

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет Информатика и вычислительная техника

Кафедра Кибербезопасность информационных систем

**ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Автор |  | | |  | Ковалев Д.П. | | | |
|  | (подпись, дата) | | |  |  | | | |
| Обозначение | ИиВТ.10.05.01.13 | | Группа | | | | ВКБ32 |  |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | | | | |  |
| Профиль | Математические методы защиты информации | | | | | | |  |
| Проверил |  | | |  | |  | |  |
|  | (подпись, дата) | | |  | |  | |  |

Ростов-на-Дону

2024

Содержание

[Лабораторная работа №1 3](#_Toc159059439)

[Лабораторная работа №2 4](#_Toc159059440)

[Лабораторная работа №3 8](#_Toc159059441)

[Лабораторная работа №4 10](#_Toc159059442)

[Лабораторная работа №5 12](#_Toc159059443)

[Лабораторная работа №6 14](#_Toc159059444)

[Лабораторная работа №7 16](#_Toc159059445)

# 17

# **Лабораторная работа №1**

**Тема:** «Изучения создания сети в Cisco Packet Tracer»

**Цель:** научиться создавать простые сетевые соединения.

**Задание 1**. Создание сети из двух ПК в программе Cisco Parket Tracer.

Нажмем на кнопку “End devices”, где находятся персональные компьютеры. На рисунке 1 также видно знак молнии, чтобы подключить наши 2 компьютера. Во вкладке “Connections” есть медные провода с обжимом.

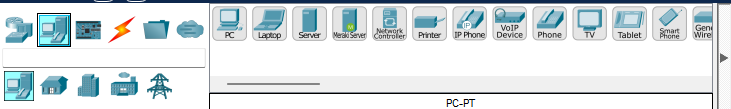


Рисунок 1 – интерфейс программы с Drag & Drop

В результате проделанных операций получилось подключение, которое представлено на рисунке 2.

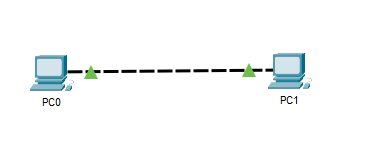


Рисунок 2 – физическое подключение 2 ПК

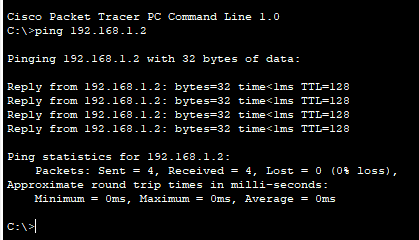
 Далее нам нужно настроить IP адрес и маску для каждого ПК, как было показано в методическом материале. Теперь можно приступить к проверке видимости другого ПК в локальной сети с помощью команды “ping 192.168.1.2”. На рисунке 3 представлен результат.

Рисунок 3 – результат команды ping

**Задание 2.** Организация Режим симуляции работы сети.

Нужно задать в рабочем пространстве программы сеть из 4х ПК и 2х хабов. Результат подключения представлен на рисунке 4.

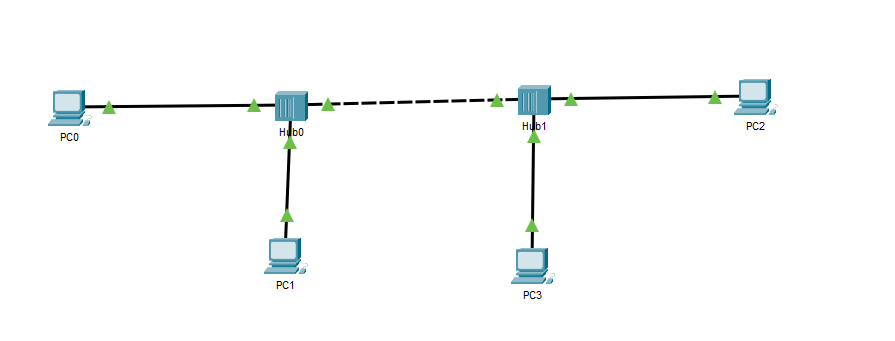


Рисунок 4 – результат подключения устройств

Далее я отключил все протоколы, оставив только ICMP (Internet Control Message Protocol) — сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных. Фильтры были отключены, как представлено на рисунке 5.

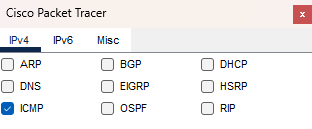


Рисунок 5 – фильтры, которые я использовал

Теперь создадим пакет ICMP, который будет передаваться от 1 ПК к другому, здесь вот создал я пакет, который будет в дальнейшем отправлен, данный конверт представлен на рисунке 6.

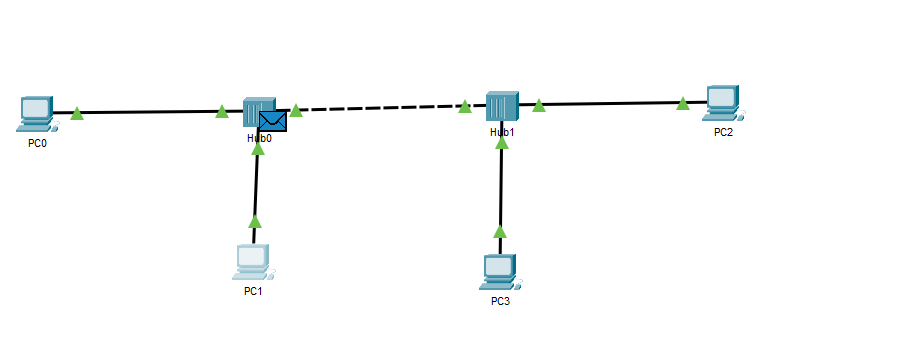


Рисунок 6 – процесс отправки пакетов с 1 ПК на другой

Чтобы отображалось в списке событий, нужно нажать на английскую букву “P” или нажать на иконку “письмо”. Дальше остается только связать два элемента. В результате должно появиться, как на рисунке 7.



Рисунок 7 – список событий после прогона пакета

**Задание 3.** Настройка сетевых параметров ПК в его графическом интерфейсе.

Настроить IP адрес и маску подсети можно, используя несколько вариантов. Первый вариант – использование GUI интерфейса, другой вариант – это использование командной строки. В нашем случае подойдет команда `ipconfig 192.168.0.1 255.255.255.0`, где первый аргумент – это IP, а второй 0 маска. Также с помощью данной команды можно проверять настройки. Ipconfig для 4 ПК представлен на рисунке 8.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 8 – ipconfig для 4 ПК

# **Лабораторная работа №2**

**Тема:** «Изучение построения сетей с различными топологиями»

**Цель:** научиться строить сеть с топологией звезда.

**Задание 1**. Моделирование сети с топологией звезда на базе концентратора.

Для начала нам нужно понять, что такое топология “Звезда”? На самом деле – это когда много объектов подключаются к центральному звену, в качестве центра, грубо говоря, может быть хаб, коммутатор или ПК. Сама топология представлена на рисунке 1.

Изображение выглядит как снимок экрана, текст, линия, диаграмма

Автоматически созданное описание

Рисунок 1 – топология звезда

Отправка пакетов в GUI делается с помощью “конвертика” в верхней панели, фото данного участка представлено на рисунке 2.



Рисунок 2 – фото раздела

После тот, как мы нажали на конверт, то мы теперь можем выбрать от какого ПК отправить к какому. В результате проделанной операции мы получим статус продвижения пакета. Все показано на рисунке 3.



Рисунок 3 – отчет о прохождении пакета

Далее требуется построить ещё одну сеть с целью изучения полезных приемов в CPT. В Cisco Packet Tracer можно копировать объекты, зажимая Ctrl и выбирая определенные объекты через ЛКМ. Новая сеть представлена на рисунке 4.

Изображение выглядит как диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 4 – сеть с использованием хаба и свитча

**Задание 2**. Моделирование сети с топологией звезда на базе коммутатора.

Нам нужно построить сеть на основе свитча. Здесь можно заметить, что свитч отправляет пакеты только определенному потребителю в прямом и обратном направлении. Сама сеть представлена на рисунке 5.

Изображение выглядит как снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 5 – топология звезда с использованием свитча

Теперь по заданию нужно построить с использованием хаба и свитча. Сеть представлена на рисунке 6.

Изображение выглядит как диаграмма, снимок экрана

Автоматически созданное описание

Рисунок 6 – сеть с использованием свитчи и хаба

Тестировать подключение будет с использованием команды ping, делая запрос от к PC1 к PC3. Результат запроса представлен на рисунке 7. Там видно, что происходит 4 запроса, в среднем задержка 2 ms, что неплохо я считаю.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – результат запроса ping от PC0 к PC3

**Задание 3.** Исследование качества передачи трафика по сети

Теперь мы здесь будем тестировать различные сети, построенные на хабе и коммутаторе по задержкам. В начале я построю сеть, где в центральной части будет хаб, все представлено на рисунке 8.

Изображение выглядит как диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 8 – строение сети, где центральный элемент – хаб

Я отправил в traffic generator PC1, а начал отправлять пакеты с 0 ПК на PC3. Потеряно где-то 6 пакетов. Я сделал поменьше запросов с той целью, чтобы долго не отчитывать. Количество потерянных пакетов представлено на рисунке 9.

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Рисунок 9 – потеря пакетов с хабом.

Теперь попробуем поменять так, чтобы у нас был свитч, а не хаб. В результате потеря пакетов составляет в количестве 4 шт., что получше намного, чем с хабом. Результат представлен на рисунке 10.

Изображение выглядит как снимок экрана, текст, Шрифт, линия

Автоматически созданное описание

Рисунок 10 – потеря пакетов со свитчем

# **Лабораторная работа №3**

**Тема:** «Знакомство с командами Cisco IOS»

**Цель:** научиться использовать командную строку Cisco Packet Tracer

**Задание 1.** Изучить работу CLI.

Столкнулся с проблемой, что нельзя выйти из настройки интерфейсов, используя команду Interface. Помогает только закрытие файла или простое ожидание. Все представлено на рисунке 1.



Рисунок 1 – попытка настроить интерфейсы

Установку пароля делал, как представлено на рисунке 2. Здесь главное в начале войти в config через команду `conf t`, а потом уже вводить настройки для установки.

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рисунок 2 – установка пароля на роутер

Теперь я проверил, что для входа в привилегированный режим нужно использовать пароль, все работает. Для выхода из данного режима нужно ввести `logout`. Результат проверки находится на рисунке 3.

Изображение выглядит как текст, Шрифт, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 3 – проверка входа в привилегированный режим

C помощью secret поменяли пароль для входа пользователя в привилегированный режим. Для сохранения всех наших операций можем воспользоваться сохранением пароля в память, используя `write memory`.

# **Лабораторная работа №4**

**Тема:** «Работа с кнопками»

**Цель:** При нажатии кнопки, запрограммировать плату на воспроизведение короткого фрагмента, выбранной Вами музыкальной композиции (количество пьезоэлементов выбирайте самостоятельно).

На рисунке 5 показана реализация музыкального фрагментам при помощи пьезоэлемента, где правая кнопка отвечает за запуск, а левая за остановку. Резисторы подключаются к кнопкам для понимажения напряжения и увеличения сопротивления.

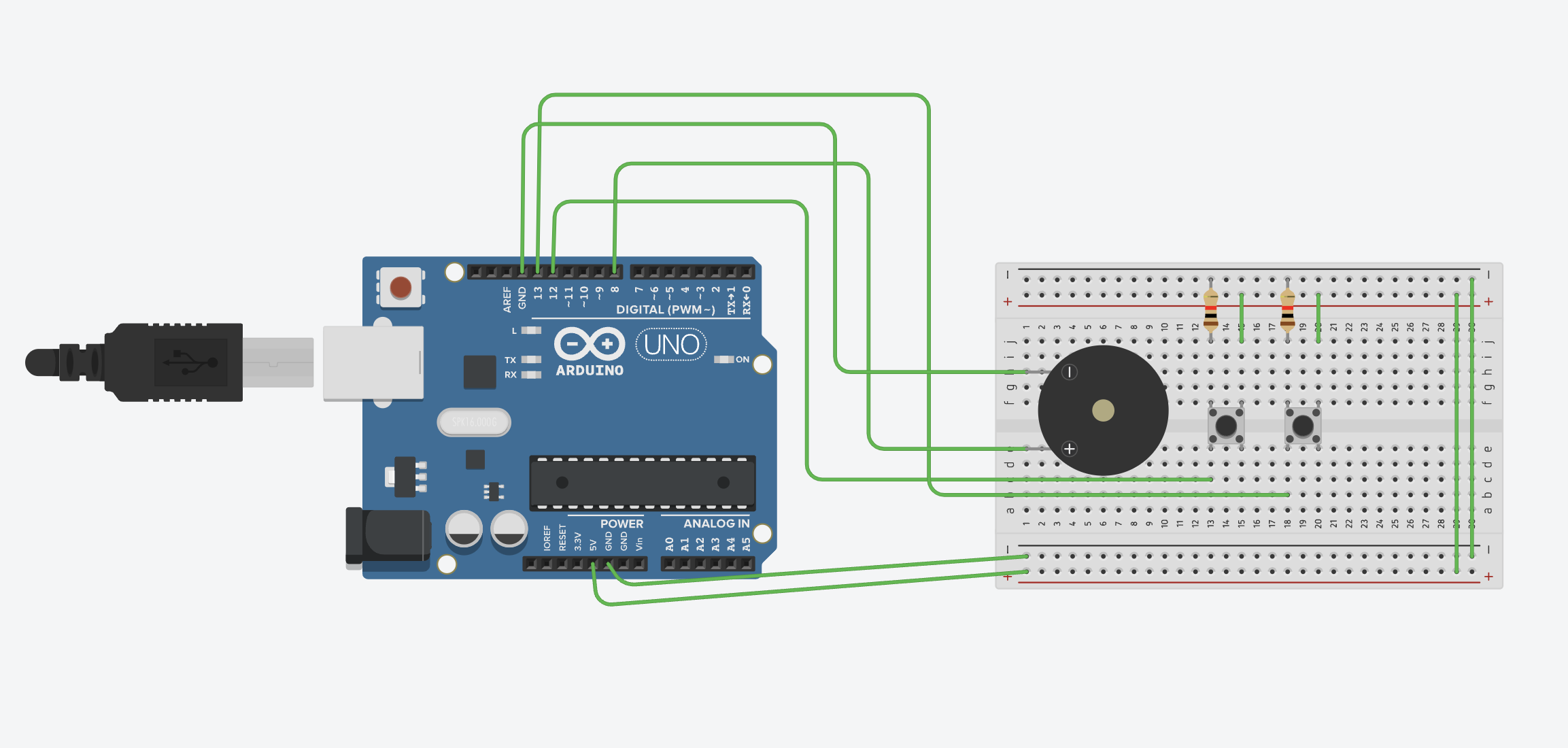
****

Рисунок 5 – Реализация задания с пьезоэлементами и кнопками

Программный код:

// C++ code

//

int playButtonPin = 13;

int stopButtonPin = 12;

int piezoPin = 8;

int notes[] = {262, 262, 294, 262, 349, 330, 262, 262, 294, 262, 392, 349, 262, 262, 523, 440, 349, 330, 294, 466, 466, 440, 349, 392, 349};

int duration[] = {4, 4, 8, 8, 8, 4, 4, 8, 8,ч 8, 4, 4, 8, 8, 4, 4, 8, 8, 8, 4, 4, 8, 8, 8, 4};

void setup() {

pinMode(playButtonPin, INPUT);

pinMode(stopButtonPin, INPUT);

pinMode(piezoPin, OUTPUT);

}

void loop() {

if (digitalRead(playButtonPin) == HIGH) {

for (int i = 0; i < sizeof(notes) / sizeof(notes[0]); i++) {

tone(piezoPin, notes[i], duration[i] \* 100);

delay(duration[i] \* 100);

if (digitalRead(stopButtonPin) == HIGH) {

noTone(piezoPin);

break;

}

}

}

}

# **Лабораторная работа №5**

**Тема:** «Работа с ЖК-дисплеем»

**Цель:** Подсоединить к плате несколько светодиодов, на экран выводить номер и цвет светодиода.

Для работы с ЖК-дисплеем в инструментах Tinkercad имеется встроенная плата. На рисунке 6 к готовой плате подключается 3 резистора к соответствующим трём светодиодам цветов светофора.

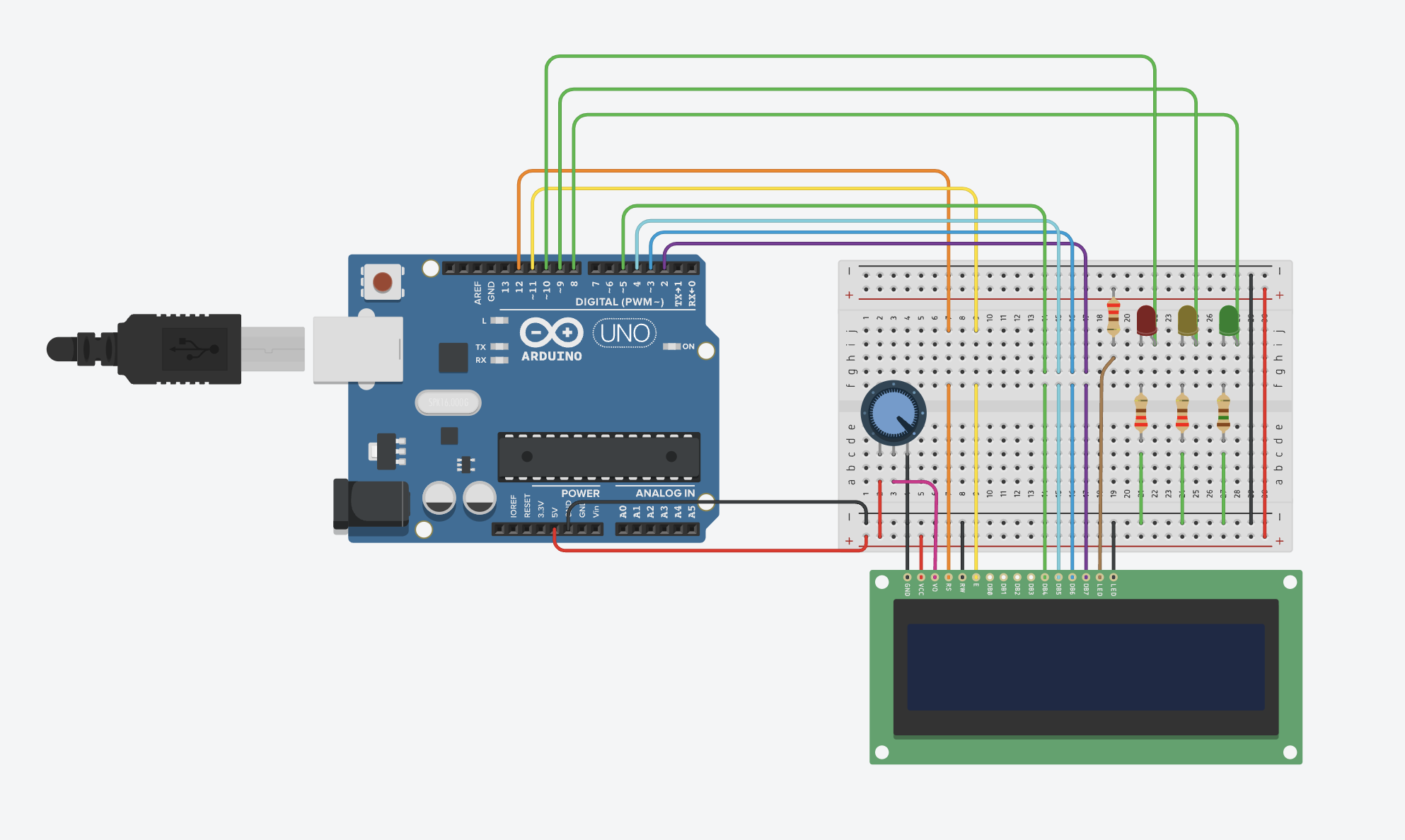


Рисунок 6 – Реализация задания с ЖК-дисплеем

Программный код:

// C++ code

//

#include <LiquidCrystal.h>

int led1 = 8;

int led2 = 9;

int led3 = 10;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup(){

lcd.begin(0, 15);

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3,OUTPUT);

}

void loop(){

digitalWrite(led1, HIGH);

digitalRead(led1) == HIGH;

lcd.setCursor(8, 1);

lcd.print("8 Green");

delay(3000);

digitalRead(led1) == HIGH;

lcd.clear();

digitalWrite(led1, LOW);

delay(1000);

digitalWrite(led2, HIGH);

digitalRead(led2) == HIGH;

lcd.setCursor(0, 1);

lcd.print("9 Yellow");

delay(3000);

digitalRead(led2) == HIGH;

lcd.clear();

digitalWrite(led2, LOW);

delay(1000);

digitalWrite(led3, HIGH);

digitalRead(led3) == HIGH;

lcd.setCursor(7, 0);

lcd.print("10 Red");

delay(3000);

digitalRead(led3) == HIGH;

lcd.clear();

digitalWrite(led3, LOW);

delay(1000);

}

# **Лабораторная работа №6**

**Тема:** «Изучение кода»

**Цель:** Написать код, задающий работу железнодорожного светофора.

На рисунке 7 представлена реализация железнодорожного светофора с помощью трёх светодиодов, четырёх резисторов и одного пьезоэлемента.

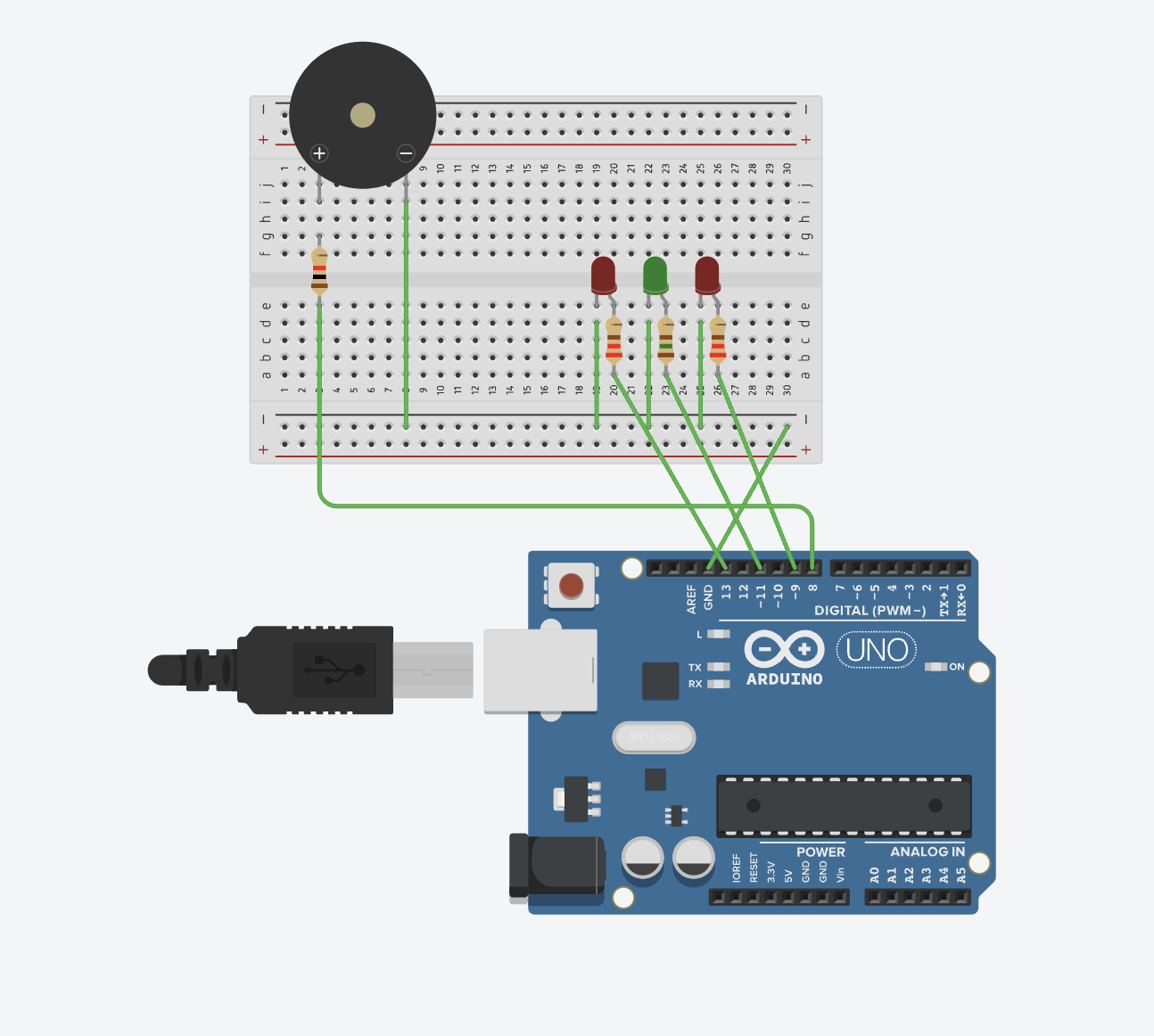


Рисунок 7 – Реализация железодорожного светофора

Программный код:

// C++ code

//

int counter;

void setup()

{

pinMode(8, OUTPUT);

pinMode(13, OUTPUT);

pinMode(9, OUTPUT);

pinMode(11, OUTPUT);

}

void loop()

{

for (counter = 0; counter < 5; ++counter) {

tone(8, 450, 500);

digitalWrite(13, HIGH);

delay(500);

tone(8, 550, 500);

digitalWrite(13, LOW);

delay(500);

tone(8, 500, 500);

digitalWrite(9, HIGH);

delay(500);

tone(8, 600, 500);

digitalWrite(9, LOW);

delay(500);

noTone(8);

}

digitalWrite(11, HIGH);

delay(5000);

digitalWrite(11, LOW);

}

# **Лабораторная работа №7**

**Тема:** «Самостоятельное кодирование платы»

**Цель:** Запрограммировать плату так, чтобы начальная температура была 40 градусов (при этом не горит ни один светодиод), потом при поднятии температуры на каждые 10 градусов зажигался один светодиод.  
(50 град.- один, 60 град.- два, 70 град.- все три светодиода).

Используя термодатчик и три светодиода на рисунке 3 показан способ реализации задания.

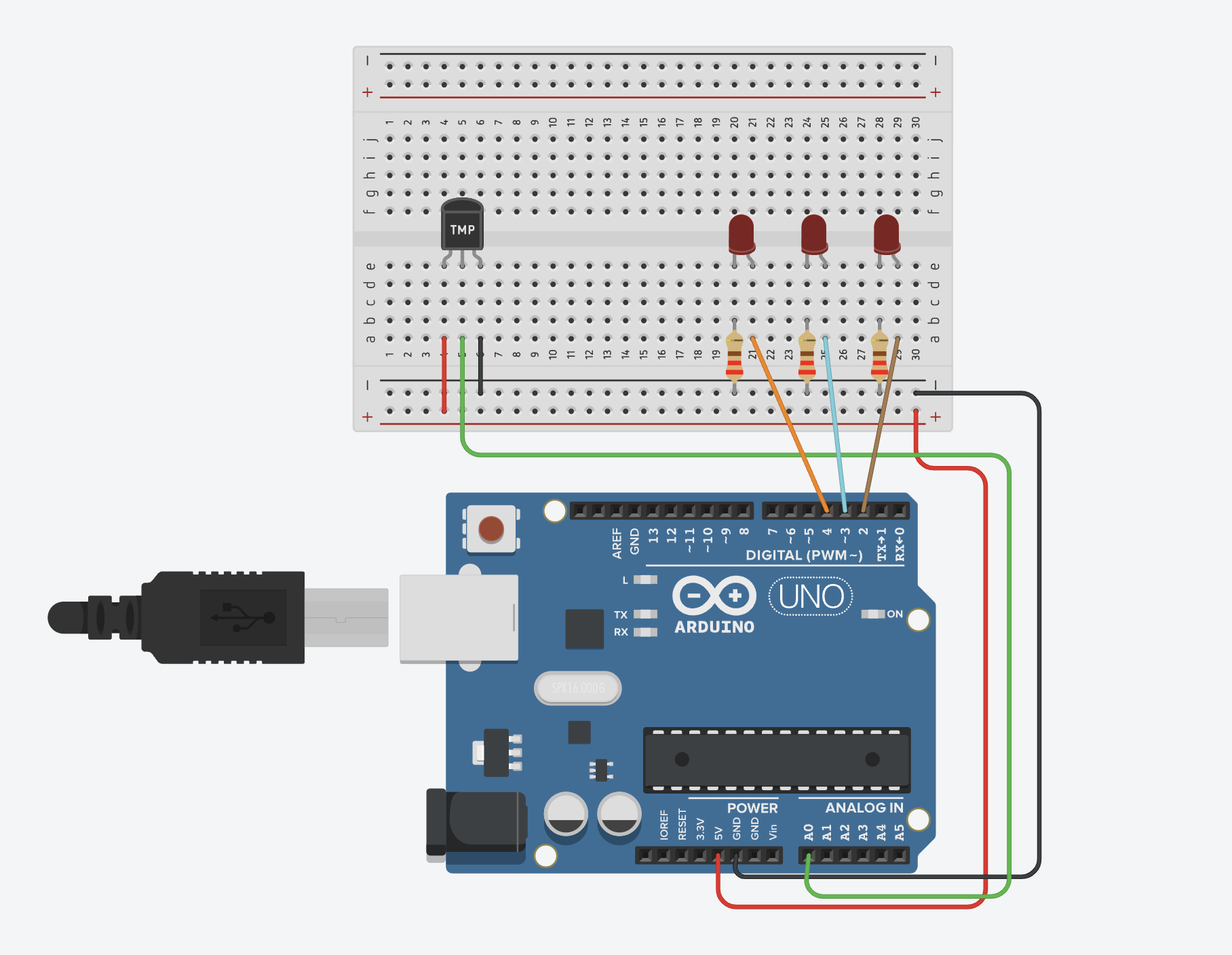


Рисунок 8 – Реализация с помощью термодатчика

Программный код:

// C++ code

//

float tempC;

void setup()

{

pinMode(A0, INPUT);

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

}

void loop()

{

tempC = map(((analogRead(A0) - 20) \* 3.04), 0, 1023, -40, 125);

if (tempC > 40) {

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

}

if (tempC > 50) {

digitalWrite(2, HIGH);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

}

if (tempC > 60) {

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, LOW);

}

if (tempC > 70) {

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

}

if (tempC >= 70) {

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

}

delay(1000);

}