одного диапазона на другой, строя пропорцию.
  // В нашем случае [0; 1023] -> [3500; 4500]. Так мы получим
  // частоту от 3,5 до 4,5 кГц.
  frequency = map(val, 0, 1023, 3500, 4500);

  // заставляем пин с пищалкой «вибрировать», т.е. звучать
  // (англ. tone) на заданной частоте 20 миллисекунд. При
  // следующих проходах loop, tone будет вызван снова и снова,
  // и на деле мы услышим непрерывный звук тональностью, которая
  // зависит от количества света, попадающего на фоторезистор
  tone(BUZZER_PIN, frequency, 20);
}
```


- ✓ Функция `map(value, fromLow, fromHigh, toLow, toHigh)` возвращает целочисленное значение из интервала `[toLow, toHigh]`, которое является пропорциональным отображением содержимого `value` из интервала `[fromLow, fromHigh]`
- ✓ Верхние границы `map` не обязательно должны быть больше нижних и могут быть отрицательными. К примеру, значение из интервала `[1, 10]` можно отобразить в интервал `[10,-5]`
- ✓ Если при вычислении значения `map` образуется дробное значение, оно будет отброшено, а не округлено
- ✓ Функция `map` не будет отбрасывать значения за пределами указанных диапазонов, а также масштабирует их по заданному правилу.
- ✓ Если вам нужно ограничить множество допустимых значений, используйте функцию `constrain(value, from, to)`, которая вернет:
 - ✓ `value`, если это значение попадает в диапазон `[from, to]`
 - ✓ `from`, если `value` меньше него
 - ✓ `to`, если `value` больше него
- ✓ Функция `tone(pin, frequency, duration)` заставляет пьезопищалку, подключенную к порту `pin`, издавать звук высотой `frequency` герц на протяжении `duration` миллисекунд
- ✓ Параметр `duration` не является обязательным. Если его не передать, звук включится навсегда. Чтобы его выключить, вам понадобится функция `noTone(pin)`. Ей нужно передать номер порта с пищалкой, которую нужно выключить
- ✓ Одновременно можно управлять только одной пищалкой. Если во время звучания вызвать `tone` для другого порта, ничего не произойдет.
- ✓ Вызов `tone` для уже звучащего порта обновит частоту и длительность звучания

Задание 4. Ответьте на следующие вопросы

1. Каким сопротивлением должен обладать фоторезистор, чтобы на аналоговый вход было подано напряжение 1 В?
2. Можем ли мы регулировать яркость светодиода, подключенного к 11-му порту, во время звучания пьезопищалки?
3. Что изменится в работе терменвокса, если заменить резистор на 10 кОм резистором на 100 кОм? Попробуйте ответить без эксперимента. Затем отключите питание, замените резистор и проверьте.
4. Каков будет результат вызова `map(30, 0, 90, 90, -90)`?
5. Как будет работать вызов `tone` без указания длительности звучания?
6. Можно ли устроить полифоническое звучание с помощью функции `tone`?

Задание 5. Самостоятельно измените существующую программу и схему

1. Уберите из программы чтение датчика освещенности и пропишите азбукой Морзе позывной SOS: три точки, три тире, три точки
2. Измените код программы так, чтобы с падением освещенности звук становился ниже (например, падал от 5 кГц до 2,5 кГц)
3. Измените код программы так, чтобы звук терменвокса раздавался не непрерывно, а 10 раз в секунду с различными паузами

Задание 7. Самостоятельное решите задач

Уберите из проекта фоторезистор и из программы чтение датчика освещенности. Настройте пьезодинамик так, чтобы он издавал любую мелодию. Мелодию необходимо заранее выбрать и согласовать с преподавателем. Мелодии не должны повторяться!