ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**Отчет по практической работе №2**

**«СПЬЕЗОЭЛЕМЕНТ, МИКРОСХЕМЫ и СВЕТОДИОДНЫЕ**

**СБОРКИ»**

по дисциплине

«Информационные технологии»

|  |  |
| --- | --- |
| Выполнил: | студент гр. ИТб-2302-02-20 Ердяков Р.А. |
| Проверила: | ст. преподаватель каф. САУ Шмакова Н. А. |

Киров 2024

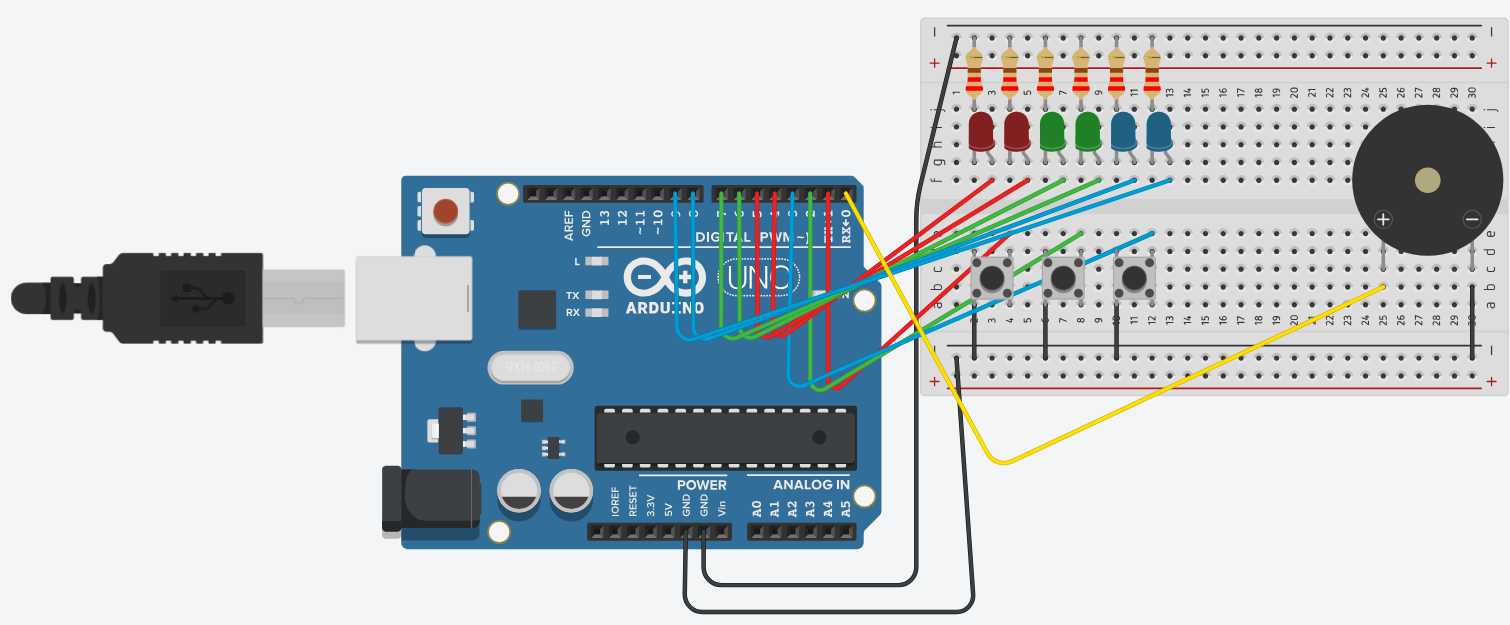
Задание № 1

**«Кнопочные ковбои»**

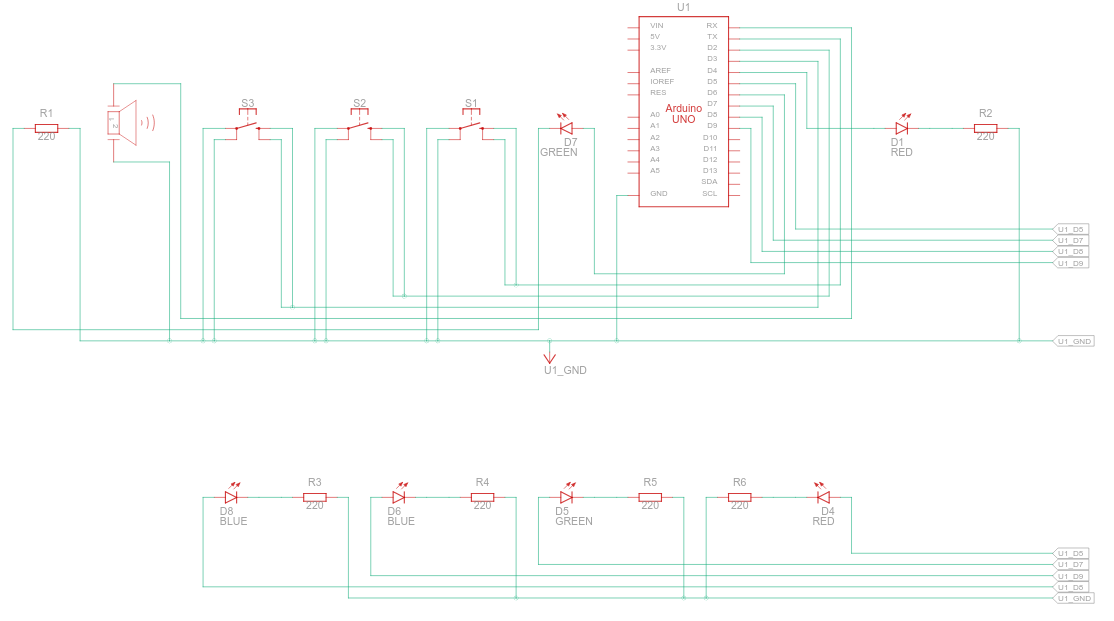
**Задание:** Добавить светодиоды в схему (всего в сборке 6 шт.) В начале игры светодиоды могут быть включены или выключены. Если светодиоды выключены, то логика такая: игрок зарабатывает бонусы за нажатие, по нажатию на кнопку, если игрок нажал первый после звукового сигнала, то он зарабатывает бонус.

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования.

**Схема сборки на макетной плате:**



**Принципиальная схема:**

****

**Листинг программы:**

#**define** BUZZER\_PIN 0 // пин с пищалкой

#**define** PLAYER\_COUNT 3 // количество игроков ковбоев

int buttonPins[PLAYER\_COUNT] = {1, 2, 3};

int ledPins[PLAYER\_COUNT] = {4, 6, 8};

int gamePoints[PLAYER\_COUNT] = {0, 0, 0};

bool game = true;

void **setup**() {

pinMode(BUZZER\_PIN, OUTPUT);

**for** (int player = 0; player < PLAYER\_COUNT; ++player) {

pinMode(ledPins[player],OUTPUT);

pinMode(ledPins[player] + 1,OUTPUT);

pinMode(buttonPins[player], INPUT\_PULLUP);

}

}

int **getPlayerLedPin**(int playerIndex)

{

int ledPin = ledPins[playerIndex];

ledPins[playerIndex] = ledPin + 1;

**return** ledPin;

}

bool **writePoint**(int playerIndex)

{

gamePoints[playerIndex] = gamePoints[playerIndex] + 1;

**return** gamePoints[playerIndex] == 2 ? true : false;

}

bool **isFalseStart**()

{

**for** (int player = 0; player < PLAYER\_COUNT; ++player) {

**if** (!digitalRead(buttonPins[player])) {

**return** true;

}

}

**return** false;

}

void **loop**() {

**while** (game) {

delay(random(2000, 7000));

**while** (isFalseStart()) {

delay(random(2000, 7000));

}

tone(BUZZER\_PIN, 3000, 250); // 3 килогерца, 250 миллисекунд

**for** (int player = 0; ;player = (player+1) % PLAYER\_COUNT) {

// если игрок номер «player» нажал кнопку...

**if** (!digitalRead(buttonPins[player])) {

// ...включаем его светодиод и сигнал победы

digitalWrite(getPlayerLedPin(player), HIGH);

delay(1000);

**if** (writePoint(player)) {

game = false;

}

**break**; // Есть победитель! Выходим (англ. break) из цикла

}

}

}

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/5ON01AoX9i7-prakticheskaya-2-knopochnye-kovboi-/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fcollections%2F4AaRg4tRxbf%2Fcircuits&sharecode=vyzAjMB8ii6XIxPM6UoC0GGXTcMnzvnMkpkZnLp4OqQ

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования.

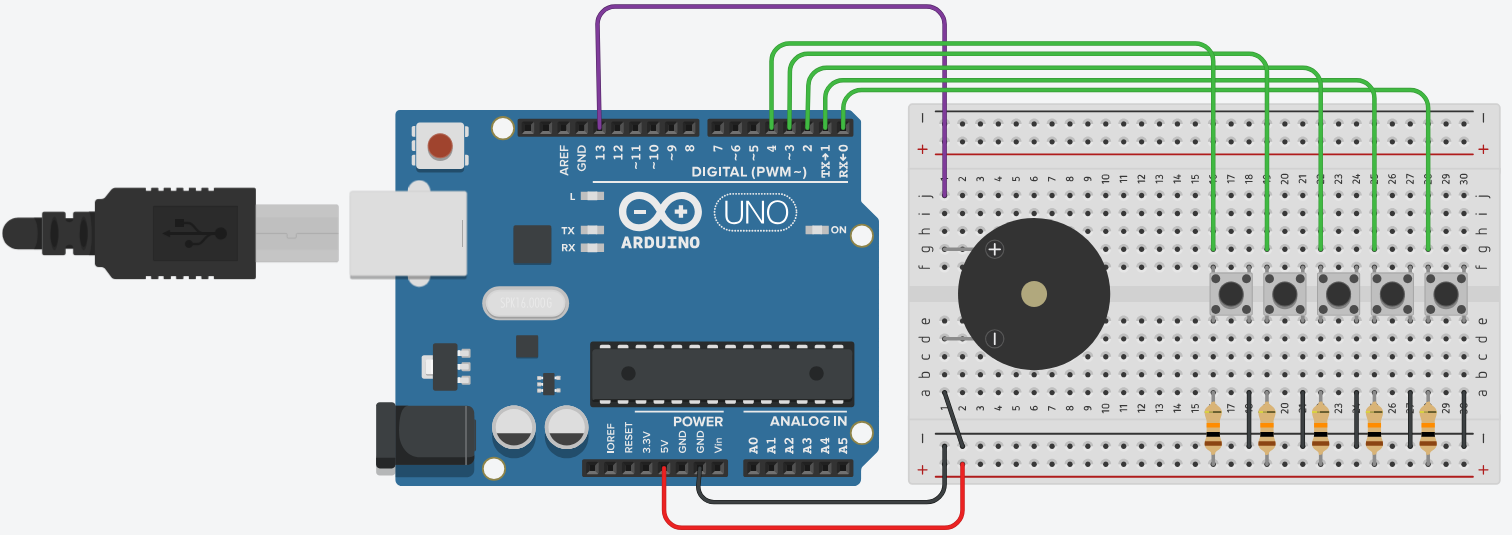
Задание № 2

**«Мерзкое пианино»**

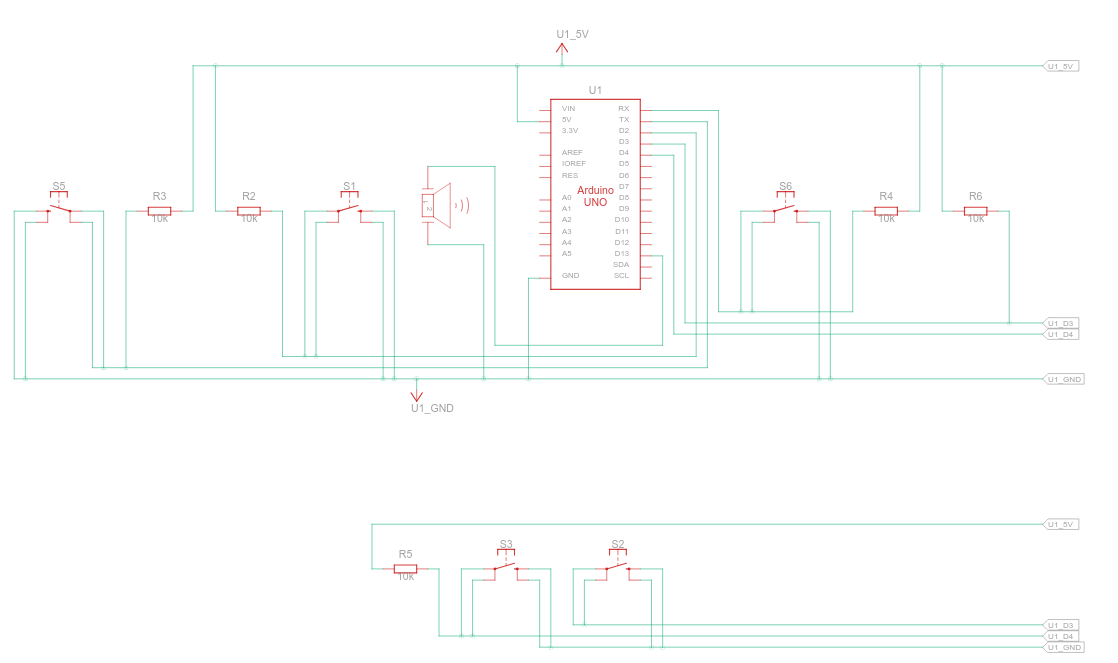
**Задание:**  Добавить в схему две кнопки. При нажатии на каждую кнопку должна проигрываться определенная частота

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования.

**Схема сборки на макетной плате:**

****

**Принципиальная схема:**

****

**Листинг программы:**

#**define** BUZZER\_PIN 13 // пин с пищалкой (англ. «buzzer»)

#**define** FIRST\_KEY\_PIN 0 // первый пин с клавишей (англ. «key»)

#**define** KEY\_COUNT 5 // общее количество клавиш

int noteFrequencies[KEY\_COUNT];

float frequencies[KEY\_COUNT];

float baseFrequency = 261.63;

int semitoneOffset[] = {0, 2, 4, 5, 7};

void **setup**() {

pinMode(BUZZER\_PIN, OUTPUT);

**for** (int i = 0; i < KEY\_COUNT; ++i) {

pinMode(FIRST\_KEY\_PIN + i, INPUT\_PULLUP);

frequencies[i] = baseFrequency \* pow(2, semitoneOffset[i] / 12.0);

}

}

void **loop**() {

**for** (int i = 0; i < KEY\_COUNT; ++i) {

int keyPin = i + FIRST\_KEY\_PIN;

boolean keyUp = digitalRead(keyPin);

**if** (!keyUp) {

tone(BUZZER\_PIN, frequencies[keyPin], 20);

}

}

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/5ukGaIyB69T-prakticheskaya-2-merzkoe-pianino/editel?returnTo=%2Fthings%2F5ukGaIyB69T-prakticheskaya-2-merzkoe-pianino&sharecode=dZ2uNbH9cbHvCHFIzydk7AxKGyd0TWsU6kISYwW7hpU

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования, собрал схему с пищалкой

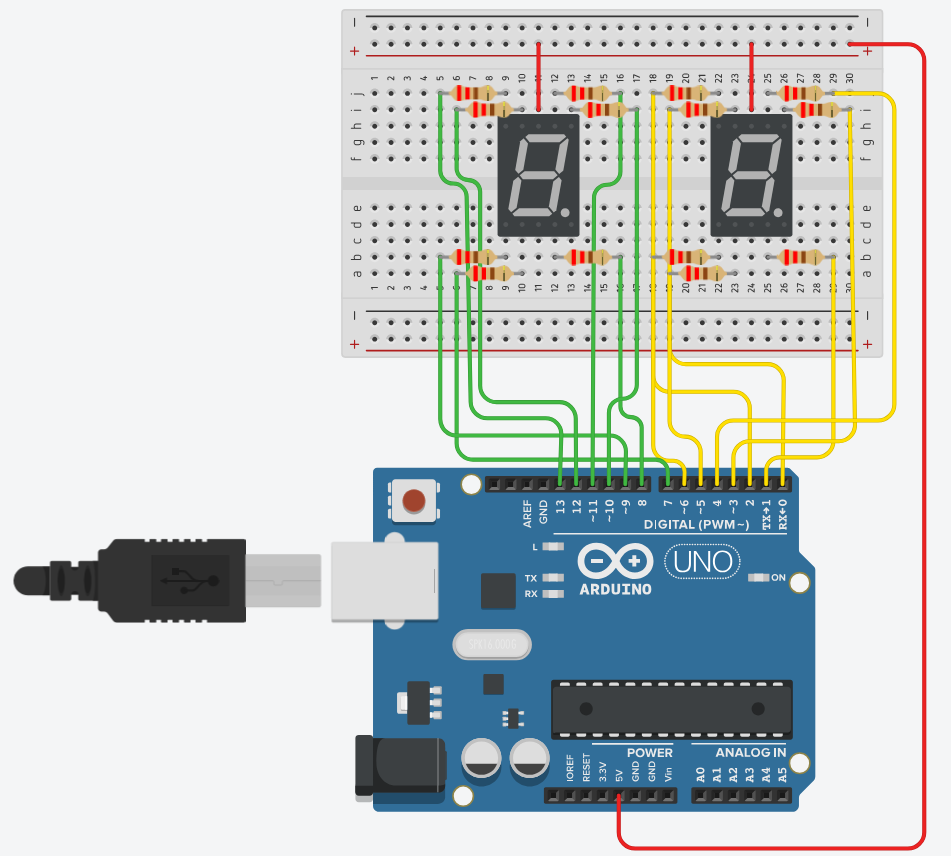
Задание № 3

**«Показатели Мерсенна»**

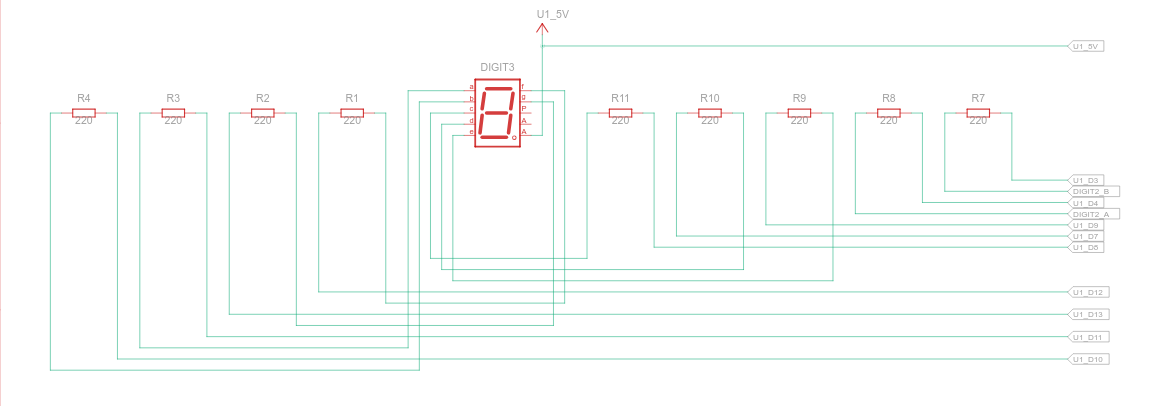
**Задание:**  Вывести последовательность Показатели Мерсенна (A000043). Сборка на общем аноде.

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования, познакомиться с работой семисигментных дисплеев.

**Схема сборки на макетной плате:**

****

**Принципиальная схема:**

**Листинг программы:**

#define UNITS\_SEGMENT\_PIN 0

#define TENS\_SEGMENT\_PIN 7

#define SEGMENT\_COUNT 7

**byte** numberSegments[10] = {

**0b00111111**, 0b00001010, 0b01011101, 0b01011011, 0b01101010,

**0b01110011**, 0b01110111, 0b00011010, 0b01111111, 0b01111011,

};

**void** setup()

{

**for** (int i = 0; i < SEGMENT\_COUNT; ++i) {

**pinMode**(i + UNITS\_SEGMENT\_PIN, OUTPUT);

**pinMode**(i + TENS\_SEGMENT\_PIN, OUTPUT);

}

}

**void** showNumber(int firstSegmentPin, int number)

{

**int** mask = numberSegments[number];

**for** (int i = 0; i < SEGMENT\_COUNT; ++i) {

**boolean** enableSegment = !bitRead(mask, i);

**digitalWrite**(i + firstSegmentPin, enableSegment);

}

}

**int** currentIndex = 2;

**bool** isPrime(long long num) {

**if** (num <= 1) return false;

**for** (long long i = 2; i \* i <= num; i++) {

**if** (num % i == 0) return 0;

}

**return** 1;

}

**unsigned** long long getMersennePrime()

{

**for** (int i = currentIndex; i <= 61; i++) {

**if** (isPrime(i)) {

**unsigned** long long mersenne = (1LL << i) - 1;

**if** (isPrime(mersenne)) {

**currentIndex** = i + 1;

**return** i;

}

}

}

**currentIndex** = 2;

**return** currentIndex++;

}

**void** loop()

{

**unsigned** long long mersennePrime = getMersennePrime();

**showNumber**(UNITS\_SEGMENT\_PIN, mersennePrime % 10);

**showNumber**(TENS\_SEGMENT\_PIN, mersennePrime / 10);

**delay**(1000);

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/6TjrYxSnsWD-prakticheskaya-2-pokazateli-mersenna/editel?returnTo=%2Fthings%2F6TjrYxSnsWD-prakticheskaya-2-pokazateli-mersenna&sharecode=X9h4KxW\_l0beNXADWl\_T3T7edskg6A9M6qlqoO6bEn8

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования, собрал схему с семисегментными дисплеями.

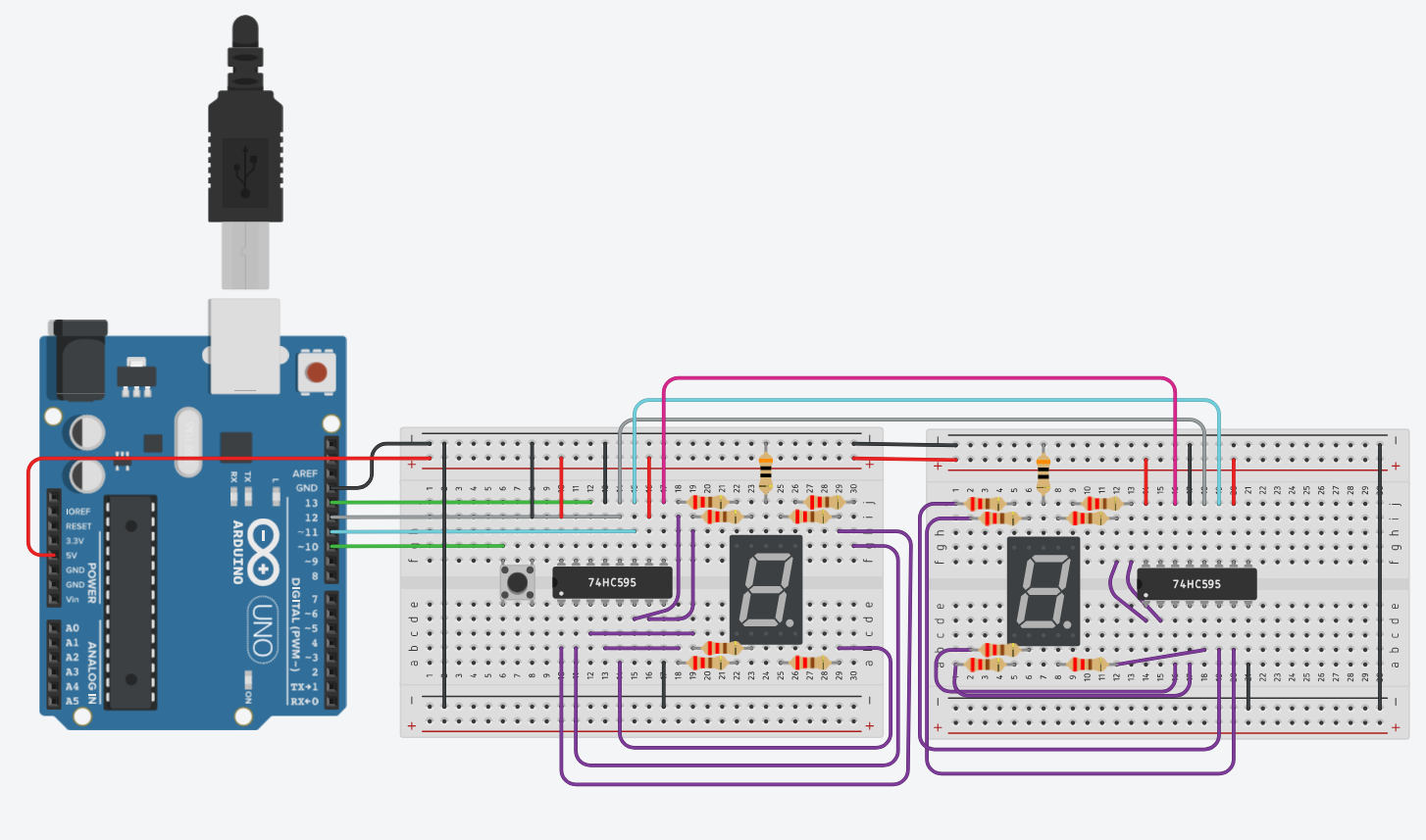
Задание № 4

**«Счетчик нажатий на сдвиговом регистре 595»**

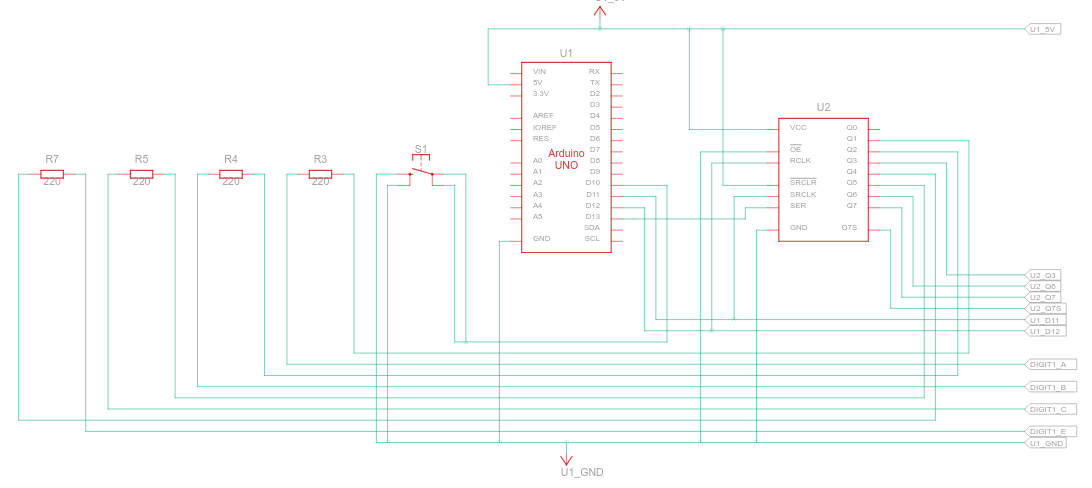
**Задание:** Вывести последовательность простых чисел (A000040), по нажатию кнопки

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования, познакомиться с работой регистра 595.

**Схема сборки на макетной плате:**



**Принципиальная схема:**

****

**Листинг программы:**

#**define** DATA\_PIN 13 // пин данных (англ. data)

#**define** LATCH\_PIN 12 // пин строба (англ. latch)

#**define** CLOCK\_PIN 11 // пин такта (англ. clock)

#**define** BUTTON\_PIN 10

boolean buttonWasUp = true;

byte segments[10] = {

0b01111101, 0b00100100, 0b01111010, 0b01110110, 0b00100111,

0b01010111, 0b01011111, 0b01100100, 0b01111111, 0b01110111

};

void **setup**()

{

pinMode(DATA\_PIN, OUTPUT);

pinMode(CLOCK\_PIN, OUTPUT);

pinMode(LATCH\_PIN, OUTPUT);

pinMode(BUTTON\_PIN, INPUT\_PULLUP);

}

int **isPrime**(int num)

{

**for** (int i = 2; i < num; i++) {

**if** (num % i == 0) {

**return** 0;

}

}

**return** 1;

}

int CURRENT\_PRIME = 2;

void **updateCurrentPrime**()

{

**for** (int i = CURRENT\_PRIME + 1; i <= 100; i++) {

**if** (isPrime(i)) {

CURRENT\_PRIME = i;

**return**;

}

}

CURRENT\_PRIME = 2;

**return**;

}

void **loop**()

{

**if** (buttonWasUp && !digitalRead(BUTTON\_PIN)) {

//delay(10);

**if** (!digitalRead(BUTTON\_PIN)) {

updateCurrentPrime();

}

}

int tens = CURRENT\_PRIME / 10;

int ones = CURRENT\_PRIME % 10;

buttonWasUp = digitalRead(BUTTON\_PIN);

digitalWrite(LATCH\_PIN, LOW);

shiftOut(DATA\_PIN, CLOCK\_PIN, LSBFIRST, segments[ones]);

shiftOut(DATA\_PIN, CLOCK\_PIN, LSBFIRST, segments[tens]);

digitalWrite(LATCH\_PIN, HIGH);

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/fcBcsnlg9RB-prakticheskaya-2-schetchik-nazhatij-so-sdvigom-registra-595/editel?returnTo=%2Fthings%2FfcBcsnlg9RB-prakticheskaya-2-schetchik-nazhatij-so-sdvigom-registra-595&sharecode=6hDj-HU9SbtthmU\_ACh1nB-0grEJZS4oX584eNfJQEo

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования, собрал схему с регистром 595.

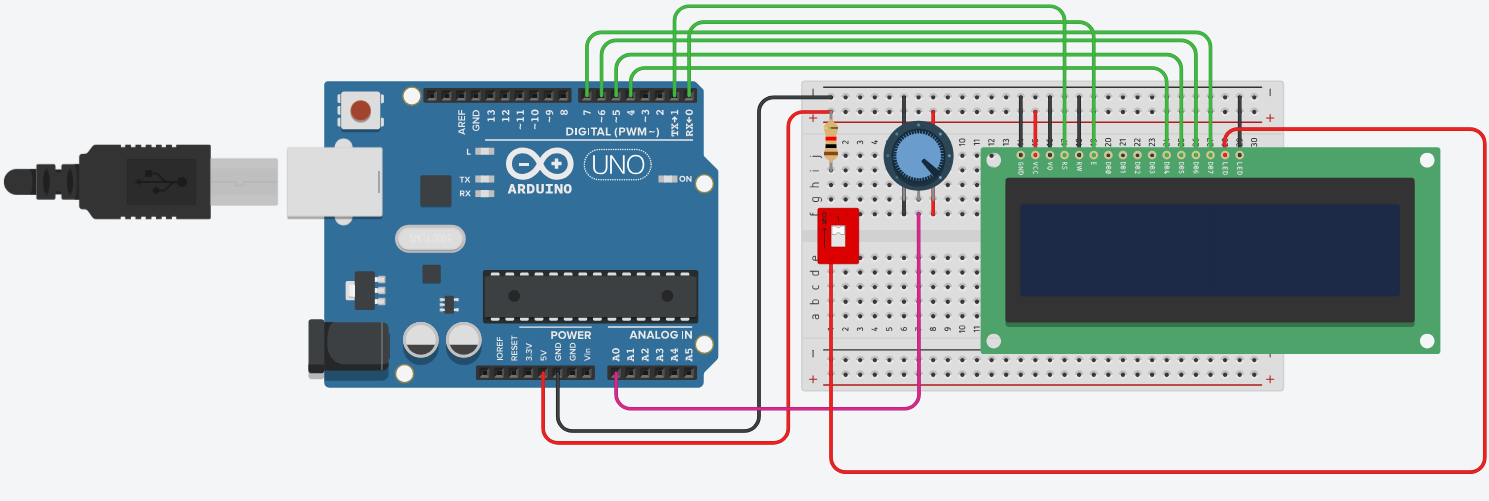
Задание № 5

**«Вывод текста»**

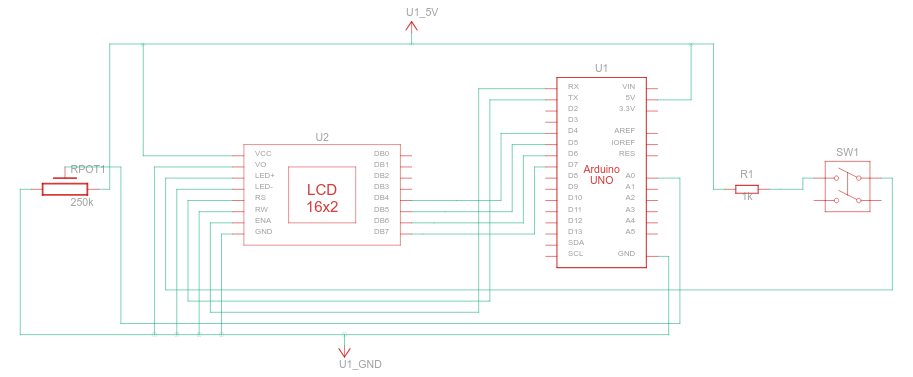
**Задание:** Вывести текст с прокруткой с помощью потенциометра. Добавить включение/выключение подстветки дисплея по нажатию тумблера.

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования, познакомиться с работой lcd дисплея.

**Схема сборки на макетной плате:**



**Принципиальная схема:**

****

**Листинг программы:**

#**include** <LiquidCrystal.h>

#**define** POT\_MIN 0

#**define** POT\_MAX 1023

#**define** COLUMNS 16

#**define** ROWS 2

// LCD: RS, E, D4, D5, D6, D7

LiquidCrystal **lcd**(1, 0, 4, 5, 6, 7);

String text = "The quick brown fox jumps over the lazy dog. I like to play with my friends. The quick brown fox jumps over the lazy dog. I like to play with";

int maxIndex = ceil((float) text.length() / COLUMNS);

void **setup**() {

lcd.begin(COLUMNS, ROWS);

}

void **loop**() {

int potValue = analogRead(A0);

int displayedLen = floor((float) text.length() / COLUMNS) - 1;

// Преобразуем значение потенциометра в диапазон индексов для текста

int startIndex = map(potValue, POT\_MIN, POT\_MAX, 0, displayedLen);

int firstIndex = startIndex;

String firstString = text.substring(firstIndex \* COLUMNS, firstIndex \* COLUMNS + COLUMNS);

int secondIndex = startIndex + 1;

String secondString = text.substring(secondIndex \* COLUMNS,

secondIndex \* COLUMNS + COLUMNS);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(firstString);

lcd.setCursor(0, 1);

lcd.print(secondString);

delay(100);

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/3V6HZo1zjW0-prakticheskaya-2-vyvod-teksta/editel?returnTo=%2Fthings%2F3V6HZo1zjW0-prakticheskaya-2-vyvod-teksta&sharecode=r-XE0iE3d3DXlZh7uAORv3q74nQ0CAnUpLMhjYwovhw

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования, собрал схему с lcd дисплеем.

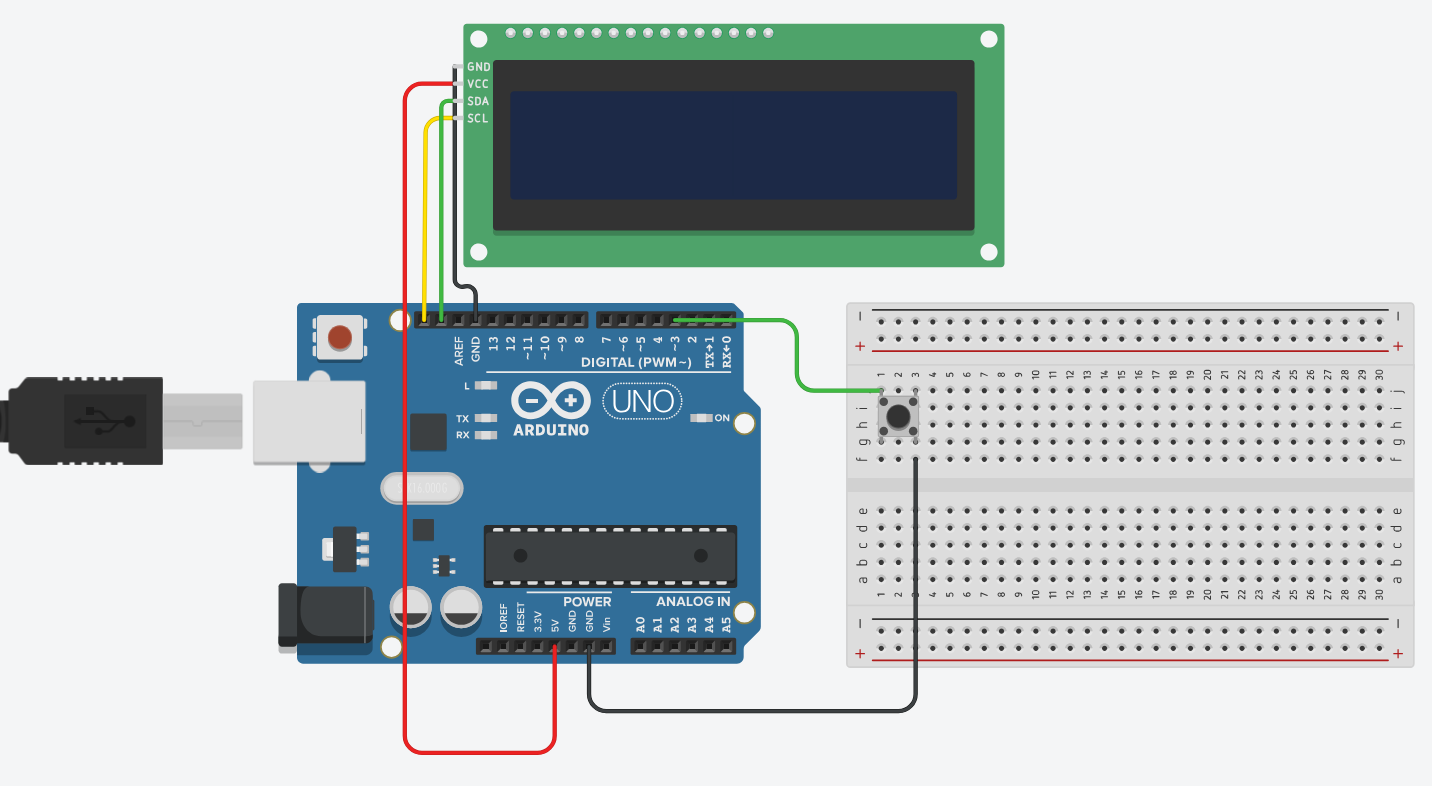
Задание № 6

**«Тестер батареек »**

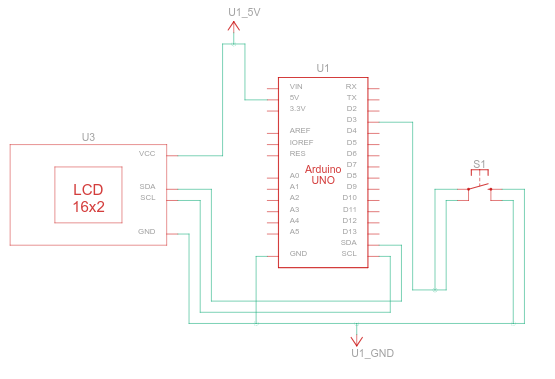
**Задание:** Текст задан строкой 20–30 символов. Организовать бегущую строку без использования циклов и встроенной функции scroll. Движение справа-налево. При нажатии кнопки должен менятся текст.

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования, познакомиться с работой I2C дисплея.

**Схема сборки на макетной плате:**



**Принципиальная схема:**

****

**Листинг программы:**

#**include** <LiquidCrystal\_I2C.h>

#**define** COLUMNS 16

#**define** ROWS 2

#**define** BUTTON\_PIN 3

LiquidCrystal\_I2C **lcd**(32, COLUMNS, ROWS);

String text = "";

String originalText = "The quick brown fox jumps over the lazy dog. ";

String altText = "Hasta la vista, baby ";

boolean isOriginalText = true;

int currentIndex = 0;

void **setup**() {

text = originalText;

lcd.init();

lcd.backlight();

pinMode(BUTTON\_PIN, INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(BUTTON\_PIN), toggleText, FALLING);

}

void **toggleText**()

{

**if** (isOriginalText) {

text = altText;

isOriginalText = false;

} **else** {

text = originalText;

isOriginalText = true;

}

currentIndex = 0;

}

String **getSubstring**()

{

int substringCapacity = text.length() - currentIndex;

**if** (substringCapacity < COLUMNS) {

String substringBegin = text.substring(currentIndex);

String result = substringBegin + text.substring(0, COLUMNS - substringCapacity);

currentIndex++;

**if** (currentIndex > text.length()) {

currentIndex = 0;

}

**return** result;

}

String result = text.substring(currentIndex++, COLUMNS + currentIndex);

**if** (currentIndex > text.length()) {

currentIndex = 0;

}

**return** result;

}

void **printMillis**()

{

lcd.setCursor(0, 1);

lcd.print(millis());

}

void **printText**()

{

lcd.setCursor(0, 0);

lcd.print(getSubstring());

}

void **loop**()

{

printText();

printMillis();

delay(100);

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/8YVtMZhnNWG-prakticheskaya-2-begushij-tekst/editel?returnTo=%2Fthings%2F8YVtMZhnNWG-prakticheskaya-2-begushij-tekst&sharecode=o\_vUKBHHBnJtMXYHEPFdtOu\_zlrHmC0NtUb7vTOYb4I

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования, собрал схему с I2C дисплеем.

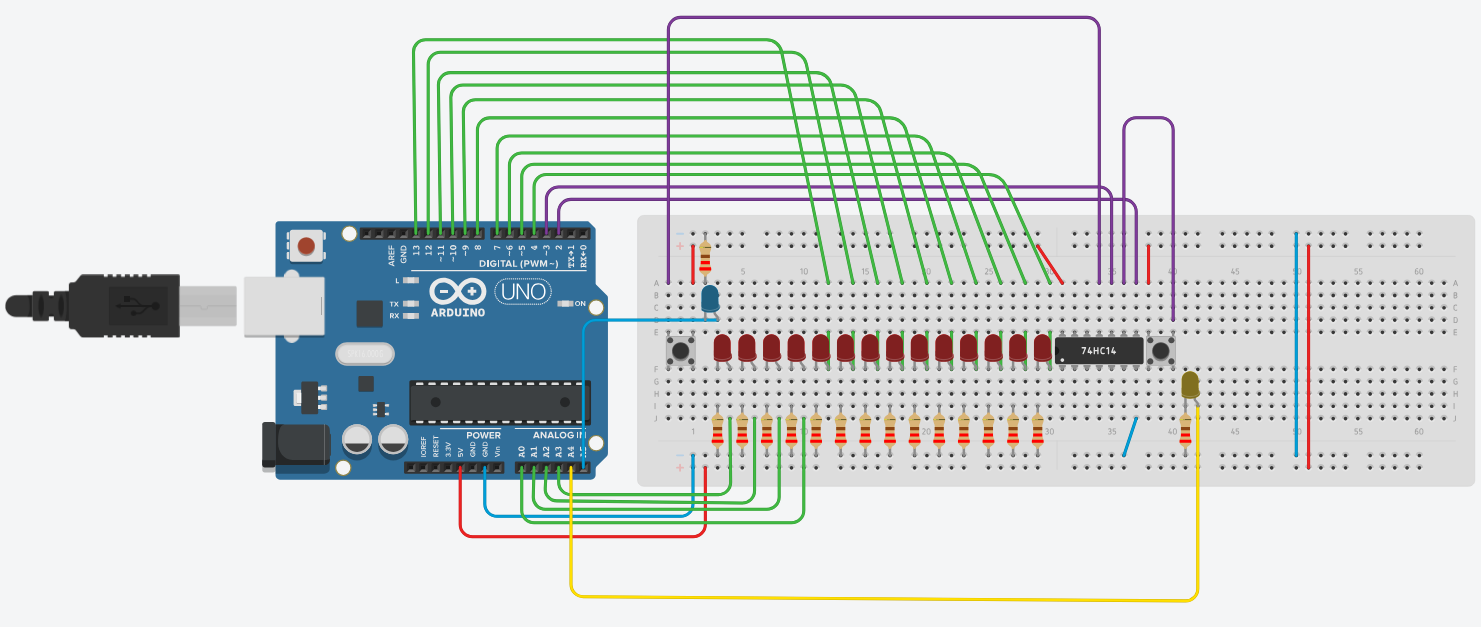
Задание № 7

**«Перетягивание каната»**

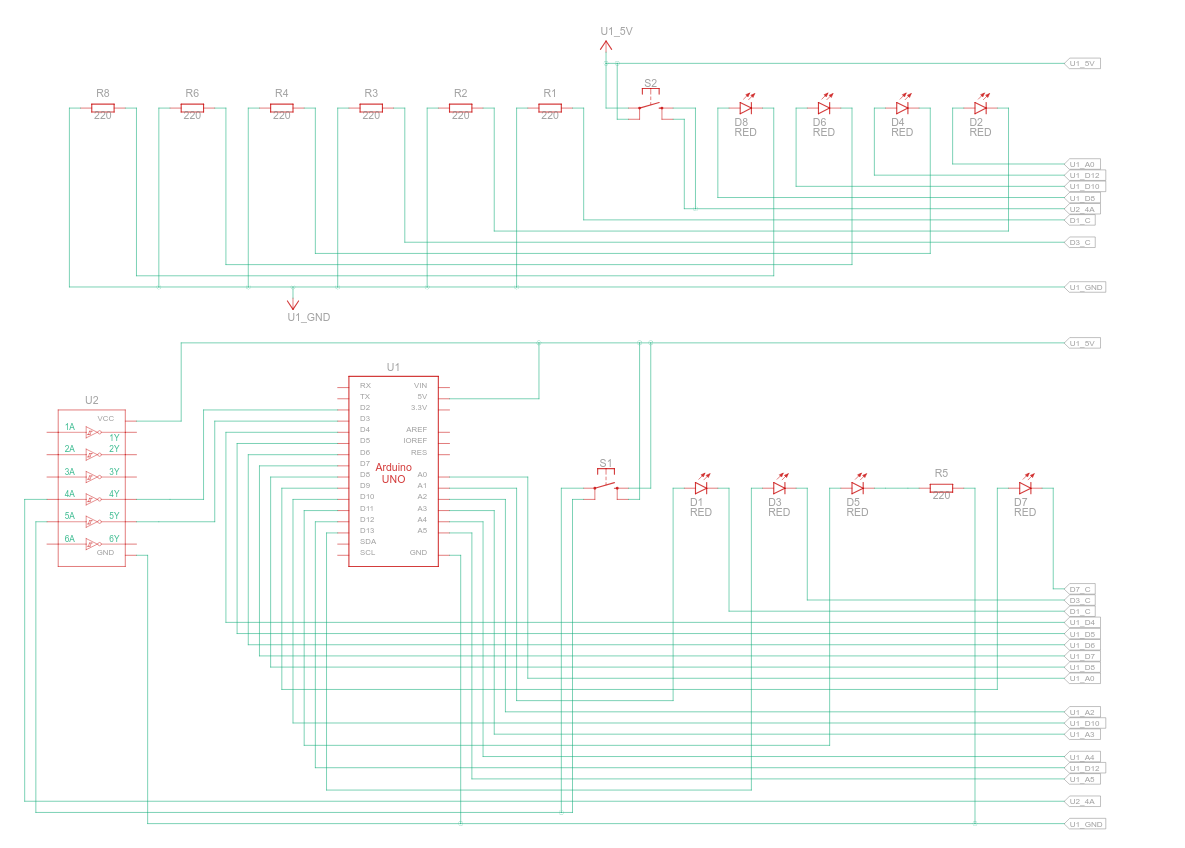
**Задание:** Добавить по 6 светодиодов каждому игроку. При двойном нажатии на кнопку светодиоды должны заговаться в сторону игрока. При достижении последнего светодиода, пользователь объявляется победителем.

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования. Изучение схемы на инвертирующем триггере Штитта.

**Схема сборки на макетной плате:**



**Принципиальная схема:**

****

**Листинг программы:**

#**define** FIRST\_BAR\_PIN 4

#**define** BAR\_COUNT 14

#**define** MAX\_SCORE 14

#**define** P1\_WIN\_LED A4

#**define** P2\_WIN\_LED A5

**volatile** int score = 0;

int additionalLeds[] = {A0, A1, A2, A3};

int **getLedPin**(int index)

{

**if** (index > 9) {

**return** additionalLeds[index - 9 - 1];

}

**return** FIRST\_BAR\_PIN + index;

}

void **setup**()

{

**for** (int i = 0; i < BAR\_COUNT; ++i) {

pinMode(getLedPin(i), OUTPUT);

}

pinMode(P1\_WIN\_LED, OUTPUT);

pinMode(P2\_WIN\_LED, OUTPUT);

attachInterrupt(INT1, pushP1, FALLING);

attachInterrupt(INT0, pushP2, FALLING);

}

void **pushP1**()

{

++score;

}

void **pushP2**()

{

--score;

}

void **loop**() {

**while** (abs(score) < MAX\_SCORE) {

int bound = map(score, -MAX\_SCORE, MAX\_SCORE, 0, BAR\_COUNT);

int left = min(bound, BAR\_COUNT / 2 - 1);

int right = max(bound, BAR\_COUNT / 2);

**for** (int i = 0; i < BAR\_COUNT; ++i) {

digitalWrite(getLedPin(i), i >= left && i <= right);

}

}

score < 0 ? digitalWrite(P1\_WIN\_LED, HIGH)

: digitalWrite(P2\_WIN\_LED, HIGH);

**while** (true) {}

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/jnPCzcVbN7a-prakticheskaya-2-peretyagivanie-kanata-svetodiod-na-vyigrysh/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits%3Fpage%3D2&sharecode=0WFOBuw7F62W6fFKaIHia8nGKixNdU15MfHqIRpTPVQ

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования, собрал схему на инвертирующем триггере Штитта.

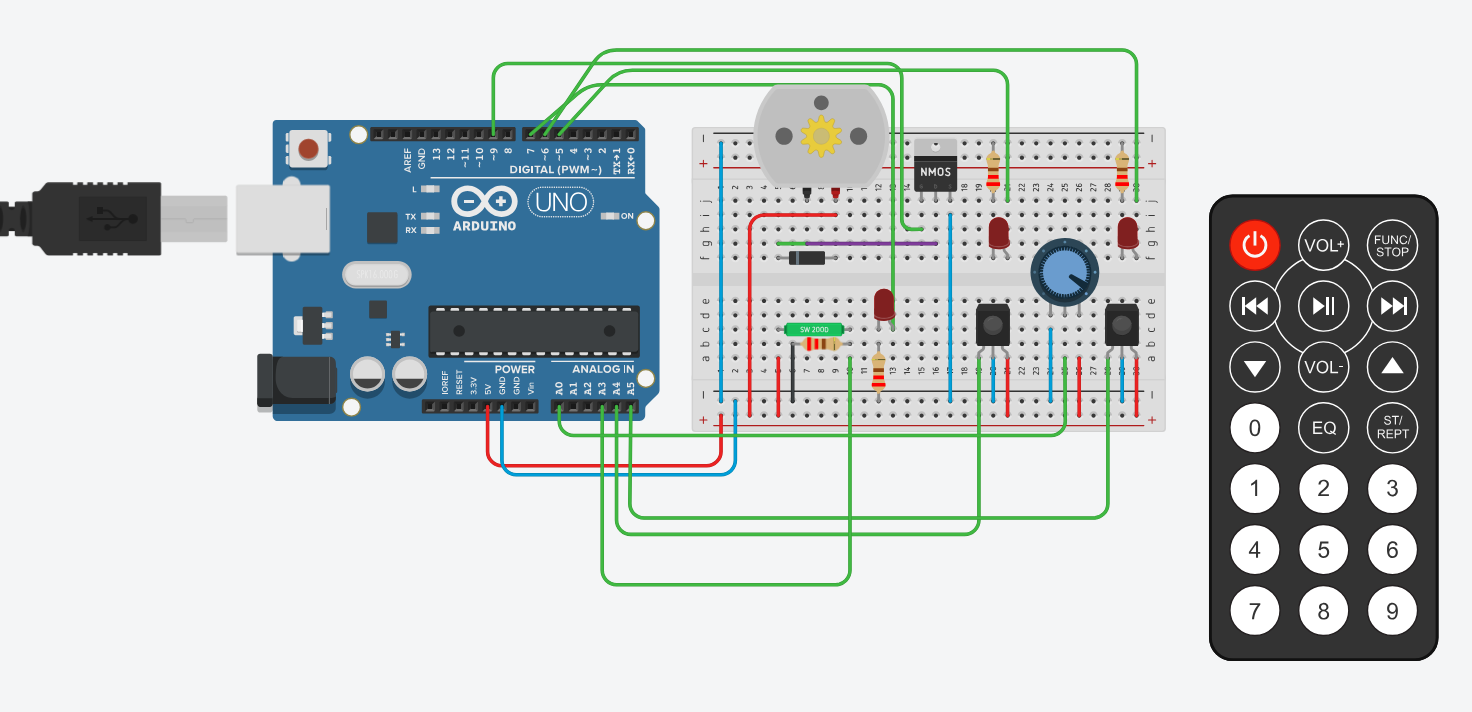
Задание № 8

**«Задание с мотором»**

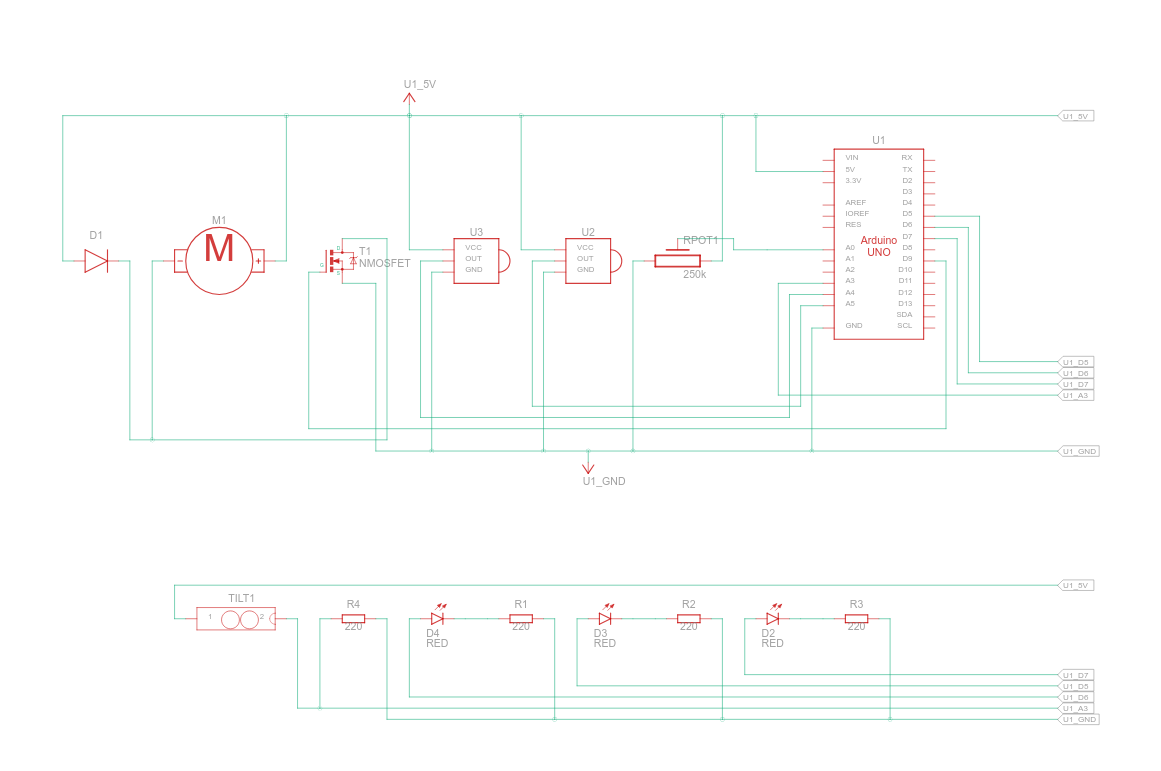
**Задание:** Добавить в схему датчик уровня и два ИК датчика. При срабатывании датчика уровня, мотор включается на половину мощности, при срабатывании датчиков ИК мотор включается на полную мощность

**Цель работы:** Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования. Изучение мотора, датчика уровня и ИК датчика.

**Схема сборки на макетной плате:**



**Принципиальная схема:**

****

**Листинг программы:**

#**define** MOTOR\_PIN 9

#**define** LEVEL\_PIN A3

#**define** LINE1\_PIN A4

#**define** LINE2\_PIN A5

#**define** LINE1\_LED\_PIN 5

#**define** LINE2\_LED\_PIN 6

#**define** LEVEL\_LED\_PIN 7

int pot = A0;

void **setup**() {

pinMode(MOTOR\_PIN, OUTPUT);

pinMode(pot, INPUT);

pinMode(LINE1\_PIN, INPUT);

pinMode(LINE2\_PIN, INPUT);

pinMode(LEVEL\_LED\_PIN, OUTPUT);

pinMode(LINE1\_LED\_PIN, OUTPUT);

pinMode(LINE2\_LED\_PIN, OUTPUT);

Serial.begin(9600);

}

void **toggleLed**(int pin, int state)

{

state == LOW ? digitalWrite(pin, HIGH) : digitalWrite(pin, LOW);

}

void **loop**()

{

int level = digitalRead(LEVEL\_PIN);

int line1 = digitalRead(LINE1\_PIN);

int line2 = digitalRead(LINE2\_PIN);

toggleLed(LEVEL\_LED\_PIN, !level);

toggleLed(LINE1\_LED\_PIN, line1);

toggleLed(LINE2\_LED\_PIN, line2);

**if** (level == HIGH && (line1 == LOW || line2 == LOW)) {

analogWrite(MOTOR\_PIN, 255);

} **else** **if** (level == HIGH) {

analogWrite(MOTOR\_PIN, 128);

} **else** {

analogWrite(MOTOR\_PIN, 0);

}

delay(100);

}

**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/98PHytirOAw-prakticheskaya-2-upravlenie-skorostyu-motora-s-datchikami/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits%3Fpage%3D2&sharecode=9q8o8pr5LZiV0iQgdPHg1RJmMFgUFx1FCRYirr4WYF4

**Вывод**

В ходе практической работы ознакомился с элементами Arduino IDE, изучил основы работы со средой для программирования, собрал схему с мотором, датчиком уровня и ИК датчиками.

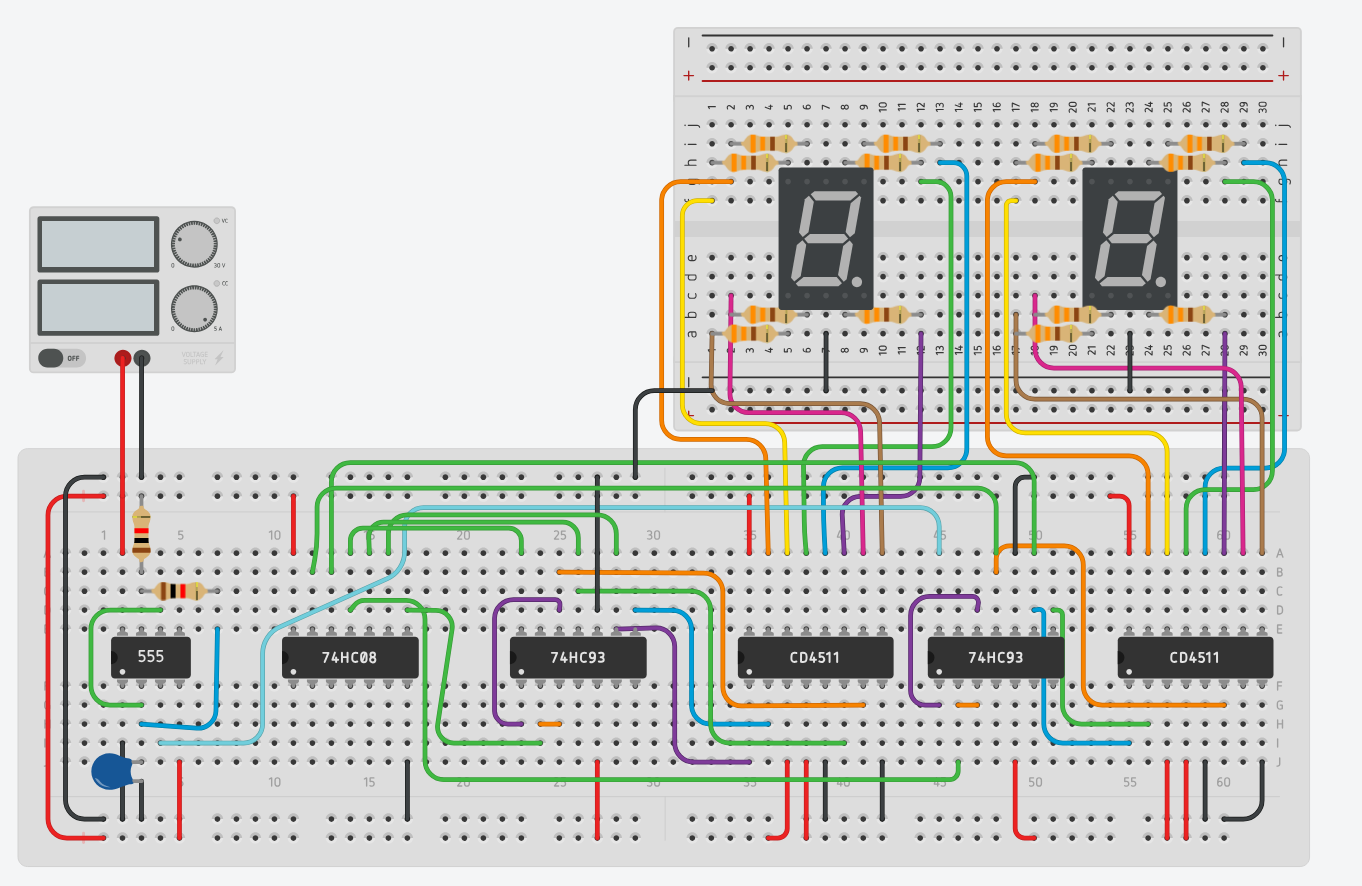
Задание № 10

**«Секундомер на два семисегментных индикатора»**

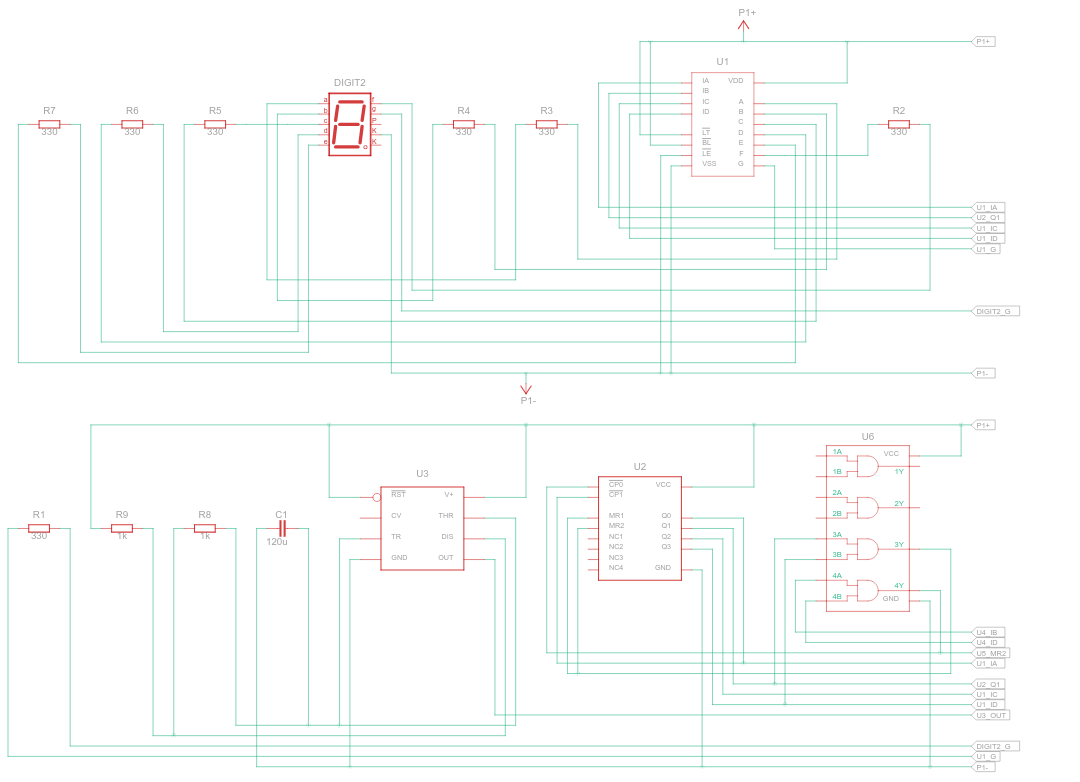
**Задание:** Без Arduino, без программирования. Только с использованием изученных микросхем и микросхем стандартной логики собрать секундомер, который выводит четные числа и производит сброс на числе 98

**Цель работы:** Собрать схему и использованием изученных микросхем и микросхем стандартной логики, без использования arduino.

**Схема сборки на макетной плате:**



**Принципиальная схема:**



**Ссылка на рабочий проект:**

https://www.tinkercad.com/things/lHNUIzzYcTm-10-rabota/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits%3Fpage%3D2&sharecode=v7GqKO33nyBJ8yBSPZ1S2\_VDy6z37Hd4oDVjKm3qqXA

**Вывод**

В ходе практической работы изучил сборку схему и использованием изученных микросхем и микросхем стандартной логики, без использования arduino.