

Лабораторная работа №2.

Пьезодинамик

Цель:

Научиться подключать и управлять пьезодинамиком (звуковым сигналом) робота-паука с помощью Arduino. Освоить функции `tone()` и `noTone()` для генерации одиночных сигналов (например, предупреждение) и простых мелодий (например, индикация включения)

Теоретическая часть

Роль звука в роботе-пауке

Оповещения о препятствиях

Сигнализация низкого заряда батареи

Индикации режимов работы

Эмоциональное взаимодействие (мелодии)

Что такое пьезодинамик?

Пьезодинамик — это электронное устройство, преобразующее электрические сигналы в звуковые колебания.

Используется для подачи звуковых сигналов: пищалок, мелодий, оповещений

Типы пьезодинамиков

Тип	Особенности	Подключение
Пассивный	Требует частотного сигнала. Поддерживает разные тона.	Через <code>tone()</code>
Активный	Издаёт один звук при подаче питания.	Прямо на пин или через <code>digitalWrite()</code>

В данной лабораторной используется **пассивный** пьезодинамик — он более универсален и подходит для обучения

Пины подключения:

Один вывод ➡ **A3**

Второй ➡ **GND** можно через резистор 100–220 Ом

Зачем резисторы?
Защищает плату и динамик от перегрузки

Функции для управления:

```
Tone(pin, frequency, duration)
```

Воспроизводит звуковой сигнал заданной частоты (в герцах) на указанном пине

`pin` — цифровой пин

`frequency` — частота звука (в Гц)

`duration` (необязательно) — длительность в мс

Пример:

```
void setup() {  
  pinMode(A3, OUTPUT); // Пин 8 как выход для динамика  
}  
  
void loop() {  
  tone(A3, 440); // Включаем ноту Ля 4-й октавы (440 Гц) на пине 8  
  delay(1000); // Звучит 1 секунду  
  noTone(8); // Выключаем звук на пине 8  
  delay(1000); // Пауза 1 секунду  
  
  // Проверка: Измените частоту ниже и загрузите код снова  
  // tone(8, 262); // До (низкий звук)  
  // tone(8, 1000); // Очень высокий звук  
}
```

Если не указана длительность, звук будет играть, пока не будет вызвана noTone()

`noTone(pin)` останавливает воспроизведение звука

Таблица нот и частот 4 октавы

Нота	Частота (Гц)
C4 – До	262
D4 – Ре	294
E4 – Ми	330
F4 – Фа	349
G4 – Соль	392
A4 – Ля	440
B4 – Си	494
C5 – До	523



Музыкальная длительность

Обозначение	Название	Длительность (мс при темпе 120)
1	Целая	1000
0.5	Половинная	500
0.25	Четвертная	250

Темп можно задать самостоятельно — например, считать одну четверть = 500 мс.

Расширенная версия (от 0 до 8 октавы):

Нота	0	1	2	3	4	5	6	7	8
C	16	33	65	131	262	523	1047	2093	4186
C#	17	35	69	139	277	554	1109	2217	4435
D	18	37	73	147	294	587	1175	2349	4699
D#	19	39	78	156	311	622	1245	2489	4978
E	21	41	82	165	330	659	1319	2637	5274
F	22	44	87	175	349	698	1397	2794	5588
F#	23	46	93	185	370	740	1480	2960	5920
G	25	49	98	196	392	784	1568	3136	6272
G#	26	52	104	208	415	831	1661	3322	6645
A	28	55	110	220	440	880	1760	3520	7040
A#	29	58	117	233	466	932	1865	3729	7459
B	31	62	123	247	494	988	1976	3951	7902

Разбор кода по частям:


```
const int notes[] = {164, 261, 220, 220, 164, 164, 246,
195, 195, 164, 164, 246, 195, 195, 146, 146, 184, 184, 195};

const int durations[] = {216, 216, 216, 108, 216, 108, 216,
216, 108, 216, 108, 216, 216, 108, 216, 216, 108, 216};

const int pauses[] = {240, 240, 240, 120, 240, 120, 240,
240, 120, 240, 120, 240, 240, 120, 240, 240, 120, 240};
```

```
void song() {  
    for(int i = 0; i <= sizeof(notes); i++){  
        tone(A3, notes[i], durations[i]);  
        delay(pauses[i]);  
    }  
}
```

Этот код предназначен для воспроизведения последовательности звуков



Массив `notes[]` содержит частоты звуков в герцах (Hz), которые будут звучать. Каждое число — это высота тона (например, 261 Гц — это нота до средней октавы).

Массив `durations[]` задаёт длительность проигрывания каждого звука в миллисекундах.

Массив `pauses[]` определяет время паузы (задержки) после каждого звука, тоже в миллисекундах. (например, `i += 5`).

Функция `song()`:

В цикле `for` переменная `i` используется для прохода по всем элементам массива нот.

Для каждой ноты вызывается функция `tone(tonePin, notes[i], durations[i]);`, которая начинает воспроизводить звук на пине `tonePin` с частотой `notes[i]` и длительностью `durations[i]`.

Затем вызывается `delay(pauses[i]);`, чтобы подождать паузу.

Практическая часть

Задание 1: Сигнал тревоги

Создайте звуковое предупреждение при обнаружении объекта.

1 Подключите пьезодинамик к пину A3 и GND.

2 Загрузите код:

```
void setup() {  
  pinMode(A3, OUTPUT); // Динамик на D8  
}  
  
void loop() {  
  // Сигнал тревоги (чередование высокого/низкого тона)  
  tone(A3, 880, 200); // Высокий тон  
  delay(300);  
  tone(A3, 440, 200); // Низкий тон  
  delay(500);  
}
```

Проверка: Измените частоты
(880 → 660, 440 → 330)



Как меняется характер
сигнала?

Задание 2: Мелодия запуска робота

Запрограммируйте звуковое сопровождение при включении паука.



Используйте частоты
из таблицы выше

Задание 3.

Сделайте проигрывание сигнала SOS, короткой известной мелодии (например, "Mario", "Star Wars", "Titanic").

Например, сигнал SOS (... --- ...)

Контрольные вопросы

1. Почему для робота-паука выбран пассивный динамик?
2. Что произойдёт, если убрать `noTone(8)` в примере с мелодией?
3. Как создать эффект вибрации паука при столкновении с помощью звука?
4. Почему в функции `tone()` нет параметра громкости?