

Эксперимент № 10. Музыкальный инструмент. Взаимодействие Piezo и Button

1. Proteus 8.1 Professional
2. <http://123D.circuits.io>
3. Arduino IDE

Список деталей для эксперимента

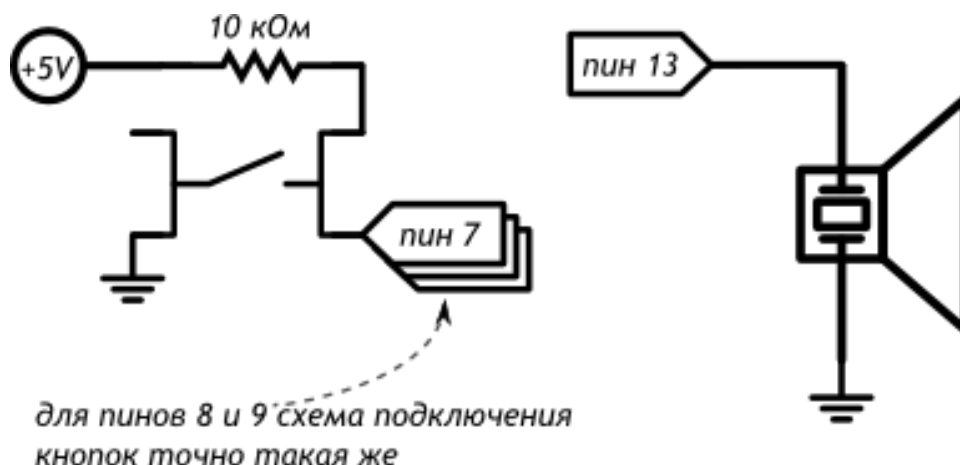
- 1 плата Arduino Uno
- 1 беспаячная макетная плата
- 3 резистора номиналом 100-220 Ом
- Пьезодинамик (Sounder(Proteus))
- RGB- светодиод
- провода «папа-папа»

Пошаговое руководство для проведения эксперимента

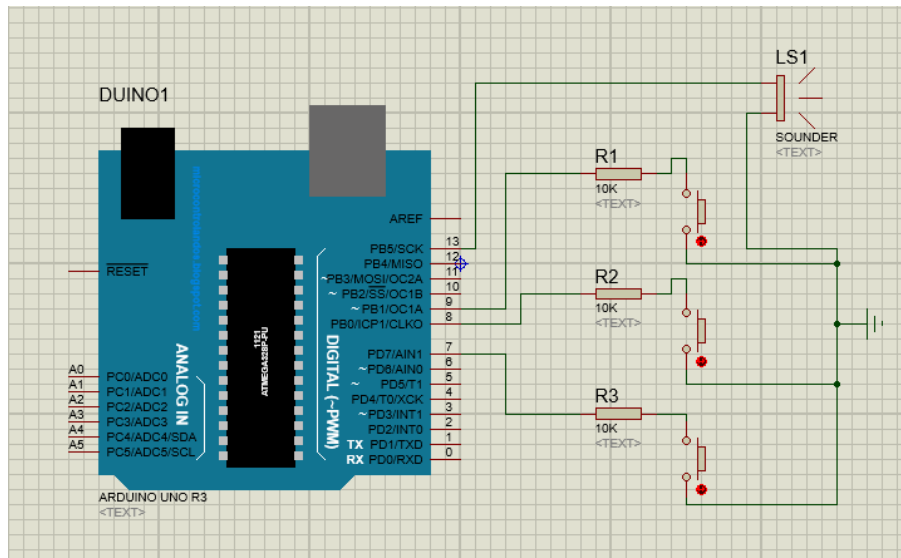
В эксперименте мы создаем маленькую клавиатуру, на которой можно сыграть несколько нот и программируем ее с помощью Arduino IDE.

Обратите внимание

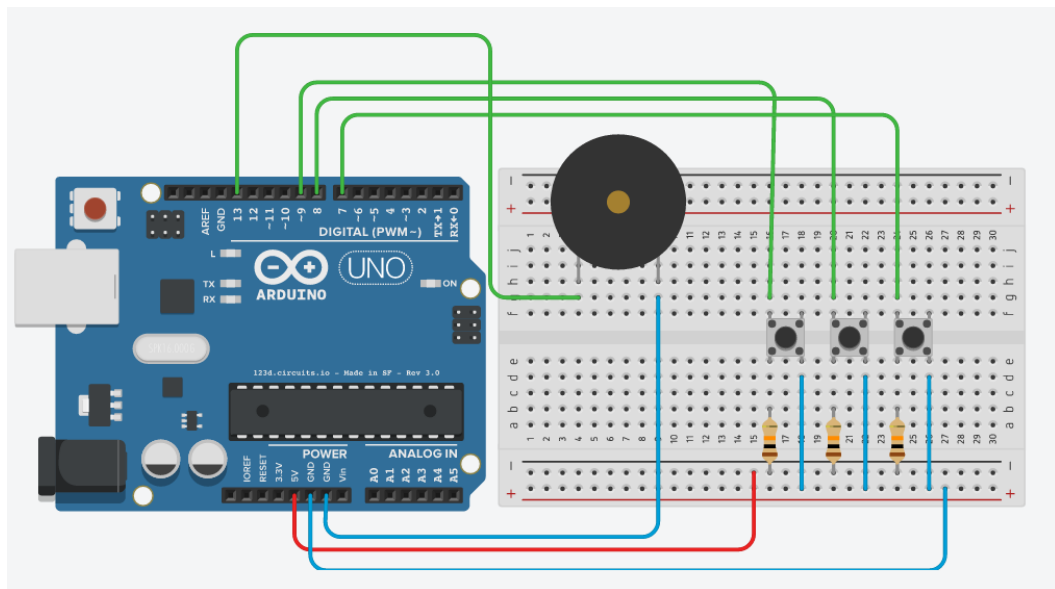
- Ножки тактовой кнопки, расположенные с одной стороны, разомкнуты, когда кнопка не нажата. Ножки, расположенные друг напротив друга на противоположных сторонах макетки находятся на одной «рельсе». Воспользовавшись этим, мы можем расположить резистор с одной стороны макетки, а провод, подключаемый к 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.



Подключение кнопок и пьезодинамика к Arduino в эксперименте Piano (принципиальная схема)



Подключение кнопок и пьезодинамика к Arduino в эксперименте Piano (Proteus)



Подключение кнопок и пьезодинамика к Arduino в эксперименте Piano (123D.Circuits)

Пример 1. Пианино (количество кнопок может быть изменено по собственному желанию)

```
#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

1. 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`.
2. 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`
3. 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.
4. Мы используем логический оператор отрицания «не» `!`. Если `keyUp` имеет значение 0, выражение `!keyUp` будет иметь значение 1 и наоборот.
5. Поскольку мы собрали схему с подтягивающим резистором, при нажатии кнопки мы будем получать на соответствующем порте 0.
6. 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 пищалкой, звучат с разной громкостью?
4. Для чего мы использовали оператор логического отрицания `!`?

Пример 2. Электронная арфа

Какая первая ассоциация при упоминании слова арфа? Деревянная рама, струны, ее звук. А что если представить темное помещение, дым, лазер и электронную музыку? Все кажется непонятным и интригующим. Теперь представляется несколько лазерных лучей, обхваченных рамой. Но если выйти за пределы данной мысли, можно представить арфу неограниченную рамкой и сделанной лишь на лазерном луче.

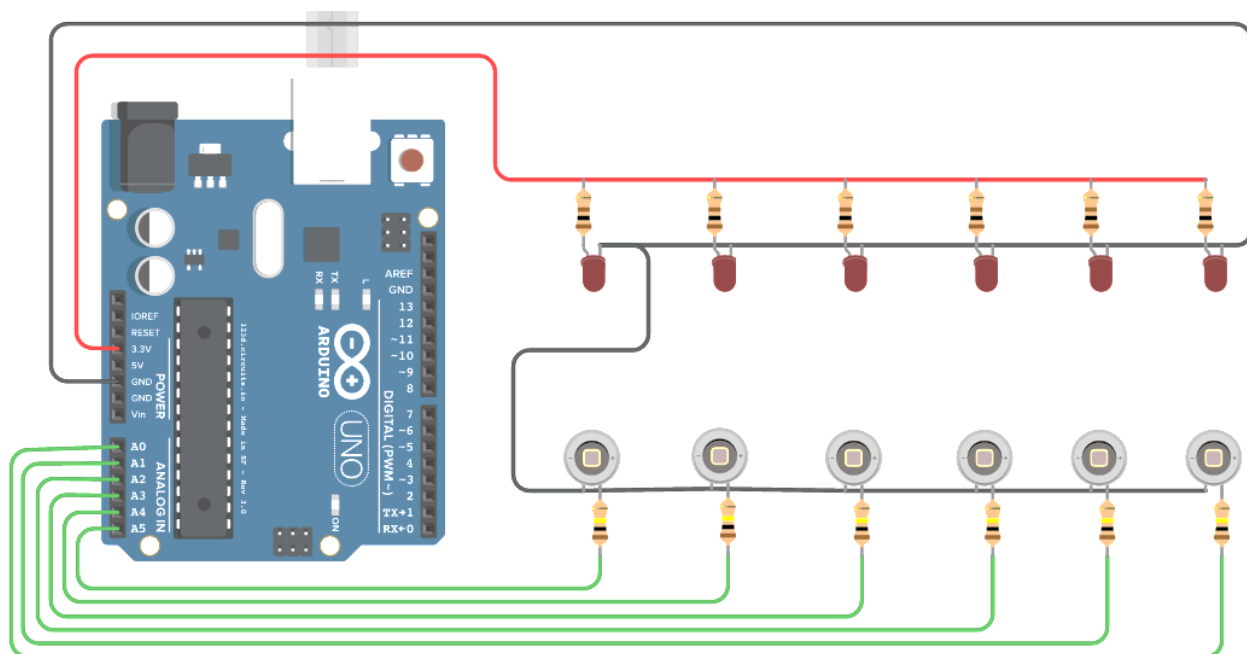
Лазерная арфа впервые была использована на концерте в Китае. Прототип лазерной арфы был разработан французом Бернардом Шайнером в 1979 году. В 1981 году, когда инструмент был впервые представлен публике, он еще был в стадии разработки. С тех пор арфа претерпевала множество изменений.

Интеллектуальную часть инструмента составляет Arduino, использующий программу MiniPiano соединяясь по каналу COM. Каждый луч может играть разные ноты, когда мы дотрагиваемся к «световым» струнам. Когда мы перемещаем руки вверх и вниз, тон ноты изменяется. Как только мы убираем руку с луча, воспроизведение ноты прекращается.

Список деталей для эксперимента

- 1 плата Arduino Uno
- Деревянная рамка
- 6 резисторов номиналом 100Ом
- 6 резисторов номиналом 100Ком
- 6 фотодиодов (фоторезисторов)
- 6 светодиода
- провода «папа-папа», «папа-мама»

В эксперименте мы создаем маленькую арфу на 6 струн, передаем аналоговый сигнал с фотодиода на Arduino и отправляем данные из Serial Port по COM-каналу на стационарный блок, реализующий воспроизведение MIDI программной реализацией с помощью средств ООП. На арфе можно сыграть несколько нот, в зависимости от количества подключенных пинов.



```
#include <SoftwareSerial.h>
int val0,val1,val2,val3,val4,val5;
```

```
void setup() {
  Serial.begin(9600); }
```

```
void loop() { // run over and over
```

```
  val0 = analogRead(0);
  if (val0>600) {
    Serial.println("1"); }
```

```
  val1 = analogRead(1);
  if (val1>600) {
    Serial.println("2"); }
```

```
  val2 = analogRead(2);
  if (val2>600) {
    Serial.println("3"); }
```

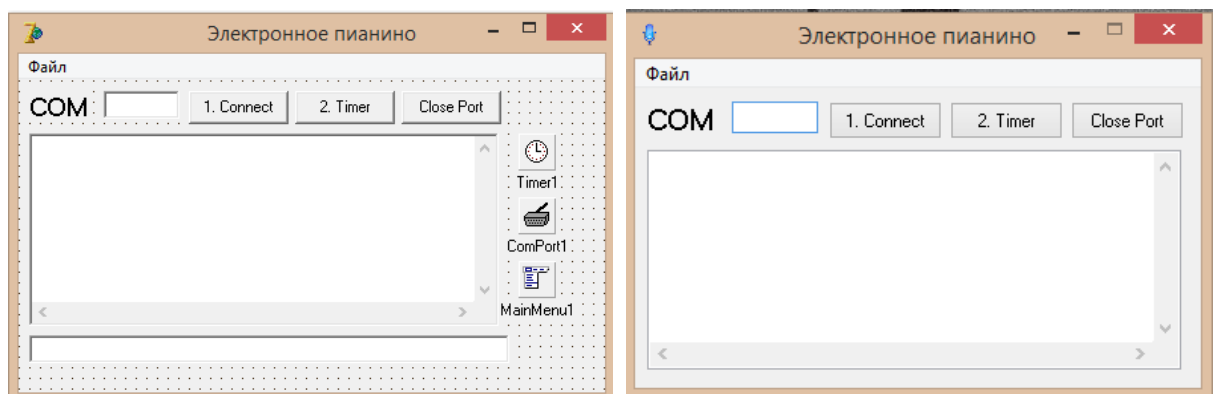
```
  val3 = analogRead(3);
  if (val3>630) {
    Serial.println("4"); }
```

```
  val4 = analogRead(4);
  if (val4>600) {
    Serial.println("5"); }
```

```
  val5 = analogRead(5);
  if (val5>653) {
    Serial.println("6"); }
```

```
  Serial.println("0");
}
```

Интерфейс стационарной программы (Delphi + набор COM компонентов)



//Звуки исполняются через встроенный midi-модуль MMsystem.

interface //Интерфейсная часть

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

MMSystem, StdCtrls, ExtCtrls, Menus, CPort; //Список используемых модулей. Добавлены
MMSystem, StdCtrls, Menus

...

var //Объявление переменных

Form1: TForm1;
hMidi :integer;
midimsg :integer; //Миди сообщение
no:integer; //Нота
pressed: array[0..11] of boolean; //Кнопка: нажата/не нажата

procedure TForm1.FormCreate(Sender: TObject);

var
i :integer;
begin
midiOutOpen(@hmidi, 0, 0, 0, 0); //Открытие миди-выхода
Form1.KeyPreview:= true;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin
midiOutClose(hmidi); //Закрытие миди-выхода
end;

**procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word; //Процедура нажатия кнопки
Shift: TShiftState);**

begin
midimsg := \$C0+\$100*46 ; // ??? t - ????? ??????????
midiOutShortMsg (hmidi, midimsg);
if key=ord('1') **then**
 begin
 no:=60; //c
 if not pressed[0] **then**
 begin
 //формируем сообщение
 midimsg := \$90 + ((no) * \$100) + (127 * \$10000) + 0;
 //отправляем его
 midiOutShortMsg (hmidi, midimsg);
 pressed[0]:=true;
 end;
 end;
 if key=ord('2') **then**
 begin
 no:=62 ; //d
 if not pressed[1] **then**
 begin
 midimsg := \$90 + ((no) * \$100) + (127 * \$10000) + 0;
 midiOutShortMsg (hmidi, midimsg);
 pressed[1]:=true;
 end;
 end;
 end;
 if key=ord('3') **then**

```

begin
no:=64 ; //e
if not pressed[2] then
  begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[2]:=true;
  end;
end;
if key=ord('4') then
  begin
no:=65; //f
if not pressed[3] then
  begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[3]:=true;
  end;
end;
if key=ord('5') then
  begin
no:=67 ; //g
if not pressed[4] then
  begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[4]:=true;
  end;
end;
if key=ord('6') then
  begin
no:=69; //a
if not pressed[5] then
  begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[5]:=true;
  end;
end;
if key=ord('7') then
  begin
no:=71 ; //b
if not pressed[6] then
  begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[6]:=true;
  end;
end;
if key=ord('8') then
  begin
no:=72 ; //c
if not pressed[7] then
  begin

```

```

midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[7]:=true;
end;
end;
if key=ord('9') then
begin
no:=74 ; //d
if not pressed[8] then
begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[8]:=true;
end;
end;
if key=ord('0') then
begin
no:=76; //e
if not pressed[9] then
begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[9]:=true;
end;
end;
if key=189 then
begin
no:=77 ; //f
if not pressed[10] then
begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[10]:=true;
end;
end;
if key=187 then
begin
no:=79 ; //g
if not pressed[11] then
begin
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg);
pressed[11]:=true;
end;
end;
end;

procedure TForm1.FormKeyUp(Sender: TObject; var Key: Word; //Процедура отпускания кнопки
Shift: TShiftState);
begin
if key=ord('1') then
begin
no:=60; //c
//формируем сообщение

```

```

midimsg := $80 + ((no) * $100) + 0 ;
//отправляем его
midiOutShortMsg (hmidi, midimsg);
pressed[0]:=false;
end;
if key=ord('2') then
begin
no:=62 ; //d
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
pressed[1]:=false;
end;
if key=ord('3') then
begin
no:=64 ; //e
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
pressed[2]:=false;
end;
if key=ord('4') then
begin
no:=65; //f
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
pressed[3]:=false;
end;
if key=ord('5') then
begin
no:=67 ; //g
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
pressed[4]:=false;
end;
if key=ord('6') then
begin
no:=69; //a
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
pressed[5]:=false;
end;
if key=ord('7') then
begin
no:=71 ; //b
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
pressed[6]:=false;
end;
if key=ord('8') then
begin
no:=72 ; //c
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
pressed[7]:=false;
end;

```

```

if key=ord('9') then
  begin
    no:=74 ; //d
    midimsg := $80 + ((no) * $100) +0 ;
    midiOutShortMsg (hmidi, midimsg);
    pressed[8]:=false;
  end;
if key=ord('0') then
  begin
    no:=76; //e
    midimsg := $80 + ((no) * $100) +0 ;
    midiOutShortMsg (hmidi, midimsg);
    pressed[9]:=false;
  end;
if key=189 then
  begin
    no:=77 ; //f
    midimsg := $80 + ((no) * $100) +0 ;
    midiOutShortMsg (hmidi, midimsg);
    pressed[10]:=false;
  end;
if key=187 then
  begin
    no:=79 ; //g
    midimsg := $80 + ((no) * $100) +0 ;
    midiOutShortMsg (hmidi, midimsg);
    pressed[11]:=false;
  end;
end;

procedure TForm1.N2Click(Sender: TObject);
begin
  close //Выход из приложения
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  ComPort1.Close;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  ComPort1.Port := 'COM'+Edit1.Text;
  ComPort1.Open;
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  Str: String;
  i:integer;
begin
  ComPort1.ReadStr(Str, Count);
  Memo1.Lines.Add(Str);

```

```
for i:=0 to Memo1.Lines.Count do if Memo1.Lines.Count>6 then memo1.Clear;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
timer1.Enabled:=true;
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);
var
i,x,y:integer;
begin
```

```
midimsg := $C0+$100*46 ;
midiOutShortMsg (hmidi, midimsg);
```

```
for i:= 0 to Memo1.Lines.Count do if Memo1.Lines[i] = '1' then begin
no:=61; //c
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg); end else
midimsg := $80 + ((no) * $100) +0 ;
midiOutShortMsg (hmidi, midimsg);
```

```
for i:= 0 to Memo1.Lines.Count do if Memo1.Lines[i] = '2' then begin
no:=62; //c
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg) end else
midimsg := $80 + ((no) * $100) +0 ;
midiOutShortMsg (hmidi, midimsg);
```

```
for i:= 0 to Memo1.Lines.Count do if Memo1.Lines[i] = '3' then begin
no:=63; //c
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg) end else
midimsg := $80 + ((no) * $100) +0 ;
midiOutShortMsg (hmidi, midimsg);
```

```
for i:= 0 to Memo1.Lines.Count do if Memo1.Lines[i] = '4' then begin
no:=64; //c
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg) end else
midimsg := $80 + ((no) * $100) +0 ;
midiOutShortMsg (hmidi, midimsg);
```

```
for i:= 0 to Memo1.Lines.Count do if Memo1.Lines[i] = '5' then begin
no:=65; //c
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg) end else
midimsg := $80 + ((no) * $100) +0 ;
midiOutShortMsg (hmidi, midimsg);
```

```
for i:= 0 to Memo1.Lines.Count do if Memo1.Lines[i] = '6' then begin
no:=77; //c
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
```

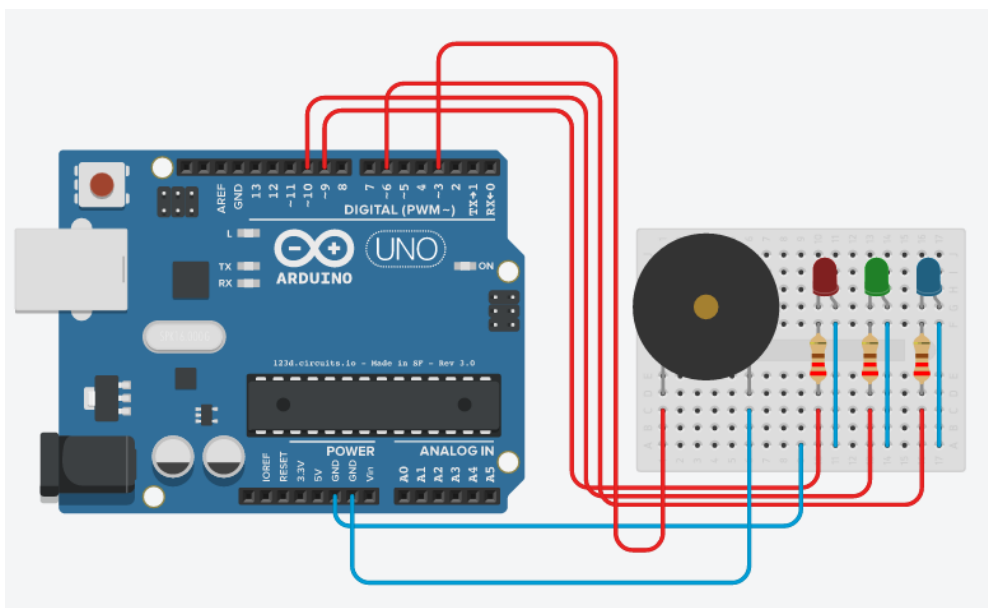
```

midiOutShortMsg (hmidi, midimsg) end else
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);

for i:= 0 to Memo1.Lines.Count do if Memo1.Lines[i] = '7' then begin
no:=67; //c
midimsg := $90 + ((no) * $100) + (127 * $10000) + 0;
midiOutShortMsg (hmidi, midimsg) end else
midimsg := $80 + ((no) * $100) + 0 ;
midiOutShortMsg (hmidi, midimsg);
end;
end.

```

Пример 3. Музыка Star Wars на Arduino и RGB цветомузыка



```

const int Pin_tone = 3; // номер порта зуммера
const byte COUNT_NOTES = 39; // Колличество нот
byte PWM_led_R, PWM_led_G, PWM_led_B;
byte pin_led_R = 9; // Номер контакта для Светодиода R
byte pin_led_G = 10; // Номер контакта для Светодиода G
byte pin_led_B = 6; // Номер контакта для Светодиода B
//частоты ноты
int frequencies[COUNT_NOTES] = {
  392, 392, 392, 311, 466, 392, 311, 466, 392,
  587, 587, 587, 622, 466, 369, 311, 466, 392,
  784, 392, 392, 784, 739, 698, 659, 622, 659,
  415, 554, 523, 493, 466, 440, 466,
  311, 369, 311, 466, 392
};
//длительность нот
int durations[COUNT_NOTES] = {
  350, 350, 350, 250, 100, 350, 250, 100, 700,
  350, 350, 350, 250, 100, 350, 250, 100, 700,
  350, 250, 100, 350, 250, 100, 100, 100, 450,
  150, 350, 250, 100, 100, 100, 450,
  150, 350, 250, 100, 750

```

```

};
void setup() {
  pinMode(pin_led_R, OUTPUT); // Настраиваем контакт на выход
  pinMode(pin_led_G, OUTPUT); // Настраиваем контакт на выход
  pinMode(pin_led_B, OUTPUT); // Настраиваем контакт на выход
}
void loop() {
  for (int i = 0; i <= COUNT_NOTES; i++ ) { // Цикл от 0 до количества нот
    tone(Pin_tone, frequencies[i]); // Включаем звук, определенной частоты
    PWM_led_R = random(0, 254); // Генерируем случайное число от 0 до 254
    PWM_led_G = random(0, 254); // Генерируем случайное число от 0 до 254
    PWM_led_B = random(0, 254); // Генерируем случайное число от 0 до 254
    analogWrite(pin_led_R, PWM_led_R); // Зажигаем светодиод на случайно сгенерированную
яркость
    analogWrite(pin_led_G, PWM_led_G); // Зажигаем светодиод на случайно сгенерированную
яркость
    analogWrite(pin_led_B, PWM_led_B); // Зажигаем светодиод на случайно сгенерированную
яркость
    delay(durations[i] * 2); // Дауза для заданой ноты
    noTone(Pin_tone); // Останавливаем звук
  }
}

```

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.
3. Подключите кнопки по схеме со стягивающим резистором и измените программу так, чтобы она продолжала работать.