ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА

КОМИТЕТ ПО НАУКЕ И ВЫСШЕЙ ШКОЛЕ

**Государственное бюджетное образовательное учреждение**

**среднего профессионального образования**

**«Санкт-Петербургский технический колледж управления и коммерции»**

|  |
| --- |
| «Утверждаю» |
| Руководитель практики от организации / колледжа |
|  |
| / Меньков А.П./ Меньков А.П |
| (подпись, Ф.И.О. руководителя практики от организации / колледжа)    «08»       июня     2020 г. |

**ОТЧЕТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**09.02.01** «**Компьютерные системы и комплексы»**

**"МАКЕТИРОВАНИЕ УСТРОЙСТВ НА БАЗЕ ПЛАТФОРМЫ ARDUINO"**

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент группы № 9СК-31  Дьячков Николай Михайлович |  | Принял  Руководитель практики  Меньков Александр Прокопьевич |
| (Ф.И.О. студента) |  | (подпись, Ф.И.О. руководителя практики от колледжа)    (оценка) |

Санкт-Петербург

2020г

**Раздел 7. Исследование систем автоматического управления**

**Цель работы:**

1. Разработка автоматического регулятора температуры;
2. Разработка бойлера;
3. Разработка учебного стенда-тренажер.

**Выполнение работы:**

**Задание 1**

Создал проект 7.1. На макетной плате разместил ЖК-экран с токоограничивающим резистором 220 Ом, контакт V0 подключил к потенциометру для регулировки контрастности дисплея. К контакту А0 подключил датчик температуры ТМР36 для вывода измеряемых значений на дисплей. Для регулировки предельной температуры установил две кнопки, подключенные к контактам 9 и 10. Превысив допустимую температуру, должен сработать пьезоэлемент, подключенный через резистор 150 Ом к контакту 8, и двигатель постоянного тока, подключенный к контакту 11, рисунок 1.

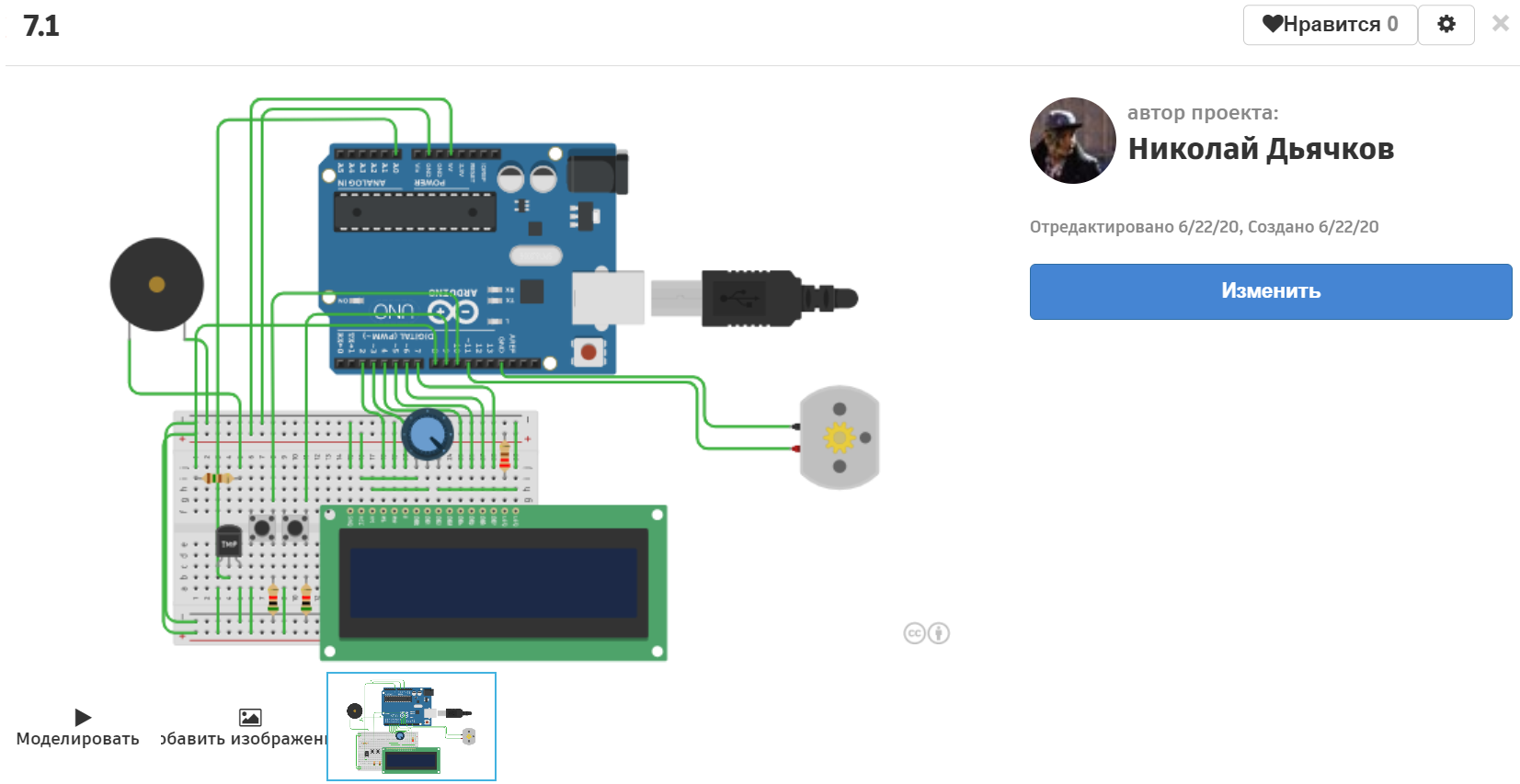


Рисунок 1. Монтажная схема

**Текст программы:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(2,3,4,5,6,7);

byte degree[8]={

B00110,

B01001,

B01001,

B00110,

B00000,

B00000,

B00000,

B00000,

};

byte fan\_on[8]={

B00100,

B10101,

B01110,

B11111,

B01110,

B10101,

B00100,

B00000,

};

byte fan\_off[8]={

B00100,

B00100,

B00100,

B11111,

B00100,

B00100,

B00100,

B00000,

};

const int SPEAKER=8;

const int DOWN\_B=10;

const int UP\_B=9;

const int FAN=11;

int set\_temp=22;

boolean one\_time=false;

int val;

void setup()

{

pinMode(FAN,OUTPUT);

pinMode(UP\_B,INPUT);

pinMode(SPEAKER,OUTPUT);

lcd.begin(16,2);

lcd.createChar(0,degree);

lcd.createChar(1,fan\_off);

lcd.createChar(2,fan\_on);

lcd.setCursor(0,0);

lcd.print("Current:");

lcd.setCursor(10,0);

lcd.write((byte)0);

lcd.setCursor(11,0);

lcd.print("C");

lcd.setCursor(0,1);

lcd.print("Set:");

lcd.setCursor(10,1);

lcd.write((byte)0);

lcd.setCursor(11,1);

lcd.print("C");

lcd.setCursor(15,1);

lcd.write(1);

}

void loop()

{

val=analogRead(0);

val=map(val,102,1023,0,450);

lcd.setCursor(8,0);

lcd.print(val);

if(digitalRead(DOWN\_B)==HIGH)

{

set\_temp--;

delay(200);

}

if(digitalRead(UP\_B)==HIGH)

{

set\_temp++;

delay(200);

}

lcd.setCursor(8,1);

lcd.print(set\_temp);

if(val>set\_temp)

{

if(!one\_time)

{

tone(SPEAKER,400);

delay(500);

one\_time=true;

}

else

{

noTone(SPEAKER);

}

digitalWrite(FAN,HIGH);

lcd.setCursor(15,1);

lcd.write(2);

}

else

{

noTone(SPEAKER);

one\_time=false;

digitalWrite(FAN,LOW);

lcd.setCursor(15,1);

lcd.write(1);

}

}

На рисунке 2 продемонстрирована работоспособность собранной схемы.

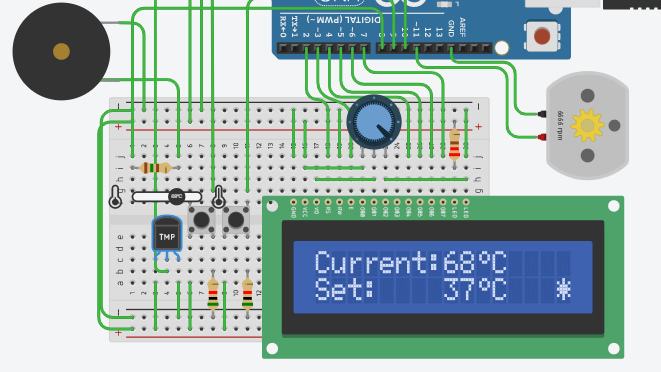


Рисунок 2. Автоматический регулятор температуры

После запуска на первой строке дисплея выведется значение температуры с датчика ТМР36. На второй строке установленный предел температуры и в конце строки значок, отображающий работу двигателя (вентилятора). Нажав на левую кнопку можно понизить предел температуры, а правой повысить. Если температура с датчика ТМР36 не превышает установленный предел, двигатель выключен, а в правом нижнем углу дисплея отображается знак «+». При превышении температуры пьезоэлемент начинает пищать, двигатель начинает вращаться, а значок в углу экрана меняется на «\*». Если температура возвращается к значению «Set:», двигатель останавливается, в углу дисплея отображается «+». Контраст ЖК-экрана можно регулировать потенциометром.

**Задание 2**

Создал проект 7.2. Для реализации полностью автоматической системы управления нагреванием воды разместил на макетной плате сервопривод, подключенный к порту вывода с ШИМ – порту 11, с автономной системой питания: батарейка 9v, стабилизатор напряжения на 5v, два полярных конденсатора по10мкФ. Для отображения действий бойлера разместил RGB-светодиод, подключенный к портам вывода с ШИМ – порты 6 и 9, и два ограничивающих резистора 220 Ом. Потенциометр для регулировки температуры нагрева номиналом 1кОм, подключаемый к порту A0. Пьезоэлемент для формирования звуковых сигналов при сменах режимов работы с ограничивающим резистором 150 Ом, подключаемый к порту 4. Кнопки ПУСК и СТОП, подключаемые к портам 2 и 3 с резисторами 5 кОм, подтягивающими к нулю, рисунок 3.

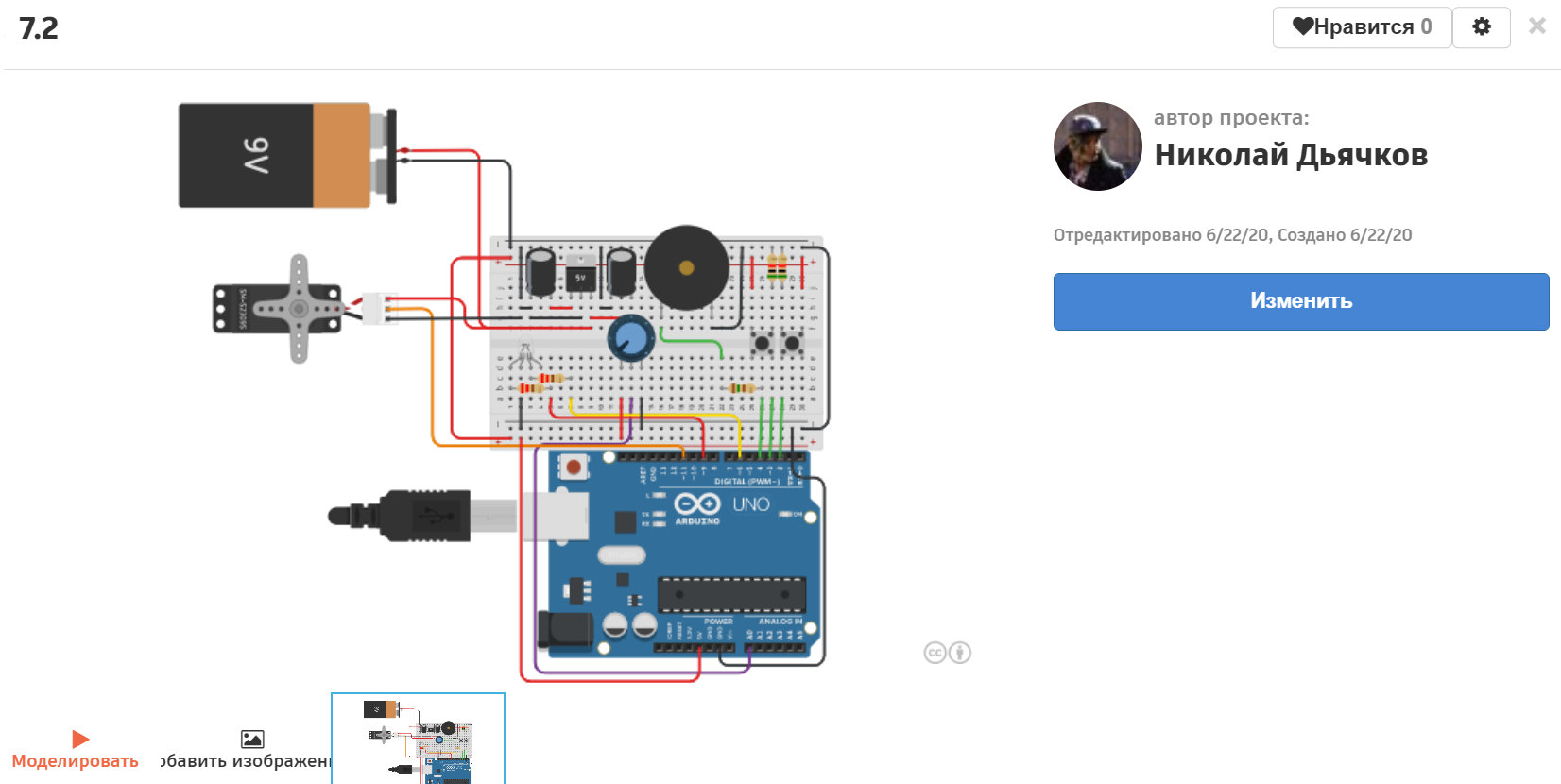


Рисунок 3. Монтажная схема

**Текст программы:**

#include <Servo.h>

const int BLUE=6;

const int RED=9;

const int SERVO=11;

const int PUSK=2;

const int STOP=3;

const int ZVUK=4;

const int T\_MAX=A0;

Servo myServo;

int val=0;

int svB=0;

int svR=0;

int i;

int tempV=10;

int tempC=50;

boolean Work=false;

void setup()

{

Serial.begin(9600);

myServo.attach(SERVO);

myServo.write(0);

pinMode(RED,OUTPUT);

pinMode(BLUE, OUTPUT);

pinMode(PUSK, INPUT);

pinMode(STOP, INPUT);

pinMode(ZVUK, OUTPUT);

}

void loop()

{

if(digitalRead(STOP)==HIGH)

Work=false;

if(Work==true)

{

Serial.println(" water set");

tone (ZVUK, 500, 100);

for (val=0; val<180; val=val+3)

{

analogWrite(RED, svR);

myServo.write(val);

delay(150);

analogWrite(BLUE,svB);

svB=svB+4;

}

tempV=10;

svB=255;

svR=0;

Serial.println(" water heating");

tone (ZVUK, 400, 100);

tempC=analogRead(T\_MAX);

tempC=map(tempC,0,1023,50,90);

for(i=255; i>=0; i=i-12)

{

analogWrite(BLUE,svB);

analogWrite(RED,svR);

svR=svR+12;

svB=svB-12;

delay(200);

tempV=map(svR, 0, 255, 10,tempC);

Serial.print(tempV);

Serial.print(" ");

}

Serial.println(tempC);

svB=0;

svR=255;

analogWrite(BLUE, svB);

Serial.println(" draining");

tone (ZVUK, 300, 100);

for(val=180; val>=0; val=val-3)

{

myServo.write(val);

delay(150);

analogWrite(RED,svR);

svR=svR-4;

}

svR=0;

tempV=10;

}

else

if(digitalRead(PUSK)==HIGH)

Work=true;

}

На рисунке 4 продемонстрирована работоспособность собранной схемы.

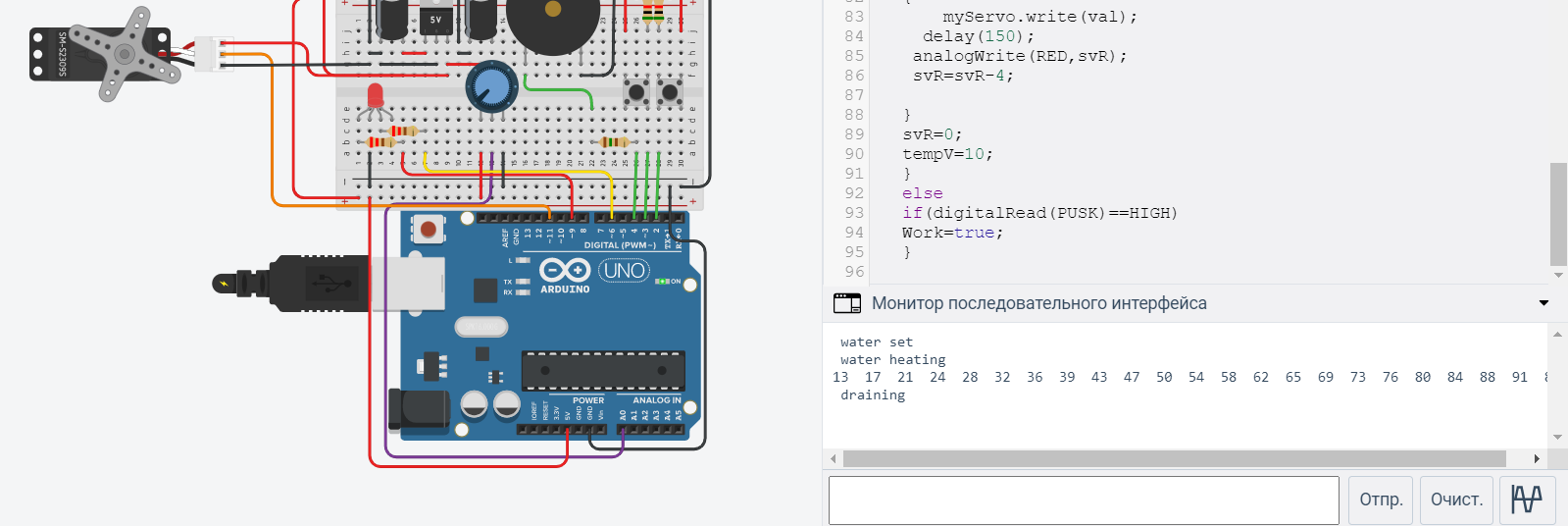


Рисунок 4. Работа бойлера

Система должна осуществлять автоматическую работу в следующих режимах:

1. Набор холодной воды с температурой 10 градусов.
2. Нагрев воды – от 10 градусов до температуры, установленной потенциометром.
3. Слив всей воды из бойлера.

После прохождения всего цикла режимов, режимы повторяются заново.

В режиме набора воды синий светодиод начинает светиться ярче; когда вода нагревается, светодиод медленно начинает смещать цвет свечения в красный; в режиме слива свечение светодиода уменьшается.

При смене режима в терминале появляется сообщение о переключении режима, а пьезоэлемент издает звуковой сигнал. При нагреве воды в терминал выводятся значения температуры, пока вода не достигнет необходимого, после чего начинается процедура слива. В процессе работы бойлера можно регулировать температуру нагрева от 40 до 90 градусов с помощью потенциометра.

**Задание 3**

Создал проект 7.3. Разместил на макетной плате три кнопки, подключенные к 10, 3 и 2 контакту контроллера. Подключил ЖК-экран к контроллеру, анод LED к цепи питания через резистор 0.3 кОм и V0 к цепи земли через резистор 0.3 кОм. Для операций с флэш-памятью использовал библиотеку EEPROM.h, рисунок 5.

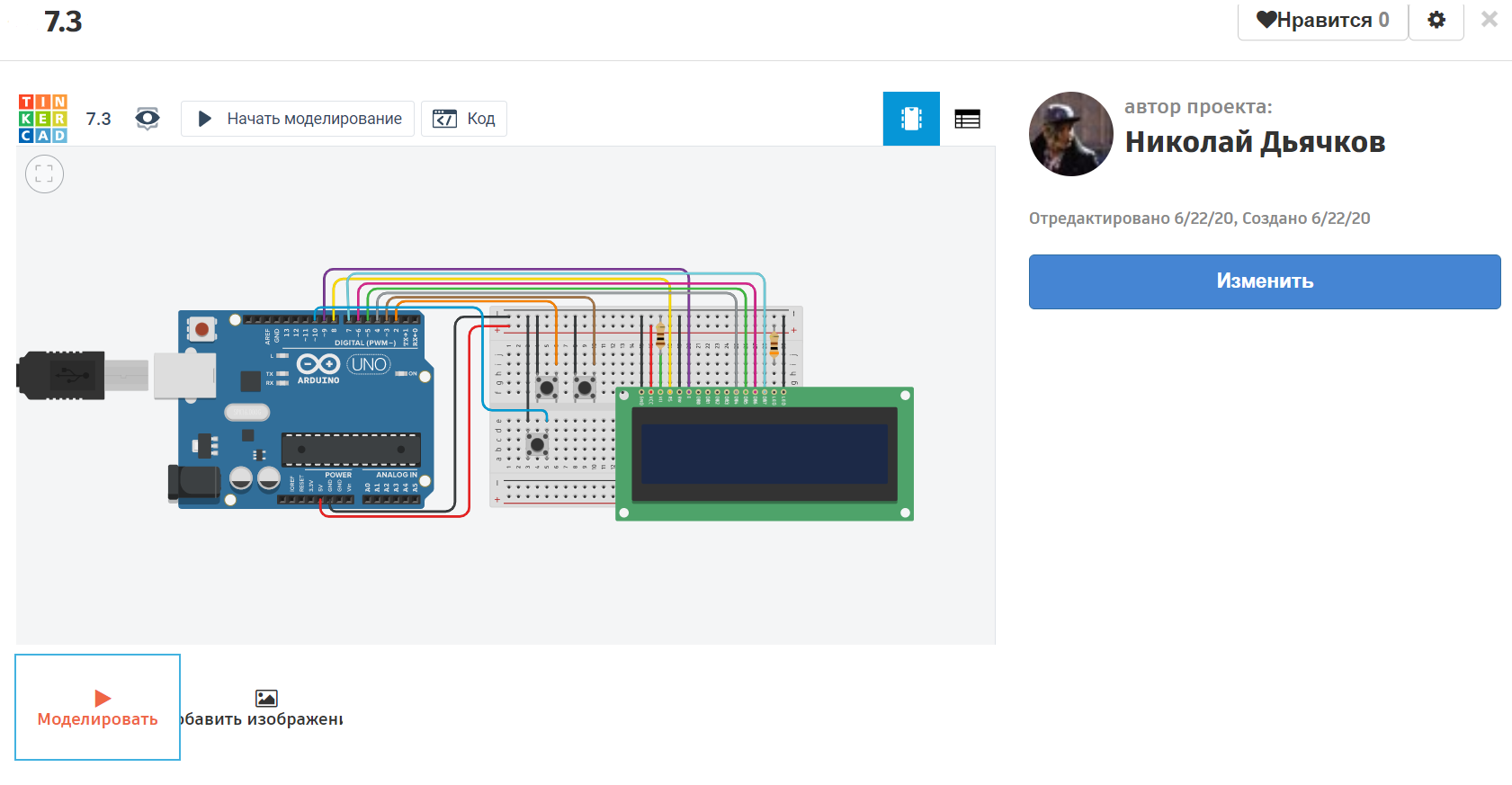


Рисунок 5. Монтажная схема

**Текст программы:**

#include <LiquidCrystal.h>

#include <EEPROM.h>

unsigned char AB[17]={0x79, 0xa6, 0x59, 0xc7, 0xe9, 0xbe,

0x3c, 0xd2, 0x4e, 0x27, 0xfa, 0x6f, 0xed,0x16,0xb7, 0x5e,0x6e};

unsigned char USL [15][6] = {"A-B=2", "B%3=0", "B-A=4", "A=0xC",

"B=0x9", "B-A=2", "B-A=9", "A>13", "B%4=0", "A+B=9", "B=0xA",

"B-A=9", "A=0xD", "B-A=5", "A%3=2"};

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

int address=0;

int PUSK=10;

unsigned char A;

int schetPRAV=0;

int schetOSHIB=0;

int timer=10;

boolean otvet;

boolean Work=false;

int zadanie;

void setup()

{

pinMode(2, INPUT);

digitalWrite(2, HIGH);

pinMode(3, INPUT);

digitalWrite(3, HIGH);

pinMode(PUSK, INPUT);

digitalWrite(PUSK, HIGH);

lcd.begin (16,2);

for (int i=0; i<17; i++)

{

EEPROM.update(address, AB[i]);

address++;

}

for (int i=0; i<15; i++)

{

for (int j=0; j<6; j++)

{

EEPROM.update(address, USL[i][j]);

address++;

}

}

attachInterrupt(0, da, FALLING);

attachInterrupt(1, net, FALLING);

}

void da()

{

if (zadanie>7)

{

otvet=bitRead(AB[16], zadanie-8);

}

else otvet=bitRead(AB[15], zadanie);

if(otvet==1)

{

schetPRAV++;

}

else

{

schetOSHIB++;

}

timer=0;

}

void net()

{

if (zadanie>7)

{

otvet=bitRead(AB[16], zadanie-8);

}

else otvet=bitRead(AB[15], zadanie);

if(otvet==0)

{

schetPRAV++;

}

else

{

schetOSHIB++;

}

timer=0;

}

void loop()

{

if(digitalRead(PUSK)==LOW)

Work=true;

if(Work==true)

{

schetPRAV=0;

schetOSHIB=0;

for (zadanie=0; zadanie<15; zadanie++)

{

timer=10;

address=zadanie;

A=EEPROM.read(address);

lcd.setCursor(0,0);

lcd.print("A=");

lcd.setCursor(2,0);

lcd.print(A/0x10, BIN);

lcd.setCursor(0,1);

lcd.print("B=");

lcd.setCursor(2,1);

lcd.print(A&0x0f, BIN);

lcd.setCursor(7,0);

address=zadanie\*6+17;

for(int j=0; j<5; j++)

{

lcd.print(char(EEPROM.read(address)));

address++;

}

lcd.setCursor(8,1);

lcd.print(schetPRAV);

lcd.print("-");

lcd.print(schetOSHIB);

lcd.print(" ");

lcd.setCursor(14,1);

lcd.print(timer);

while (timer>0)

{

delay(1000);

if(timer==0)break;

timer=timer-1;

lcd.setCursor(14,1);

lcd.print(timer);

lcd.setCursor(15,1);

lcd.print(" ");

}

lcd.clear();

}

Work=false;

lcd.setCursor(8,1);

lcd.print(schetPRAV);

lcd.print("-");

lcd.print(schetOSHIB);

}

}

На рисунке 6 продемонстрирована работоспособность собранной схемы.

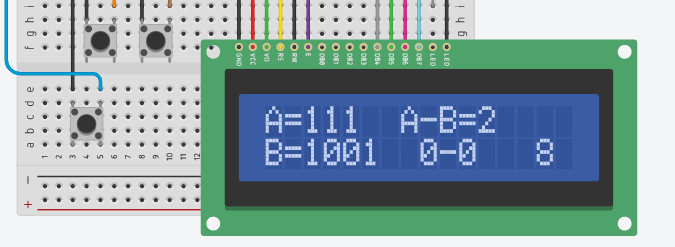


Рисунок 6. Работа стенда-тренажер

При работе надо нажать нижнюю кнопку, после чего запустится тестирование. В тестировании задаются только вопросы с ответами «да» или «нет», за ответы на которые отвечают соответственно кнопки левая (да) и правая (нет). После ответов на все 15 заданий выполнение теста завершится, а на экране останутся количества правильных и неправильных ответов. Чтобы начать тестирование с начала, нужно нажать на нижнюю кнопку.

**Вывод:** ознакомился с системами автоматического управления.