

Зуммер

Монтаж зуммера еще проще — ему не нужен токоограничивающий резистор (рис. 6.31).

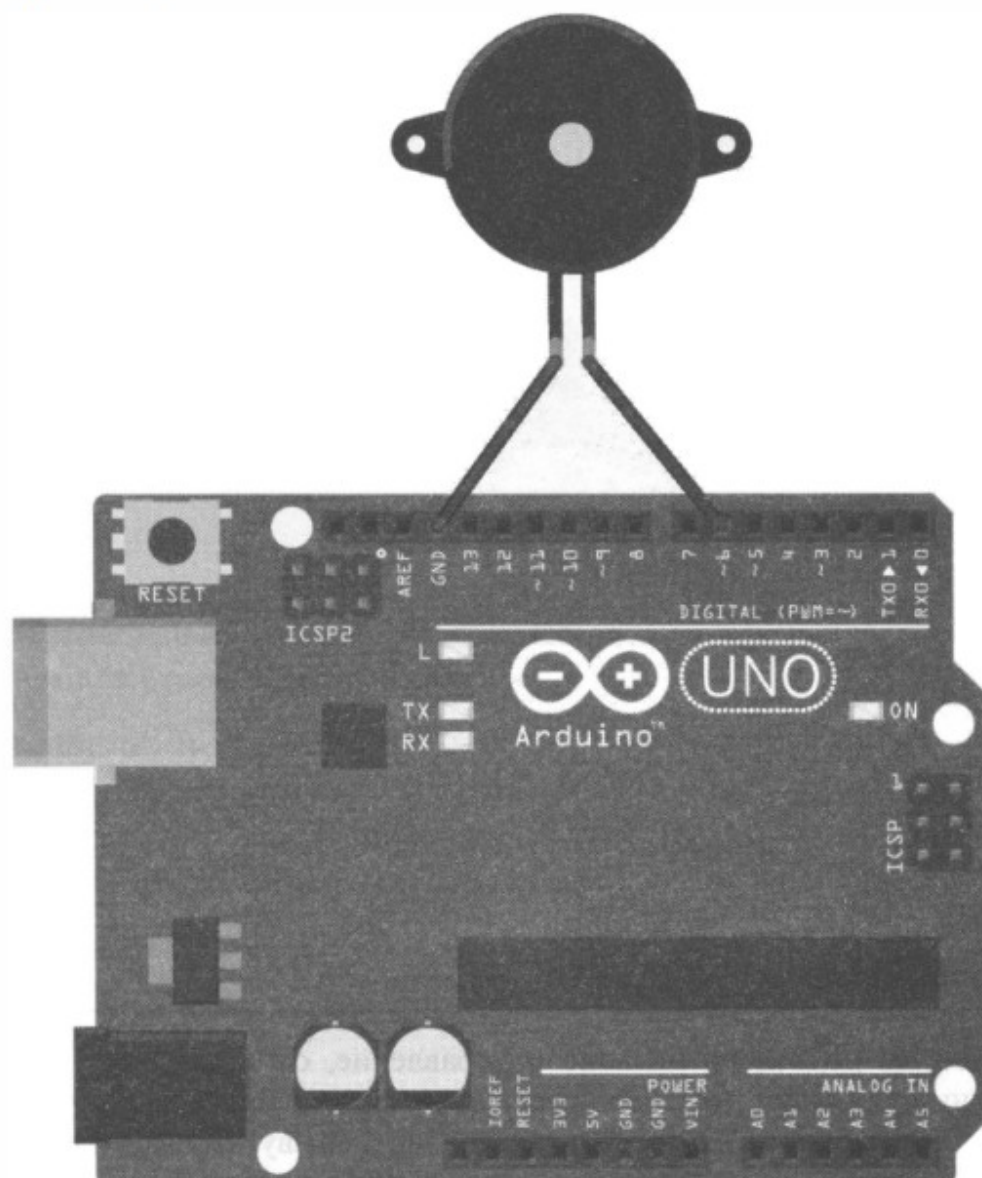


Рис. 6.31. Подключение зуммера к Arduino

Для тестирования работы зуммера можно использовать программу из листинга 6.3. Контакт зуммера (в данном случае это ножка 6) переводим в режим вывода, а затем командой `tone` (номер пина, частота) заставляем его издавать писк. Команда `noTone` (номер пина) отменяет звук. Отмечу, что можно использовать только один зуммер для Arduino одновременно. В программе зуммер, прерываясь, гудит на частоте 1000 Гц в течение 2 секунд, затем на 1 секунду замолкает.

Листинг 6.3. Тестирование зуммера

```
//Создадим переменные для хранения номеров используемых пинов/портов Arduino.
int In1, In2, In3, In4, InZ;
//Настройка
void setup() {
    // Присвоим переменным номера пинов Arduino.
    InZ = 6;
    pinMode(InZ, OUTPUT);
    In1 = 2;
    In2 = 3;
    In3 = 4;
    In4 = 5;

    // Переведем эти пины/порты в режим вывода.
    pinMode(In1, OUTPUT);
    pinMode(In2, OUTPUT);
    pinMode(In3, OUTPUT);
    pinMode(In4, OUTPUT);
}
//Тело программы
void loop() {
    for (int i = 0; i < 10; i++)
    {
        tone(InZ, 1000);
        delay(100);
        noTone(InZ);
        delay(100);
    }
    delay(1000);
}
```

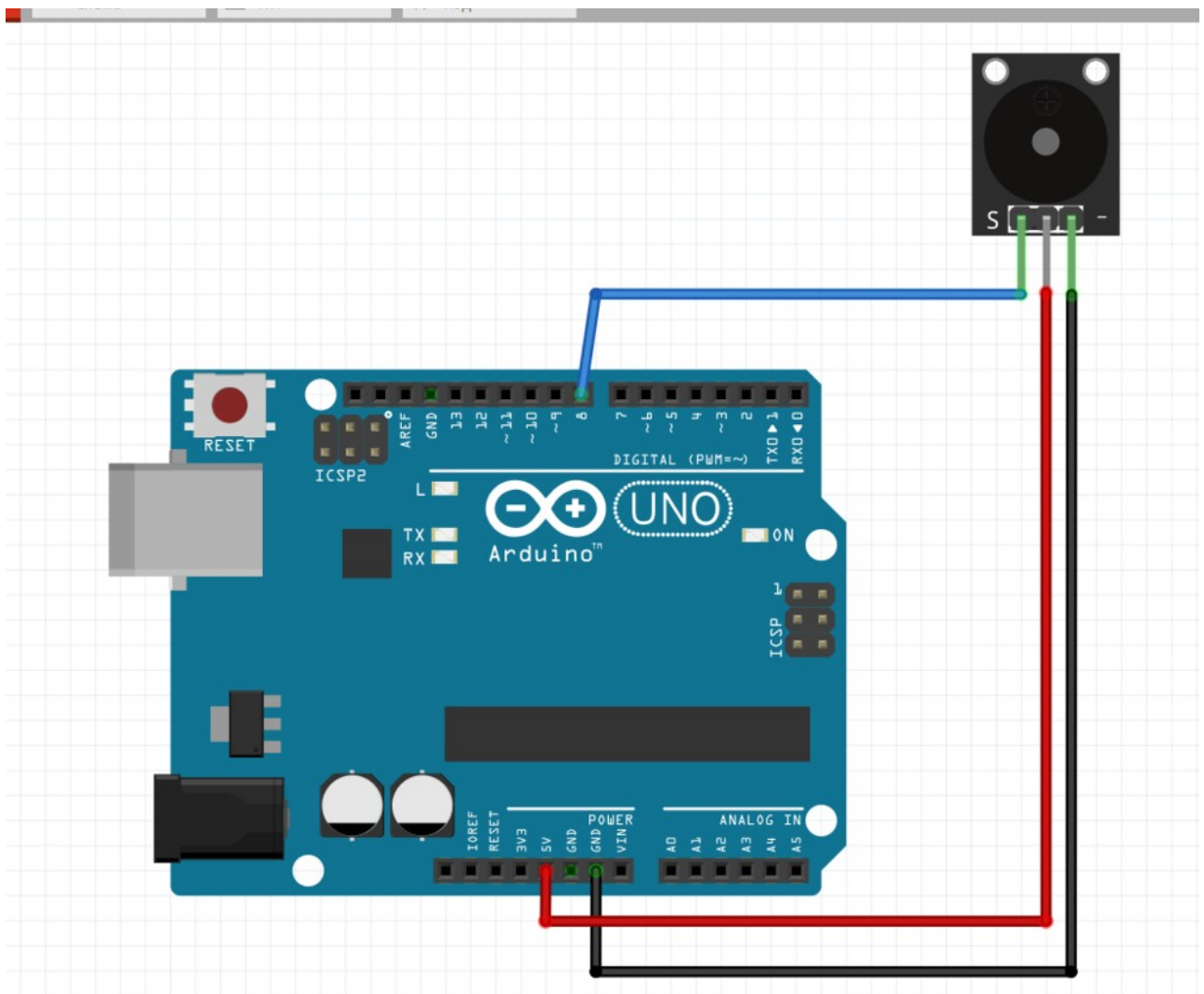
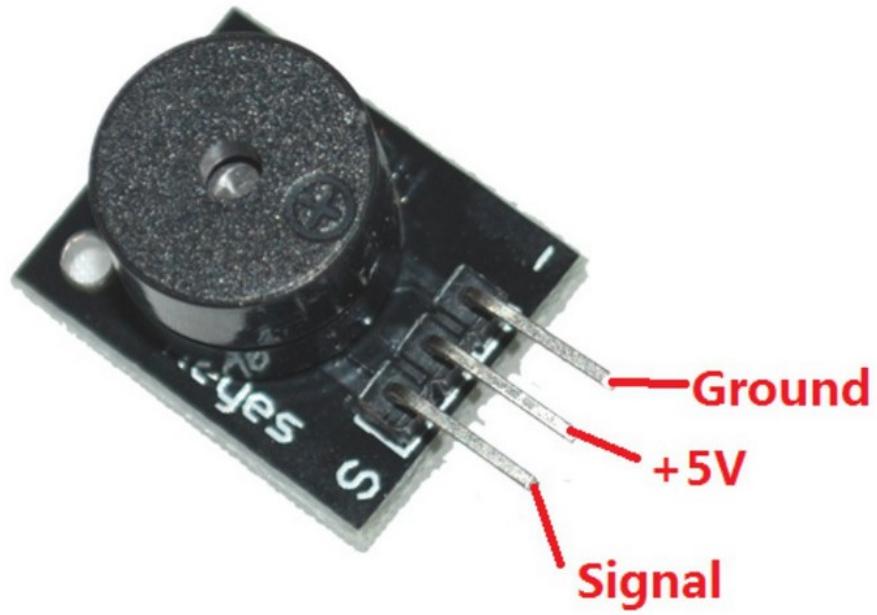
Программа:

```
int InZ;

void setup() {
    InZ = 6; //Подключение зуммера к 6 порту
    pinMode(InZ, OUTPUT);
}

void loop() {
    for (int i=0; i <= 10; i++){
        tone(InZ, 2000);
        delay(100);
        noTone(InZ);
        delay(100);
    }
    delay(1000);
}
```

Модуль пассивного зуммера



Код программы:

```
const int pin = 8;           // Указываем порт I/O

void setup()
{
  pinMode(pin, OUTPUT);     // Указываем порт, как выход
}

void loop(){
  digitalWrite(pin, HIGH);  // Отключаем зуммер, устанавливаем вывод в HIGH
  delay(200);               // Пауза
  digitalWrite(pin, LOW);  // Включаем зуммер, устанавливаем вывод в LOW
  delay(200);               // Пауза
}
```

Примеры мелодий для зуммера

Для того, чтобы разнообразить работу с новым проектом, добавить в него «развлекательный» элемент, пользователи придумали задавать определённый набор частот звука, делая его созвучным некоторым знаменитым композициям из песен и кинофильмов. Разнообразные скетчи для таких мелодий можно найти в интернете. Приведем пример мелодии для пьезопищалки для одного из самых узнаваемых треков «nokia tune» из ставших легендарными мобильных Nokia. Файл pitches.h можно сделать самим, скопировав его содержимое так, как указано в этой статье на официальном сайте.

```
#include "pitches.h"
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

int noteDurations[] = { 4, 8, 8, 4, 4, 4, 4, 4 };

void setup() {}

void loop() {

  for (int thisNote = 0; thisNote < 8; thisNote++) {

    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8);
  }
  delay(1000);
}
```

Код pitches.h

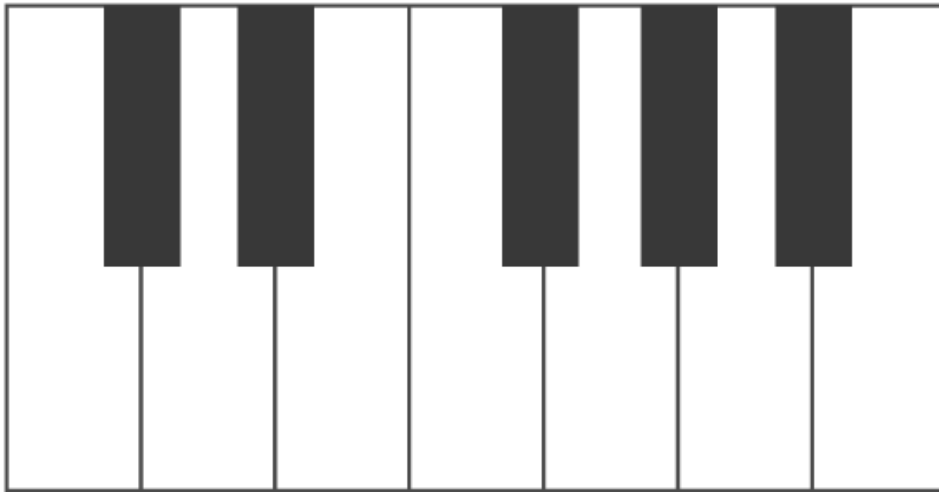
** Public Constants*

*****/

```
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
```

```
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
```

При написании собственных мелодий пригодится знание частот нот и длительностей интервалов, используемых в стандартной нотной записи.



TONE		FREQUENCY (in hertz)		DELAY VALUE
Do (C)	→	261.63	→	1915 μ s
Re (D)	→	293.66	→	1700 μ s
Mi (E)	→	329.63	→	1519 μ s
Fa (F)	→	349.23	→	1432 μ s
Sol (G)	→	392.00	→	1275 μ s
La (A)	→	440.00	→	1136 μ s
Si (B)	→	493.88	→	1014 μ s

Частота нот для пищалки Ардуино

Программа (Do, Re, Mi, Fa, Sol, La, Si, Do)

```
#define do_ 262  
#define re_ 294  
#define mi_ 330  
#define fa_ 349  
#define sol_ 392
```

```
#define la_ 440
#define si_ 494

#define ddo_ 1915
#define dre_ 1700
#define dmi_ 1519
#define dfa_ 1432
#define dsol_ 1275
#define dla_ 1136
#define dsi_ 1014

int pin = 8;

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

void loop() {

  delay(1000);

  tone(pin, do_);
  delay(ddo_);
  noTone(pin);

  tone(pin, re_);
  delay(dre_);
  noTone(pin);

  tone(pin, mi_);
  delay(dmi_);
  noTone(pin);

  tone(pin, fa_);
  delay(dfa_);
  noTone(pin);

  tone(pin, sol_);
  delay(dsol_);
  noTone(pin);

  tone(pin, la_);
  delay(dla_);
  noTone(pin);

  tone(pin, si_);
  delay(dsi_);
  noTone(pin);

  tone(pin, do_);
```



```
    delay(ddo_);  
    noTone(pin);  
  
    delay(1000);  
}
```