МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Дата сдачи на проверку:

«__»_____ 2025 г.

Проверено:

«__»_____ 2025 г.

Отчёт по лабораторной работе №10
по дисциплине
«Информатика»
«Пьезоэлемент, микросхемы»

Разработал студент гр. ИВТб-1301-05-00 _____/Черкасов А. А./
(подпись)
Проверил преподователь _____/Шмакова Н.А./
(подпись)

Работа защищена «__»_____ 2025 г.

Киров
2025

Цель

Закрепить на практике знания о пьезоэлементе и микросхемах.

Задания

1. Азбука Морзе. Используйте мелодии для пьезодинамика из вложения (можно выбрать любые 2). Добавьте световую индикацию (не менее 3-х светодиодов). Добавьте кнопку для переключения мелодий.
2. Терменвокс. При падении освещённости звук уменьшается. Звуковой сигнал непрерывен. Первая октава.
3. Мерзкое пианино. 5 кнопок. Без резистора. Входной сигнал - switch/case. Малая октава.
4. Перетягивание каната. 14 светодиодов. Стягивающий резистор. Сигнализация на светодиоде. 1 нажатие для следующего перехода.

Решение

Задание 1

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 1.1.

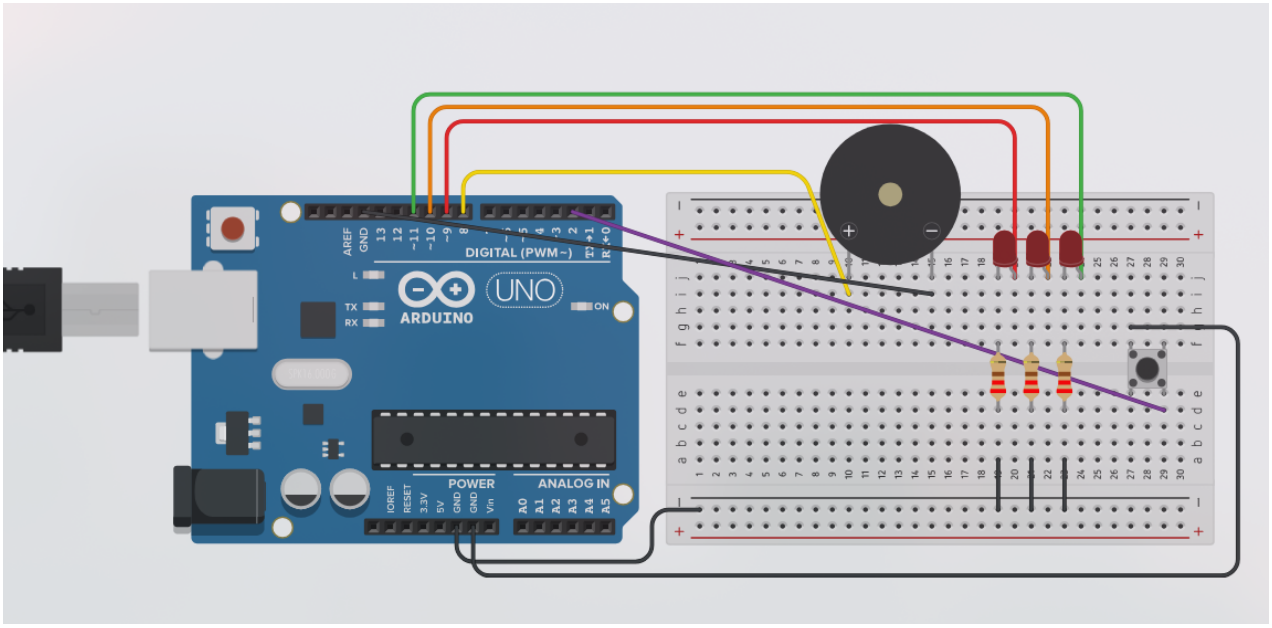

Рисунок 1.1 — Схема сборки задания №1

Принципиальная схема

Принципиальная схема представлена на рисунке 1.2.

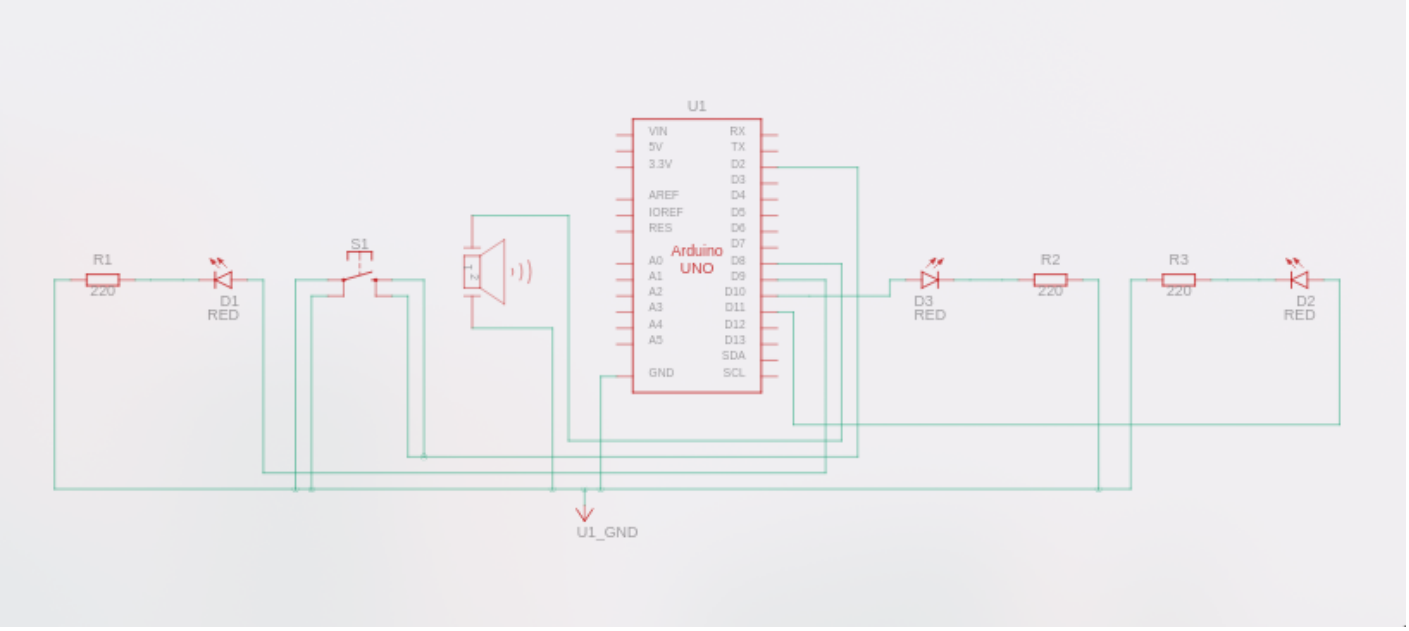

Рисунок 1.2 — Принципиальная схема задания №1

Перейти на Tinkercad

Исходный код решения задания представлен в приложении А1.

Задание 2

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 2.1.

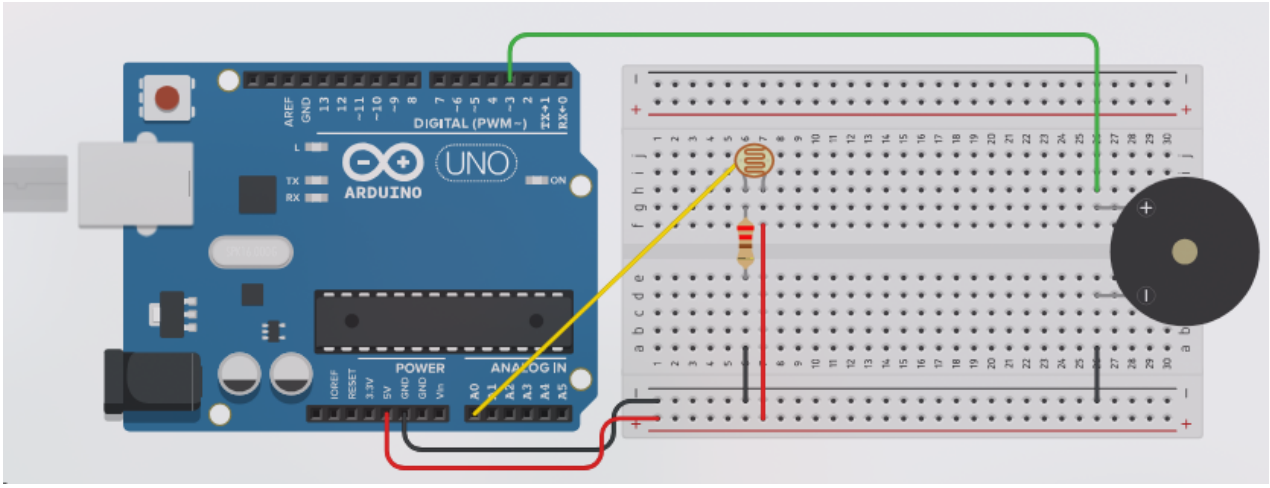

Рисунок 2.1 — Схема сборки задания №2

Принципиальная схема

Принципиальная схема представлена на рисунке 2.2.

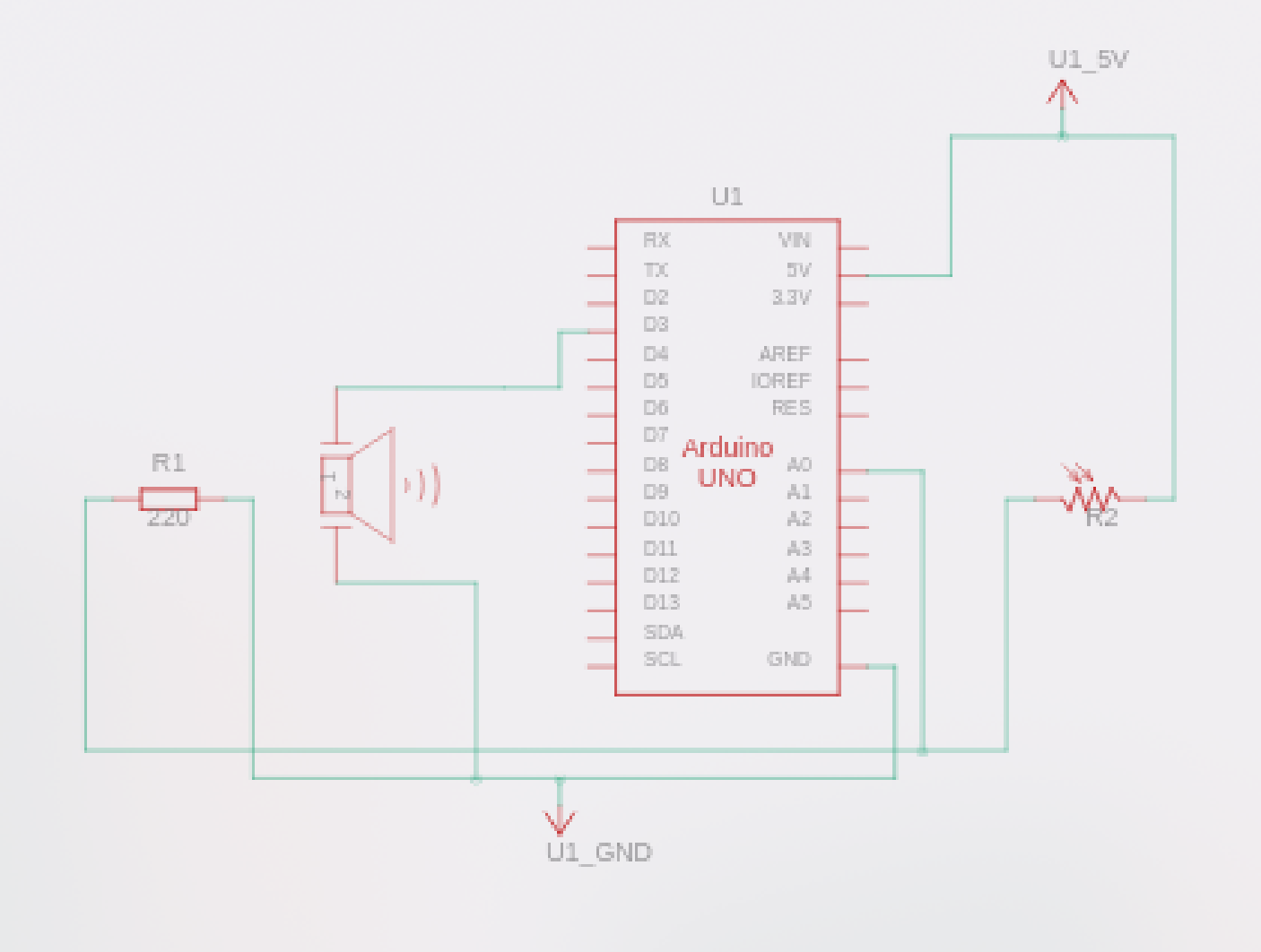

Рисунок 2.2 — Принципиальная схема задания №2

Перейти на Tinkercad

Исходный код решения задания представлен в приложении А2.

Решение

Задание 3

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 3.1.

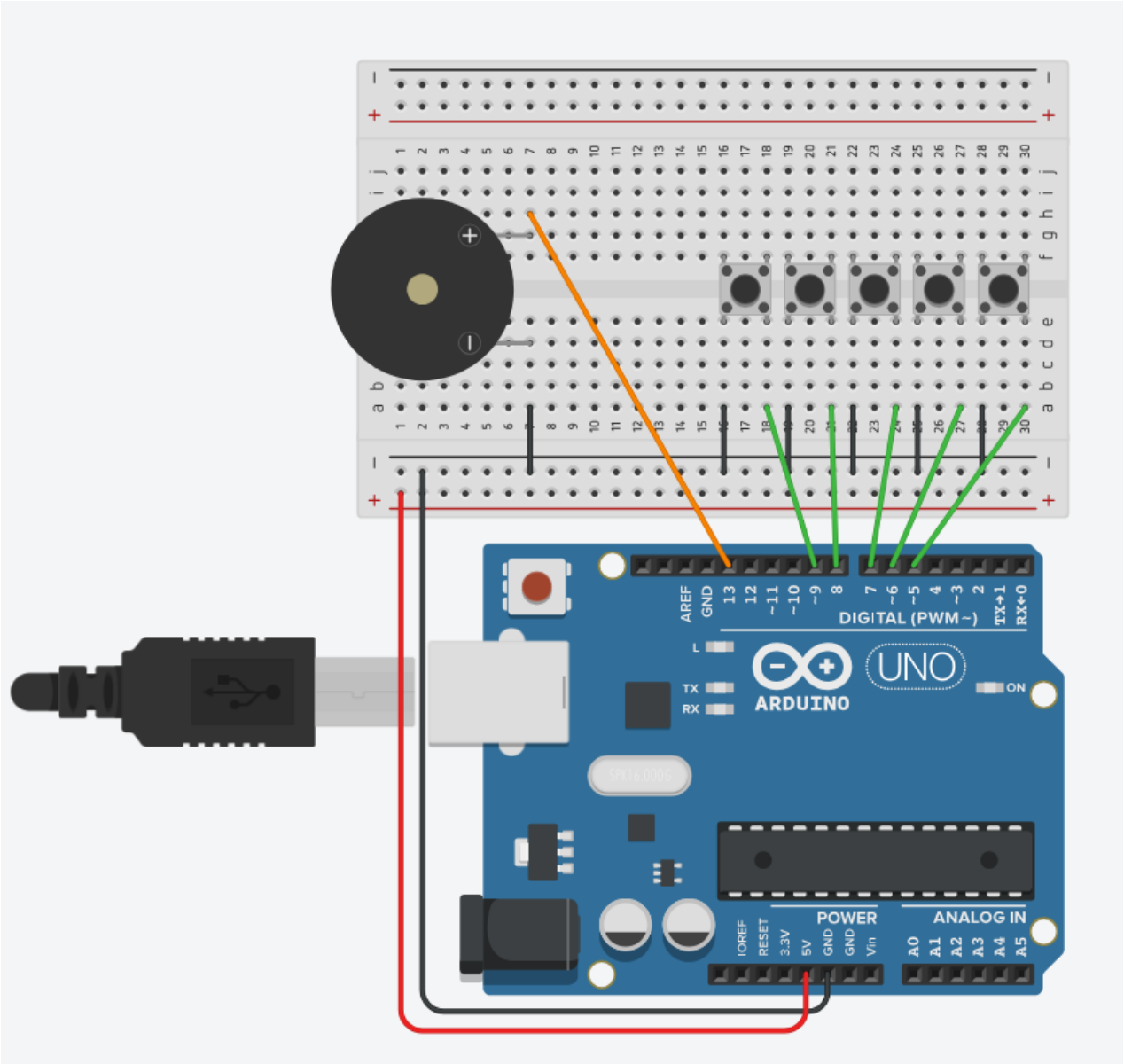

Рисунок 3.1 — Схема сборки задания №3

Принципиальная схема

Принципиальная схема представлена на рисунке 3.2.

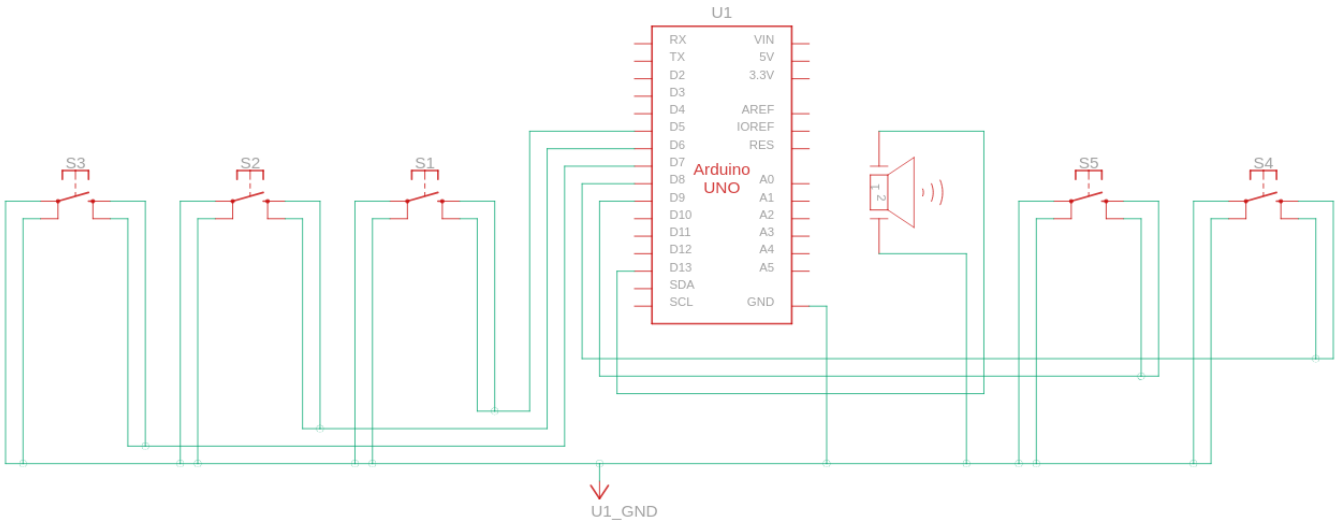

Рисунок 3.2 — Принципиальная схема задания №3

Перейти на Tinkercad

Исходный код решения задания представлен в приложении А3.

Решение

Задание 4

Схема сборки на макетной плате

Схема сборки на макетной плате представлена на рисунке 4.1.

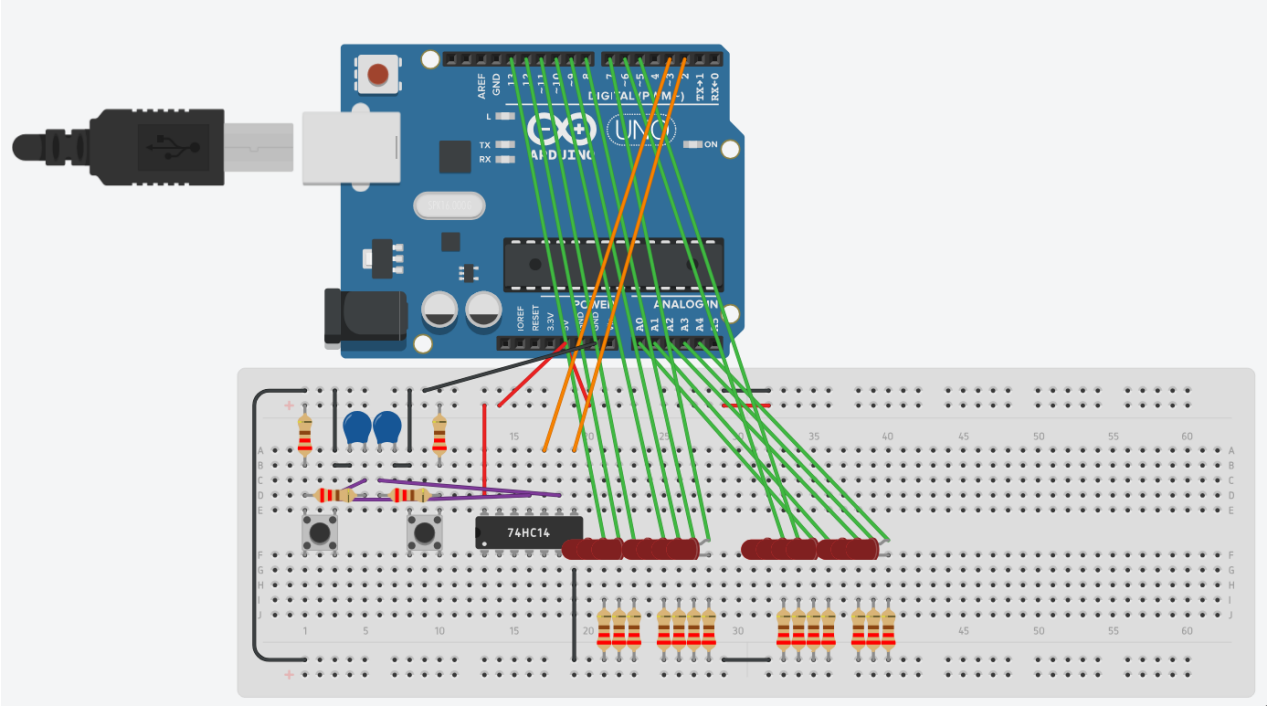

Рисунок 4.1 — Схема сборки задания №4

Принципиальная схема

Принципиальная схема представлена на рисунке 4.2 и 4.3.

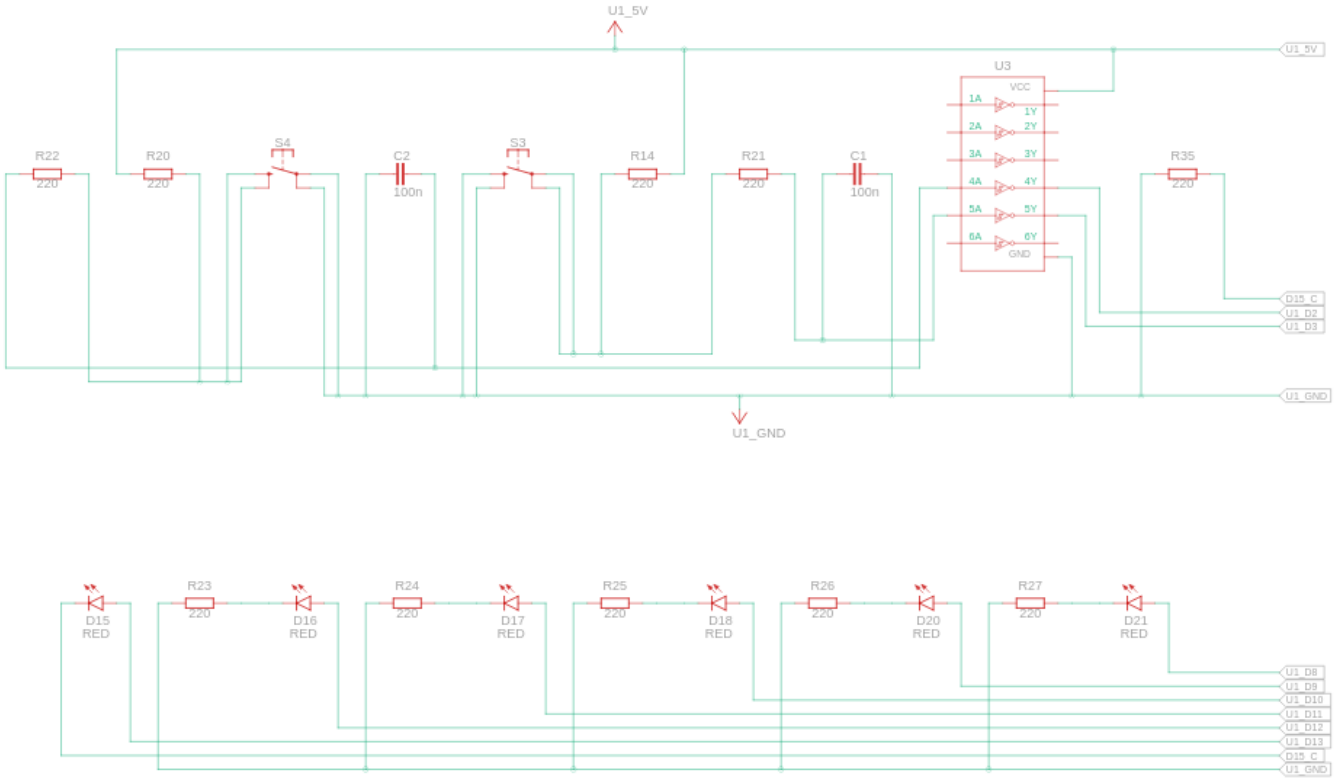

Рисунок 4.2 — Принципиальная схема задания №4

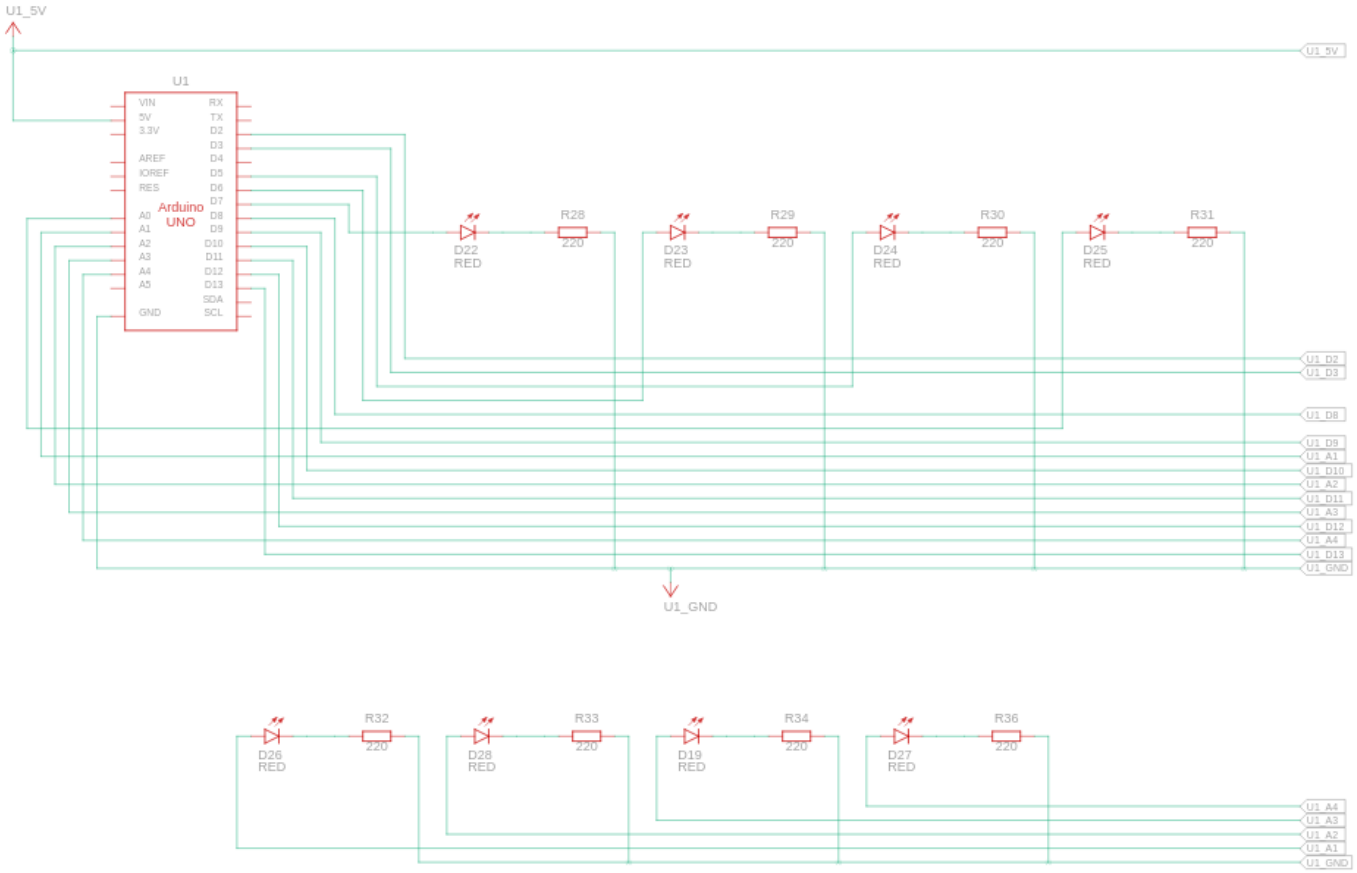

Рисунок 4.3 — Принципиальная схема задания №4

Перейти на Tinkercad

Исходный код решения задания представлен в приложении А4.

Вывод

В результате выполнения лабораторной работы была достигнута поставленная цель — закрепление на практике знаний о пьезоэлементе и микросхемах на примере работы с платформой Arduino. Были успешно реализованы четыре различных задания, демонстрирующих разнообразные способы работы с пьезоэлектрическими излучателями и интегральными микросхемами:

1. Азбука Морзе — воспроизведение мелодий на пьезодинамике с одновременной световой индикацией на трех светодиодах и возможностью переключения между мелодиями с помощью кнопки;
2. Терменвокс — непрерывная генерация звука первой октавы с изменением громкости в зависимости от уровня освещенности;
3. Мерзкое пианино — пятикнопочное пианино малой октавы с использованием оператора switch/case для обработки входных сигналов без подтягивающих резисторов;
4. Перетягивание каната — игра с использованием 14 светодиодов и микросхемы для управления последовательностью включения, с сигнализацией о переходах.

В ходе работы были изучены принципы работы пьезоэлектрических элементов для генерации звуковых сигналов, освоены методы управления интегральными микросхемами, а также получены практические навыки создания интерактивных устройств с использованием датчиков и исполнительных механизмов. Были использованы следующие ключевые функции Arduino:

tone(pin, frequency, duration) – генерация тонов заданной частоты и длительности

noTone(pin) – остановка генерации тона

analogRead(pin) – чтение аналогового значения с датчика

digitalRead(pin) – чтение состояния цифрового входа

digitalWrite(pin, value) – установка цифрового уровня сигнала

pinMode(pin, mode) – настройка режима работы пина

delay(ms) – задержка выполнения программы

Serial.begin(speed) – инициализация последовательного порта

Приложение А1. Исходный код решения для задания №1

```
#include <Arduino.h>

const int buzzerPin = 8;
const int buttonPin = 2;
const int ledPin1 = 9;
const int ledPin2 = 10;
const int ledPin3 = 11;
int marioNotes[] = {659, 659, 0,   659, 0,   523, 659, 0,   784, 0,
                    392, 0,   523, 0,   392, 0,   330, 0,   440, 0,
                    494, 0,   466, 440, 0,   392, 659, 784, 880, 0,
                    698, 784, 0,   659, 0,   523, 587, 494, 0};
int marioDurations[] = {50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,
                        50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50,
                        50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50};
int mashNotes[] = {880, 783, 698, 783, 0, 698,  0,    783,  0,   698, 0,   587,
                   0,   698, 587, 698, 0, 587,  0,    493,  0,   698, 587, 698,
                   0,   587, 0,   698, 0, 587,  0,    523,  0,   698, 0,   698,
                   523, 698, 0,   987, 0, 1046, 987,  1046, 0,   987, 0,   1046,
                   0,   987, 0,   880, 0, 880,  1046, 880,  1046};
int mashDurations[] = {500, 500,  250, 500,  250, 250, 250, 250, 250, 250,
                       250, 1000, 250, 250,  250, 250, 250, 250, 250, 1000,
                       250, 250,  250, 250,  250, 250, 250, 250, 250, 250,
                       250, 250,  250, 1000, 250, 250, 250, 250, 250, 250,
                       250, 250,  250, 250,  250, 250, 250, 250, 250, 250,
                       250, 250,  250, 250,  250, 250, 250, 250, 250, 250};
int currentMelody = 0;
bool isPlaying = false;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;
void setup() {
  pinMode(buzzerPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
```

```
  // Настройка прерывания на кнопку
  attachInterrupt(digitalPinToInterrupt(buttonPin), buttonPressed, FALLING);
}
void loop() {
  if (!isPlaying) {
    isPlaying = true;
    if (currentMelody == 0) {
      playMelody(marioNotes, marioDurations, sizeof(marioNotes) / sizeof(int));
    } else {
      playMelody(mashNotes, mashDurations,
                 sizeof(mashNotes) / sizeof(int)); // MASH
    }
    isPlaying = false;
  }
}

void playMelody(int notes[], int durations[], int length) {
  for (int i = 0; i < length; i++) {
    int frequency = notes[i];
    int duration = durations[i] * 3;
    if (frequency > 0) {
      tone(buzzerPin, frequency, duration);
    } else {
      noTone(buzzerPin);
    }
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin3, LOW);
    delay(duration / 3);
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, HIGH);
    digitalWrite(ledPin3, LOW);
    delay(duration / 3);
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin3, HIGH);
    delay(duration / 3);
```

```
    if (!isPlaying) {
      noTone(buzzerPin);
      return;
    }
  }
  noTone(buzzerPin);
}
void buttonPressed() {
  if (millis() - lastDebounceTime > debounceDelay) {
    lastDebounceTime = millis();
    isPlaying = false;
    currentMelody = (currentMelody + 1) % 2;
  }
}
```

Приложение А2. Исходный код решения для задания №2

```
#include <Arduino.h>

#define BUZZER_PIN 3
#define LDR_PIN A0
const int notes[] = {262, 294, 330, 349, 392, 440, 494};
void setup() {
  pinMode(BUZZER_PIN, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int val = analogRead(LDR_PIN);
  Serial.println(val);
  int noteIndex = map(val, 1, 310, 0, 6);
  int frequency = notes[noteIndex];
  tone(BUZZER_PIN, frequency);
  delay(200);
}
```

Приложение А3. Исходный код решения для задания №3

```
#include <Arduino.h>

#define BUZZER_PIN 13
#define BUTTON_PIN_1 5
#define BUTTON_PIN_2 6
#define BUTTON_PIN_3 7
#define BUTTON_PIN_4 8
#define BUTTON_PIN_5 9
const int notes[] = {131, 147, 165, 175, 196};
void setup() {
  pinMode(BUZZER_PIN, OUTPUT);
  for (int i = BUTTON_PIN_1; i <= BUTTON_PIN_5; i++) {
    pinMode(i, INPUT_PULLUP);
  }
}
void loop() {
  for (int i = 0; i < 5; i++) {
    if (digitalRead(BUTTON_PIN_1 + i) == LOW) {
      playNote(notes[i]);
      return;
    }
  }
  noTone(BUZZER_PIN);
}
void playNote(int frequency) {
  tone(BUZZER_PIN, frequency);
  delay(100);
}
```

Приложение А4. Исходный код решения для задания №4

```
#include <Arduino.h>

#define LED_COUNT 14
#define BUTTON1_PIN 3
#define BUTTON2_PIN 2
int ledPins[LED_COUNT] = {13, 12, 11, 10, 9, 8, 7, 6, 5, A0, A1, A2, A3, A4};
int currentLed = LED_COUNT / 2 - 1;
void setup() {
  for (int i = 0; i < LED_COUNT; i++)
    pinMode(ledPins[i], OUTPUT);
  pinMode(BUTTON1_PIN, INPUT);
  pinMode(BUTTON2_PIN, INPUT);
  digitalWrite(ledPins[currentLed], HIGH);
}
void loop() {
  if (digitalRead(BUTTON1_PIN) && currentLed > 0)
    moveLed(-1);
  if (digitalRead(BUTTON2_PIN) && currentLed < LED_COUNT - 1)
    moveLed(1);
  if (currentLed == 0 || currentLed == LED_COUNT - 1)
    victory();
}
void moveLed(int dir) {
  digitalWrite(ledPins[currentLed], LOW);
  currentLed += dir;
  digitalWrite(ledPins[currentLed], HIGH);
  delay(100);
}
void victory() {
  for (int i = 0; i < LED_COUNT; i++)
    digitalWrite(ledPins[i], i < LED_COUNT / 2 ? currentLed == 0
                                               : currentLed == LED_COUNT - 1);
}
```