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**

**«Микросхемы, дисплей и мониторы»**

по дисциплине

«Информационные технологии»  
Чётный вариант

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

Киров 2023

Цель работы: ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования, а также сборка схем с пьезодинамиком, микросхемами, дисплеем и моторами.

Задания

1. Кнопочные ковбои.

Формулировка задания: Добавить “полосу бонусов”. Для каждого игрока в начальном состоянии есть 3 светодиода, которые не горят. Нажатием на кнопку игрок получает “бонус”, если нажал быстрее соперника. Игра до 3х очков.

Схема сборки на макетной плате представлена на рисунке 1. Принципиальная схема представлена на рисунке 2.

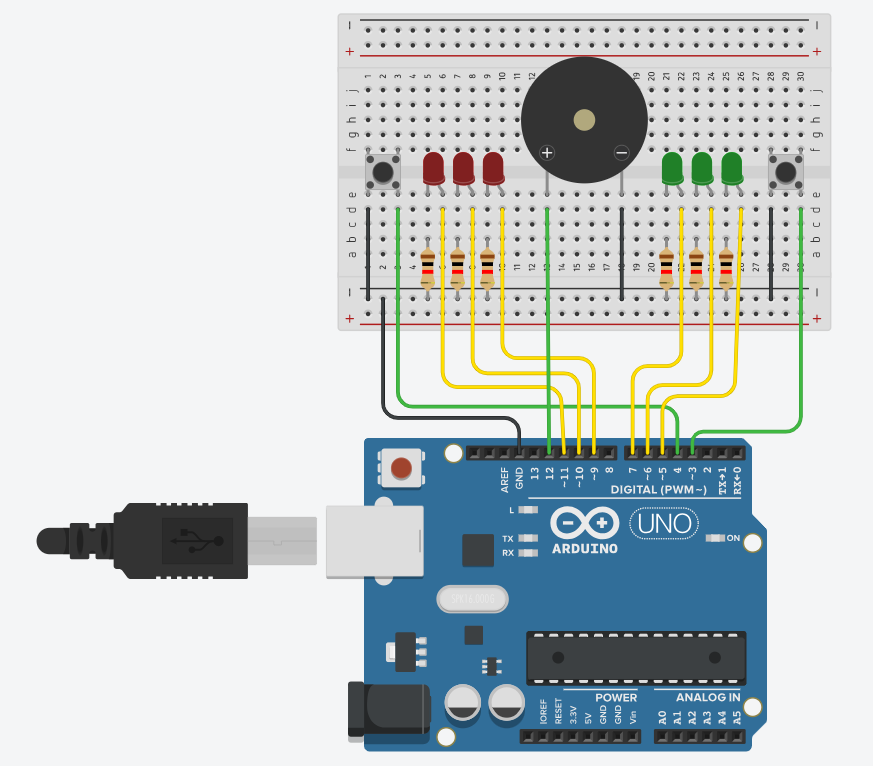


Рисунок 1 – Кнопочные ковбои

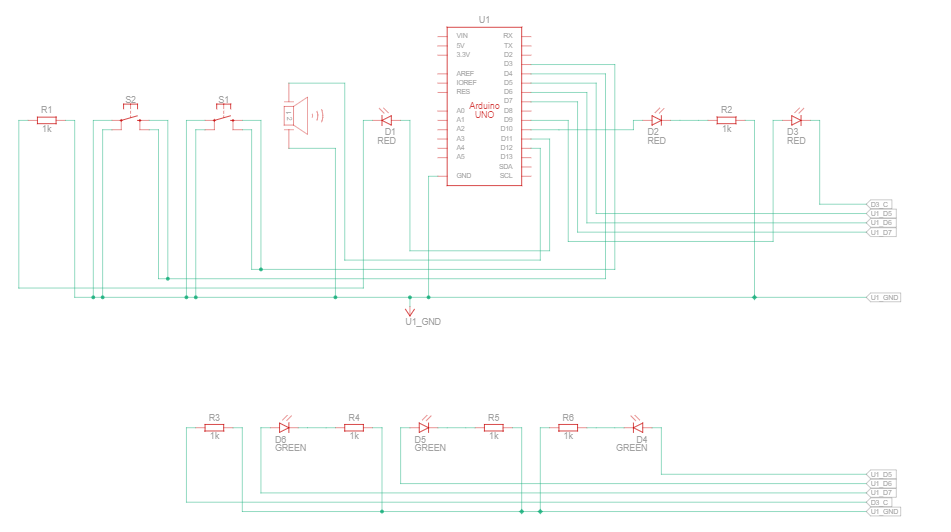


Рисунок 2 – Принципиальная схема «Кнопочные ковбои»

Код, необходимый для работы данной схемы, представлен в листинге 1.

Листинг 1

#define BUZZER\_PIN 12 // пин с пищалкой

#define PLAYER\_COUNT 2 // количество игроков-ковбоев

int right = 0;

int k = 0;

bool t=true;

int left = 0;

int buttonPins[PLAYER\_COUNT] = {3, 4}; int

ledPins[PLAYER\_COUNT] = {5, 9};

void setup()

{

pinMode(BUZZER\_PIN, OUTPUT);

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

pinMode(ledPins[player], OUTPUT); pinMode(buttonPins[player],

INPUT\_PULLUP);

}

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(10, OUTPUT);

pinMode(11, OUTPUT);

}

void loop()

{

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 на 1 сек

if (ledPins[player]==5) {

digitalWrite(ledPins[player]+right,HIGH);

right++;

}

else {

digitalWrite(ledPins[player]+left,HIGH);

left++;

}

tone(BUZZER\_PIN, 4000, 1000); delay(1000);

if (left == 3 | right == 3) {

tone(BUZZER\_PIN, 6000, 2000);

while(true);

}

break;

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

}

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/eiDKkp2gRfi-terrific-snaget/editel?sharecode=6RvwbYArUkKa4PYwlWrkbBBU2kiaexZUBMWTVcDir9s>

1. Мерзкое пианино

Формулировка задания: Подключите кнопки по схеме со стягивающим резистором и измените программу так, чтобы она продолжала работать. Опрос кнопок сделать через switch/case. В каждом case не более одной строки. Реализация должна быть без использования цикла в функции loop().

Схема сборки на макетной плате представлена на рисунке 3. Принципиальная схема представлена на рисунке 4.

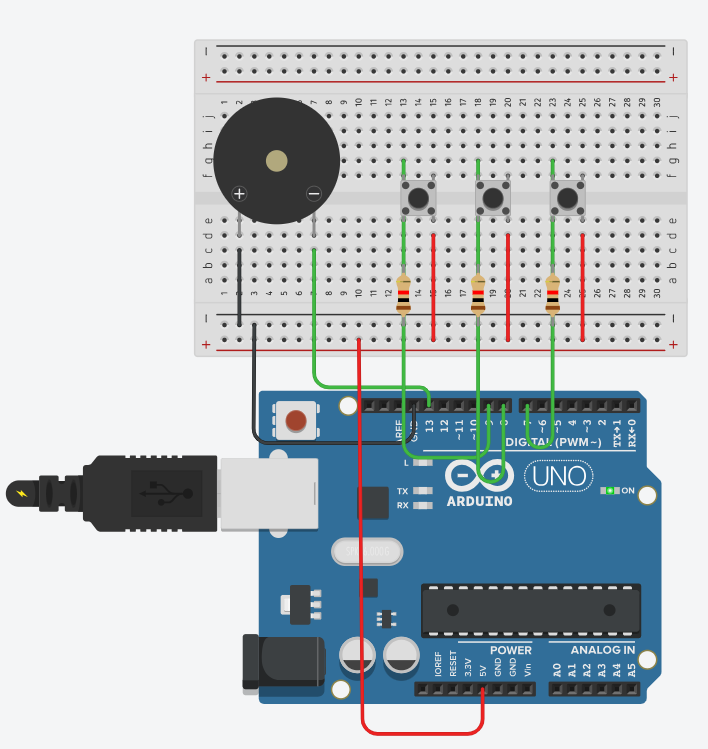


Рисунок 3 – Мерзкое пианино

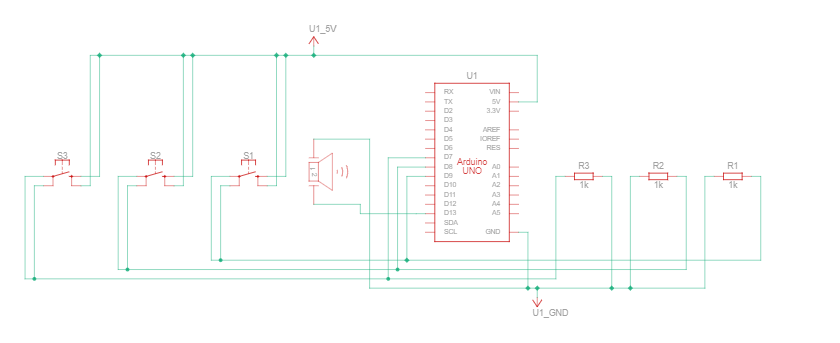


Рисунок 4 – Принципиальная схема «Мерзкое пианино»

Код, необходимый для работы данной схемы, представлен в листинге 2.

Листинг 2

#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

int frequency = 0;

void setup(){

pinMode(BUZZER\_PIN, OUTPUT);

}

void loop(){

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

int keyPin = i + FIRST\_KEY\_PIN;

boolean keyUp = digitalRead(keyPin);

if (keyUp) {

switch (keyPin) {

case 7: {

frequency = 1000;

break;

}

case 8: {

frequency = 2000;

break;

}

case 9: {

frequency = 3000;

break;

}

}

tone (BUZZER\_PIN, frequency,1);

}

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/cpZ0TdUr0GM-neat-snaget/editel?sharecode=NfpDpLhArEKSpsnyxjB_iNzNMk71ORa07p0SbY0CpJk>

1. Секундомер

Формулировка задания: Добавить в схему второй семисегментный индикатор. Сделать счетчик от 99 до 0. Без микросхем.

Схема сборки на макетной плате представлена на рисунке 5. Принципиальная схема представлена на рисунке 6.

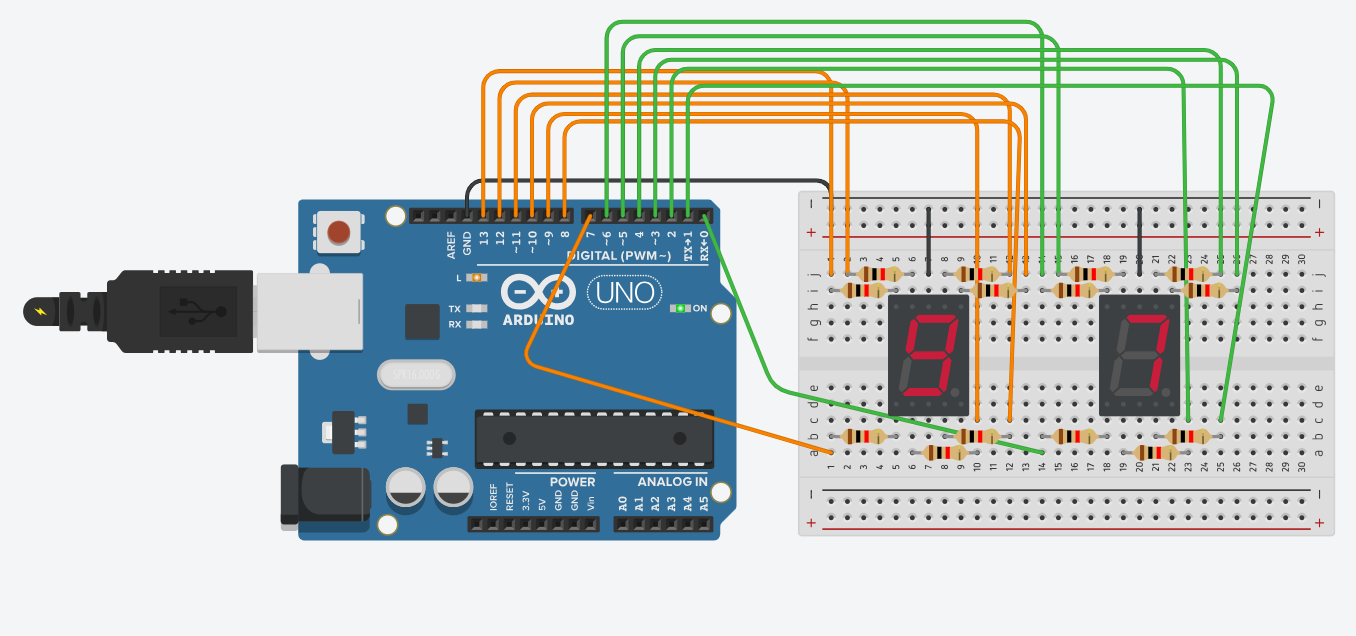


Рисунок 5 – Секундомер

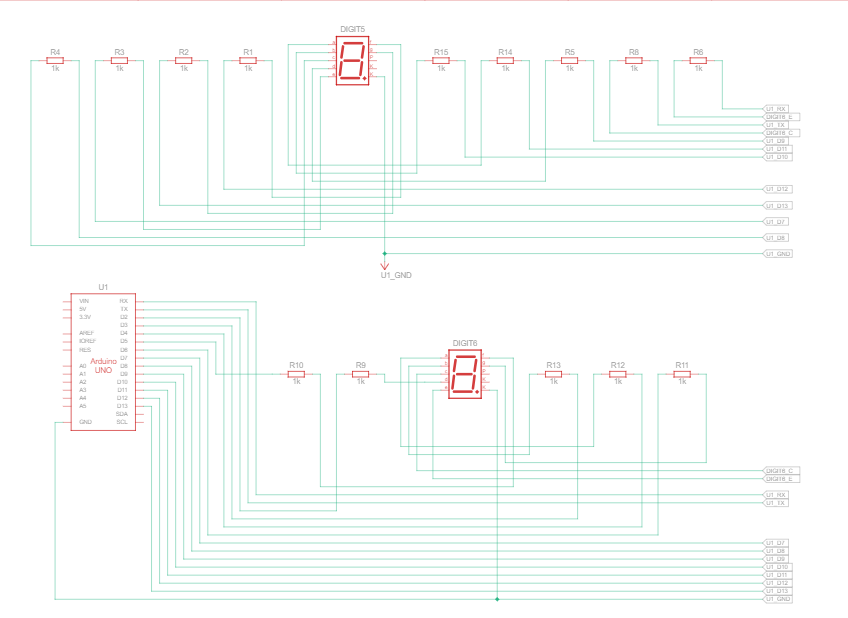


Рисунок 6 – Принципиальная схема «Секундомер»

Код, необходимый для работы данной схемы, представлен в листинге 3.

Листинг 3

#define FIRST\_SEGMENT\_PIN 0

#define SEGMENT\_COUNT 6

#define FIRST\_SEGMENT\_PIN1 7

#define SEGMENT\_COUNT1 13

int number = 99;

byte numberSegments[10] = {

0b00111111, 0b00001010, 0b01011101, 0b01011110, 0b01101010,

0b01110110, 0b01110111, 0b00011010, 0b01111111, 0b01111110,

};

void setup()

{

for (int i = 0; i < 14; ++i)

pinMode(i + FIRST\_SEGMENT\_PIN, OUTPUT);

}

void loop()

{

number--;

delay(1000);

int mask2 = numberSegments[number/10];

int mask1 = numberSegments[number%10];

for (int i = 0; i < 7; ++i) {

boolean enableSegment1 = bitRead(mask1, i);

boolean enableSegment2 = bitRead(mask2, i);

digitalWrite(i + FIRST\_SEGMENT\_PIN, enableSegment1);

digitalWrite(i + FIRST\_SEGMENT\_PIN1, enableSegment2);

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/0NpKkOjoHuF-neat-gogo/editel?sharecode=vAG2ZJ7LGgAaDkdYZ5e5x7Sg_tp2WxCzeDbQweVBBV0>

1. Счётчик нажатий

Формулировка задания: Добавить в схему второй семисегментный индикатор. Выводить по нажатию простые числа в пределах 100. Числа получать условием или функцией, а не задавать заранее.

Схема сборки на макетной плате представлена на рисунке 7. Принципиальная схема представлена на рисунке 8.

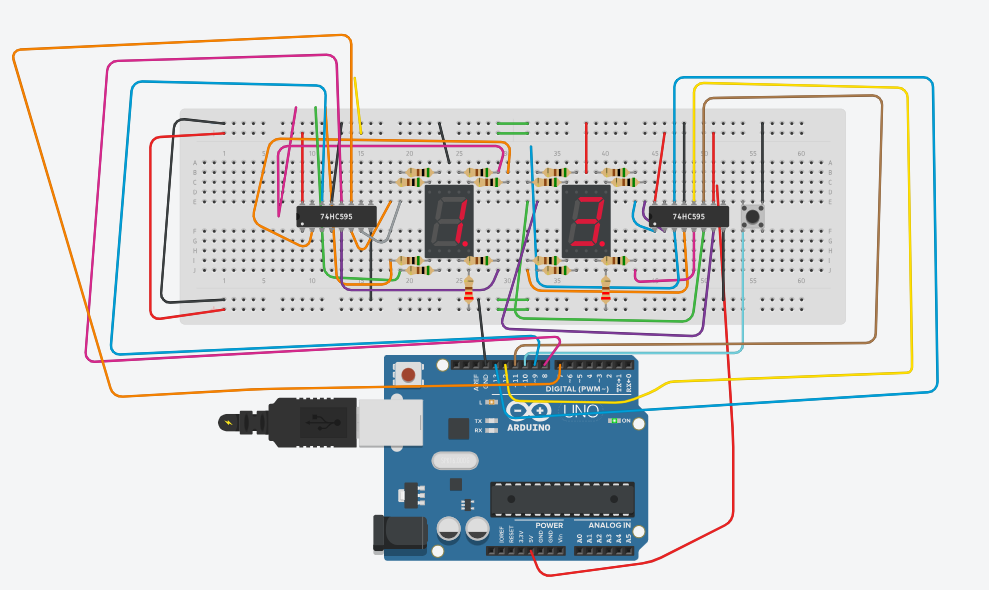


Рисунок 7 – Счётчик нажатий

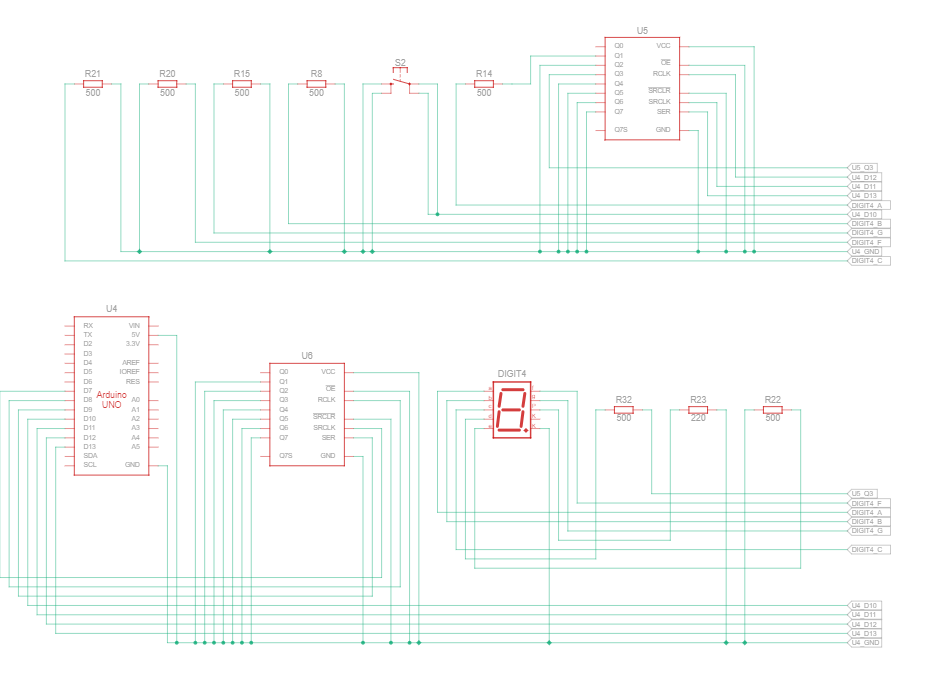


Рисунок 8 – Принципиальная схема «Счётчик нажатий»

Код, необходимый для работы данной схемы, представлен в листинге 4.

Листинг 4

#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 DATA\_PIN2 9 // пин данных (англ. data)

#define LATCH\_PIN2 8 // пин строба (англ. latch)

#define CLOCK\_PIN2 7 // пин такта (англ. clock)

#define BUTTON\_PIN 10

int clicks = 0;

int clicks2 = 0;

int i = 2;

int chislo = 0;

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(DATA\_PIN2, OUTPUT);

pinMode(CLOCK\_PIN2, OUTPUT);

pinMode(LATCH\_PIN2, OUTPUT);

pinMode(BUTTON\_PIN, INPUT\_PULLUP);

}

void loop()

{

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

delay(10);

if (!digitalRead(BUTTON\_PIN)) {

if (i>97) {

i = 2;

}

if (isSimple(i)) {

chislo = i;

i++;

}

else while(isSimple(i) == false) {

i++;

}

if (isSimple(i)&& i!=3) {

chislo = i;

i++;

}

}

}

buttonWasUp = digitalRead(BUTTON\_PIN);

digitalWrite(LATCH\_PIN, LOW);

digitalWrite(LATCH\_PIN2, LOW);

shiftOut(DATA\_PIN, CLOCK\_PIN, LSBFIRST, segments[chislo%10]);

shiftOut(DATA\_PIN2, CLOCK\_PIN2, LSBFIRST, segments[chislo/10]);

digitalWrite(LATCH\_PIN, HIGH);

digitalWrite(LATCH\_PIN2, HIGH);

}

static bool isSimple(int N)

{

//чтоб убедится простое число или нет достаточно проверить не делитсья ли

//число на числа до его половины

for (int i = 2; i <= (N / 2); i++) {

if (N % i == 0)

return false;

}

return true;

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/3MYk82FlCYl-amazing-fyyran/editel?sharecode=el_pq_VZsBAn3Op_ZiASHSWec9DP8xJ9QA0h-TFs1Sc>

1. Тестер батареек

Формулировка задания: Добавить потенциометр в схему для прокрутки текста, не менее 5 строк.

Схема сборки на макетной плате представлена на рисунке 9. Принципиальная схема представлена на рисунке 10.

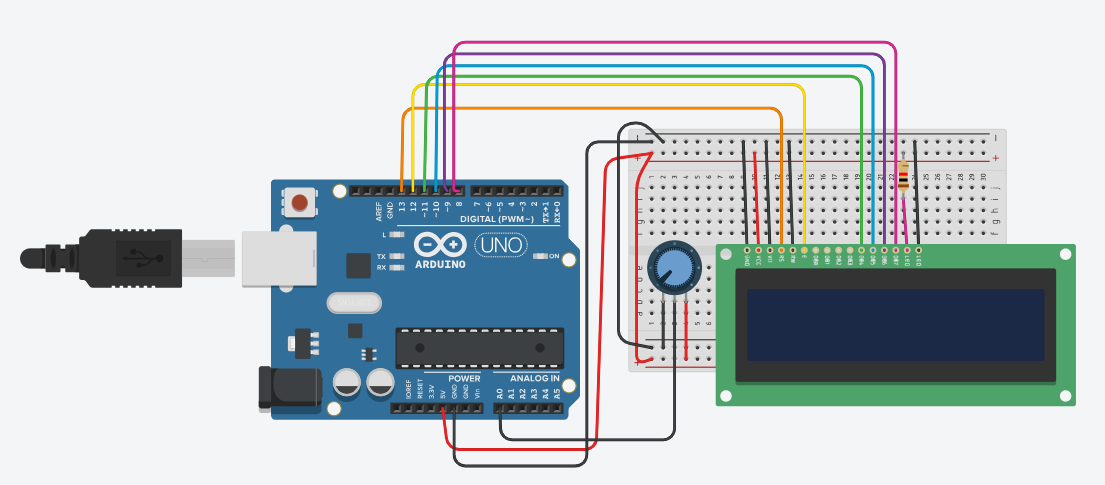


Рисунок 9 – Тестер батареек

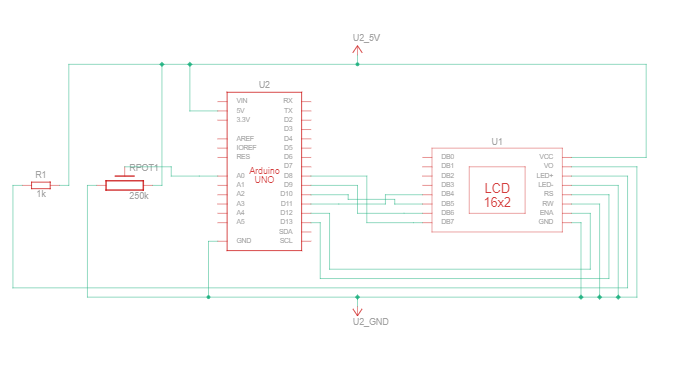


Рисунок 10 – Принципиальная схема «Тестер батареек»

Код, необходимый для работы данной схемы, представлен в листинге 5.

Листинг 5

#include <LiquidCrystal.h>

#define DIODE\_DROP 0.7

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

String s= "Hello, today i will show my arduino project for get good mark. I hope that my work will be passed. I tried make good work. good lick see you next time.";

void setup() {

lcd.begin(16, 2);

}

void loop()

{

int anRead = analogRead(0);

int countSkip = anRead / 48;

int cursor = countSkip \* 32;

for (int i = 0; i < 16;i++) {

lcd.setCursor(i, 0);

char val = s[cursor];

lcd.print(val);

cursor++;

}

for (int i = 0; i < 16;i++) {

lcd.setCursor(i, 1);

char val = s[cursor];

lcd.print(val);

cursor++;

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/2FnoCIQFoIZ-tester-batareek/editel?sharecode=e7CrTK0sQpxxMDDsR8sNQwcaMP85TxOmaC_-LeHjIHw>

1. Тестер батареек 2

Формулировка задания: Изменить схему и программу таким образом, чтобы получилась бегущая строка длиной 20 символов слева направо.

Схема сборки на макетной плате представлена на рисунке 11. Принципиальная схема представлена на рисунке 12.

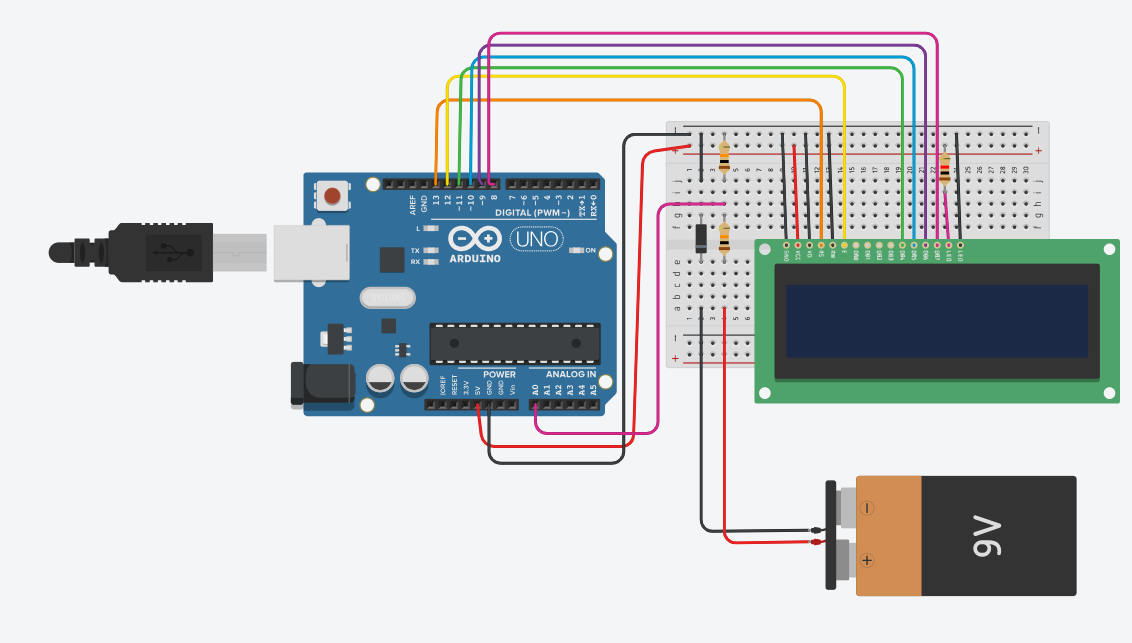


Рисунок 11 – Тестер батареек

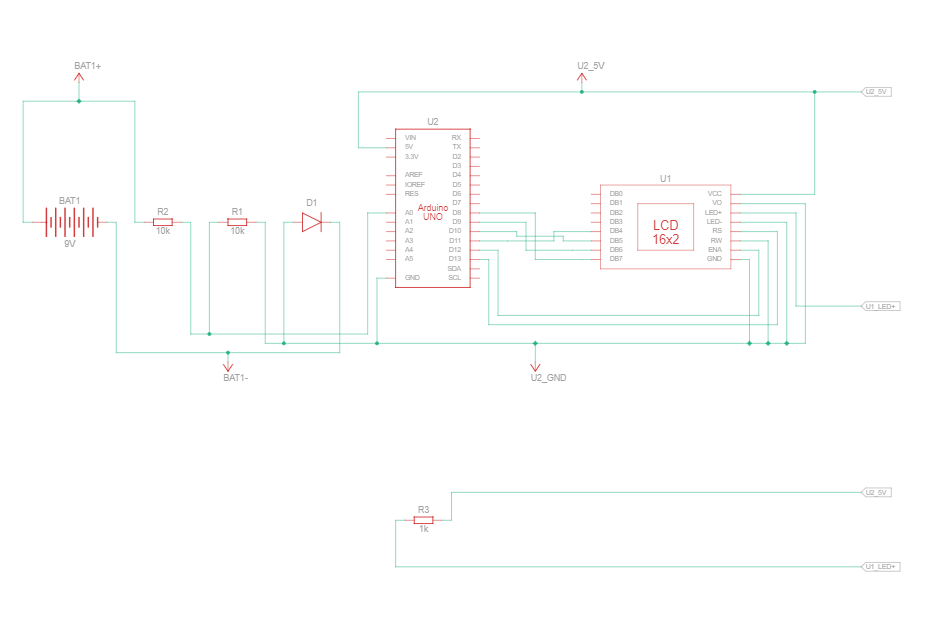


Рисунок 12 – Принципиальная схема «Тестер батареек»

Код, необходимый для работы данной схемы, представлен в листинге 6.

Листинг 6

String s = "hello, it's text for output to display";

#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

int i = 0;

int i1 = 0;

void setup()

{

lcd.begin(12, 1);

}

void loop()

{

int strLen = s.length();

if(i<=16){

lcd.setCursor(i, 0);

lcd.print(s);

delay(300);

lcd.clear();

i++;

}

else{

for (int i = strLen; i >=0; i--) {

for (int i1 = 2 +i; i1 <= strLen - 1; i1++) {

lcd.print(s[i1]);

}

delay(300);

lcd.clear();

}

i=0;

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/jgRgJjSlAV0-tester-batareek-2/editel?sharecode=UhSSB4OqYLFocWQfY5iBT2NZ8cQ-tmH54Fk0g0s9A-I>

1. Перетягивание каната

Формулировка задания: Добавить в схему дисплей. Изменить программу таким образом, чтобы управлять “канатом” на дисплее.

Схема сборки на макетной плате представлена на рисунке 13. Принципиальная схема представлена на рисунке 14.

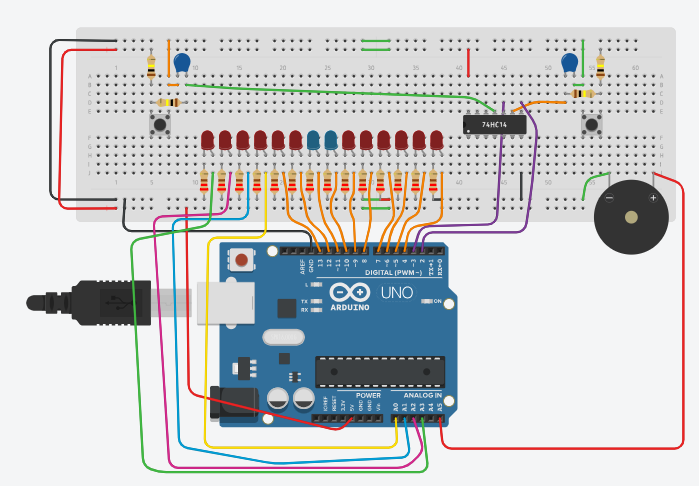


Рисунок 13 – Перетягивание каната

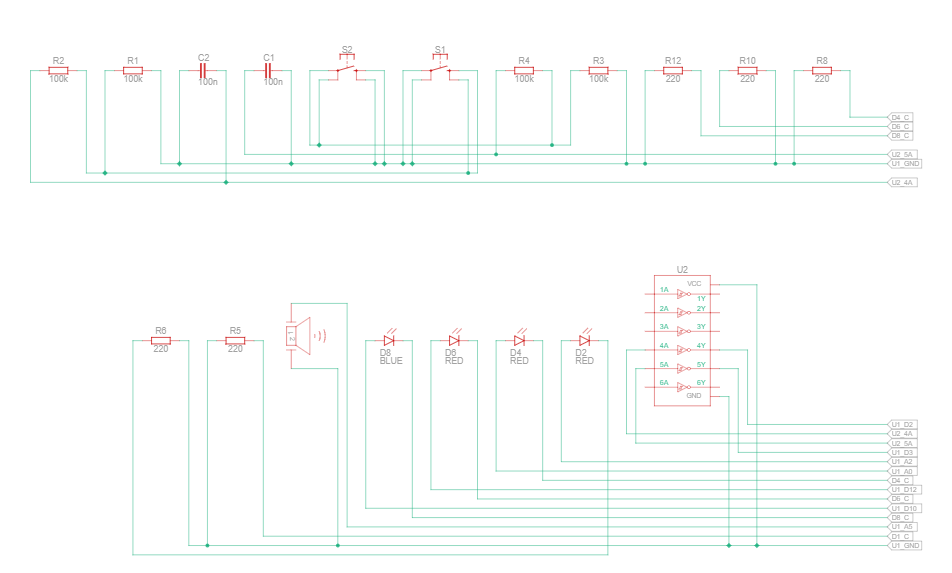


Рисунок 14 – Принципиальная схема «Перетягивание каната»

Код, необходимый для работы данной схемы, представлен в листинге 7.

Листинг 7

#define BUZZER\_PIN A5

#define FIRST\_BAR\_PIN 4

#define BAR\_COUNT 14

#define MAX\_SCORE 28

volatile int score = 0;

void setup()

{

Serial.begin(9600);

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

pinMode(i + FIRST\_BAR\_PIN, OUTPUT);

pinMode(BUZZER\_PIN, OUTPUT);

attachInterrupt(INT1, pushP1, FALLING);

attachInterrupt(INT0, pushP2, FALLING);

}

}

void pushP1() { ++score; }

void pushP2() { --score; }

void loop()

{

tone(BUZZER\_PIN, 2000, 50);

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

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

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

Serial.println(score);

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

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

digitalWrite(i + FIRST\_BAR\_PIN, i >= left && i <= right);

}

}

tone(BUZZER\_PIN, 4000, 50);

while (true) {}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/fnn7Bj5Ytur-brave-jaagub/editel?sharecode=kR6vGfOqKDircQ2VUa3lO12Xbg30e7mgJUFtIulQu9E>

1. Управление скоростью мотора

Формулировка задания: Управляйте скоростью мотора при помощи полевого p-канального МОП транзистора.

Схема сборки на макетной плате представлена на рисунке 15. Принципиальная схема представлена на рисунке 16.

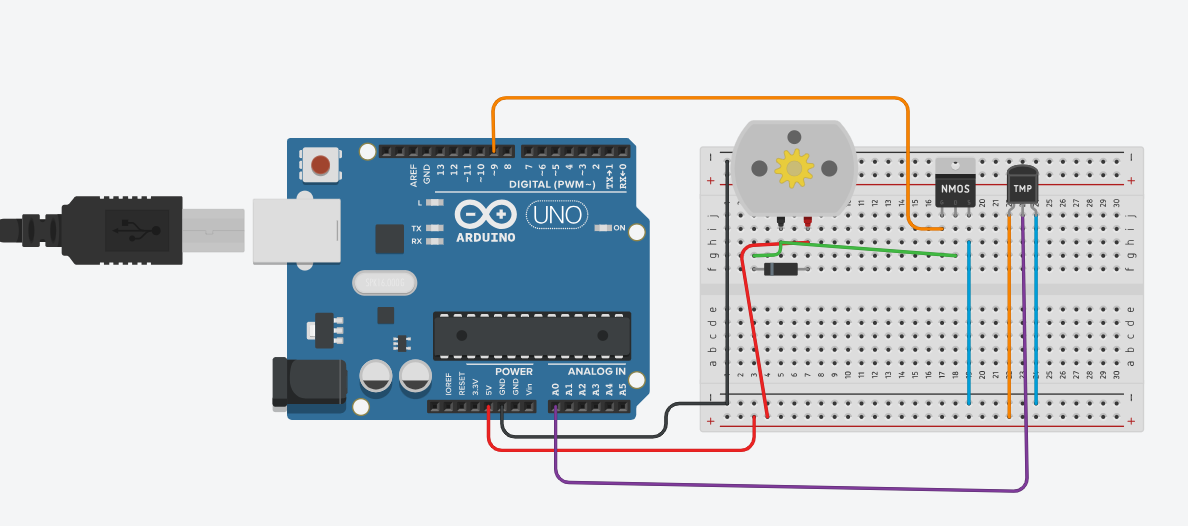


Рисунок 15 – Управление скоростью каната

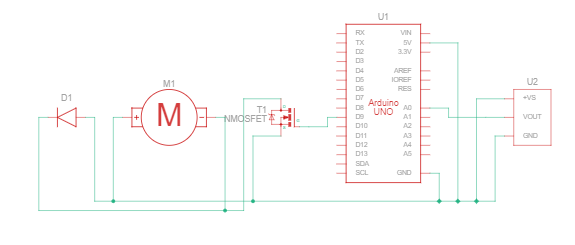


Рисунок 16 – Принципиальная схема «Управление скоростью каната»

Код, необходимый для работы данной схемы, представлен в листинге 8.

Листинг 8

#define MOTOR\_PIN 9

int tmp = A0;

void setup(){

pinMode(MOTOR\_PIN, OUTPUT);

pinMode(tmp, INPUT);

Serial.begin(9600);

}

void loop(){

int val = analogRead(tmp) / 4;

Serial.println(val);

analogWrite(MOTOR\_PIN, val);

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/3eqJzQQrjis-grand-krunk-fulffy/editel?sharecode=ruYrlg2tV3EVnbj4agjqL5ZsGkdOvJWlRgNl08uvNtY>