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»  
Нечётный 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.

Добавить “полосу здоровья”. Для каждого игрока в начальном состоянии горят 3 светодиода/жизни. Нажатием на кнопку игрок отнимает “жизнь” соперника.

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

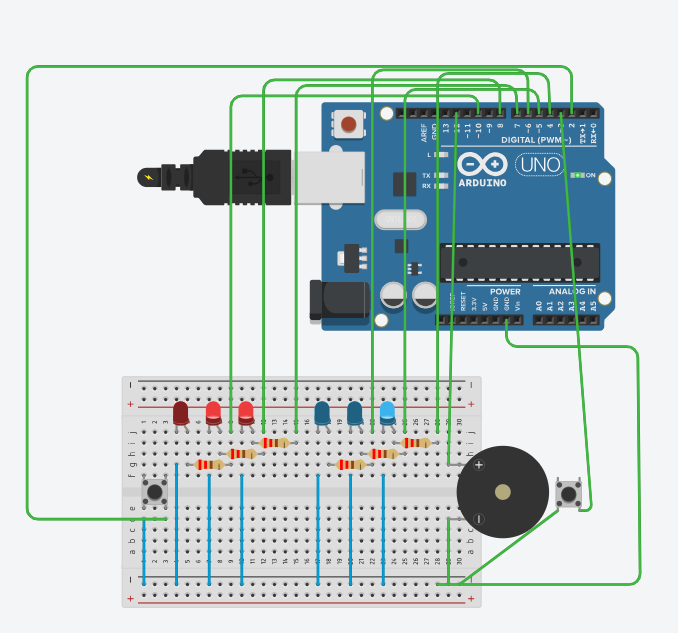


Рисунок 1 – Кнопочные ковбои

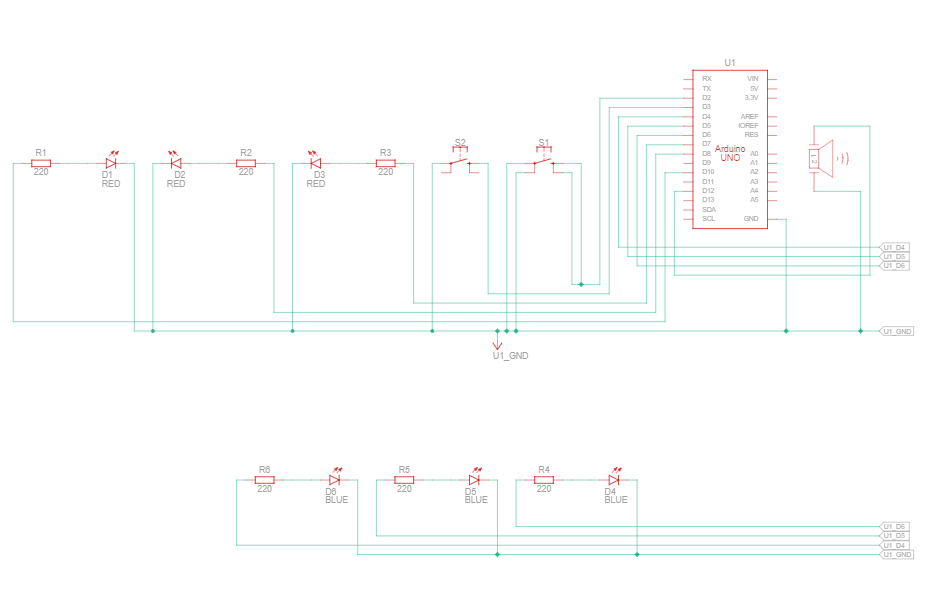


Рисунок 2 – Принципиальная схема «Кнопочные ковбои»

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

#define LED\_COUNT 3 // кол-во светодиовов для одного игрока

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

int players\_led[PLAYER\_COUNT][LED\_COUNT] = {{4,5,6},{7,8,10}};

int players\_health[PLAYER\_COUNT] = {2,2}; // Жизнь в диапазон от -1 до 2

int count\_kill = 0; // Кол-во убитых игроков

// Отбираем 1 жизнь у игрока с индексом player

void HealthMinus(int player){

digitalWrite(players\_led[player][players\_health[player]], LOW);

players\_health[player]--;

}

// Звук победы

void SoundWin(){

for(int i = 0; i <= 2; i++){

tone(BUZZER\_PIN, 4000, 200);

delay(400);

}

}

// Возвращаем все в исходное состояние

void Clear(){

count\_kill = 0;

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

for(int j =0; j < LED\_COUNT; ++j){

digitalWrite(players\_led[i][j], HIGH);

}

players\_health[i] = 2;

}

}

void setup()

{

pinMode(BUZZER\_PIN, OUTPUT);

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

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

pinMode(players\_led[player][i], OUTPUT);

digitalWrite(players\_led[player][i], HIGH);

}

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

}

}

void loop()

{

while (true){

// Даём сигнал «пли!», выждав случайное время от 2 до 7 сек

delay(random(2000, 7000));

tone(BUZZER\_PIN, 3000, 250);

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

// Запрещаем убитым игрокам участвовать в новом раунде

if(players\_health[player] != -1){

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

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

// Отнимаем жизни у всех проигравших игроков

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

if(i != player){

HealthMinus(i);

// Проверяем, умер ли проигравший игрок

if (players\_health[i] == -1){

count\_kill++;

}

}

}

tone(BUZZER\_PIN, 4000, 1000);

delay(1000);

break;

}

}

}

//Если в живых остался лишь один игрок

if(count\_kill == PLAYER\_COUNT-1){

SoundWin();

break;

}

}

Clear();

delay(2000);

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/g9w0ADTv4dS-neat-borwo-esboo/editel?sharecode=r0s1grrBRVWiMbqMeiaYEPk3Qmp6NwwvMbgV5A7ABnQ>

1. Мерзкое пианино

Формулировка задания: Добавьте еще 2 кнопки и измените программу так, чтобы можно было извлечь 5 различных нот. Значение нот задать в массиве и без использования цикла в функции loop().

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

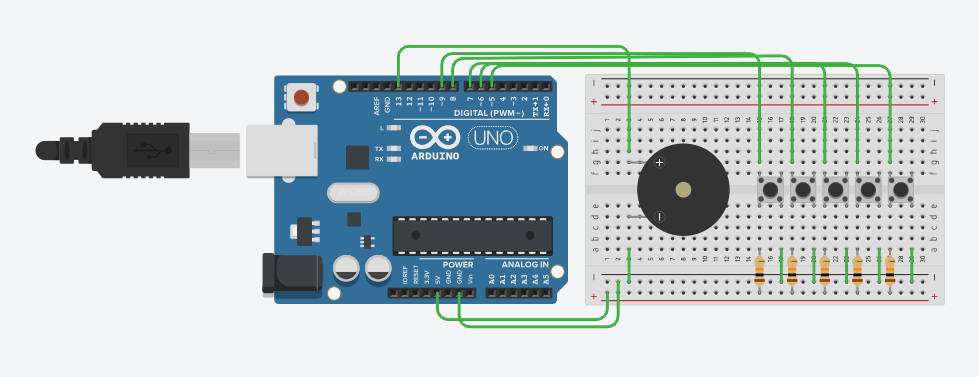


Рисунок 3 – Мерзкое пианино

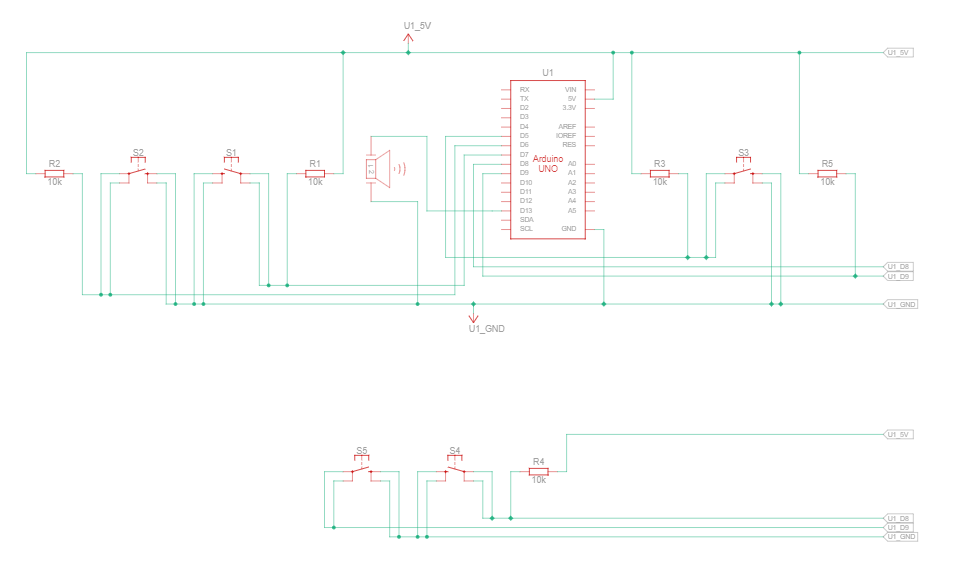


Рисунок 4 – Принципиальная схема «Мерзкое пианино»

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 5 // первый пин с клавишей (англ. «key»)

#define KEY\_COUNT 5 // общее количество клавиш

int i = 0;

int noty[] = {523,587,659,698,784};

void setup()

{

pinMode(BUZZER\_PIN, OUTPUT);

for(int i = 5; i <=9; i++) pinMode(i, INPUT\_PULLUP);

}

void loop()

{

boolean keyUp;

int frequency;

if (i == 5) {

i = 0;

return;

}

int keyPin = i + FIRST\_KEY\_PIN;

keyUp = digitalRead(keyPin);

if (!keyUp) {

frequency = noty[i];

tone(BUZZER\_PIN, frequency, 20);

return;

}

i++;

loop();

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/gc2t8RRKzgX-funky-vihelmo-snicket/editel?sharecode=fIuJZypmnO7HMhUCNAH27Jy5AVILgPLImNi231JdB7c>

1. Секундомер

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

Схема сборки на макетной плате представлена на рисунке 5. Принципиальная схема представлена на рисунке 6 и 7.

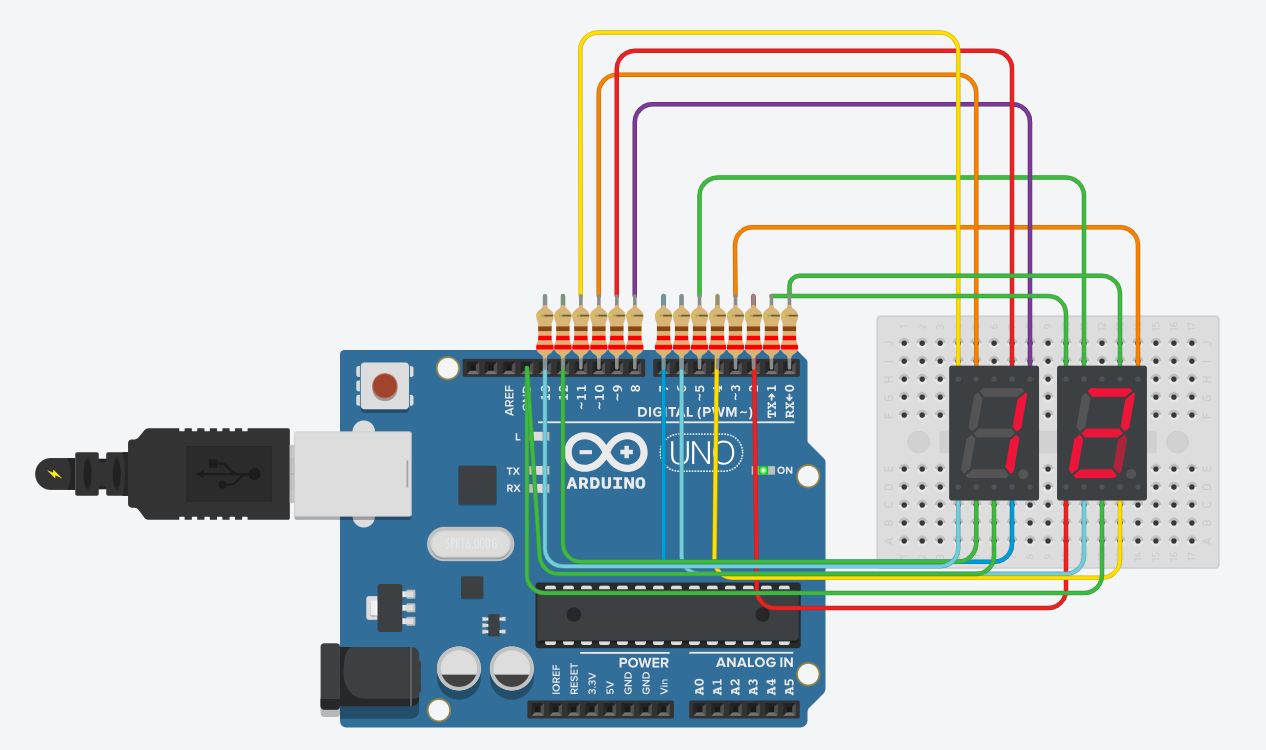


Рисунок 5 – Секундомер

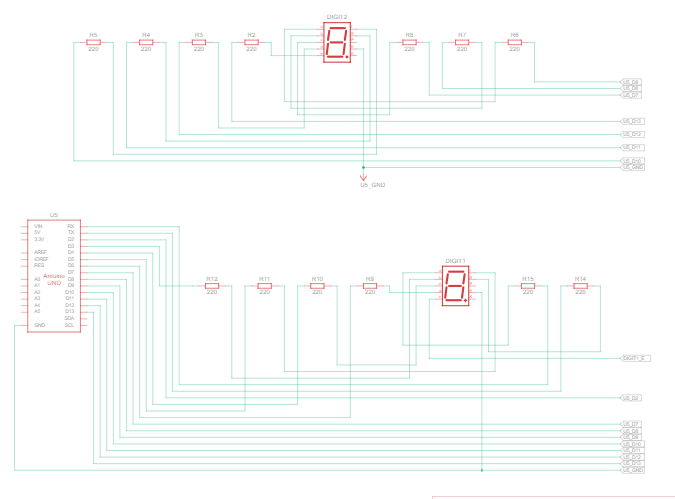


Рисунок 6 – Принципиальная схема «Секундомер»



Рисунок 7 – Принципиальная схема 2 «Секундомер»

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

Листинг 3

#define A1 9

#define B1 8

#define C1 7

#define D1 12

#define E1 13

#define F1 10

#define G1 11

#define A2 0

#define B2 3

#define C2 4

#define D2 6

#define E2 2

#define F2 5

#define G2 1

byte secunda=0;

unsigned long time;

void setup()

{

pinMode(A1, OUTPUT);

pinMode(B1, OUTPUT);

pinMode(C1, OUTPUT);

pinMode(D1, OUTPUT);

pinMode(E1, OUTPUT);

pinMode(F1, OUTPUT);

pinMode(G1, OUTPUT);

pinMode(A2, OUTPUT);

pinMode(B2, OUTPUT);

pinMode(C2, OUTPUT);

pinMode(D2, OUTPUT);

pinMode(E2, OUTPUT);

pinMode(F2, OUTPUT);

pinMode(G2, OUTPUT);

}

void loop()

{

if(millis()-time>1000)

{

time = millis();

secunda++;

if(secunda > 99) secunda = 1;

byte c1 = secunda/10;

byte c2 = secunda%10;

show(c1, 1);

show(c2, 2);

}

delay(10);

}

void show(byte n, byte e)

{

digitalWrite(e==1?A1:A2, LOW);

digitalWrite(e==1?B1:B2, LOW);

digitalWrite(e==1?C1:C2, LOW);

digitalWrite(e==1?F1:F2, LOW);

digitalWrite(e==1?G1:G2, LOW);

digitalWrite(e==1?E1:E2, LOW);

digitalWrite(e==1?D1:D2, LOW);

switch(n)

{

case 1:

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

break;

case 2:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?G1:G2, HIGH);

digitalWrite(e==1?E1:E2, HIGH);

digitalWrite(e==1?D1:D2, HIGH);

break;

case 3:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?G1:G2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

digitalWrite(e==1?D1:D2, HIGH);

break;

case 4:

digitalWrite(e==1?F1:F2, HIGH);

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?G1:G2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

break;

case 5:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?F1:F2, HIGH);

digitalWrite(e==1?G1:G2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

digitalWrite(e==1?D1:D2, HIGH);

break;

case 6:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?F1:F2, HIGH);

digitalWrite(e==1?E1:E2, HIGH);

digitalWrite(e==1?D1:D2, HIGH);

digitalWrite(e==1?G1:G2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

break;

case 7:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

break;

case 8:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?F1:F2, HIGH);

digitalWrite(e==1?E1:E2, HIGH);

digitalWrite(e==1?D1:D2, HIGH);

digitalWrite(e==1?G1:G2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

break;

case 9:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?F1:F2, HIGH);

digitalWrite(e==1?D1:D2, HIGH);

digitalWrite(e==1?G1:G2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

break;

case 0:

digitalWrite(e==1?A1:A2, HIGH);

digitalWrite(e==1?B1:B2, HIGH);

digitalWrite(e==1?F1:F2, HIGH);

digitalWrite(e==1?E1:E2, HIGH);

digitalWrite(e==1?D1:D2, HIGH);

digitalWrite(e==1?C1:C2, HIGH);

break;

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/dOQxxMrz7Ar-amazing-bombul/editel?sharecode=CeEJDmkY-zmYPqMQYTPaumUucyhnaB50QXXMIZ7igo8>

1. Счётчик нажатий

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

Схема сборки на макетной плате представлена на рисунке 8. Принципиальная схема представлена на рисунке 9 и 10.

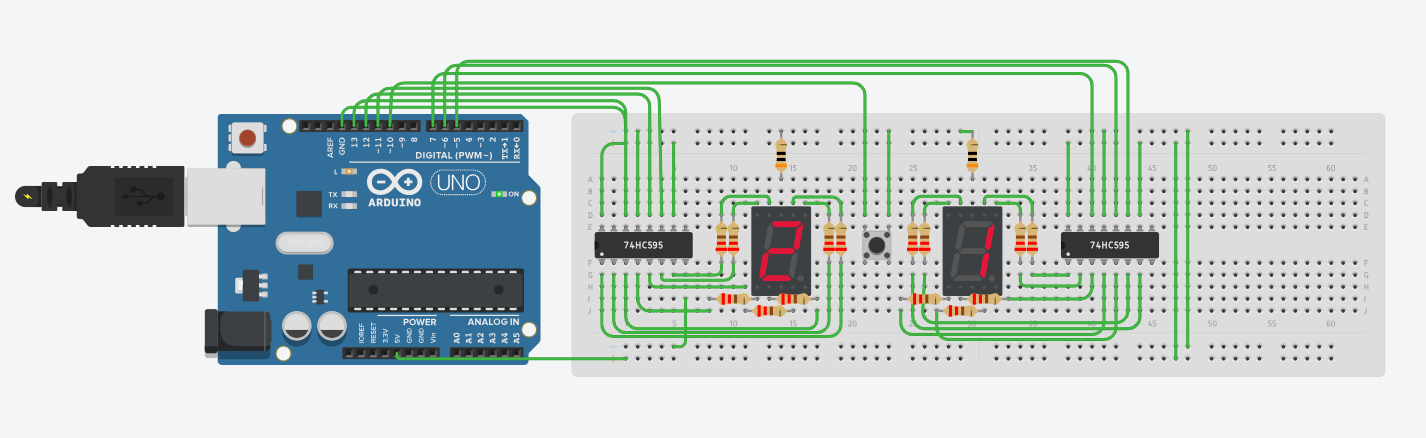


Рисунок 8 – Счётчик нажатий

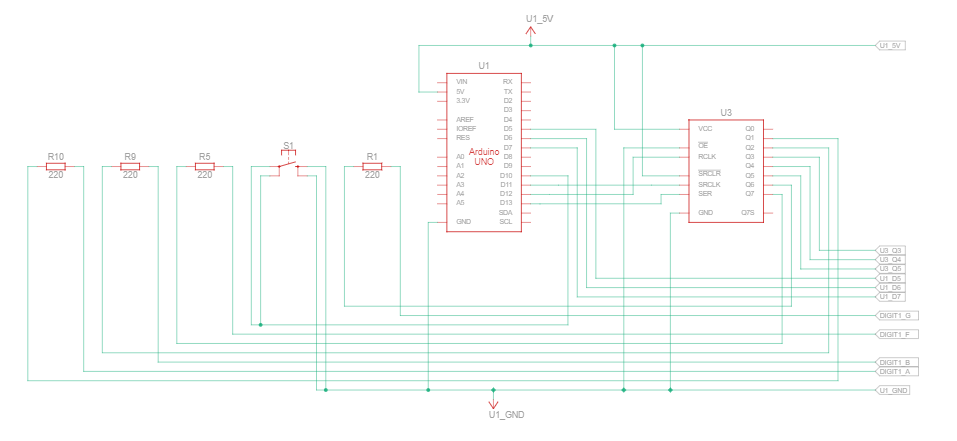


Рисунок 9 – Принципиальная схема «Счётчик нажатий»

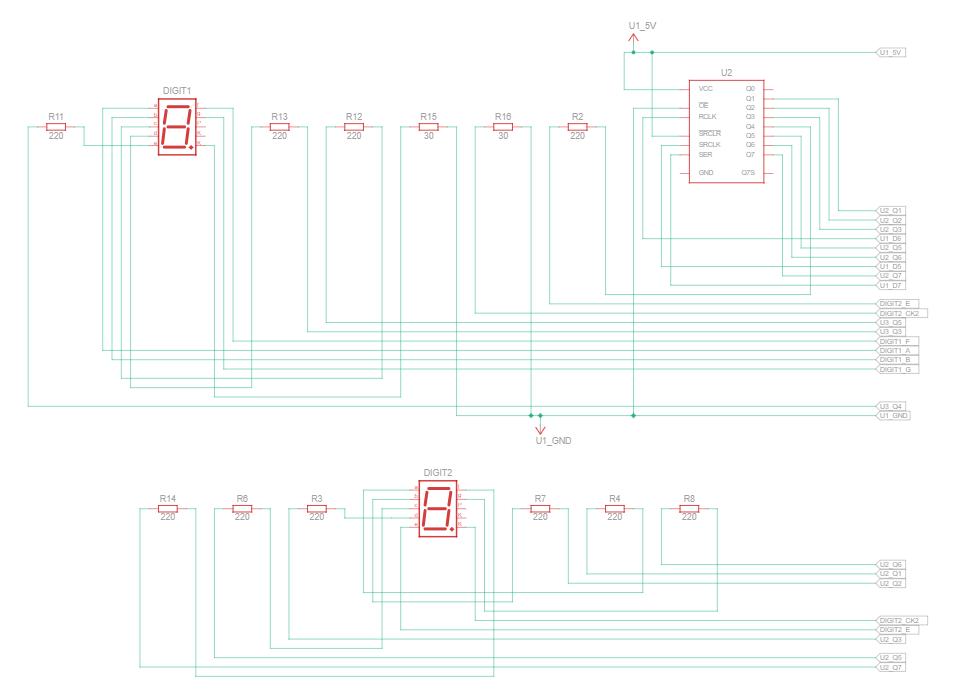


Рисунок 10 – Принципиальная схема 2 «Счётчик нажатий»

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

Листинг 4

#define DATA\_PIN2 13 // пинданных (англ. data)

#define LATCH\_PIN2 12

#define CLOCK\_PIN2 11

#define DATA\_PIN1 7

#define LATCH\_PIN1 6

#define CLOCK\_PIN1 5

#define BUTTON\_PIN 10

int c=0, a=0, b=1;

boolean buttonWasUp =true;

byte segments[10]={

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

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

};

void setup()

{

pinMode(DATA\_PIN1, OUTPUT);

pinMode(CLOCK\_PIN1, OUTPUT);

pinMode(LATCH\_PIN1, 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)) {

c = a + b;

a = b;

b = c;

if(c==89){

a=0; b=1;

}

}

}

buttonWasUp = digitalRead(BUTTON\_PIN);

digitalWrite(LATCH\_PIN2, LOW);

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

digitalWrite(LATCH\_PIN2, HIGH);

digitalWrite(LATCH\_PIN1, LOW);

shiftOut(DATA\_PIN1, CLOCK\_PIN1, LSBFIRST, segments[c%10]);

digitalWrite(LATCH\_PIN1, HIGH);

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/chc6E6Pb5wd-fantastic-wolt/editel?sharecode=88peXeF5tPV_bMh2Dj2ARVQvPPk08FsN3xyiD0rQJn4>

1. Тестер батареек --------

Добавить две кнопки (вверх/вниз) для прокрутки текста не менее 5 строк.

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

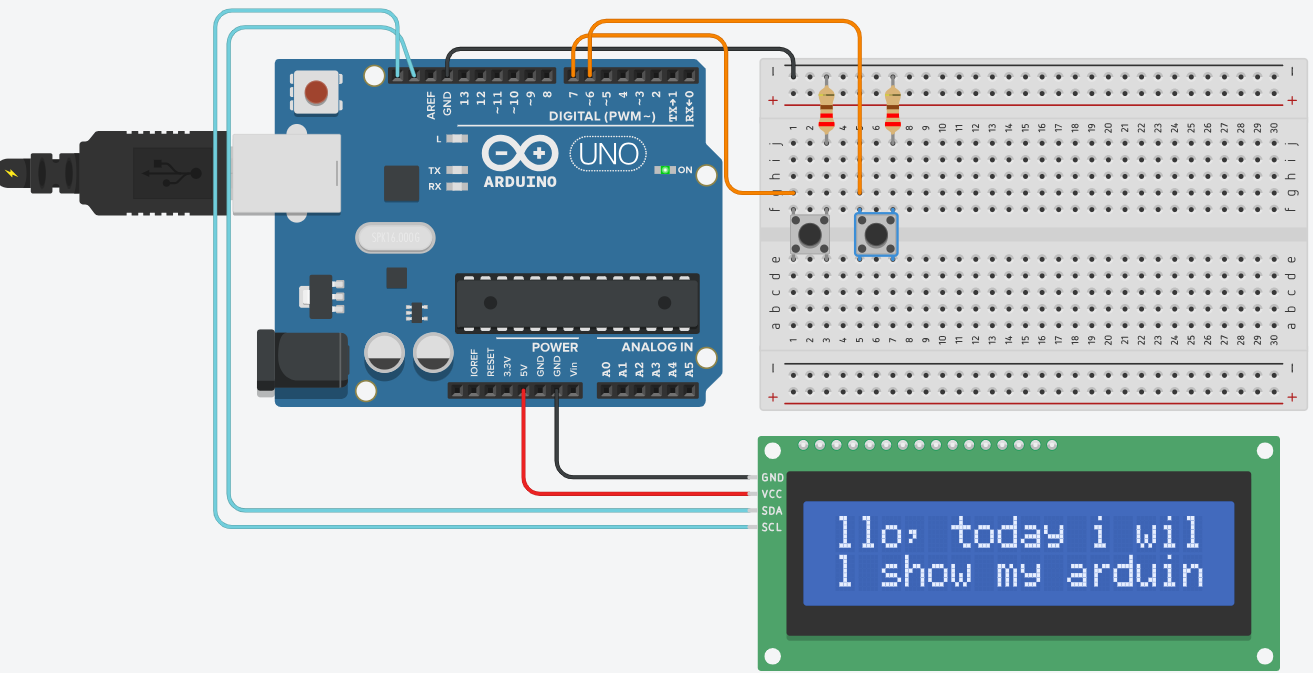


Рисунок 11 – Тестер батареек

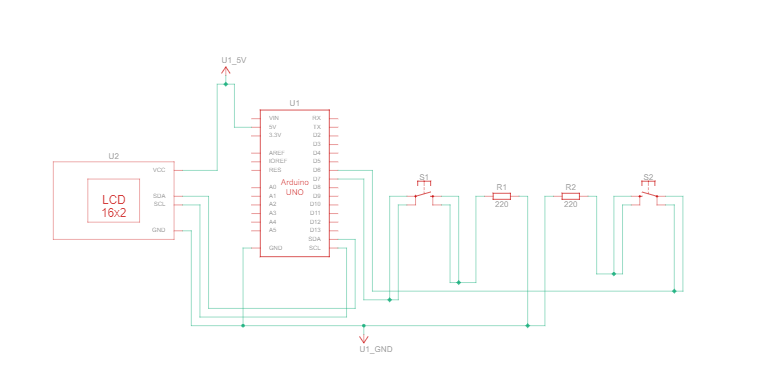


Рисунок 12 – Принципиальная схема «Тестер батареек»

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

Листинг 5

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

String textToShow = "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 luck see you next timeuihuhkjhguhkjiho;jjkugiuvugtlnukjvuhjkygkbuukihuyygoihjhkguikbiluhoiuhioinuiygblbuiyglbiuytgbkiuygvb,kikuygvlbuoygbkuigbliougbolugblouygbljhgoiuyybjkygjbhiyuj.";

int textLength;

int scrollPosition = 0;

const int BTN\_UP = 7;

const int BTN\_DOWN = 6;

const int LINES\_TO\_DISPLAY = 2;

const int CHARACTERS\_PER\_LINE = 16;

void setup() {

lcd.init();

lcd.backlight();

lcd.clear();

textLength = textToShow.length();

pinMode(BTN\_UP, INPUT\_PULLUP);

pinMode(BTN\_DOWN, INPUT\_PULLUP);

}

void loop() {

if (digitalRead(BTN\_UP) == LOW) {

delay(10); // Для предотвращения дребезга кнопки

lcd.clear();

if (scrollPosition < textLength - LINES\_TO\_DISPLAY \* CHARACTERS\_PER\_LINE) {

scrollPosition+=16;

}

}

if (digitalRead(BTN\_DOWN) == LOW) {

delay(10); // Для предотвращения дребезга кнопки

lcd.clear();

if (scrollPosition > 0) {

scrollPosition-=16;

}

}

for (int i = 0; i < LINES\_TO\_DISPLAY; i++) {

lcd.setCursor(0, i);

lcd.print(textToShow.substring(scrollPosition + i \* CHARACTERS\_PER\_LINE,

min(scrollPosition + (i + 1) \* CHARACTERS\_PER\_LINE, textLength)));

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/7RHUaCI7bkw-incredible-stantia-fulffy/editel?sharecode=SVdzSYRQPw3eZRxrjKnPWE2G-GSnIqyRvQ2j4VhiCCI>

1. Тестер батареек 2

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

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

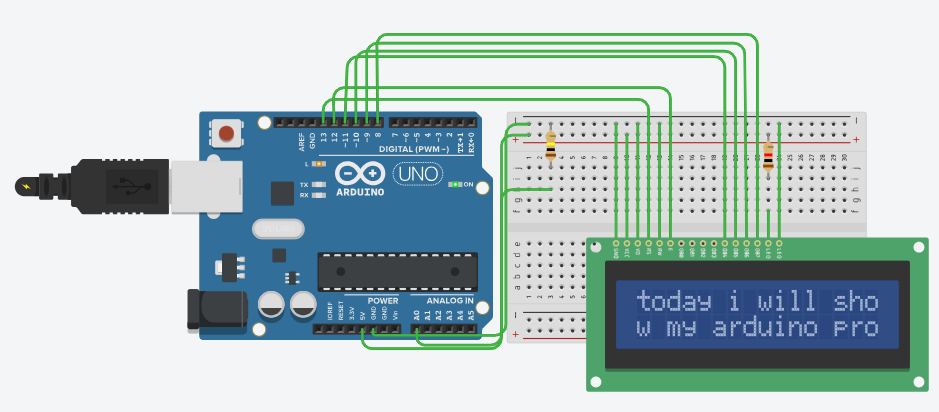


Рисунок 13 – Тестер батареек

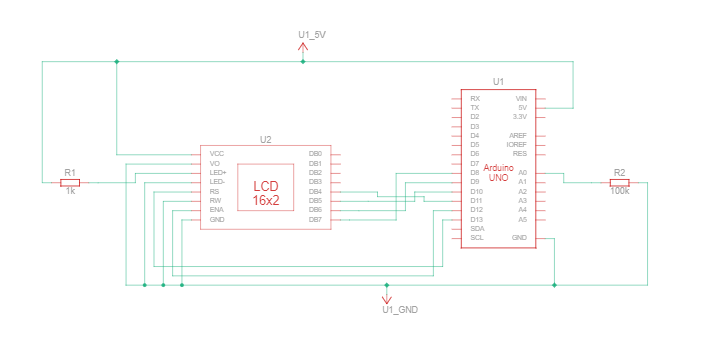


Рисунок 14 – Принципиальная схема «Тестер батареек»

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

Листинг 6

#include <LiquidCrystal.h>

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

String textToShow = "Hello, today i will show my arduino project for get good mark.";

int textLength;

int scrollPosition = 0;

void setup() {

lcd.begin(16, 2);

textLength = textToShow.length();

}

void loop() {

lcd.clear();

if (scrollPosition < textLength + 16) {

int endPos = min(textLength, scrollPosition + 16);

lcd.setCursor(0, 0);

lcd.print(textToShow.substring(scrollPosition, endPos));

lcd.setCursor(0, 1);

lcd.print(textToShow.substring(scrollPosition + 16, min(scrollPosition + 32, textLength)));

}

delay(200); // Для предотвращения дребезга кнопки

scrollPosition++;

if (scrollPosition > textLength + 16) {

scrollPosition = 0;

}

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/0IkPvjNzCVR-tremendous-juttuli-lappi/editel?sharecode=27dtXvSsz8oKe3CCj0SMtIVYVtRa7SxgmiHXaH1SRGE>

1. Перетягивание каната ---

Формулировка задания: Добавить в схему мотор или сервопривод. Изменить программу таким образом, чтобы при управлении “канатом” дополнительно менялась скорость и направление вращения двигателя/сервопривода.

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

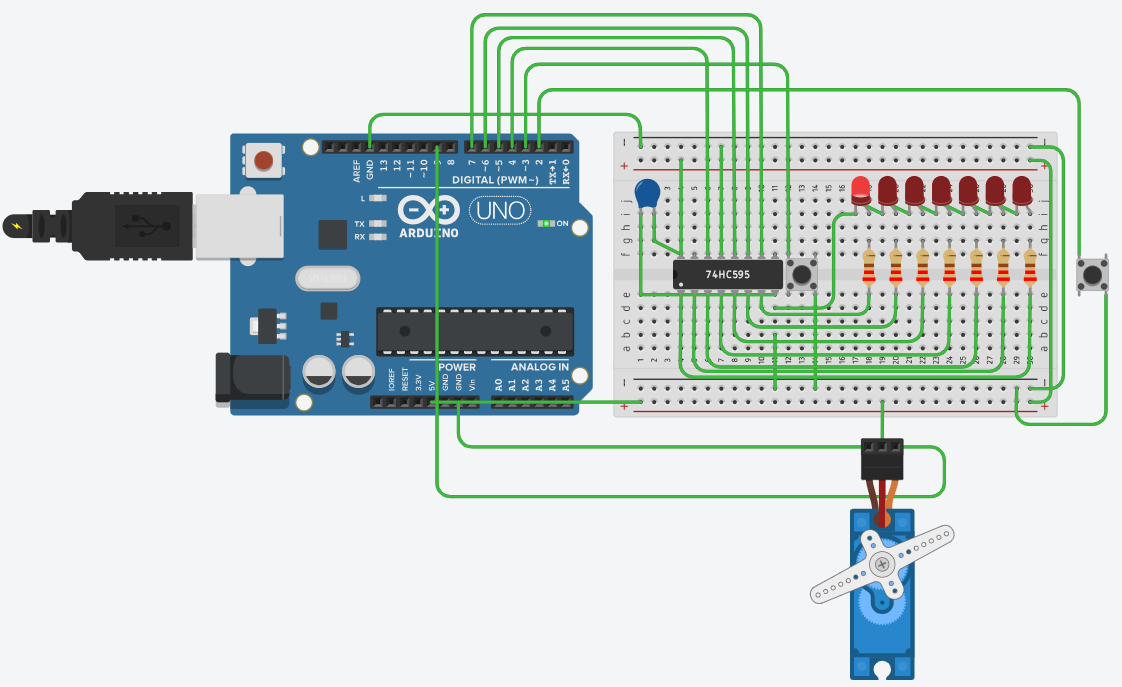


Рисунок 15 – Перетягивание каната

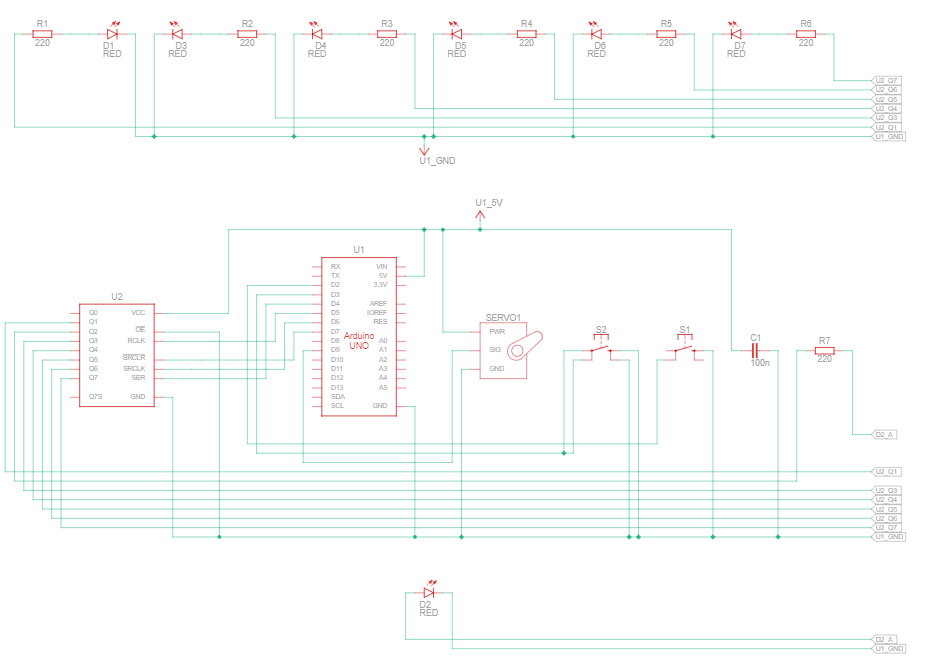


Рисунок 16 – Принципиальная схема «Перетягивание каната»

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

Листинг 7

#define DS 4

#define ST\_CP 5

#define SH\_CP 6

#define MR 7

#define BUT1\_PIN 3

#define BUT2\_PIN 2

#define PRESS\_COUNT 1

#include <Servo.h>

int pos = 90, step = 2;

Servo servo\_9;

void setup() {

servo\_9.attach(9);

pinMode(BUT2\_PIN, INPUT\_PULLUP);

pinMode(BUT1\_PIN, INPUT\_PULLUP);

pinMode(DS, OUTPUT);

pinMode(ST\_CP, OUTPUT);

pinMode(SH\_CP, OUTPUT);

pinMode(MR, OUTPUT);

digitalWrite(MR, LOW);

delay(10);

digitalWrite(MR, HIGH);

delay(100);

}

byte prevOneState = HIGH;

byte prevTwoState = HIGH;

byte ledPos = (1 << 4);

int press\_count[2] = { 0, 0 };

void loop() {

byte oneState = digitalRead(BUT1\_PIN);

byte twoState = digitalRead(BUT2\_PIN);

if (oneState == LOW && prevOneState == HIGH) {

press\_count[0]++;

}

if (twoState == LOW && prevTwoState == HIGH) {

press\_count[1]++;

}

prevOneState = oneState;

prevTwoState = twoState;

if (press\_count[0] == PRESS\_COUNT) {

if (ledPos < 128) {

ledPos \*= 2;

pos += 10;

}

press\_count[0] = 0;

}

if (press\_count[1] == PRESS\_COUNT) {

if (ledPos > 2) {

ledPos /= 2;

pos -= 10;

}

press\_count[1] = 0;

}

shift(ledPos);

}

void shift(byte b) {

for (int i = 7; i >= 0; i--) {

digitalWrite(DS, ((b >> i) & 1));

digitalWrite(SH\_CP, HIGH);

digitalWrite(SH\_CP, LOW);

servo\_9.write(pos);

}

digitalWrite(ST\_CP, HIGH);

digitalWrite(ST\_CP, LOW);

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/keeT3CBOOAC-glorious-curcan/editel?sharecode=78CN7pGhrEPunVpOUDBIY7oT0y6wfaUXV056KogyQlg>

1. Управление скоростью мотора

Формулировка задания: Управляйте скоростью мотора при помощи биполярного n-p-n транзистора.

Схема сборки на макетной плате представлена на рисунке 17. Принципиальная схема представлена на рисунке 18.

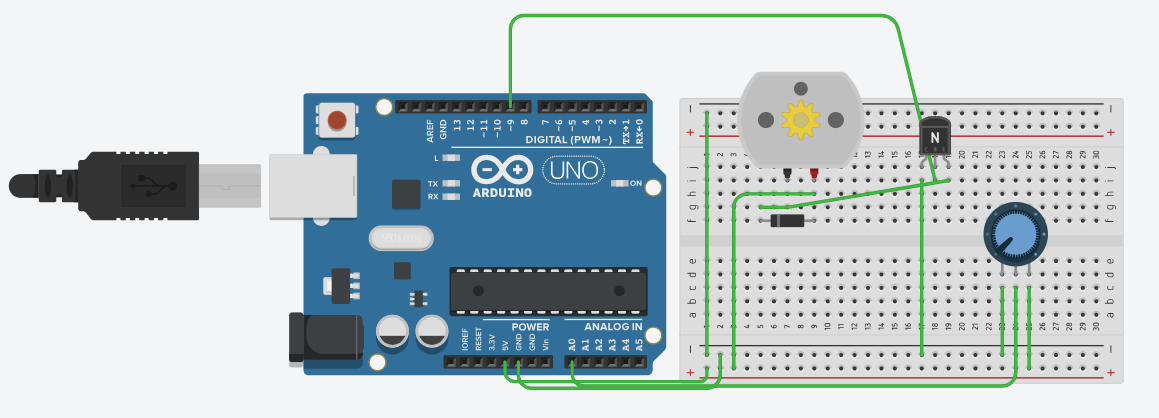


Рисунок 17 – Управление скоростью каната

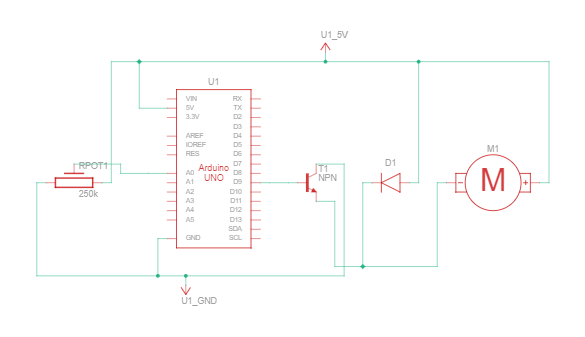


Рисунок 18 – Принципиальная схема «Управление скоростью каната»

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

Листинг 8

#define MOTOR\_PIN 9

int pot = A0;// пин потенциометра

void setup(){

pinMode(MOTOR\_PIN, OUTPUT);

pinMode(pot, INPUT);

Serial.begin(9600);

}

void loop(){

int val = analogRead(pot) / 4;

Serial.println(val);

analogWrite(MOTOR\_PIN, val);

}

Ссылка на готовый проект:

<https://www.tinkercad.com/things/jMIzqsbR81L-bodacious-habbi-waasa/editel?sharecode=Rc8YwmaFtl-gUshtwJaYkxhvCmFGWRV6lzLLuiqwRHw>

1. Секундомер на таймере 555

Формулировка задания: Сделать генератор случайных чисел, без программирования ( в пределах 10).

Схема сборки на макетной плате представлена на рисунке 19. Принципиальная схема представлена на рисунке 20.

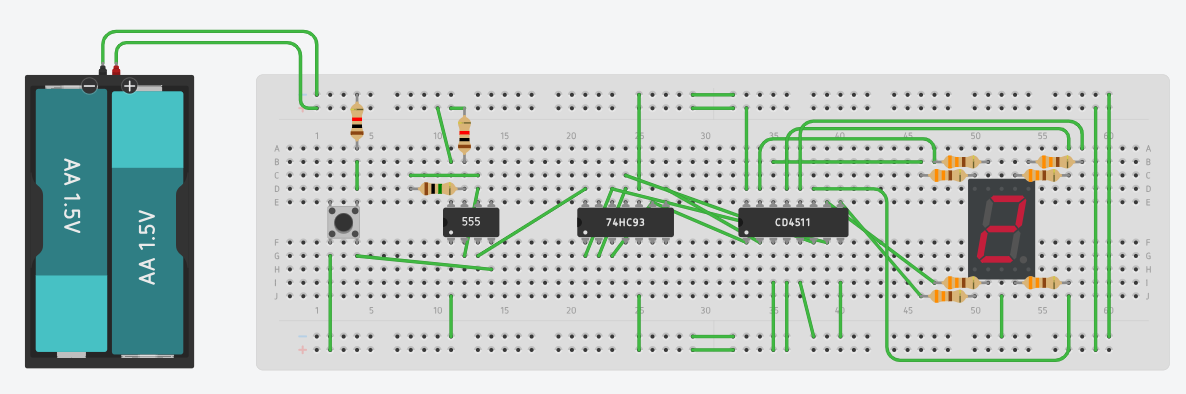


Рисунок 19 – Секундомер на таймере 555

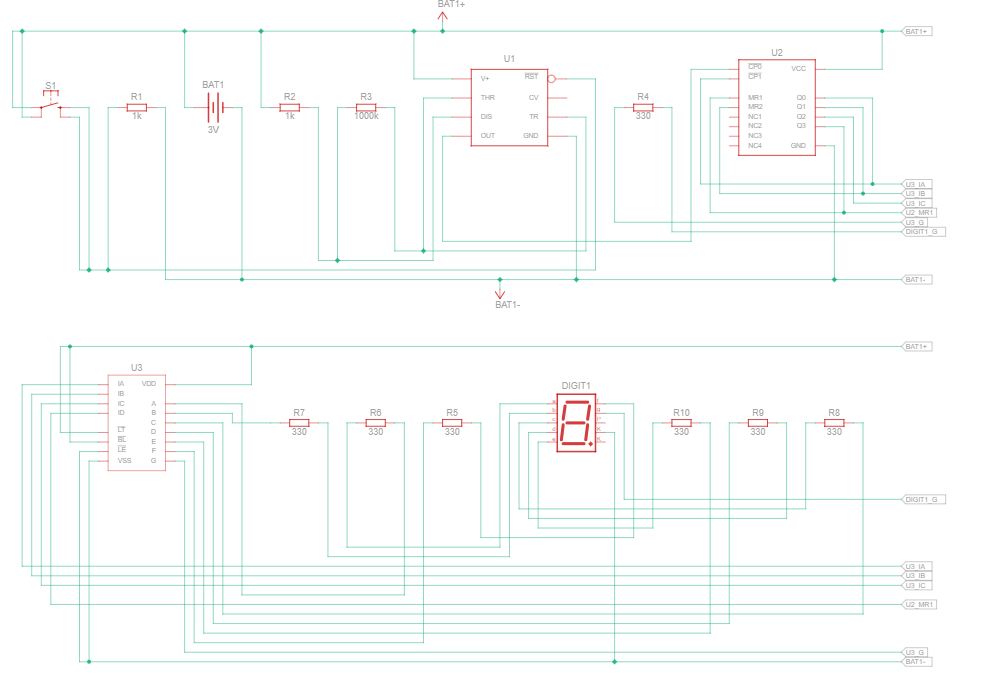


Рисунок 20 – Принципиальная схема «Секундомер на таймере 555»

Код, необходимый для работы данной схемы не нужен.

Ссылка на готовый проект:

<https://www.tinkercad.com/things/iD4TnqEx4Ib-sekundomer-na-tajmere-555/editel?returnTo=%2Fdashboard%3Ftype%3Dcircuits%26collection%3Ddesigns&sharecode=kSSGwKbBwU0uM6dqBp3CZb9PzOVKC_Q3gyxQiUbE5VU>

1. Секундомер на таймере 555 – 2

Формулировка задания: Сделать секундомер до 99 только четных чисел.

Схема сборки на макетной плате представлена на рисунке 21. Принципиальная схема представлена на рисунке 22 - 23.

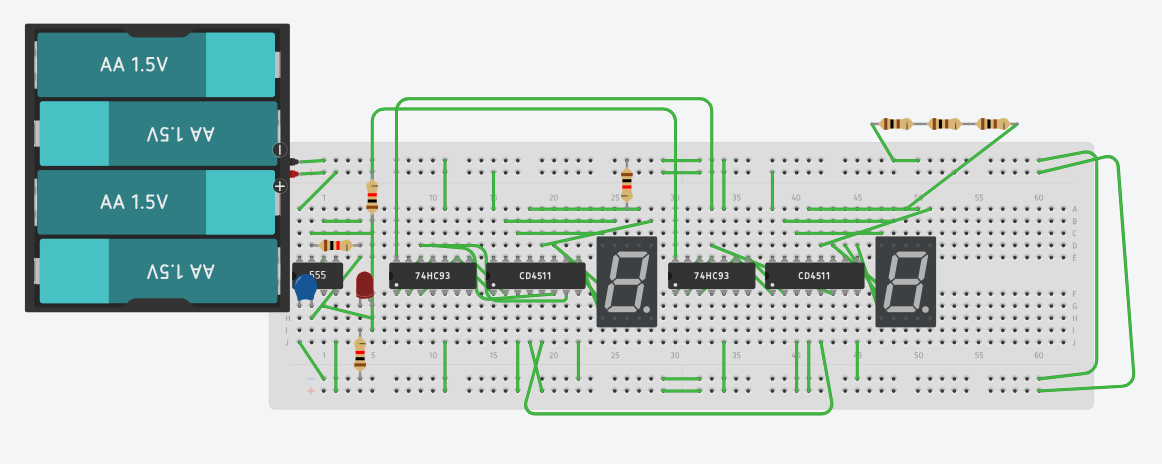


Рисунок 21 – Секундомер на таймере 555

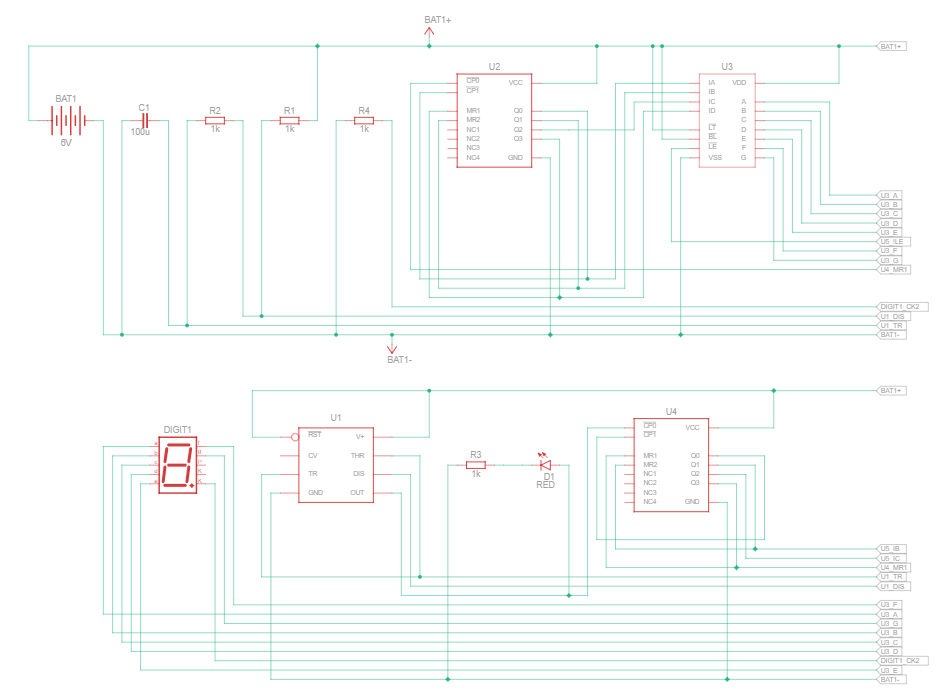


Рисунок 22 – Принципиальная схема «Секундомер на таймере 555»

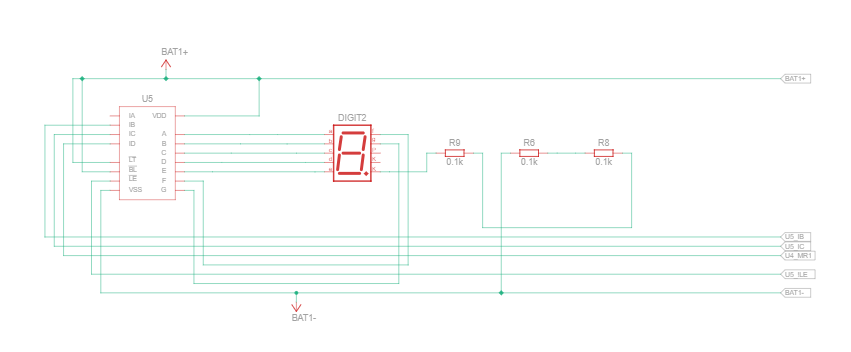


Рисунок 23 – Принципиальная схема 2 «Секундомер на таймере 555»

Код, необходимый для работы данной схемы не нужен.

Ссылка на готовый проект:

<https://www.tinkercad.com/things/jXqEgdVGyOy-terrific-duup-jaagub/editel?sharecode=aPcBI-Ky83FCHCK2FrUJQ_7nO_CrRpWi2IK1hSuU2E8>

Вывод

В ходе выполнения практической работы было произведено ознакомление с микросхемами, дисплеями и мониторами Arduino, а также написаны программы на специальном для этого языке программирования.